A BENCHMARKING FRAMEWORK FOR SENSITIVITY AND COMPARATIVE ANALYSIS OF ENERGY HARVESTING STRATEGIES VIA RETRACTABLE WIND ENERGY CONVERSION SYSTEMS

by

Guy Gadola

B.S. in Electrical Engineering, University of Pittsburgh, 1991

M.S. in Mathematics (with the Computer Science Option),

Youngstown State University, 2006

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of Arts and Sciences in partial

fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2019

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Guy Gadola

It was defended on

July 30, 2019

and approved by

Daniel Mossé, Professor, Computer Science, University of Pittsburgh

Alex K. Jones, Professor, Electrical and Computer Engineering, University of Pittsburgh

Rami Melhem, Professor, Computer Science, University of Pittsburgh

Greg Reed, Professor, Electrical and Computer Engineering, University of Pittsburgh

Taieb Znati, Professor, Computer Science, University of Pittsburgh

Dissertation Director: Daniel Mossé, Professor, Computer Science, University of Pittsburgh

A BENCHMARKING FRAMEWORK FOR SENSITIVITY AND COMPARATIVE ANALYSIS OF ENERGY HARVESTING STRATEGIES VIA RETRACTABLE WIND ENERGY CONVERSION SYSTEMS

Guy Gadola, PhD

University of Pittsburgh, 2019

Wind power is well known for being variable. Our main insight is that one can take advantage of variability by appropriately building wind-energy harvesters that may be stowed/retracted when winds are calm. We refer to harvesters that can be deployed and retracted on command as retractable wind-energy harvesters (RWEHs). Among other advantages, stowed harvesters do not block views, do not constrain avian life, and do not make noise, and thus can increase the neighborliness of harvesting wind near or within a residential community.

RWEH control algorithms help owners to achieve the neighborliness that might be required by an RWEH hosting community while helping RWEHs' efficiency. The stowing requirements, or operation limitation agreements (OLAs), specify conditions when the retractable harvesters should be stowed (e.g., when it is not windy).

In this work, we contribute a suite of benchmarks to compare RWEH control algorithms, three families of control algorithms, and a simulator with which to run the algorithms. The benchmark suite provides workloads formed from the following workload components: 1. specifications of a harvester to be controlled, 2. a set of historical windspeeds from 30 weather stations, and 3. a variety of stowing requirements.

We derived OLAs from a survey of 304 respondents in which survey-takers were asked whether they would support RWEHs viewable from where they live and when the RWEHs should be hidden or stowed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

PREFACE

There are many to thank for their help, patience, and encouragement. I especially would like to thank my parents. Their patient support has been and is tremendous, as I pursued the goals of working in higher education and advancing renewable energy.

Thanks go to the Computer Science Department staff. A special thanks go to Keena Walker, the Computer Science Department and Graduate Support Administrator, for helping me to register and to complete the proper paperwork.

I would also like to express my appreciation for the late Dr. Levitan's encouragement. He agreed to be a reference for me when I was an undergraduate. Years later, he and Dr. Chiarulli accepted me into the Computer Engineering graduate program. Thanks also go to Dr. Chiarulli for guiding me as I became a real graduate student, which included becoming a teaching assistant. In that capacity, I appreciated being part of the teaching teams; thank you to all the team members, which included my fellow TA's and the following CS 447 lecturers: Dr. Childers, Dr. Misurda, and Dr. Zhang.

Thanks go to Dr. Chrysanthis for co-authoring a paper with me about transporting renewable energy, which was presented with the help of the Mascaro Center for Sustainable Innovation (MCSI), which provided a travel grant. Thank you.

I also thank all the members of my research committee. I greatly appreciate the committee's patient consideration of the proposed work and their guidance in the focusing of it. Of special note is Dr. Znati who helped me to develop some of the metrics presented herein and for providing a textbook to me on discrete-event system simulation.

With great appreciation, I would like to thank the director this dissertation, Dr. Daniel Mossé, for graciously guiding, for considering of my ideas, for providing great honest feedback, for answering questions, for reading the rough drafts, for asking very insightful questions, and for improving the iterations of this and other work, which included a business competition and co-authorship of papers on retractable wind-energy harvesters. Thank you again.

1.0 THESIS AND INTRODUCTION

The thesis of this work, *A Benchmarking Framework for Sensitivity and Comparative Analysis of Energy Harvesting Strategies via Retractable Wind Energy Conversion Systems*, is that it is possible to create such a framework that meets the objectives/goals, and that rises to the challenges addressed in the following section.

1.1 OBJECTIVES/GOALS, AND CHALLENGES

1.1.1 Overarching goals and approach

The overarching goals and wishes of this research are to lower the cost of renewable energy, to protect customers from grid disturbances, and to improve stability and resiliency of the grid (defined in Section 1.2.1). (Some of those overarching outcomes, which are above the scope of this research, could be measured by counting the number of disturbances, percentage of customers affected by grid disturbances, and time-to-recovery from disturbances.)

Toward those ends, this dissertation seeks to help developers of retractable-harvester control algorithms to meet or exceed the terms of operation limitation agreements (OLAs) between harvester-hosting communities and harvester operator(s). For example, such an OLA could specify the type of harvester a community hosts and the limits in which the harvester(s) must operate.

The approach is to discuss what solutions retractable harvesters provide (reducing visual and noise pollution as well as reducing the need for long-distance transmission lines) and to contribute the following aids for developers of retractable-harvester control algorithms:

1. Metrics to gauge the performance of retractable-harvester control algorithms
2. A set of preliminary retractable-harvester control algorithms
3. Benchmarks or workloads to provide common bases on which to compare algorithm performance
4. A simulation environment in which to run the algorithms.

1.1.2 Specific goals

The specific goals of this work are to answer the following questions:

Question 1: What operation limitation agreements (OLAs) and weather conditions approximate actual field conditions of retractable-harvester control algorithms?

Question 2: Does the metric SCNetNorm (Equation 4.5) sufficiently measure how well a retractable-harvester control algorithm controls a harvester, within a operation limitation agreement (OLA) between the harvester operator and the harvester-hosting community?

Question 3: Which energy harvesting strategies (implemented by the preliminary control algorithms and simulator contributed by this work) performed best in the approximate actual field conditions found in Question 1 as measured by the metric examined in Question 2 (and another metric based on time-of-day electricity prices)?

1.1.3 Challenges

1.1.3.1 Anticipating actual agreements Challenges included the gathering of expressed preferences from residents across the U.S.A. that might reasonably be embodied into actual OLA's between harvester operators and harvester-hosting communities. Such actual OLA's would likely be derived through coalitions of stakeholders, discussions with legal teams, and harvester-operation vendors. Since, to our knowledge, since RWEHs seem to still be at the developmental and envisioned stages, actual OLA's do not yet exist. Thus, a challenge is to anticipate actual terms of such agreements.

We seek to approximate potentially actual agreements via survey results and the Pareto principle (which is demonstrated by our earlier work).

1.1.3.2 Anticipating actual wind conditions

Are there trends in windspeeds? Can past wind conditions predict future ones? What is a good way to analyze a variety of wind conditions covering at least approximately 40% of the U.S. population?

We examine those questions herein.

1.1.3.3 Anticipating actual electricity-price profiles

Are there trends in hourly electricity prices provided by an independent system operator (ISO)? Is the ratio between the morning peak and afternoon peak prices changing? We search for some trends those electricity prices herein.

1.2 WHY RETRACTABLE-ENERGY HARVESTERS

1.2.1 Self-sufficient microgrids

The U.S. Energy Information Administration (EIA) defined the *electrical power grid* as "[a] system of synchronized power providers and consumers connected by transmission and distribution lines and operated by one or more control centers" [22]. The U.S. electrical grid has a total length of high-voltage transmission lines over 150,000 miles [9, Table 2]. Transmission lines are shown in Figure 1 on page 7 in the context of the traditional grid.

It has been suggested that the electric grid is especially vulnerable to cascading failures because its organization is geographic [92], "[g]iven its age, some existing lines have to be replaced or upgraded and new lines will need to be constructed to maintain the electrical system's overall reliability," and that a challenge to improving the grid includes "[e]nsuring that the network of long-distance transmission lines reaches renewable energy generation sites where high-quality wind and solar resources are located, which are often far from areas where demand for electricity is concentrated" [100].

One way to protect customers from extended grid failures is to arrange customers (and electricity generation systems) into a small subset of the power grid which can operate independently from the grid at times. That grid subset is called a *microgrid*. "Microgrids are localized grids that can disconnect from the traditional grid to operate autonomously and help mitigate grid disturbances to strengthen grid resilience" [69].

Long-distance transmission lines have some drawbacks. New long-distance transmission lines require approval and rights-of-way, complicate construction cost recovery (a line in one state benefits another) [100]. "[T]ransmission lines needed to carry renewable energy hundreds of miles, from remote areas where it's captured to cities where it can be used, are expensive to build and sometimes opposed by people living in their path" [96]. Existing lines are easy targets for a physical attack, but "can also be repaired quickly unless there is a coordinated widespread attack. Even then, the transmission lines can be repaired almost as soon as replacement towers can be delivered" [14, Chapter 5]. "Since the last occurrence of a major geomagnetic storm in 1921, the [United States'] high voltage (HV) and extra high voltage (EHV) systems have increased in size over tenfold. Longer transmission lines that span greater surface potentials act as conductors for the geomagnetically induced current (GIC) that can devastate the electrical grid. GIC poses the risk of catastrophic damage to EHV transformers and can lead to long-term outages of large portions of the grid" [24].

One way to reduce the need for long-distance transmission lines is to harness renewable energy by wind-energy-conversion devices (e.g., wind turbines) **locally**. Harnessing local wind energy has at least one advantage over harnessing remote wind energy: it does not use long-distance transmission lines. We assume that a microgrid may also harness solar energy locally and also assume that locally harnessed solar energy might not meet all the energy needs of some microgrids.

1.2.2 Wind energy harvesters that are retractable

Since we are assuming that solar energy might not meet all the energy needs of some microgrids, we are assuming that a microgrid seeking to be self-sufficient will look also to another local, renewable energy source, wind.

However, some wind harvesting projects have been rejected because, in least in part, there were concerns that wind turbines ruin landscapes. Three examples follow:

1. A county council in Ireland rejected a wind turbine for a business park; "In considering

the scale of the turbine relative to the houses, the decision said it 'would result in a visually overscaled, unbalanced overdominant, confusing and incongruous development' within the area" [87].

2. A hillside wind farm proposal in Northern Ireland, which drew concerns from the Northern Ireland Tourist Board, was rejected partly because of its predicted noise and negative visual effects; there was also a concern about its possible electro-magnetic impact on a communications network [49].

3. "In her proposed ruling, [a Maryland public utility law judge] found that a wind farm's adverse impacts — the effect noise and shadow flickers would have 'on the esthetic of local communities on and around Dan's Mountain' — would outweigh any benefits" [1].

Opponents of those three proposed projects might have agreed to the projects if the project proposals would have offered operation limitation agreements (OLAs) that limit the visual impact of the proposed wind harvesters. Reducing the visual impact of wind harvesters is discussed in the next section.

1.2.2.1 Reducing the visual impact of wind harvesters One way to reduce the visual impact of wind harvesters is to strategically stow and deploy them. For example, at the start of a several-hour strong breeze, the harvester would be deployed and then after the strong breeze subsides into a light breeze, the harvester would be retracted. Examples of retraction-suitable wind harvesting technologies are given in Appendix A.6, and include an energy kite [48], a towered turbine [75], a buoyant turbine [4], and other devices.

1.2.2.2 Controlling the retractable harvesters In this work, we are not concerned with algorithms that control internal aspects of the harvesters, such as blade-pitch angles, but are concerned with algorithms that stow and deploy the harvesters. Such "stowing-and-retracting" control algorithms include crisp algorithms (e.g., when the windspeed is greater than 10 knots deploy and when the windspeed is less than 8 knots retract) and fuzzy algorithms [80, p. 410] (e.g., when it is "windy," deploy; when "calm," retract). Restricting deployment to windy weather is part of four standard operation limitation agreements which we introduce in Section 4.1. The algorithms can be configured via machine learning (e.g., neuro-fuzzy modeling[1]).

1.2.2.3 Time-of-day electricity pricing The U.S. Energy Information Administration (EIA) notes, "Electricity demand is usually highest in the afternoon and early evening (peak hours), and costs to provide electricity are usually higher at these times" [21]. How well can a harvester supply energy when demand is high? To help answer that question, we define a metric in Section 4.1.5 that takes into account time-of-day electricity pricing. The EIA defines *time-of-day pricing* as "[a] special electric rate feature under which the price per kilowatthour depends on the time of day" [22].

1.2.3 Benchmarks

This benchmark suite we are building provides various data for the control algorithms, similar to benchmarks in computer architecture (e.g., PARSEC [7]) which rely on simulations to determine whether a technique is efficient. So that control algorithms can train themselves to process testing data for each weather station ws, training data is provided by this work for each weather station ws of a set of 30 weather stations from across the U.S. (Appendix A.15). Some algorithms might use weather forecasts. Thus, the benchmarks include simulated forecasted windspeeds. Because some algorithms might use electricity pricing, the benchmarks include time-of-day electricity price data.

1.2.4 Why these benchmarks and metrics

No other retractable-harvester benchmark suite exists, to our knowledge, that uses OLAs that limit when and/or how much time retractable harvesters may be visible. (Please see Section 2 for related work and a literature search.) To help future retractable-harvester operators meet or exceed the terms of such OLAs, this work contributes benchmarks (Section 4.2)

[1] "[T]he term *neuro-fuzzy modeling* refers to the way of applying various learning techniques developed in the neural network literature to [learning fuzzy if-then rules for] fuzzy inference systems" [39]. "[A]n integration of neural networks and fuzzy systems can yield systems, which are capable of learning [fuzzy if-then rules] and decision making" [11].

and metrics (Section 4.1) to measure how well those algorithms perform. The contributed benchmarks and metrics can be used by designers of future retractable-harvester control algorithms to advance the state of the art of time-restricted retractable-harvester control. The metrics can also be used by harvester-hosting communities to define incentives (Section 4.2.0.1) for retractable harvester operators.

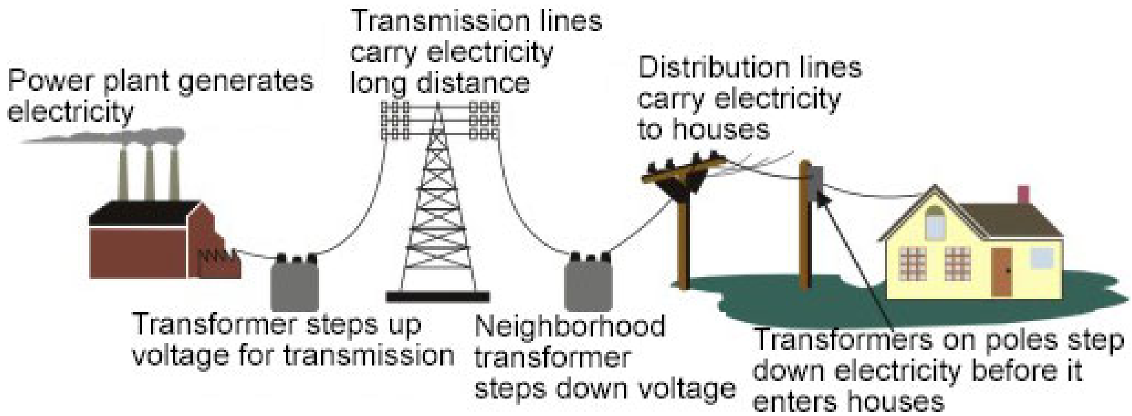

Figure 1: Electricity from power plant to home [99].

2.0 STATE OF THE ART AND ITS SHORTCOMINGS

Related to this dissertation work, which involves feeding minute-by-minute windspeeds into a retractable wind turbine simulator, are the sub-hourly wind-data sets described in Section 2.4 and the wind turbine simulators described in Section 2.5. We discuss a benchmark related to this work in Section 2.2.2. Because this dissertation work involves surveys of persons about wind energy projects, we discuss related surveys in the following section.

2.1 SURVEYS ABOUT WIND ENERGY PROJECTS

"[A] sizable literature has developed on the public perception of wind energy" [90], which includes, for example, the **National Survey of Attitudes of Wind Power Project Neighbors** [33] which "[collected] data from a broad-based and representative sample of individuals living near U.S. wind power projects. The aim was to widen the understanding of how U.S. communities are reacting to the deployment of wind turbines, and to provide insights to those communities considering wind projects" [33]. The National Survey of Attitudes of Wind Power Project Neighbors includes a project where "[in] 2015 and 2016, [data was collected] from 1,705 randomly drawn individuals living within 5 miles of all U.S. wind projects, with oversampling being done within 1 mile. The findings indicate an overall positive attitude toward the nearby turbines, including for those living even as close as $\frac{1}{2}$ mile. Roughly 8% of the population had negative attitudes within 5 miles. In an examination of a broad set of possible correlates to attitudes, it was found that neither demographic nor local wind project characteristics were significantly related. Attitudes were significantly correlated with compensation, sensory perceptions of the nearby turbines, planning process perceptions, and attitudes toward wind turbines in general" [33]. A peer-reviewed version of the survey findings [27] state "jurisdictions should consider developing procedures that ensure citizens are consulted and heard and establish benchmarks or best practices for developer interaction with communities and citizens" and "[o]ur findings might be best summed up as: '[Attitudes of wind power project neighbors correlate to] the public process, the developer, aesthetics and general wind power attitude/clean energy values' [27].

The National Survey of Attitudes of Wind Power Project Neighbors is different from our work in that The National Survey of Attitudes of Wind Power Project Neighbors did not inquire about retractable wind turbines. We are aware of no other national survey besides our own that gathered data about attitudes toward retractable-harvesters. (Note that the The National Survey of Attitudes of Wind Power Project Neighbors found that attitudes of permanently-deployed wind turbines correlate to aesthetics, which retractable-wind turbines address by stowing.)

Another example from the body of literature on the public's attitudes toward wind energy is a **regional survey** which "was developed to explore perceptions of wind energy in the [Texas] region as well as general attitudes about energy and the environment.... Findings support the view that the use of [Not-In-My-Backyard (Nimby) phenomenon] does not adequately explain the attitudes of local wind farm opposition" [90]. The regional survey found that "slightly less than the majority of respondents (47.2%) indicat[ed] that wind turbines are unattractive and a notable amount of respondents (26.7%) indicat[ed] that wind turbines are an attractive feature of the landscape" [90] and found "nearly half of all respondents (46.6%) willing to support wind farms on their property and a very small portion (13.8%) that do not support wind farms at all" [90]. The two authors of the regional study, Swofford and Slattery, ask, "So what factors are individuals basing their attitudes on? Krohn and Damborg (1999) argue that the positive acceptance of wind power is largely based on public attitudes regarding the benefits of *wind energy*, while the negative opposition of wind power is largely based on public attitudes regarding the negative aspects of *wind turbines*" [90]. Swofford and Slattery note that their findings seem to agree with Krohn and Damborg, but temper that agreement by pointing out that "there will always be additional factors influencing attitudes that are unique to locale" (e.g., earning income from turbines on private land) [90]. Swofford and Slattery suggest that the region's communities be taught about benefits of wind energy and, citing another work, state, "Communities that are in close proximity to wind farms are typically inadequately informed about wind energy projects and are often excluded from decision making and the planning process" [90].

Our standard OLAs presented in this work can help educate communities because a standard OLA can provide a solid frame of reference to which to compare a proposed project. For example, if community A is considering implementing an OLA that community B has already adopted, then community A can learn lessons about the exact OLA they are considering by learning from community B's specific experience with that same OLA. If community A is considering to enter into an OLA that is not standard, then it seems less likely that community A would find another community that has entered into that non-standard OLA.

Another advantage of communities considering standard OLAs over non-standard is that when communities ask for bids from various vendors on standard OLAs, then the vendors will develop bids that can potentially be re-used, and possibly develop price lists posted to websites that communities can compare without asking for bids. (That comparison shopping might result in lower renewable energy prices.)

2.2 BENCHMARKS INVOLVING WIND TURBINES

2.2.1 Benchmarking the control of rotor speed

Of the thirteen benchmark problems for "non-linear system identification and control" that Kroll and Schulte offer [45], one benchmark problem is for a wind turbine [45, Section 3.12], which involves keeping a modeled turbine's rotor speed and generator torque within limits. "The control performance is assessed for two operating ranges: The partial and the full load range. In the partial load range the rotor speed is regulated to obtain optimum energy efficiency.... In the full load range the requirement change to a set-point controller to limit the energy production even with increasing wind speed and keep also the mechanical loads and pitch activity to a minimum" [45]. Kroll and Schulte refer to using a "stepgust simulation" and a "stochastic wind field with given mean wind speeds and turbulence intensities" to assess "extreme loads" and "reduction of fatigue loads," respectively [45]. Their benchmark problem is concerned with controlling a turbine's internals that are modeled by a reference 5 GW offshore turbine model provided by the National Renewable Energy Laboratory (NREL).

Such internal modeling is beyond the scope of our benchmark suite. As we explain in Section A.3, we model an envisioned retractable harvester model via a power curve (Section A.3.1) and functions that calculate how much energy the retractable harvester uses to deploy and retract and a function that models the harvester's deployment speed and degree.

2.2.2 Benchmarking stochastic control of energy storage devices

An algorithm and benchmark by Salas and Powell was published in 2017 [81]. Salas and Powell "present and benchmark an approximate dynamic programming algorithm that is capable of designing near-optimal control policies for a portfolio of heterogenous storage devices in a time-dependent environment, where wind supply, demand, and electricity prices may evolve stochastically."

"[Salas and Powell's] model works on a time scale of five minutes, which means 288 time periods over a daily cycle. The time scale is fixed by the grid operator, PJM, which updates electricity prices every five minutes" [81]. "[They] benchmark against optimal policies for the full problem" [81]. Salas and Powell "benchmark against optimal policies... on deterministic and stochastic time-dependent problems for a one-device system, which include the presence of exogenous information such as wind, prices, and demand" and "set forth this set of problems as a library that may be easily used to test the performance of other algorithms" [81]. Salas and Powell's "test problems each consist of 2,000 periods" (spanning approximately one week). "For the deterministic benchmarks, [Salas and Powell] designed... test problems... where the electricity prices, renewable energy and energy demand evolve deterministically over time. [Salas and Powell] consider four different dynamics: sinusoidal, constant, step, or fluctuating with no particular pattern" [81].

Instead of a five-minute time scale/step, the work presented herein uses a finer one-minute time step. Our windspeed data is minute-by-minute. Where Salas and Powell's model uses a finer resolution than we do is electricity prices. Salas and Powell model five-minute electricity prices. We provide 1-hour electricity prices and 1-minute wind data for 11 years. Salas and Powell's test problems each span approximately one week. Salas and Powell's deterministic benchmarks are determined by functions ("sinusoidal, constant, step, or fluctuating"). Conversely, our benchmarks are derived from actual minute-by-minute weather data and from actual hour-by-hour electricity prices over 11 years. (An advantage of using actual historical conditions instead of simplified functions is that the simplifications might not contain information that is important during actual field operation. That information is more likely to be in our benchmarks because our benchmarks are closer to actual field conditions. The actual conditions would probably include a real-time data feed from wind speed sensors.)

In a related paper [40], Salas and Powell along with Jiang[1] et al. mention that energy storage and inventory management are closely related [40]. For example, Harsha and Dahleh's work on energy storage [32] refers to Federgruen and Yang's paper on inventory control [25]. We note that inventory management is related to meeting delivery deadlines, which is related to task scheduling in computer science. Hence, we direct the reader who is interested in energy storage to also consult inventory management and computer-task scheduling research. (We indirectly address energy storage via a metric (Equation 4.7 on page 58) that measures how often a retractable-harvester control algorithm harnesses energy when energy is and is not needed.)

2.3 STOCHASTIC CONTROL ALGORITHMS

Powell, in an invited review [76], "provide[s] a modeling framework [for stochastic optimization] with which we can create a single canonical model that describes all of the [following] problems" [76]: "Decision trees… Stochastic search… Optimal stopping… Optimal control… Markov decision processes… Approximate/adaptive/neuro-dynamic programming…

[1] An interesting fact is that Dr. Jiang is an assistant professor in the Department of Industrial Engineering here at the University of Pittsburgh [89].

Reinforcement learning…Online algorithms…Model predictive control…Stochastic programming…Robust optimization…Ranking and selection…Simulation optimization… Multiarmed bandit problems…[and] Partially observable Markov decision processes" [76]. The problems involve various fields including "business, science, engineering, economics and finance, health and transportation" [76].

Our benchmarks differ from stochastic problems in that our benchmarks provide deterministic problems. In addition to deterministic problems, we provide a Weibull windspeed distribution for each of 30 weather stations that can be used to develop stochastic solutions within Powell's canonical model.

2.4 SUB-HOURLY WIND DATA

2.4.1 Automated Surface Observing System (ASOS) data set DSI 6405

The Automated Surface Observing System (ASOS) data set DSI 6405 provides an average of windspeeds of the previous two minutes [58, p. 3] nearly every minute. An advantage of using DSI 6405 is that it is freely available. However, it has some quality issues such as missing minutes and repeated timestamps. We describe how we derived the training and test data from the minute-by-minute ASOS data set DSI 6405 in Appendix A.4.1 and Appendix A.7.1.

2.4.2 Minute-by-minute windspeed data from NREL's M2 tower

The National Renewable Energy Laboratory (NREL) provides minute-by-minute windspeed and direction data from its M2 tower in Colorado measured at six heights ranging from 2 to 80 meters [38].

A drawback of using data only from Colorado is that Colorado weather does not necessarily represent weather experienced by major population centers in coastal states (e.g., hurricanes). We solve that drawback by using, as noted later in this document, data from 30 weather stations from across the U.S. (Appendix A.15) instead of just one location.

2.4.3 Fifteen-minute windspeed data from MADIS

The Meteorological Assimilation Data Ingest System (MADIS) [61] quality-checks windspeed data. However, according to MADIS Support, MADIS-checked one-minute data is not available and MADIS runs every five minutes and preserves 15-minute values.

An advantage of using 1-minute windspeeds instead of 15-minute values is that 1-minute windspeeds better simulate a real-time data feed that a control algorithm might use to control a retractable harvester. As mentioned above, the data feed would probably include real-time anemometer data. Each anemometer would most likely be sending data at one-minute intervals or less.

2.5 WIND TURBINE SIMULATORS

2.5.1 WISDEM

"The Wind-Plant Integrated System Design and Engineering Model (WISDEM) is a set of models for assessing overall wind plant cost of energy (COE)" [67]. Documentation for WISDEM implies that WISDEM uses a Weibull distribution to model windspeeds instead of historical minute-by-minute windspeeds as evidenced by the following documented WISDEM input parameters [68], where each parameter is set to an example annotated value having a description field:

```
wind_speed_50m = Float(8.35, units = 'm/s', iotype='in', desc='mean
    annual wind speed at 50 m height', group='Plant_AEP')
weibull_k= Float(2.1, iotype='in', desc = 'weibull shape factor for
    annual wind speed distribution', group='Plant_AEP')
```

Wiebull distributions may not fit actual distributions of windspeeds exactly. Instead of simulating windspeeds, we use actual historical minute-by-minute windspeed data (and interpolate gaps) as noted later in this work (Section 4.2).

2.5.2 HOMER

HOMER is an acronym for Hybrid Optimization of Multiple Energy Resources [34]. HOMER software products are used to "[e]valuate design options for both off-grid and grid-connected power systems for remote, stand-alone, and distributed generation applications" [64].

HOMER allows the inputting of wind data in a time series, but only one year's worth. "HOMER [Pro] can accept one year of data at timesteps down to a minute" [36].

A disadvantage of using one year of wind data instead of multiple years is that it takes more than one year of wind data to detect seasonal trends. The current version of our simulator (Section 3.2.1) can handle 11 years (and with relatively slight modification can handle many more). We describe the 11 years of data we provide in Section 4.2.

3.0 MODES OF OPERATION DEFINED BY OPERATIONAL LIMITATION AGREEMENTS (OLA'S)

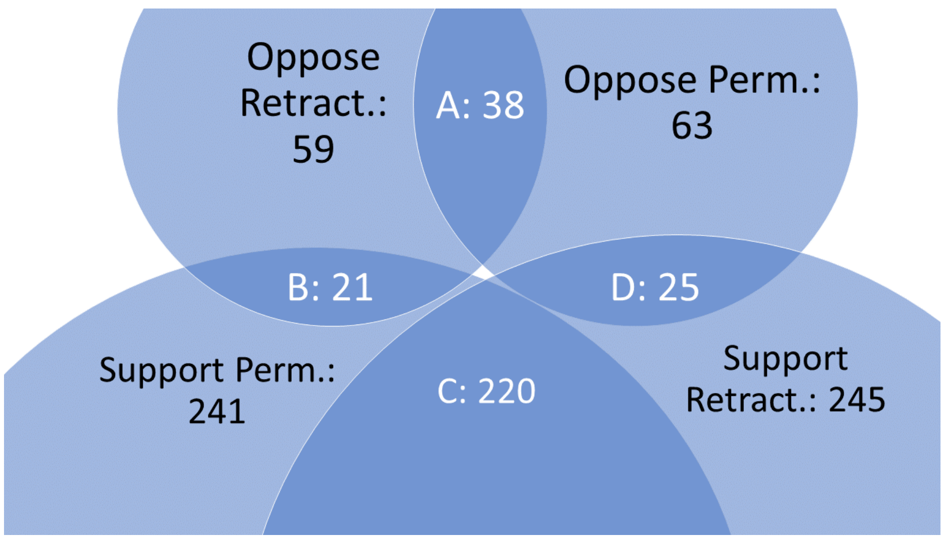

Figure 2: Overlapping groups of survey respondents

3.1 DERIVING POTENTIAL OPERATION LIMITATION AGREEMENTS

3.1.1 Survey results

To derive operation limitation agreements (OLAs) between harvesting-hosting communities and harvester operators, we obtained 304 interview results from persons across the USA via a survey company[1]. The full results of our five question survey are given in Appendix A.9.

[1]Survata, Inc. (www.survata.com)

(Because none of the questions mention the financial benefits a harvester-hosting community might receive, we are assuming that those anticipated benefits did not obfuscate the preferences of the interviewees.) Results are listed here in parentheses following each question and are represented in Figure 2 on the preceding page, where each of the four major groups below is represented by a circle (The figure is cropped intentionally). Each circle's diameter is determined by the size of the group, which is given in Figure 2 in black font. Intersections of the circles, which are labeled in white font, represent the four intersections, A, B, C, and D, described here:

1. Would you support or oppose a wind turbine project if you could always see the installed turbines from where you live?

   (a) Support (241/304 or 79%) (Of the 241 that support permanently visible harvesters, 220 also support retractable turbines under certain conditions: We label the "support-both" intersection/group as "C" in Figure 2.)

   (b) Oppose (63/304 or 21%) (Of the 63 that oppose permanently visible harvesters, 38 also oppose retractable turbines: We label the "oppose-both" intersection/group as "A" in Figure 2. Of the 38 persons in Group A, five persons explicitly commented negatively about the appearance or visual impact of wind turbines. It is possible that those five persons might compromise by accepting retractable wind turbines that are highly limited in visibility if those persons received financial benefits. Four members of Group A made comments indicating a lack of knowledge about retractable wind turbines (e.g., "i just don't understand their purpose." Those four members might assent if they were to learn more about turbines that are retractable. At least three members of Group A indicated that they oppose both types of turbines because better options exist (e.g., "Solar is better" and "Better options.")).

2. Would you support or oppose a wind turbine project viewable from where you live that uses only retractable wind turbines? Retractable wind turbines are able to be deployed and retracted when specified. E.g., deploy when "windy" and retract when "calm."

   (a) Support retractable wind turbines under certain conditions (245/304 or 81%) (Of the 245 that support retractable turbines under certain conditions, 25 oppose permanent deployment. We label the "oppose-permanent-and-support-retractable" group as "D" in Figure 2.)

(b) Oppose retractable wind turbines (59/304 or 19%) (Of the 59 who oppose retractable wind turbines, 21 support permanent deployment. We label the "support-permanent-and-oppose-retractable" group as "B" in 2. One of the members of Group B seems to actually make a case for retractable turbines by commenting, "If they are retractable, they will kill fewer birds. . . .")

3. When should the retractable wind turbines be retracted and hidden? Check all that apply. (Please note that 220 of the 245 persons responding to this question also support turbines that are permanently deployed. Statistics for the remaining 25 person are enclosed below in curly braces. The statistics for both groups shown graphically in Figure 3.)

   (a) When it's not windy (138/245 or 56%){9/25 or 36%}
   (b) When the month is not March (8/245 or 3%) {4/25 or 4%}
   (c) During every morning (13/245 or 5%) {3/25 or 12%}
   (d) During every afternoon (13/245 or 5%) {4/25 or 16%}
   (e) During every night (29/245 or 12%) {7/25 or 28%}
   (f) When birds are migrating (100/245 or 41%) {6/25 or 24%}
   (g) During every weekend (12/245 or 5%) {4/25 or 16%}
   (h) After it has been visible a certain proportion the month (14/25 or 6%) {2/25 or 8%}
   (i) Other (15/245 or 6%) {0/25 or 0% }(Three of the answers were related to high winds.[2]) The fifteen respondents left these short-answers (and these comments in response to Question 5 below):

      1. "Windy days storms" (comment: "No comments")

[2]Because some interviewees do not want to see turbines during high winds, we included the restriction "Windspeed is TOO HIGH" in OLAs 1-4, which are all windspeeds above the harvester's cut-out threshold as defined by the harvester's power curve (a power curve is shown in Figure 30)

2. "Never" (comment: "None")
3. "Doesn't matter" (comment: "None at this time")
4. "Doing [sic] a hurricane" (comment: "Only during a hurricane, it would be scary seeing them move fast because of the strong winds.")
5. "During a storm" (comment: "no comment")
6. "No need to retract them, we need wind turbines 24/7" (comment: "I have no opposition against any sort of wind turbines. We should use them en masse every day.")
7. "No idea" (comment: "Don't know much about turbines")
8. "never" (comment: "we should always use the turbines, they should never be hidden")
9. "not sure" (comment: "no comment")
10. "whenever the turbine doesn't need to be in use" (comment: "Don't feel one way or the other. If turbines create energy at a low cost, wouldn't matter if I could see it or not.")
11. "weather" (comment: "do not have one")
12. "who cares, turbines are good" (comment: "nope")
13. "don't know" (comment: "none")
14. "it doesn't bother me either way" (comment: "clean energy is the future....it has to be")
15. "doesn't matter" (comment: "I don't oppose any forms of turbines")

4. After what percentage of the month that the wind turbine has been visible should the turbine be hidden? (Of the 14 interviewees that indicated "After it has been visible a certain proportion the month" excluding the person that answered 0% of the month[3], the average is 49% of the month. In the next section, we derive OLAs from

[3]We excluded that 0%-of-the-month response because 0%-of-the-month could indicate that the respondent did not understand the question. To Question 1 of the survey, which is "Would you support or oppose a wind turbine project if you could always see the installed turbines from where you live?" (Table 24), the respondent indicated support. Thus, the respondent supports permanently deployed harvesters, which are

the survey results presented above. Since the 14 interviewees are only 6% of the 245 interviewees who support retractable wind turbines, we, in the next section, do *not* restrict deployment on the basis of how long the harvester has been visible in the two OLAs that we named "basic," but the basic OLAs still restrict deployment on the basis of windspeed and quiet hours.) { The two "oppose-permanent-support-retractable', Group-D, persons that indicated "After it has been visible a certain proportion the month" answered 40% and 10%, which average 25%.}

5. If you have any comments, please share them here. (Especially, if you oppose all types of wind turbines including retractable wind turbines, why?)

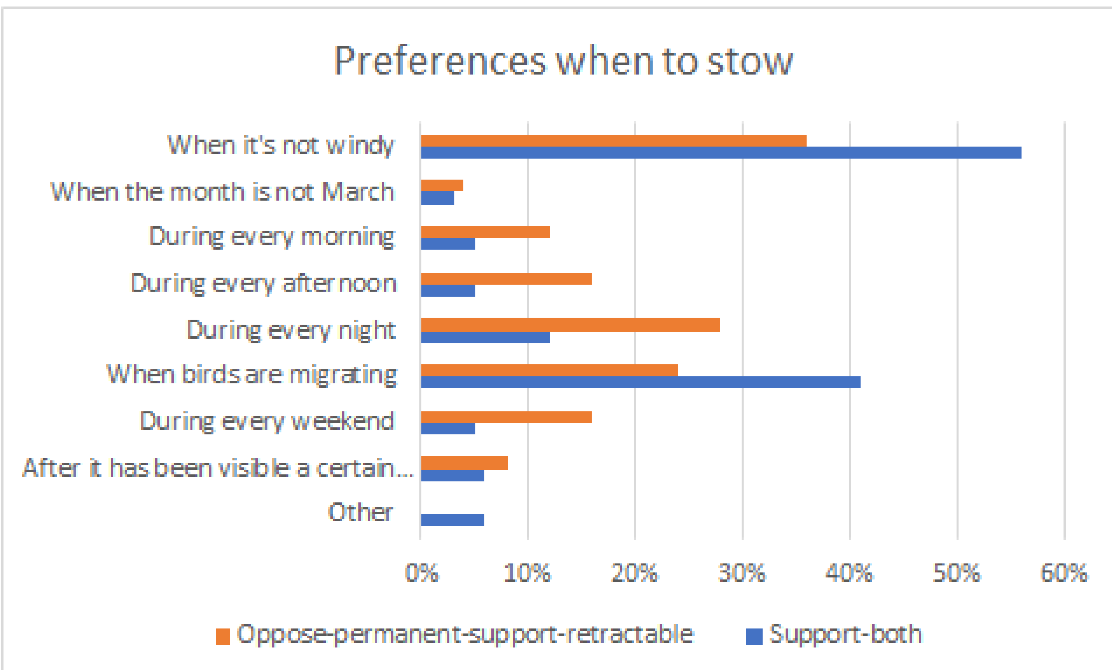

Figure 3: Preferences of those who support retractable harvesters.

deployed 100% of the month. But, 0% of the month was the respondent's answer.

3.1.2 Using a threshold to derive OLA conditions

The survey results can lead to the following OLA portion where 10% or more[4] of survey interviewees in each category (if that group of interviewees has a plurality) chose the following conditions to retract or keep the harvester hidden:

3.1.2.1 Requirement 1 of 2: "When it's not windy" (138/245 or 56%). Fifty-six percent of respondents chose to retract the harvester when the weather is not windy. To define when the weather is not windy, we first find a windspeed threshold for each station. After we find the windspeed threshold, we allow each algorithm to choose the size of the window of the running windspeed average for each month. Prior to each month, the harvester operator should give the hosting community the monthly running-windspeed-average-window size (defined in Section 3.2.1) so that the hosting community can measure compliance with the when-it's-not-windy requirement. When the running windspeed average is one (1) knot below the windspeed threshold defined in the first step, we consider the weather to be not windy. (Conversely, when the running windspeed average is at or above the windspeed threshold defined in the first step, we deem the weather to be windy.)

Note: The "When it's not windy" requirement prevents algorithms from deploying until the running average windspeed reaches the lowest windspeed deemed to be windy k_{ws} at the pertinent weather station ws and prevents algorithms from *remaining deployed* when the running average windspeed drops below the lowest windspeed deemed to be windy less 1 knot: $k_{ws} - 1$. (We explore offsets greater than the 1-knot offset in Section 4.6.4 for the Aging algorithms for OLAs 3 and 4.)

Finding the windspeed threshold for each station We use the method detailed in Appendix A.2.1. There, a method is described to create a membership function that assigns a membership value (inclusively ranging from 0 to 1) in the fuzzy set NOT WINDY AT KBOS to each windspeed in a universe of discourse. (KBOS refers to a specific weather station at Boston's Logan International Airport.) For each weather station ws, we use the

[4] A community might choose x to be 10% because it might be useful to build a coalition and/or to be neighborly. A threshold x% that is not greater than 50% can make sense where no other equally-or-larger percentage of the community opposes the x%. For example, we are assuming that the group wanting to stow harvesters during every night is larger than a group wanting to deploy harvesters every night.

training data in our benchmark specific to that weather station to create its fuzzy set NOT WINDY AT ws. (Thus, in the benchmark suite, we provide 30 membership functions, one for each weather station.)

Then, from that fuzzy set, we create a lambda-cut set. (A general definition of a lambda-cut set can be found in Ross [80].) In this case, the lambda-cut set is the set of all windspeeds in the fuzzy set NOT WINDY having membership values of λ or higher. We chose λ to be 0.9 because lower values did not perform well in initial testing (during relatively early development). In Appendix A.2.4, we describe how to create a lambda-cut set and list the "Lowest Windspeed Deemed to Be Windy" for each station when λ is 0.9. By choosing $\lambda = 0.9$, each "Lowest Windspeed Deemed to Be Windy" happens to be either a Gentle Breeze or Moderate Breeze on the Beaufort Scale, as shown Table 11 in Appendix A.2.4. In Appendix A.2.4, five other values for lambdas are briefly explored. In other words, we determine how lambda affects the range of the "Lowest Windspeed Deemed to Be Windy" in a brief sensitivity analysis on lambda.

There are other ways to define the NOT_WINDY fuzzy set for each station besides relativizing each set to historical windspeeds at each station. For example, another approach is to derive one NOT_WINDY fuzzy set to be shared by all stations from an absolute scale such as the Beaufort Scale [60]. An advantage of using an absolute scale that is that only one membership function is required for all stations. A disadvantage of a one-threshold-fits-all approach is that the universal membership function is not embedded with historical windspeed information that is specifically local. Thus, if the local historical information is useful, the algorithm would need to extract that local historical information itself during training. We adopt the individualized membership functions, one for each station, because it is embedded with historical windspeed information.

3.1.2.2 Requirement 2 of 2: "During every night" (29/245 or 12%). Twelve percent of survey respondents chose to stow the harvester during every night[5]. Because noise

[5]We are interpreting "During every night" to be noise-restricted hours as defined by some municipalities (Appendix A.10). Survey results might have been different if instead of the option "During every night," we had used "During quiet hours" since Schwarz notes that "[s]elf-reports of... attitudes are strongly influenced by features of the research instrument, including question wording, format, and context" [83]. Although such an investigation is outside the scope of this present work, we are assuming that possible future revisions

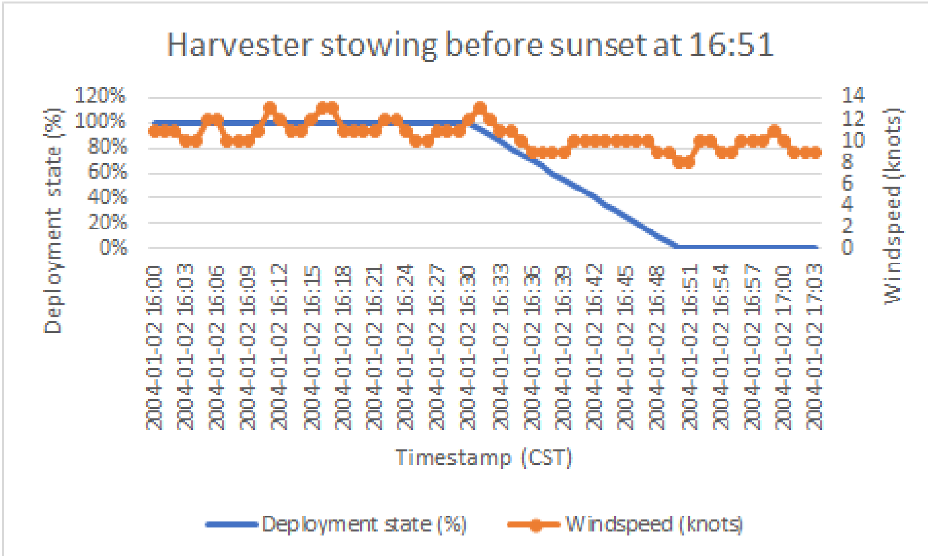

Figure 4: An envisioned harvester stowing before sunset at St. Louis at 16:51 regardless of windspeed

was a factor in some opposition to proposed wind-power projects [49] [1], we are assuming that at least some of the respondents who chose to keep harvesters hidden every night did so because they were concerned about noise. Thus, for each station, we derived definitions of *every night* or *quiet hours* from each station's corresponding city's noise ordinance's definition of nighttime. The derived *quiet hours* are in Appendix A.10. For example, quiet hours derived from Boston's and Minneapolis's municipal codes begin at 6 p.m. and end at 7 a.m. each weekday and quiet hours include the entire weekend. Pittsburgh's quiet hours start at 10 p.m. and end at 7 a.m. everyday. St. Louis's quiet hours begin at sunset and end at 6 a.m. and include the entire day of Sunday. Thus, a harvester in St. Louis must be stowed before sunset regardless of the windspeed-deployment are retraction thresholds (which, in this case, are 7 knots and 6 knots, respectively) if the harvester is observing the "During every night" restriction, as illustrated in Figure 4.

of the benchmarks presented herein will incorporate new survey data gathered in the context of specific actual or proposed retractable-harvester installations. Thus, we are proposing that future surveys ask some questions repeatedly using various "wording[s], format[s], and context[s]."

3.1.2.3 Out-of-scope requirement: "When birds are migrating" (100/245 or 41%).

Forty-one percent of respondents chose to retract the harvester when birds are migrating. Although bird migration is outside the scope of this present work, our finding the relatively strong concern for birds among interviewees provides a motivation to consider adding bird migration indicators or data to a future version of this benchmark suite.

Bird migration indicators or data could be based on weather radar. Weather radar can detect birds [29]. A 2017 report states, "Recent changes in the data delivery and processing timescale for [NEXRAD weather surveillance network] Level II information via Amazon Web Services (`https://aws.amazon.com/noaa-big-data/nexrad/`) has greatly increased the potential of using the level II product for responsive, near real-time analyses" [85]. We envision that real-time bird migration data feeds will someday be provided by organizations specializing in bird migration forecasting (e.g., the BirdCast project[6], which is associated with the Cornell [University] Lab of Ornithology).

3.1.3 Sets of operational limitations

From survey results, previous work, and the Pareto principle, we derive the following three sets of operational limitations:

Set Alpha A Basic allows a harvester to remain deployed 100% of month as long as the following conditions are met:

- when windy (but not during extremely windy conditions (Appendix A.3.2)) and
- during daytime.

Set Beta B Operational-constraint set B is the same as set A except that set B has the additional requirement that the harvester must not be visible more than 8760 minutes of each month, approximately 20%. A reason that we chose 20% is that it is lower percentage of the total time and that we found in an earlier work that "80% of the wind available over our two-month sample period can be extracted by wind harvesters deployed merely 20% of that time" [51] and is an example of the Pareto principle.

[6] http://birdcast.info

Set Gamma Γ Transition-limited OLAs 5 and 6 are the same as OLAs 3 and 4, respectively, except that OLAs 5 and 6 limit the number of state-transitions to two per month. The two states are the deployed state and the stowed state. Thus, the two possible state-transitions are stowed-to-deployed state-transition and the deployed-to-stowed state-transition. A reason to choose two state-transitions per month instead of unlimited state-transitions is that it greatly simplifies the determination of the "optimum" deployment and retraction timestamps. Also, two state-transitions describe a single wind-harvesting event: A harvester deploys, harvests energy (ideally), and then retracts. A single duration-limited (e.g., limited to approximately 20% of the month) wind-harvesting event per month is a relatively straightforward allowance to convey to potential harvesting-hosting communities: the harvester will deploy once per month and/or retract once per month.

Set Gamma Γ Set Γ has the following operational constraints:

- not during extremely windy conditions (Appendix A.3.2)),
- not visible more than more than 8760 minutes (20%) of each month, and
- not more than two (or three in case harvester needs to lower during extremely windy conditions) state changes (i.e., raises/lowers) each month

Table 1 shows the attributes of each of the six standard OLAs, where the dashed-line between the odd-numbered and even-numbered OLAs is meant to emphasize that the only difference between the odd-numbered and even-numbered OLAs is the reward/penalty function. Odd-numbered and even-numbered OLAs pass SCNetNorm() and SCMQNetNorm(), respectively, to I().

Table 1: The standard OLAs

OLA Numbers / Attribute	1. & 2. Basic OLAs	3. & 4. Intermediate OLAs	5. & 6. Transition-limited OLAs

Table 1: (continued)

Number of times harvester may change states each month	Unlimited	Unlimited	2
The weather is not windy	Yes	Yes	No
Nightly visibility ban is in effect	Yes	Yes	No
Windspeed is too high (i.e., windspeed is above cut-out threshold)	Yes	Yes	No
Harvester has exhausted its visibility allocation, which is x minutes of each month	$x = 100\%$	$x = 8760\,\text{min}$	$x = 8760\,\text{min}$
month(s) of year	all	all	all
Version of OLA that uses $I(SCNetNorm(a, wrk), b)$ (Defined in Section 4.2.0.1) as reward/penalty function where $b \in [1, 100]$ is chosen by each harvester-hosting community	OLA 1	OLA 3	OLA 5
Version of OLA that uses $I(SCMQNetNorm(a, wrk), c))$ (Defined in Section 4.2.0.1) as reward/penalty function where $c \in [1, 100]$ is chosen by each harvester-hosting community	OLA 2*	OLA 4*	OLA 6*

*OLAs 2, 4, and 6 use a different reward/penalty function than OLAs 1, 3, and 5.

3.2 VARIOUS ENERGY HARVESTING STRATEGIES; INTRODUCING THE ALGORITHMS

In this section, we develop energy harvesting strategies:

- Static: These algorithms do not change settings (e.g., deployment and retraction thresholds) while running.
- Dynamic: The algorithms may change settings during the simulation:
  - Aging (with and without weather prediction), which may lower the windspeed retraction threshold as the month progresses,
  - Fuzzy-Crisp Hybrid (with and without weather prediction), which uses crisp logic to determine when to retract to ensure that the algorithm complies with its controlling set of operational limitations (e.g., stow during quiet hours $\in A$) and which uses fuzzy logic to determine when to deploy,

  where each of the dynamic algorithms have two operating modes:

  - Non-predictive mode: At each time-step of the simulation, the simulator provides to the algorithm current, but not predicted, wind data from the sequence of windspeeds in the benchmark that the simulator is using.
  - Predictive mode: At each time-step t of the simulation, the simulator provides to the algorithm current and simulated predicted day-ahead windspeeds from a sequence, hereafter called S, of windspeeds in the benchmark. Specifically, for each minute-by-minute time-step t of the simulator, the simulator will pass to the algorithm the windspeed $s[t] \in S$ and the distorted windspeed $d(x, s[t+x])$ where

    • $s[t+x] \in S$ and

    • $x = 1440$ minutes is the time-horizon of the prediction (i.e., $x = 1440$ minutes or one day beyond time-step t) and where

    • function $d()$ distorts each future windspeed to simulate a windspeed-prediction accuracy[7].

[7] In Section 4.2.0.3, we describe how we simulate the prediction of windspeeds.

3.2.1 The simulator: configuring and running the control algorithms

We are providing an environment in which to train and to test the control algorithms. The environment consists of input data and a Java program. In Figure 5, the Java program is represented by the rectangle labeled "Simulator" and the input data are represented by the horizontal arrows pointing to the rectangle. The arrow having no fill and labeled "Algorithm-configuration Arguments →" represents settings or values which can affect the algorithm's performance (e.g., window size of windspeed running average, windspeed deployment threshold). (We discuss a way to find those values in the next paragraph.) The output of the simulator is labeled "→ Metrics. . . ."

In this section, we describe parameters and values with which we experiment. The purpose of the settings are to tune the performance of the algorithm that the simulator is running. To that end, we explore sets of parameter values. The values are described and exploration results are referenced via Table 4 on page 84 and Table 9 on page 105 in the context of different explorations. The tables also include a purpose for each exploration.

Window size of windspeed running average: Recall that the benchmark suite provides windspeed data that is minute-by-minute (which is labeled "Multi-year Minute-by-Minute Windspeeds" in Figure 5) and that we interpolated any gaps so that no minutes are missing. At every minute, the file contains a windspeed that we call a *sample*. The window size m of the windspeed running average is the number of samples (which is identical to the number of minutes because we interpolated any gaps) that the algorithm sums to calculate a running average of windspeed samples at each timestep of the simulation.

Averaging the sequence of windspeeds mitigates momentary winspeed changes (e.g., spikes). If a control algorithm does not filter out spikes, the algorithm could momentarily unstow a fully retracted harvester, wasting visibility time. Likewise, if a control algorithm does not filter out transient dips in windspeeds, the control algorithm could momentarily retract partially a fully deployed harvester, wasting opportunities to harvest.

The following is how the simulator uses the running average size m, where m is the size of the moving window measured in minutes. The simulator reads samples one at a time each minute. Let t_{now} and t_{prev} be the minute timestamps of the samples that the simulator would read most recently and next-to-most recently, respectively. The simulator inserts the most recently read samples into a set of samples having timestamps in the range $(t_{now} - m, t_{now}]$.

Mathematically, the running average having window size m minutes can be defined as follows: Let t_i be the minute timestamp of each windspeed w_i for $i = 0, 1, 2, \ldots, n$ where n tends toward infinity. Let $t_{now} = \max(t_i)$. Define the set

$$M = \{w_i \mid i = 0, 1, 2, \ldots, (m-1) \text{ and } (t_i > (t_{now} - m))\}.$$

Hence, our running average having window size m is defined as

$$\frac{\sum_{j=1}^{|M|} w_j \in M}{|M|}.$$

Values: Tables 4 and 9 show the various sequences of window sizes of the running windspeed average as part of various explorations, including the following examples:

- Exploration 1: We tested values in the sequence [1 (step 30) 121], which causes 5 iterations, for all 30 stations, for all algorithms, for all OLAs. The window size sequence starts at 1 minute, ends at 2 hours, with half-hour increments. The relatively large step size was conducive to completing a level of testing that is spread across all 30 stations, all algorithms, and all OLAs within a previous schedule.
- Exploration 2: For all 30 stations, one algorithm, and one OLA (OLA 3), we extended and tightened the sequence: [1 (step 1) 361]. We extended the sequence to 361 to determine if the upper limit of Exploration 1, i.e., 121 minutes, was reasonable. If the training routine were to often find that values greater than 121 result in the Aging algorithm's best NetNorm or MQNetNorm scores, then we would suggest that the training routine should extend its search space beyond Exploration 1's upper limit of 121. (Recall that OLA 3 limits a harvester's visibility to approximately 20% of each month. Thus, unfiltered spikes might have a more detrimental effect on OLA 3 than OLA 1, which allows the harvester to be visible 100% of non-quiet hours if windspeeds are within a specified range.)

Windspeed deployment threshold or y-intercept: The windspeed deployment threshold argument, which has a unit of knots, is passed to the algorithm Static. Static does not change the argument's value during the entire simulation. The windspeed deployment threshold tells Static when to deploy if other conditions are being satisfied (e.g., the harvester has time remaining in its monthly visibility allotment.) The y-intercept argument is used by the algorithm Aging as part of a linear equation dependent upon how much time remains in the month. The training routine may find a different y-intercept for each month of the year.

Values: We explored different window sizes of windspeed deployment thresholds and y-intercepts in multiple explorations, as shown in Table 4 and Table 9.

Fuzzy-Crisp Hybrid's deployment membership value in fuzzy set: As stated elsewhere, the Fuzzy-Crisp Hybrid algorithm uses crisp code to retract and fuzzy code to deploy. When the fuzzy-code-produced membership value in a combined fuzzy set reaches the value specified by this parameter, the fuzzy code directs the harvest to deploy (but the fuzzy code is overridden by the crisp code if the harvester must remain retracted.)

Values: Please see Tables 4 and 9.

Lambda (λ): As stated in Section 3.1.2.1, the lambda-cut set of the fuzzy set NOT WINDY is the set of all windspeeds having membership values of λ or higher in the fuzzy set NOT WINDY. The membership function that we are using assigns to low windspeeds high membership values in NOT WINDY and assigns to high windspeeds low membership values. As windspeed increases, the membership value in NOT WINDY decreases. (An example of a mapping of windspeeds to membership values is plotted in Appendix A.2.1.) The lambda cut partitions all windspeeds into two crisp subsets: CRISP NOT WINDY and CRISP WINDY. The x-axis location of the boundary between the set CRISP NOT WINDY and the set CRISP WINDY decreases as λ increases. Thus, as λ increases, the lowest windspeed deemed to be windy decreases.

Values: Please see Tables 4 and 9.

Retraction Threshold Difference: Section 4.6.4 describes a sensitivity analysis we did on the retractable threshold difference.

Values: Please see Tables 4 and 9.

Information about the other algorithms can be found in Appendix 3.4. Some algorithms have arguments (e.g., window size of windspeed running average, y-intercept of linear function that maps time remaining in month to windspeed deployment threshold) that are arrays containing settings for each month of the year.

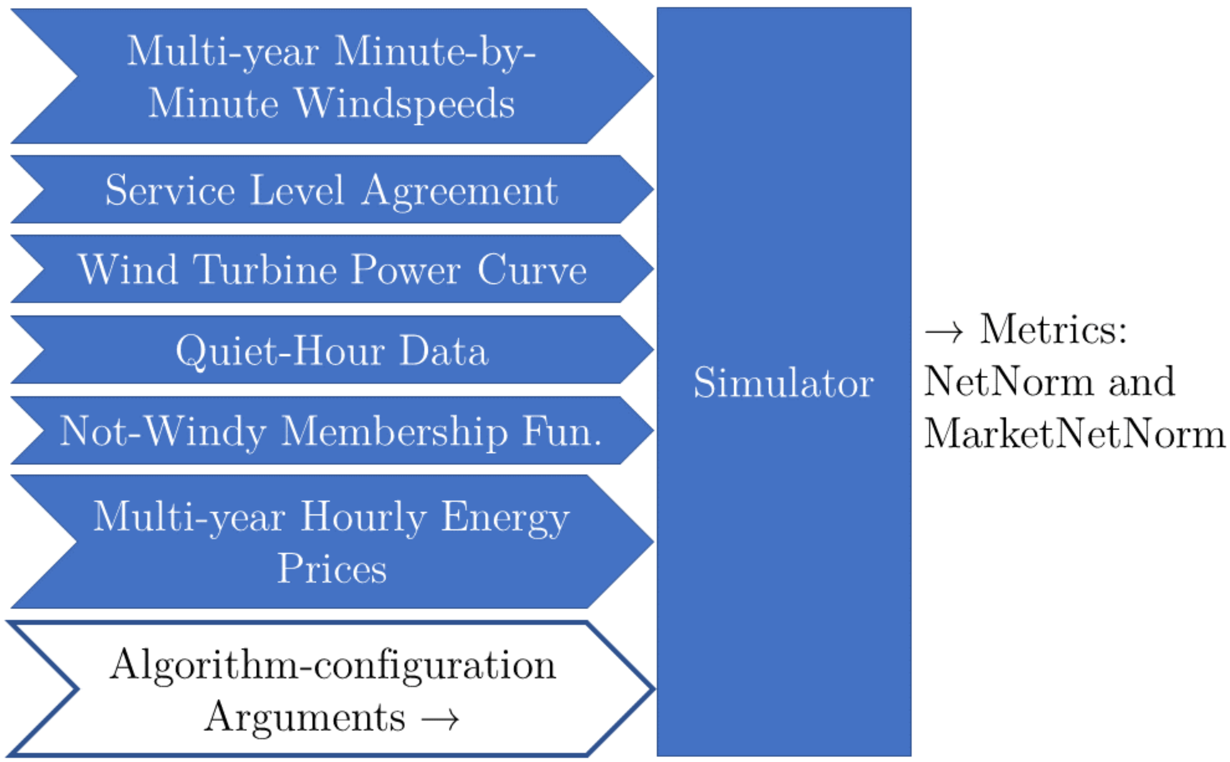

Figure 5: Environment in which to train and to test the control algorithms.

A way to find algorithm-configuration arguments or values is to search a configuration space. In other words, a way to train is to search. The process of searching a configuration space is represented on the left-hand side of Figure **??**. Searching might involve using nested loops to find sets of configuration values that maximize the output metrics during training. Thus, searching the configuration space might involve running the simulator hundreds of iterations.

The center of Figure **??** represents that training (which we implemented by searching) found two sets of configuration values: one set maximized the MQNetNorm metric; the other set maximized NetNorm. (Both sets might happen to be identical.)

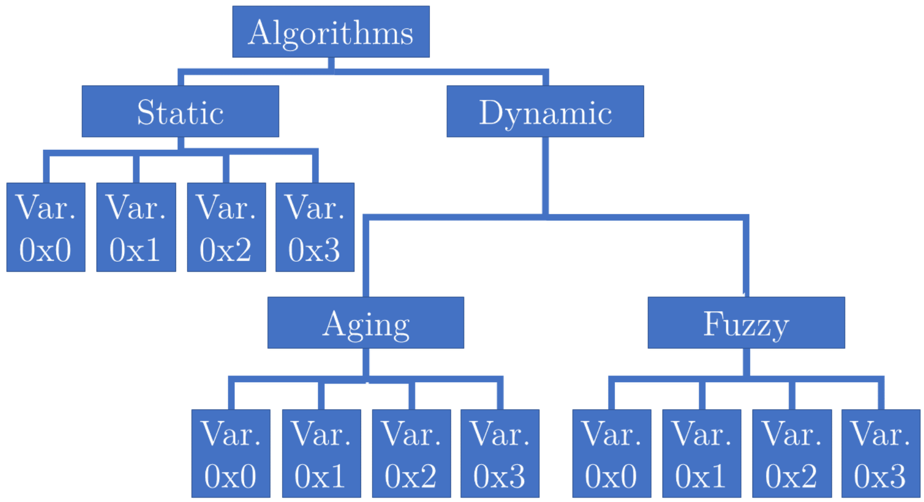

Figure 6: Categories of algorithms we wrote and tested and their variants ("Var.")

The right-hand side of Figure **??** represents the test phase, in which the simulator is run only twice. One run uses the configuration values that maximized the MQNetNorm metric during training. The second run uses the configuration values that maximized NetNorm.

We have described the environment in which to run the algorithms. We discussed training (via searching). Let us now turn to the goal of the algorithms and give an overview of the three algorithm categories that we are examining below.

3.2.2 Goal and overview of the control algorithms

The control algorithms for retractable harvesters fall into the following categories as shown in Figure 6:

- Static: These algorithms do not change deployment and retraction thresholds while running.
- Dynamic: The algorithms may change deployment and retraction thresholds during the simulation:
  - Aging (with and without weather prediction), which may lower the windspeed re-

traction threshold as the month progresses,

– Fuzzy-Crisp Hybrid (with and without weather prediction), which uses crisp logic to determine when to retract to ensure that the algorithm complies with its controlling OLA and which uses fuzzy logic to determine when to deploy,

where each of the dynamic algorithms have two operating modes:

– Non-predictive mode: At each time-step of the simulation, the simulator provides to the algorithm current, but not predicted, wind data from the sequence of windspeeds in the benchmark that the simulator is using.

– Predictive mode: At each time-step t of the simulation, the simulator provides to the algorithm current and simulated predicted day-ahead windspeeds from a sequence, hereafter called S, of windspeeds in the benchmark. Specifically, for each minute-by-minute time-step t of the simulator, the simulator will pass to the algorithm the windspeed $s[t] \in S$ and the distorted windspeed $d(x, s[t+x])$ where

  - $s[t+x] \in S$ and
  - $x = 1440$ minutes is the time-horizon of the prediction (i.e., $x = 1440$ minutes or one day beyond time-step t) and where
  - function $d()$ distorts each future windspeed to simulate a windspeed-prediction accuracy[8].

3.3 VARIANTS IN EACH CATEGORY OF ALGORITHMS

Each algorithm category (Static, Aging, Fuzzy-Crisp Hybrid) has the following variants:

Variant	Use Weather Prediction	Transitions Limited
0x0	0	0
0x1	0	1
0x2	1	0
0x3	1	1

where the

[8]In Section 4.2.0.3, we describe how we simulate the prediction of windspeeds.

- "Use Weather Prediction" column describes whether (1) or not (0) the variant uses weather prediction, and the
- "Transitions Limited" column describes whether (1) or not (0) the variant must limit the number of state transitions that the harvester it is controlling may make per month, which it must do to comply with OLAs 5 and 6 (Section 3.1.3).

3.4 ALGORITHM CATEGORIES

3.4.1 Static (with and without weather prediction)

The Static algorithm category is comprised of the simplest of the algorithms we wrote and benchmarked. We refer to each member of this category as "Static." Static keeps the same deployment and retraction thresholds as well as the same window size used to calculate the running windspeed average (described in Section 3.2.1) throughout the testing phase.

To train Static (i.e., to find the algorithm-configuration values that Static will use to process a specific workload, we explored the two-dimensional design space described in Section 3.7. Once trained, Static will not change its deployment and retraction windspeed thresholds and windows size of the running average windspeed.

3.4.2 Aging (with and without weather prediction)

The Aging algorithms may change the deployment and retraction thresholds of windspeeds as the month ages. To change the deployment and retraction thresholds, we pass the total number of minutes remaining in the month to a linear function $d()$ that returns the deployment-windspeed threshold. The linear function $d()$ has a slope and y-intercept which are determined during the training phase (Section 3.7) for each month of the year. Examples of $d()$ explored during the training phase are plotted in Figure 7 for Revision 1.1 and Figure 8 for Revision 1.2. (We chose the retraction-windspeed threshold to be $d() - 1$ knots.) The points on the x-axis are the minutes until the month ends (Revision 1.1) or minutes having expired during the month (Revision 1.2). The y-axis indicates the windspeed at which the algorithm deploys (if no other restrictions are in effect, such as quiet hours. The minimum value for each $d()$ is the lowest windspeed deemed to be windy at the pertinent weather station (Appendix A.2.4) in order to ensure that Aging complies with OLAs 1 - 4, which restrict deployment to windy weather. (OLAs 5 and 6 do not restrict deployment to windy conditions.)

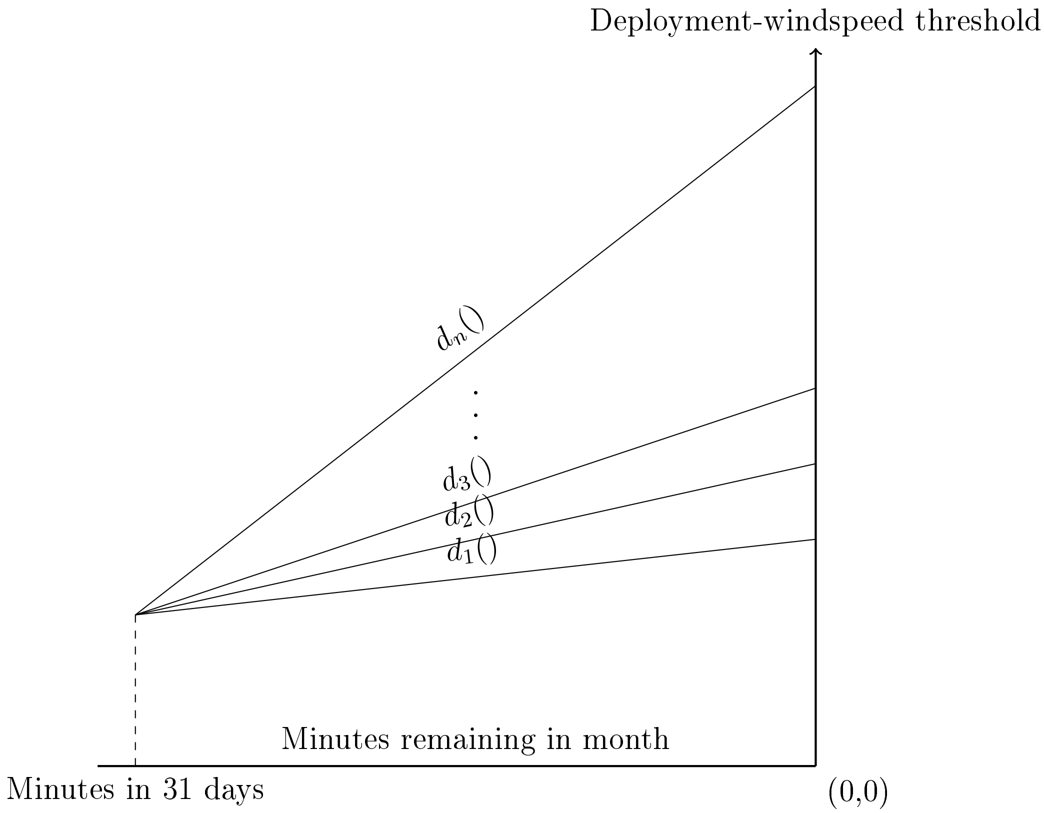

Figure 7: Iterations of deployment-threshold function $d()$ explored during training phase of Aging algorithms (Rev. 1.1) (drawing not to scale)

3.4.3 Fuzzy-Crisp Hybrid (with and without weather prediction)

The Fuzzy-Crisp Hybrid algorithms have crisp code and fuzzy code [80]. The fuzzy code combines membership values in fuzzy sets. The combination depends on the variant (Section 3.3) of the Fuzzy-Crisp Hybrid algorithm we are testing. In the following sections, we explain the combination we use for each variant.

To comply with each OLA, we have a crisp component of the algorithm that retracts the harvester before it can violate the time-visible limit of the OLA. (Crisp part: If the time that harvester has been visible plus the time it takes the harvester to retract is greater than a certain amount, or if the minutes until the month ends equals or is less than than the time it takes for the harvester to retract, then retract.) Outside of those OLA-violating conditions, we use the fuzzy logic described in the following sections, in which the fuzzy sets are delineated by quotation marks. (Thus, we named these algorithms *Fuzzy-Crisp Hybrid*, but because these algorithms do have a fuzzy part, we sometimes refer to these algorithms as simply *Fuzzy*.)

3.4.3.1 Fuzzy-Crisp variant 0x0 If the weather is "Windy" now and (if the time of day is "Not Approaching Quiet Hours" or if there are "Plenty of Allocated Visibility Minutes Remaining"), then deploy. (That is, even if it is approaching quiet hours but there is plenty of visibility time left and the weather is windy, go ahead and deploy.) Otherwise retract. Or retract if the crisp conditions of the OLA will be violated if the algorithm does not retract.

3.4.3.2 Fuzzy-Crisp variant 0x1 If the weather is "Windy" now or if the time of the month is "Approaching the Use-Visibility-Allocation-or-Lose-It Point" (i.e., if the weather is fuzzily approaching the crisp point where the visibility allocation (e.g., x minutes) equals the amount of month remaining), then deploy. Retract if the crisp visibility-time condition of the state-transition-limited OLAs (OLAs 5 and 6) will be violated if the algorithm does not retract.

3.4.3.3 Fuzzy-Crisp variant 0x2 If the weather will be "Very Windy" tomorrow and there are "Not Plenty of Allocated Visibility Minutes Remaining," then retract.

Otherwise, use the same rules used by variant 0x0.

3.4.3.4 Fuzzy-Crisp variant 0x3 If the weather is "Windy" now and the weather will be "Windy" tomorrow or if the time of the month is "Approaching the Use-Visibility-Allocation-or-Lose-It Point" then deploy.

Otherwise, use the same rules used by variant 0x1.

3.4.3.5 Membership functions In the Appendix are descriptions or examples of the membership functions for the following fuzzy sets:

Fuzzy set	Figure
Not Windy	Figure 41 on page 261
Not Very Windy	Figure 41
Plenty of Allocated Visibility Minutes Remaining	Figure 42 on page 261
Approaching Near the Start of Quiet Hours	Figure 43 on page 262
Approaching the Use Visibility Allocation or Lose It Point	Figure 44 on page 262

3.4.3.6 Combined degree of membership We combine the fuzzy sets as described above linguistically in the sections on variants of the fuzzy algorithms (Sections 3.4.3.1 to 3.4.3.4). We mapped the linguistic words *or* and *and* to the mathematical functions max and min, respectively [39, Equations 5 and 6]. For example, the conditional clause in the conditional sentence,

> If the weather will be "Very Windy" tomorrow and there are "Not Plenty of Allocated Visibility Minutes Remaining," then retract,

maps to the mathematical expression

$$\min(\mu_{vw}(p), 1 - \mu_{pvr}(c)) \tag{3.1}$$

where

- $\mu_{vw}(p)$ is the degree of membership of p in the fuzzy set "Very Windy,"
- p is the predicted day-ahead windspeed,
- $\mu_{pvr}(c)$ is the degree of membership of c in the fuzzy set "Plenty of Allocated Visibility Minutes Remaining," and
- c is how many minutes of visibility allocation that the harvester has already consumed.

We call the result of that mathematical operation (Equation 3.1) a *combined degree of membership*.

Let us translate a more complex linquistic expression to a mathematical one. The condition for deploying in the following paragraph

> If the weather is "Windy" now and (if the time of day is "Not Approaching Quiet Hours" or if there are "Plenty of Allocated Visibility Minutes Remaining"), then deploy. Otherwise retract. Or retract if the crisp conditions of the OLA will be violated if the algorithm does not retract.

maps to the mathematical expression

$$\min(\mu_{vw}(n), \max(1 - \mu_{aqr}(t), \mu_{pvr}(c))) \tag{3.2}$$

where

- $\mu_w(n)$ is the degree of membership of n in the fuzzy set "Windy,"
- n is windspeed now,
- $\mu_{aqr}(t)$ is the degree of membership of t in the fuzzy set "Approaching Quiet Hours,"
- t is current time,
- $\mu_{pvr}(c)$ is the degree of membership of c in the fuzzy set "Plenty of Allocated Visibility Minutes Remaining," and
- c is how many minutes of visibility allocation that the harvester has already consumed.

3.5 PSEUDO-CODE

Pseudo-code of the most recent and penultimate revisions of each of the variants of each category described above are given in Appendix A.1. In the following section, we examine selection portions of actual Java code.

3.6 A LOOK INTO SELECTED CODE OF THE ALGORITHMS

3.6.1 Code for the Aging algorithms ($Dynamic_1$ and $Dynamic_{1P}$)

In this section, we describe the code in the Aging category or family of algorithms. The Aging family has variants that allow the harvester to deploy and retract an unlimited number of times each month. The variants that allow "unlimited state transitions" are 0x0, which does not use weather prediction, and 0x2, which does use weather prediction. Variants 0x0 and 0x2, which may be used to comply with OLAs 1 through 4, use the code we describe directly below.

3.6.1.1 For OLAs allowing unlimited state transitions OLAs 1 through 4 require the harvester to be stowed when the weather is not windy. Line 17 in Listing 3.1 on the next page is where the code tests whether or not the windspeed average is below the retraction threshold, which is calculated in Line 9, and is one knot less than the deployment threshold retrieved from a date-dependent, linear function at Line 8. That date-dependent function is what makes the Aging family sensitive to the day of month. As the month wears on, the deployment threshold may increase linearly (Rev. 1.1), as shown in Figure 7 on page 35, or may decrease linearly (Rev. 1.2), as shown in Figure 8 on page 49, or remain constant. In Rev. 1.2, as explained in Section 3.4.2, the deployment threshold on the last day of the month is the lowest windspeed deemed to be windy at the pertinent weather station. The y-intercept is determined during training.

For example, suppose that the training determines that the y-intercept for January at station KATL is 17 knots, as training did when it trained Aging variant 0x0 to process OLA 3 at KATL (Table 41). The deployment threshold starts at 17 knots on January 1 and decreases linearly reaching 7 knots on January 31 because the lowest windspeed deemed to be windy at KATL is 7 knots (Appendix A.2.4 on page 156).

Because OLAs 3 and 4 limit the amount of time a harvester may be visible each month, Line 28 tests whether the sum of the harvester's time visible this month and the time it takes the harvester to retract is above a threshold. The sum is normalized by the number of minutes in the month. Hence, the variable holding the sum is named with the prefix *fraction*. If `fractionVisbilePlusTimeToRetractMonthly` is greater than 0.99 (which is specified on Line 24), the code directs the harvester to retract (Line 30), via the method of the `harvester` object named `.resetMode()`. The algorithm knows the number of minutes that the harvester is permitted to be visible each month by reading a .properties file.

OLAs 1 through 4 mandate that the harvester be stowed during quiet hours. Stowing during quiet hours is implemented via the `if` statement starting on Line 58. The boolean variable `bDuringNightlyVisibilityBan` is set by a variable in the .properties file and, if set, instructs the algorithm to check quiet hours if transitions are unlimited. Because the harvester takes some time to retract (explained in Appendix A.3.5), which is called TIME_TO_RETRACT_MINUTES in the code, the algorithm looks ahead that number of minutes to determine whether quiet hours will be starting then (Line 60) or whether quiet hours have begun (Line 62). If either is the case, the algorithm directs the harvester to retract/remain retracted (Line 67).

Listing 3.1: Code of Aging that processes one windspeed sample for OLAs *not* limiting state transitions

```
1  private void processOneSampleTransitionsUnlimited(Workload.WindspeedSample sample, boolean bUseWeatherPrediction) {
2
3   final int RETRACTION_THRESHOLD_DIFFERENCE = 1;
4
5   // Determine what control signal to output.
6   // Use visibility-time-remaining to control deployment threshold
7
8   int deploymentThresholdKnots = getDeploymentThresholdKnots(sample.date);
9   int retractionThresholdKnots = deploymentThresholdKnots - RETRACTION_THRESHOLD_DIFFERENCE;
10
11  double windspeed_knots_average = ra.updateRunningAverage(sample.windspeed_knots, running_average_window_size);
12
13  if (windspeed_knots_average > deploymentThresholdKnots) {
14
15   harvester.setMode();
16
17  } else if (windspeed_knots_average < retractionThresholdKnots) {
```

```java

  harvester.resetMode();

 }

 // check amount of time used per month
 final float FRACTION_VISIBLE_TIME_THRESHOLD = (float) 0.99;
 double fractionVisbilePlusTimeToRetractMonthly =
  harvester. getFractionVisbilePlusTimeToRetractMonthly(ws.iUsedAllItsAllocatedVisibilityMinutesPerMonth);

 if (fractionVisbilePlusTimeToRetractMonthly > FRACTION_VISIBLE_TIME_THRESHOLD){

  harvester.resetMode();

 }

 // If harvester has somewhat nearly exhausted is allocated visibility time for the month and
 // tomorrow will be much windier than today,
 // then retract to save visibility time
 if (bUseWeatherPrediction) {

  final double MUCH_WINDIER = 1.25;
  final double FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED = 0.64;
  double windspeed_knots_average_future =
   ra.updateRunningAverage(sample.windspeed_knots_predicted_one_day, running_average_window_size);

  if (
   (fractionVisbilePlusTimeToRetractMonthly >
    FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED) &&
   (windspeed_knots_average_future > (windspeed_knots_average*MUCH_WINDIER))
   ){
    //System.out.println("Retracting because future windspeeds are much windier...");
    harvester.resetMode();
   }
  }

 // if within retraction time of quiet hours or during quiet hours, then retract or remain retracted
 if (ws.bDuringNightlyVisibilityBan &
   (
```

```
      !noiseAllowedFutureRetractionTime.bIsNoiseAllowed(sample.date.
         plusMinutes(harvester.TIME_TO_RETRACT_MINUTES))
      | // do not short circuit
      !(noiseAllowed.bIsNoiseAllowed(sample.date))
      )
     ){
      harvester.resetMode();
    }

    harvester.processMode(sample, ep, true);
  }
```

3.6.1.2 For OLAs limiting state transitions to two per month

Because OLAs 5 and 6 limit the harvester to only two state transitions per month yet allow the harvester to remain deployed even when the weather is not windy, the Aging algorithm uses the code shown in Listing 3.2 instead of Listing 3.1 for OLAs 5 and 6.

Listing 3.2: Code of Aging that processes one windspeed sample for OLAs limiting state transitions

```

private void processOneSampleTransitionsLimited(Workload.
   WindspeedSample sample,
 boolean bUseWeatherPrediction) {

 final int RETRACTION_THRESHOLD_DIFFERENCE = 1;

 // Determine what control signal to output.
 // Use visibility-time-remaining to control deployment threshold

 int deploymentThresholdKnots = getDeploymentThresholdKnots(sample.
    date);

 double windspeed_knots_average = ra.updateRunningAverage(sample.
    windspeed_knots, running_average_window_size);

 final double MUCH_WINDIER = 1.25;
 double windspeed_knots_average_future =
   ra.updateRunningAverage(sample.windspeed_knots_predicted_one_day,
       running_average_window_size);

 if (
  (harvester.getMinutesVisibleMonthly() < 1) &&
```

```
20        (
21         (DateStatistics.getMinutesInMonthRemaining(sample.date) < ws.
              iUsedAllItsAllocatedVisibilityMinutesPerMonth)
22         ||
23         // harvester has not yet been visible this month
24         (
25          (windspeed_knots_average > deploymentThresholdKnots) &&
26           (!(bUseWeatherPrediction &&
27             (windspeed_knots_average_future > (windspeed_knots_average*
                MUCH_WINDIER)) // much windier tomorrow
28             )
29           )
30         )
31        )
32      )
33    {
34      harvester.setMode();
35    }
36
37    // check amount of time used per month
38    final float FRACTION_VISIBLE_TIME_THRESHOLD = (float) 0.99;
39    if (harvester. getFractionVisbilePlusTimeToRetractMonthly(ws.
         iUsedAllItsAllocatedVisibilityMinutesPerMonth) >
40     FRACTION_VISIBLE_TIME_THRESHOLD){
41
42       harvester.resetMode();
43     }
44
45    // if within retraction time of end of month, then retract
46    if (DateStatistics.getMinutesInMonthRemaining(sample.date) <=
47            harvester.TIME_TO_RETRACT_MINUTES
48     ) {
49       harvester.resetMode();
50    }
51      harvester.processMode(sample, ep, false);
52  }
```

3.6.2 How the algorithms use weather prediction

3.6.2.1 Static and Aging variant using weather prediction and is not transition-limited (i.e., variant 0x2)

As shown in the code above starting at Line 14 and repeated in the snippet below (which is shared by both Static and Aging), Static and Aging variant 0x2 uses weather prediction in an effort to save visibility time. To save visibility time, when the harvester has somewhat nearly exhausted its allocated visibility time for the month and tomorrow will be much windier than today, the harvester will retract. Thus, if a day is particularly windy, the harvester will still retract, using 20 minutes of visibility time without harvesting. Tomorrow, the harvester uses another 20 minutes without harvesting to deploy.

```java

// If harvester has somewhat nearly exhausted is allocated visibility time for the month and
// tomorrow will be much windier than today,
// then retract to save visibility time
if (bUseWeatherPrediction) {

  final double MUCH_WINDIER = 1.25;
  final double FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED = 0.64;
  double windspeed_knots_average_future =
    ra.updateRunningAverage(sample.windspeed_knots_predicted_one_day, running_average_window_size);

  if (
    (fractionVisbilePlusTimeToRetractMonthly >
   FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED) &&
  (windspeed_knots_average_future > (windspeed_knots_average*MUCH_WINDIER))
  ){

   //System.out.println("Retracting because future windspeeds are much windier...");
   harvester.resetMode();

  }

}
```

3.6.2.2 Static and Aging variant using weather prediction and is transition-limited (i.e., variant 0x3)

As shown in the code snippet directly below (which is shared by both Static and Aging), Static and Aging variant 0x3 uses weather prediction when the harvester has not been visible yet during the month. Then, even when the rolling average windspeed surpasses the windspeed-deployment threshold, the harvester will not deploy if tomorrow is predicted to be much windier (defined as 1.25 times windier) than today.

```
final double MUCH_WINDIER = 1.25;
double windspeed_knots_average_future =
 ra.updateRunningAverage(sample.windspeed_knots_predicted_one_day,
    running_average_window_size);

if (
  (harvester.getMinutesVisibleMonthly() < 1) &&
  (
   (DateStatistics.getMinutesInMonthRemaining(sample.date) < ws.
      iUsedAllItsAllocatedVisibilityMinutesPerMonth)
   ||
  // harvester has not yet been visible this month
   (
    (windspeed_knots_average > deploymentThresholdKnots) &&
     (!(bUseWeatherPrediction &&
       (windspeed_knots_average_future > (windspeed_knots_average*
          MUCH_WINDIER)) // much windier tomorrow
       )
     )
   )
  )
 )
{
  harvester.setMode();   //deploy
}
```

3.6.2.3 Fuzzy variant using weather prediction and is not transition-limited (i.e., variant 0x2)

As shown in the code snippet below, Fuzzy variant 0x2 uses weather prediction in an effort to save visibility time. To save visibility time, if the harvester is running out of allocated visibility time for the month and tomorrow will be much windier than today, then allow an intermediate membership value to be less than 1. That intermediate membership value becomes the maximum value of the resulting membership value. That resulting membership value must be above a certain threshold in order for the harvester to deploy. Thus, weather prediction might cause a harvester not to deploy even when a day is particularly windy, but tomorrow will be very windy and the harvester is running out of visibility time.

```
private double getResultingMembershipValueTransitionUnlimited(double windspeed_knots_average,
  Workload.WindspeedSample sample, boolean bUseWeatherPrediction, double windspeed_knots_average_future) {

 double membershipValueConditional;

 if (bUseWeatherPrediction) {
```

```
  // If very windy tomorrow and running out of time, retract
  // -or-
  // Allow deployment if not very windy tomorrow
  ////  -or-
  // if not running out of time


  membershipValueConditional = Math.max(
    not(windy.getMembershipValueForVeryWindy((int) windspeed_knots_average_future)),
    harvester.getMembershipValueForPlentyOfAllocatedVisibilityMinutesRemaining(
      ws.iUsedAllItsAllocatedVisibilityMinutesPerMonth
      )
    );

 } else {

  membershipValueConditional = 1;

 }

 final int MINUTES_BEFORE_QUIET_HOURS_X_INTERCEPT = 120; // upgrade: set during training

 // If windy and (if not approaching quiet hours or if fraction of time spent
 // stowed is low)

 double resultingMembershipValue = Math.min(membershipValueConditional,

   Math.min(windy.getMembershipValueForWindy((int) windspeed_knots_average),
     Math.max(
       not(noiseAllowed.getMembershipValueForApproachingQuietHours(sample.date,
         MINUTES_BEFORE_QUIET_HOURS_X_INTERCEPT)),
       harvester.getMembershipValueForPlentyOfAllocatedVisibilityMinutesRemaining(
         ws.iUsedAllItsAllocatedVisibilityMinutesPerMonth)))

 );

 return (resultingMembershipValue);

}
```

3.6.2.4 Fuzzy variant using weather prediction and is transition-limited (i.e., variant 0x3)

As shown in the code snippet below, Fuzzy variant 0x3 uses weather prediction to only deploy (i.e., consume a transition from being stowed to being deployed) only when both today and tomorrow are windy. Thus, if a windy day is followed by a calm day, the harvester will not deploy, missing the windy day.

```
private double getResultingMembershipValueTransitionLimited(double
    windspeed_knots_average,
  Workload.WindspeedSample sample, boolean bUseWeatherPrediction,
      double windspeed_knots_average_future) {

 double windspeed_knots_average_to_use;


 if (bUseWeatherPrediction) {

  // must be windy today and windy to-morrow
  windspeed_knots_average_to_use = Math.min(windspeed_knots_average,
      windspeed_knots_average_future);
```

```java

  } else {

   windspeed_knots_average_to_use = windspeed_knots_average;

  }

  double resultingMembershipValue = Math.max(
    windy.getMembershipValueForWindy((int)
        windspeed_knots_average_to_use),
    DateStatistics.getMembershipValueForApproachingUseItOrLoseItPoint
        (sample.date, harvester, ws));

  return (resultingMembershipValue);

}
```

3.7 TRAINING METHOD: EXPLORING A TWO-DIMENSIONAL DESIGN SPACE

To train algorithm a to process a specific workload wrk, we run a multiple times over the training partition of wrk. The number of times that we run a is governed by nested loops where we iterate an inner-loop variable $y_{intercept}$ (defined in the sections immediately below) and an outer-loop variable r where r is the size of the windspeed-running-average window.[9]

We explore the following windspeed-running-average window sizes: $r \in \{1, 31, 61, 91, 121\}$, which we express in shorthand as $r \in [1 \text{ (step 30) } 121]$ for all the algorithms. The units are minutes (or, equivalently, samples of windspeeds since we provide a windspeed each minute).

With respect to the metric that we shall use in the testing phase (i.e., $NetNorm(a, wrk)$ 4.1.3 or $MQNetNorm(a, wrk)$ 4.1.5.3), we record the best $y_{intercept}$ and windspeed-running-average size r for each month of the year (for the cases of training the Aging and Fuzzy algorithms) or for the entire training period (for the case of training the Static algorithms).

[9] In this release of the control algorithms, we use a relatively simple training or parameter-tuning method in that each iteration uses a constant step size. More sophisticated parameter-tuning methods exist that use local search [2]. We did not use local search in this release.

3.7.1 Context of $y_{intercept}$ for the Static and Aging category of algorithms

The variable $y_{intercept}$ is part of a linear equation that returns a deployment-windspeed threshold when given the number of minutes remaining in the month. For the Aging algorithm, the linear equation's slope m is less than or equal to 1. When $m < 1$, the deployment threshold decreases as the month wears on. However, for the Static algorithm, the slope of the linear equation is zero; The deployment threshold remains constant throughout the testing phase.

For the Static and Aging algorithms, the values we explore for the inner-loop variable $y_{intercept}$ are $\{w, w+10, w+20, w+30\}$, which we express in shorthand as $y_{intercept} \in [w$ (step 10) $w+30]$, where $w \in \mathbb{N}$ is the least windspeed deemed to be windy (defined in A.2.4) at weather station ws and where the units of w are knots.

3.7.2 Context of $y_{intercept}$ for the Fuzzy category of algorithms

The variable $y_{intercept}$ is part of a linear equation that returns a deployment threshold of the combined degree of membership (Section 3.4.3.6). The slope of the equation is zero. That is, the deployment-degree-of-membership threshold remains constant throughout each month.

For the Fuzzy algorithms, the value we use for the variable $y_{intercept}$ (i.e., the deployment-degree-of-membership threshold in the fuzzy set NOT WINDY) is 0.5 which maps to the lowest windspeed we deem to be windy (Appendix A.2.4).

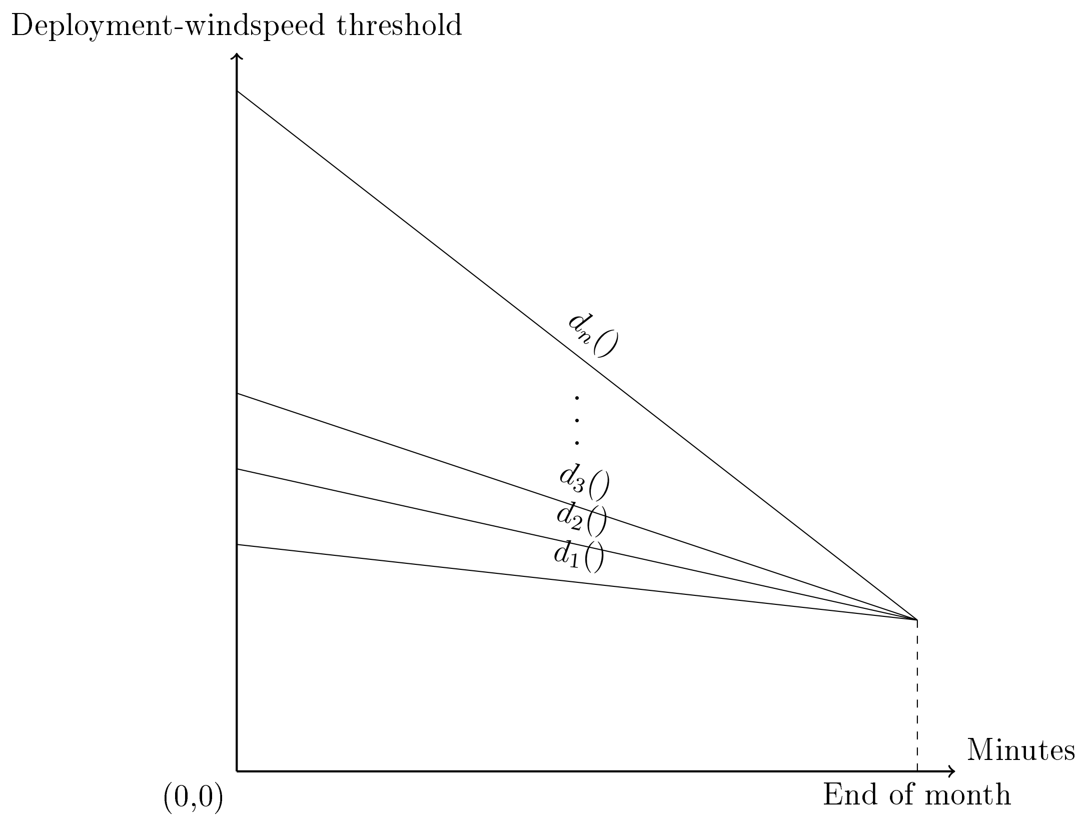

Figure 8: Iterations of deployment-threshold function $d()$ explored during training phase of Aging algorithms (Rev. 1.2) (drawing not to scale)

4.0 PERFORMANCE ANALYSIS

In this chapter, we present metrics by which to measure the performance of retractable-wind-harvester control algorithms and a set of various benchmarks or workloads. The metrics and benchmarks together form a suite to cover a variety of operating scenarios.

4.1 METRICS

When should an operationally-limited (e.g., limited-by-operation-limitation-set-A) control algorithm deploy and retract? That answer depends on what metric that algorithm is trying to optimize. How do we know whether a given metric M measures how well a retractable-harvester control algorithm controls a harvester? We address that question by listing criteria in the next section:

4.1.1 Criteria

Let the union of an operation limitation set with a metric M form an operational limitation agreement (OLA) between an RWEH-hosting community and an RWEH vendor. Suppose that an algorithm a is operating that OLA while processing a workload wrk.

1. If an abstract metric $M(a, wrk)$ satisfies the following sufficient (and not necessarily necessary) criteria, then $M(a, wrk)$ measures how well a retractable-harvester control algorithm $a \in A$ controls a harvester:
   a. The metric $M(a, wrk)$ reflects the fact of whether or not algorithm a violates the OLA when processing workload wrk.

b. If a complies with the OLA, then the value of $M(a, wrk)$ depends on how much energy that the harvester controlled by algorithm a nets.

c. $M(a, wrk) > M(b, wrk)$ if and only if, without loss of generality, algorithm a processes workload wrk better than algorithm b processes the same workload where *better* is defined as

• a netting more energy than b in the case where both a and b both comply with the OLA or

• a complying with the OLA over the entire workload wrk and b violating the OLA during some point during b's processing of wrk.

4.1.2 Applying the criteria to a specific metric, SCNetNorm

Now that we have listed the criteria, let us ask the following question: *Question 2:* Does the metric SCNetNorm (Equation 4.5) sufficiently measure how well a retractable-harvester control algorithm controls a harvester, within a operation limitation agreement (OLA) between the harvester operator and the harvester-hosting community?

Example OLA: A community will host a retractable harvester if the harvester will comply with the following three conditions:

- be visible (i.e., not fully stowed) less than t_{viz} minutes each month,
- be hidden at night between $T_{NightStart}$ and $T_{DayStart}$ (e.g., between 10 P.M. and 7 A.M. to facilitate amateur astronomy and to reduce noise), and
- be a retractable version of a typical towered wind turbine[1] (e.g., a Vestas V90 turbine)

without any other restrictions (e.g., day of month, month of year). Let the requirements listed directly above along with a function defining incentives and penalties form an example OLA between the harvester operator and the hosting community.

[1] A community might choose a towered retractable harvester over an airborne wind energy system (Appendix A.6.2) because a towered harvester's potential space in which it might operate is smaller than that of an airborne system. For example, one specific airborne system "flies autonomously in loops averaging 250 meters in diameter" [48]. Thus, it might be easier to obtain the necessary permissions to operate a (more familiar) towered harvester versus an airborne one.

Conversely, a community might choose a flying harvester over a towered harvester because windspeeds typically increase with altitude [17, p. 668] (and a flying harvester might provide entertainment value).

4.1.3 The NetNorm metric

To determine how well a control algorithm a controls the harvester, we compare a to an algorithm that keeps harvester hm permanently deployed called *Permanent*. Let us normalize the net energy harvested by a to the net energy harvested by Permanent and call the following normalized-to-permanent metric "Net Energy Normalized" or "NetNorm", which is the featured metric:

$$NetNorm(a, wrk) = \frac{NetEnergy(a, wrk)}{E_{\text{Harvested}}^{(\text{Permanent}, wrk)}} \tag{4.1}$$

where the NetNorm metric's numerator is the following simpler metric:

$$NetEnergy(a, wrk) = E_{\text{Harvested}}^{(a,wrk)} - E_{\text{DeployandRetract}}^{(a,wrk)} \tag{4.2}$$

where, in the ordered pair (a, wrk), $wrk = \{OLA, ws, hm\}$ where, in turn,

- OLA indicates the operation limitation agreement constraining control algorithm a,
- ws is the weather station code,
- hm is the model of the harvester that the algorithm a is controlling,

and where

- $E_{\text{DeployandRetract}}^{(a,wrk)}$ is the total energy that the wind harvester being controlled by control algorithm a used to deploy and retract when processing workload wrk. (Some harvesters consume energy to deploy. For example, the Makani energy kite has propellers that consume energy to lift the wing to a particular energy-harvesting height [48]. Conversely, it is conceivable that $E_{\text{DeployandRetract}}^{(a,wrk)} = 0$ if a harvester were to use airfoils to deploy and gravity to retract. Also, $E_{\text{DeployandRetract}}^{(a,wrk)} = 0$ when algorithm a =Permanent.), and
- $E_{\text{Harvested}}^{(a,wrk)}$ is the energy harvested by the wind harvester being controlled by control algorithm a processing workload wrk. Note that a can be set to the Permanent deployment denoting the energy harvested by the wind harvester processing workload wrk if it were permanently deployed. (To assist the users of this benchmark suite, $E_{\text{Harvested}}^{(\text{Permanent},wrk)}$ for each station's training and testing data is provided in Table 65 on page 265 for the envisioned towered harvester controlled by the algorithms described herein.)

4.1.4 Measuring OLA compliance

4.1.4.1 OLA-compliance function We are assuming that an actual OLA would specify penalties and rewards for violating the agreement and for surpassing the agreement, respectively. Although outside the scope of this work, an example of a penalty metric could translate each minute that the harvester is out of compliance (i.e., the harvester is visible beyond its limits) into points. Each point would equate to a fine that the harvester-operator must pay to the harvesting community.

For the purposes of this work, we define an OLA Compliance Corrector function $SCC(a, wrk)$ that simply returns 1 (which indicates a boolean value of `true`) if control algorithm a meets all the requirements of an OLA and otherwise returns 0 (which indicates a value of `false`). That is, we define

$$SCC(a, wrk) = \begin{cases} 1, \text{if } a \text{ controlling } hm \text{ satisfies } OLA \text{ over the entire} \\ \quad \text{windspeed testing file of station } ws \\ 0, \text{otherwise,} \end{cases} \tag{4.3}$$

where wrk contains the specification of the harvester model hm, the weather station ws, and the operation limitation agreement OLA (i.e., $wrk = \{OLA, ws, hm\}$).

4.1.4.2 OLA-compliance-measuring metrics We define OLA-compliance-measuring versions of NetEnergy (Equation 4.2), NetNorm (Equation 4.1), and MQNetNorm (defined below in Equation 4.8 on page 58), which return the ordered pairs:

$$SCNetEnergy(a, wrk) = (SCC(a, wrk), NetEnergy(a, wrk)), \tag{4.4}$$

$$SCNetNorm(a, wrk) = (SCC(a, wrk), NetNorm(a, wrk)), \tag{4.5}$$

and

$$SCMQNetNorm(a, wrk) = (SCC(a, wrk), MQNetNorm(a, wrk)), \tag{4.6}$$

respectively.

Note that a very simple mapping of the ordered pair (x, y) where $x \in \{0, 1\}$ and $y \in \mathbb{R}$ to the real number system $\mathbb{R}$ where $(0, y)$ is mapped to $0 \in \mathbb{R}$ and where $(1, y)$ is mapped to $y \in \mathbb{R}$ results in $(0, y)$ being mapped to a higher number in $\mathbb{R}$ (namely, $0 \in \mathbb{R}$) than the number to which $(1, y)$ is mapped (i.e., the number $y \in \mathbb{R}$) if $y < 0$. In other words, such a simple mapping would interpret an OLA-violating performance as higher than an OLA-complying performance. To avoid that problem, we define orderings of the ordered pairs themselves in the next section.

4.1.4.3 Orderings of OLA-compliance-measuring metrics We need to define orderings of the OLA-compliance-measuring metrics in order to know how to compare the values of those metrics. In Section 4.1.6, we use order $R_{X_{wrk}}$, defined immediately below.
Let A be the set of all retractable-harvester control algorithms.
Let X_{wrk} be the set of all ordered pairs generated by $SCNetEnergy(a, wrk)$ for every $a \in A$.
That is, let $X_{wrk} = \{SCNetEnergy(a, wrk) | \forall a \in A\}$.
Also, let $Y_{wrk} = \{SCNetNorm(a, wrk) | \forall a \in A\}$.

We define the following order $R_{X_{wrk}}$ on set X_{wrk} of ordered pairs where the ordered pair

$$x_a = (SCC(a, wrk), SCNetEnergy(a, wrk)) \in X_{wrk} \text{ and } a \in A$$

and the relation of $x_a \in X_{wrk}$ to $x_b \in X_{wrk}$ is given by the following cases where algorithm $b \in A$:

- If $SCC(a, wrk) = 1$ and $SCC(b, wrk) = 0$, then $x_a > x_b$. That is, any compliant algorithm always scores higher than a non-compliant algorithm.
- If $SCC(a, wrk) = 0$ and $SCC(b, wrk) = 1$, then $x_a < x_b$ (which is abstractly identical to the first bullet, but included here for clarity).
- If $SCC(a, wrk) = 0$ and $SCC(b, wrk) = 0$, then $x_a = x_b$. That is, the score of all non-compliant algorithms is the same.
- If $SCC(a, wrk) = 1$ and $SCC(b, wrk) = 1$, then x_a relates to x_b indentically to how $SCNetEnergy(a, wrk)$ and $SCNetEnergy(b, wrk)$ relate. That is, $x_a < x_b$ if and only if $SCNetEnergy(a, wrk) < SCNetEnergy(b, wrk)$, and $x_a = x_b$ if and only if $SCNetEnergy(a, wrk) = SCNetEnergy(b, wrk)$.

The order $R_{Y_{wrk}}$ on set Y_{wrk} is identical to the order $R_{X_{wrk}}$ except where $R_{X_{wrk}}$ refers to the NetEnergy metric the order $R_{Y_{wrk}}$ refers to the NetNorm metric.

Because in this dissertation, we are assuming that our example algorithms fully comply with the pertinent OLAs, the SCNetNorm() metric reduces to the NetNorm() metric, the ordering $R_{Y_{wrk}}$ reduces to the typical ordering of the real number system $\mathbb{R}$ when we evaluate our example algorithms herein using the NetNorm() metric.

4.1.5 Cost-dependent complementary metrics

On a grid itself, the amount of energy being injected into the grid must always match the amount of energy leaving it [63]. Matching involves adjusting inputs into the grid (generation) and/or outputs (loads). In the category of outputs or loads, we include dump (or diversion) loads, which can receive excess energy when, for example, normal loads become unavailable or when normal loads cannot absorb available energy [82, p.181]. "Wholesale electricity markets sometimes result in prices below zero. That is, sellers pay buyers to take the power. This situation arises because certain types of generators, such as nuclear, hydroelectric, and wind, cannot or prefer not to reduce output for short periods of time when demand is insufficient to absorb their output. Sometimes buyers can be induced to take the power when they are paid to do so" [23].

The following algorithm-performance-measuring metrics, which are complementary to the NetNorm metric from Section 4.1.3, seek to quantify the degree to which an algorithm a processes a workload wrk to cause a retractable energy harvester to help or hurt the effort to balance electricity demands with supplies in the immediate energy market: Market Quadrants Scores, Market Quadrants Matching Percentage, and MQNetNorm (all defined below). To define those metrics, we first define the following two states:

- *helping* as generating electricity when electricity prices are positive and using energy when electricity prices are negative, and
- *hurting* as using electricity when electricity prices are positive and generating energy when electricity prices are negative.

4.1.5.1 Market Quadrants Scores (MQS) The metric "Market Quadrants Scores" or "MQS" uses the variable price of energy over time t. MQS is a two-dimensional matrix describing the frequency distribution of the number of minutes that the combination of an electricity market and harvester is in the following four cases, which are shown in Figure 9 as four quadrants of a co-ordinate system where $E_{Price}(t)$ and $NetEnergy(a, wrk, t)$ are on the horizontal and vertical axes, respectively:

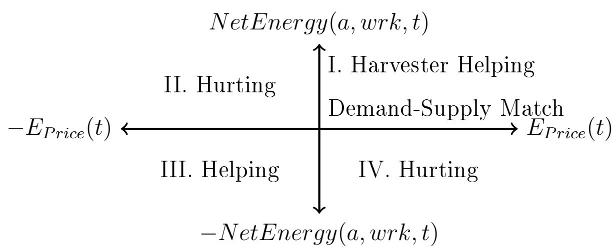

Figure 9: Quadrants indicating whether harvester is helping directly to match energy supply with demand or is directly hurting the matching effort

I. Harvester **Helping** Demand-Supply Match: The market is demanding energy and harvester is generating energy; $(E_{Price}(t) > 0)$ and $(NetEnergy(a, wrk, t) > 0)$.

II. Harvester **Hurting** Demand-Supply Match: The market has excess energy and the harvester is generating energy; $(E_{Price}(t) < 0)$ and $(NetEnergy(a, wrk, t) > 0)$.

III. Harvester **Helping** Demand-Supply Match: The market has excess energy and the harvester is consuming energy; $(E_{Price}(t) < 0)$ and $(NetEnergy(a, wrk, t) < 0)$.

IV. Harvester **Hurting** Demand-Supply Match; The market is demanding energy and the harvester is consuming energy: $(E_{Price}(t) > 0)$ and $(NetEnergy(a, wrk, t) < 0)$.

where

- t is the minute-by-minute time-step of the windspeed datafile where t ranges from 0 to $n - 1$ where n is the number of minute-by-minute time-steps in the windspeed datafile that algorithm a is processing (e.g., `testingKPIT2009-2014in.csv` has 3,154,946 lines)
- $E_{Price}(t)$ is the price per kWh of energy at time-step t in Canadian dollars[2], and

[2]The price is in Canadian dollars because we derived $E_{Price}(t)$ from data available from Ontario's Inde-

- $NetEnergy(a, wrk, t)$ is $NetEnergy(a, wrk)$ (defined by Equation 4.2 on page 52.) earned or expended during time-step t

4.1.5.2 Market Quadrants Matching Percentage (MQMP)

In the co-ordinate system shown in Figure 9, when algorithm a processes workload wrk in a simulation, each simulated minute can be plotted as a point. Each point not on an axis is either in a "helping" quadrant or a "hurting" quadrant. To calculate a percentage we call the *Market Quadrants Matching Percentage* (MQMP), we count the number of points in the "helping" quadrants and then divide by the sum of points in the "helping" and "hurting" quadrants (Equation 4.7). (Points on the axes are not counted when we total the number of points in each quadrant type, as shown in the following example.)

For example, suppose algorithm a (which is a very early version of an algorithm that we present herein) processes workload wrk which has exactly 2,630,774 minutes (which is approximately 5 years worth) and each minute falls in the seven categories listed in Table 2 on page 59, which are also listed here:

I.	Harvester Generating Energy When Grid Needs Energy
II.	Harvester Generating Energy When Grid Has Excess Energy
III.	Harvester Using Energy When Grid Has Excess Energy
IV.	Harvester Using Energy When Grid Needs Energy
−Horiz. Axis.	Harvester Idle When Grid Has Excess Energy
+Horiz. Axis.	Harvester Idle When Grid Needs Energy Minutes
Vert. Axis.	Energy Price Is Zero.

We sum the minutes in the "helping" quadrants (I and III); the sum is 244,047. The sum of the minutes in the "hurting" quadrants (II and IV) is 42,898. The harvester is idle for 2,343,829 minutes because algorithm a is complying with an OLA. (For example, the OLA limits a harvester's operation to 8760 minutes of each month, which is approximately 20% of each month, and mandates that the harvester must be stowed when windspeeds are below a

pendent Electricity System Operator [37]. The prices have not been corrected for inflation; Inflation does not affect the MQMP metric (next section) and the MQNetNorm metric (Section 4.1.5.3) because the only characteristic of $E_{Price}(t)$ that those two metrics use is its sign.

certain threshold.) We divide the "helping sum" by the aggregate of the helping and hurting sums to calculate the following Market Quadrants Matching Percentage:

$$
\begin{aligned}
MQMP(a, wrk) &= \frac{\text{Minutes Harvester Is Helping the Matching}}{\text{Minutes Harv. Is Helping} + \text{Minutes Harv. Is Hurting}} \\
&= \frac{244{,}047}{244{,}047 + 42{,}898} \\
&= 0.85 \\
&= 85\%.
\end{aligned} \tag{4.7}
$$

The MQMP metric indicates how well the harvester is using its active time to help a grid operator to match electricity loads with supply directly. An $MQMP(a, wrk)$ score of 85% indicates that algorithm a makes the harvester help 85% of the time that the harvester was active (*active* means sending energy to or consuming energy from the grid) and electricity prices were not zero. Note that Table 2 indicates that the harvester was idle 89.1% of the time which not only includes the time that the harvester was stowed, but also the time that the harvester is visible yet neither sending energy to nor consuming energy from the grid.

4.1.5.3 MQNetNorm We multiply the Market Quadrants Matching Percentage (MQMP) (Equation 4.7) by $NetNorm(a, wrk)$ (Equation 4.1) to create a version of $NetNorm(a, wrk)$ that reflects how well an algorithm a helps to balance energy supplies with demand.

For example, suppose that $NetNorm(a, wrk) = 0.24$ and that the $MQMP(a, wrk)$ for algorithm a processing workload wrk is 0.85. Then the product of those two metrics is

$$
\begin{aligned}
MQNetNorm(a, wrk) &= MQMP(a, wrk) \times NetNorm(a, wrk) \\
&= 0.85 \times 0.24 \\
&= 0.2,
\end{aligned} \tag{4.8}
$$

which is 15% lower than the NetNorm score because the harvester was helping to match energy supplies with demand only 85% of the time the harvester was active.

The MQMP and MQNetNorm scores might be used by a harvester-hosting community that altruistically wants to help match supplies with demand. Also, since hurting the effort

Table 2: Example itemized results of an algorithm a's processing of a workload wrk

Quadrant	Description	Minutes	Helping	Hurting	Idle
I	Harvester Generating Energy When Grid Needs Energy	244,028	244,028		
II	Harvester Generating Energy When Grid Has Excess Energy	433		433	
III	Harvester Using Energy When Grid Has Excess Energy	19	19		
IV	Harvester Using Energy When Grid Needs Energy	42,465		42,465	
−Horiz. Axis	Harvester Idle When Grid Has Excess Energy	2128			2128
+Horiz. Axis	Harvester Idle When Grid Needs Energy Minutes	2,341,701			2,341,701
Vert. Axis	Energy Price Is Zero	0			0
	Total	2,630,774	244,047	42,898	2,343,829
	Percentage of All Minutes	100%	9.3%	1.6%	89.1%

to match supply with demand is defined as producing electricity when electricity prices are negative, a harvester-hosting community might use the MQMP and MQNetNorm scores as part of a larger effort to provide ancillary services[3] to electric utilities.

4.1.6 Analysis of SCNetNorm

Question 1 (Section 4.1.2) is, Does the metric SCNetNorm (Equation 4.5) sufficiently measure how well a retractable-harvester control algorithm controls a harvester, within a operation limitation agreement (OLA) between the harvester operator and the harvester-hosting community? To answer Question 1, we follow these two steps: 1.) prove that SCNetEnergy (Equation 4.4) meets the criteria we listed in Section 4.1.1 on page 50, and then 2.) apply the results of that proof to show that those criteria are met by SCNetNorm (Equation 4.5).

4.1.6.1 Proofs

1. $SCNetEnergy(a, wrk)$ (Equation 4.4) is dependent on whether or not a violates the OLA. $SCNetEnergy(a, wrk)$ returns $(1, x)$ if control algorithm a meets all the requirements of the OLA and otherwise returns $(0, x)$ where $x \in \mathbb{R}$.
2. Thus, the metric $SCNetEnergy(a, wrk)$ is dependent on whether or not a violates the OLA. That is, the metric $SCNetEnergy(a, wrk)$ satisfies criterion a.
3. If a satisfies the OLA, then $SCNetEnergy(a, wrk) = (1, NetEnergy(a, wrk))$, whose value essentially is $NetEnergy(a, wrk)$ in the context of order R_x in Section 4.1.4.3. Thus, we see that $SCNetEnergy(a, wrk)$ is determined by $NetEnergy(a, wrk)$ thereby satisfying criterion b. (For example, if both a and b satisfy the OLA and if a nets more energy than b, then $SCNetEnergy(a, wrk) > SCNetEnergy(b, wrk)$ since

$$(1, NetEnergy(a, wrk)) > (1, NetEnergy(b, wrk))$$

as defined by order R_x in Section 4.1.4.3.)

[3]"Services that ensure reliability and support the transmission of electricity from generation sites to customer loads. Such services may include load regulation. . . ." [22]. For example, when electricity prices are negative the community may want to be increasing their power consumption instead of harnessing wind energy.

4. If a satisfies the OLA during its processing of the workload wrk and b violates the workload during some point, then $SCNetEnergy(a, wrk) > SCNetEnergy(b, wrk)$ as defined by order R_x.
5. Thus, the two items directly above (Items 3 and 4) imply that $SCNetEnergy(a, wrk)$ satisfies criterion c.
6. Because $SCNetEnergy(a, wrk)$ satisfies criteria a, b, and c, then by premise 1 above, $SCNetEnergy(a, wrk)$ measures how well a control algorithm a controls a retractable-harvester.

Now, let us prove that $SCNetNorm(a, wrk)$ (Equation 4.5) measures how well a control algorithm a controls a retractable-harvester. Directly above, we have proven that $SCNetEnergy(a, wrk)$ does. By Equations 4.5 and 4.1, we can express $SCNetEnergy(a, wrk)$ as

$$SCNetEnergy(a, wrk) = \\ (SCC(a, wrk), (NetNorm(a, wrk)) \times (E_{\text{Harvested}}^{(\text{Permanent}, wrk)})) \quad (4.9)$$

because $NetNorm(a, wrk)$ is merely $NetEnergy(a, wrk)$ divided by the positive constant $E_{\text{Harvested}}^{(\text{Permanent}, wrk)}$. Thus, in the seven-point line of reasoning given directly above, we can replace $NetEnergy(a, wrk)$ with the product $(NetNorm(a, wrk)) \times (E_{\text{Harvested}}^{(\text{Permanent}, wrk)})$. Without loss of generality, we can assume that $E_{\text{Harvested}}^{(\text{Permanent}, wrk)} = 1$, therefore causing the the seven-point line of reasoning to apply directly to $NetNorm(a, wrk)$, hence, proving that $SCNetNorm(a, wrk)$ measures how well a control algorithm a controls a retractable-harvester.

4.1.6.2 Future work: Market Quadrants Matching Percentage of Energy (MQMPEnergy) The MQMP metric (Section 4.1.5.2) counts minutes. The MQMPEnergy metric counts energy, as shown in the following equation:

$$MQMPEnergy(a, wrk) = \frac{\text{Helping kWh}}{\text{Helping kWh} + \text{Hurting kWh}}. \quad (4.10)$$

where

- “Helping kWh” are the kilowatt-hours that the harvester is providing to the grid when the grid needs energy or that the harvester is consuming from the grid when the grid has excess energy and
- “Hurting kWh” are the kilowatt-hours that the harvester is providing to the grid when the grid has excess energy or that the harvester is consuming from the grid when the gird needs energy. For future work, we recommend that MQMPEnergy (Section 4.1.6.2) be considered for addition to future versions of the benchmark suite.

4.1.6.3 Future work: Providing current energy-price profiles We provide hourly prices for each hour in 2004 to 2014, inclusive, in this version of the benchmark suite to be used with the MQMP metric (Section 4.1.5.2), which is concerned only with the sign of the electricity price. Because, as we illustrate in Section 4.2.0.2, that the number of hours that electricity prices are negative have been tending to increase (Figure 14 on page 75), we suggest that future versions of this benchmark suite include current energy-price profiles in order to verify and detect trends in hourly electricity prices. Also, because future metrics might take into account not only the price's sign, but also the price's “amplitude” each hour and price profiles seem to be changing (We explore changes in price profiles immediately below), it seems important to include price data for years contemporary to future releases of this benchmark suite.

Electricity-price profiles show changes per year. We averaged and plotted hourly Ontario electricity prices [37] per hour-of-day per year between the years 2003 and 2017, inclusive, to produce Figure 10 on page 64, which illustrates that prices tend to be decreasing and the shapes of the profiles are possibly changing. Note that the prices tend to have two daily peaks. We plot the hours of the first and second peaks in Figure 11 on page 65. Note that the first peak was at or after the 11:00 hour until 2015, when the first peak was at the 9:00 or 10:00 hour. Note in Figure 12 on page 66 that a trend could be developing where the second peak's dominance of the first peak is growing (but 2015 does not fit that trend).

We explore the proportions plotted in Figure 12 to determine whether or the modified Mann-Kendall test detects a trend. Let the null hypothesis H_0 be that the proportions do *not* exhibit yearly monotonic trend. Let the alternative hypothesis H_A be that the proportions do. We found that if we set the significance level A to 0.05, then we behave in a manner consistent with our conclusion that no monotonic trend is present in year-to-year proportion of the second daily peak price or the first daily peak price even though there is a 0.05 possibility that our conclusion is wrong. The significance level of 0.05 is the probability that we commit a Type I error, i.e., that we falsely reject the null hypothesis H_0 : No monotonic trend exists. For details, please see Appendix A.19. (Thus, taken with Figure 10, hypothesis H_0 implies that the amplitudes of the first and second daily peaks tend to move upwards or downwards together.)

4.2 BENCHMARKS

Question 1: What operation limitation agreements (OLAs) and weather conditions approximate actual field conditions of retractable-harvester control algorithms?

We describe herein a benchmark suite we created for evaluating retractable-harvester-control algorithms. That is, in this section, we describe a suite of workloads. As shown in Figure 13, each workload has data derived from a weather station (e.g., a set of windspeeds), a harvester model, and a operation limitation agreement or OLA (i.e., a set of deployment restrictions and a reward/penalty function).

Because we are assuming that actual sets of operational limitations will be influenced by the preferences of persons living in potential harvesting-hosting communities, we used a marketing survey company to gather responses from over 300 U.S. survey takers (Section 3.1). Each workload is comprised of the following combination of choices:

- An OLA from the set of OLAs derived by the process described in Section 3.1 (OLAs 1-4) and partially derived via the fact that persons tend to ignore non-novel stimuli [28], e.g., persons living next to busy train tracks tend to ignore the sounds of trains on those tracks (OLAs 5 and 6).
- A weather station ws from the set of 30 weather stations from which we have formatted eleven years of minute-by-minute windspeed data: The first nine of the eleven years of windspeeds are training data; the balance are test data. (Reasoning for dividing the data

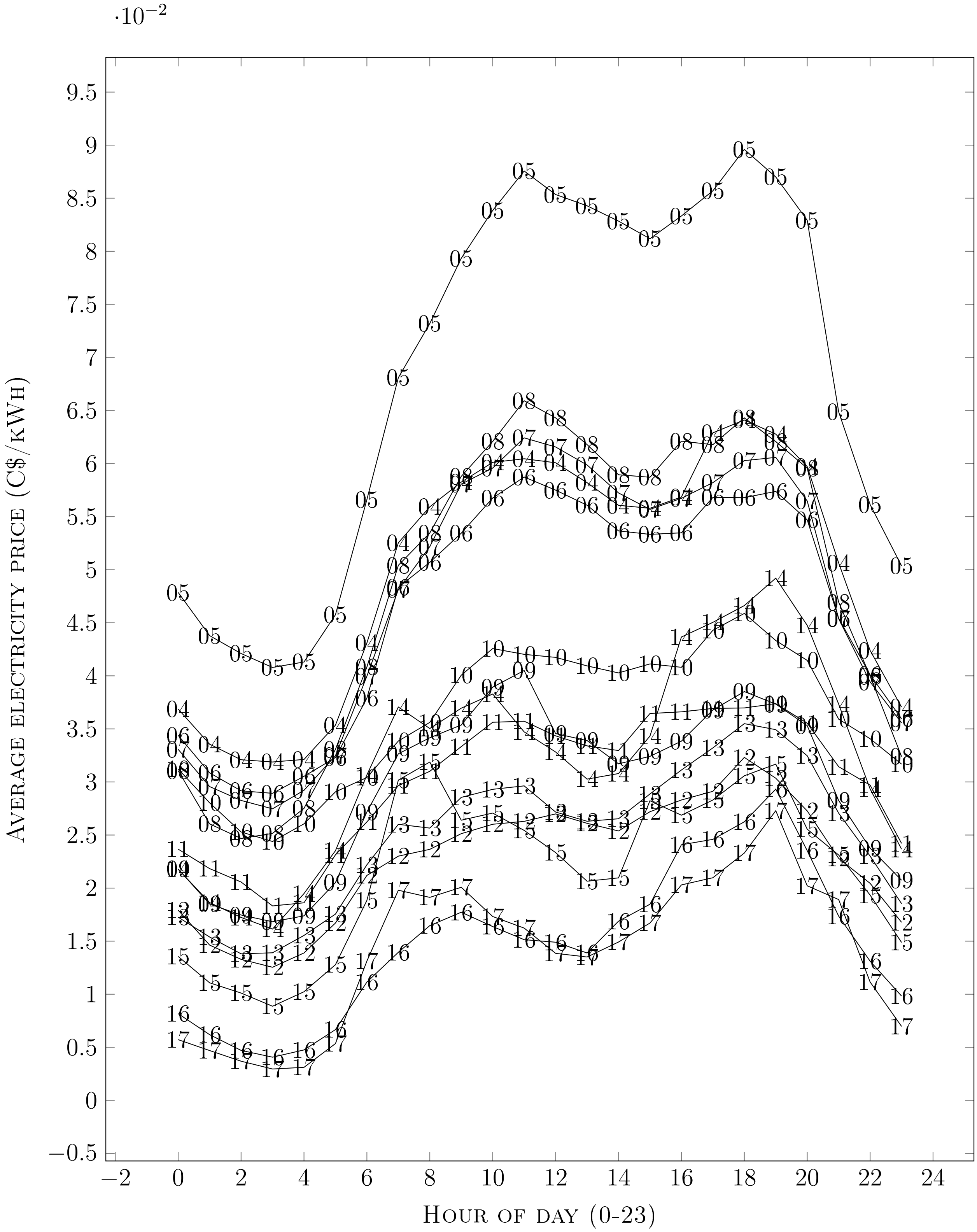

Figure 10: Average electricity price at hour of day for individual years 2004-2017

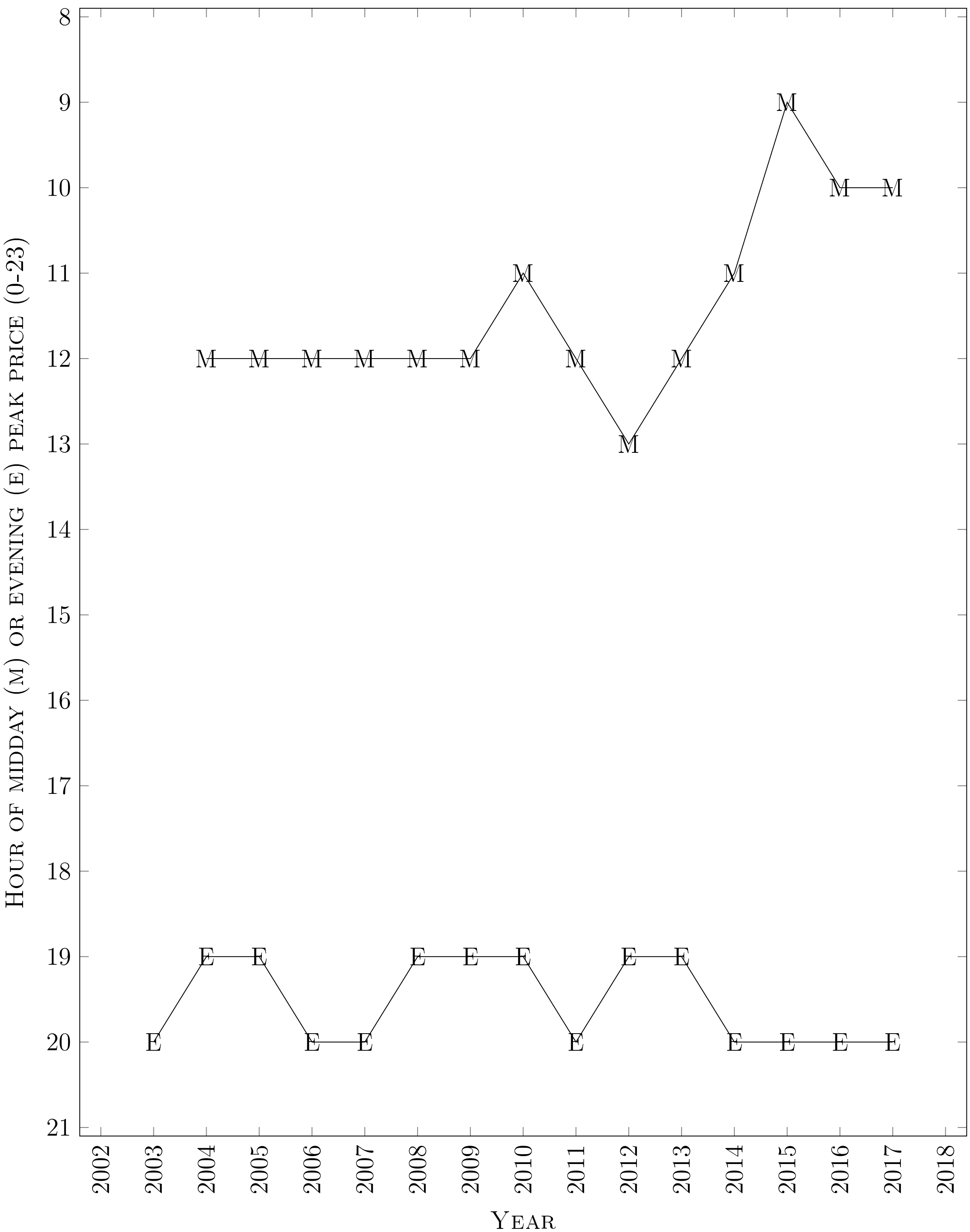

Figure 11: Hour when average electricity price peaks at midday and evening per year

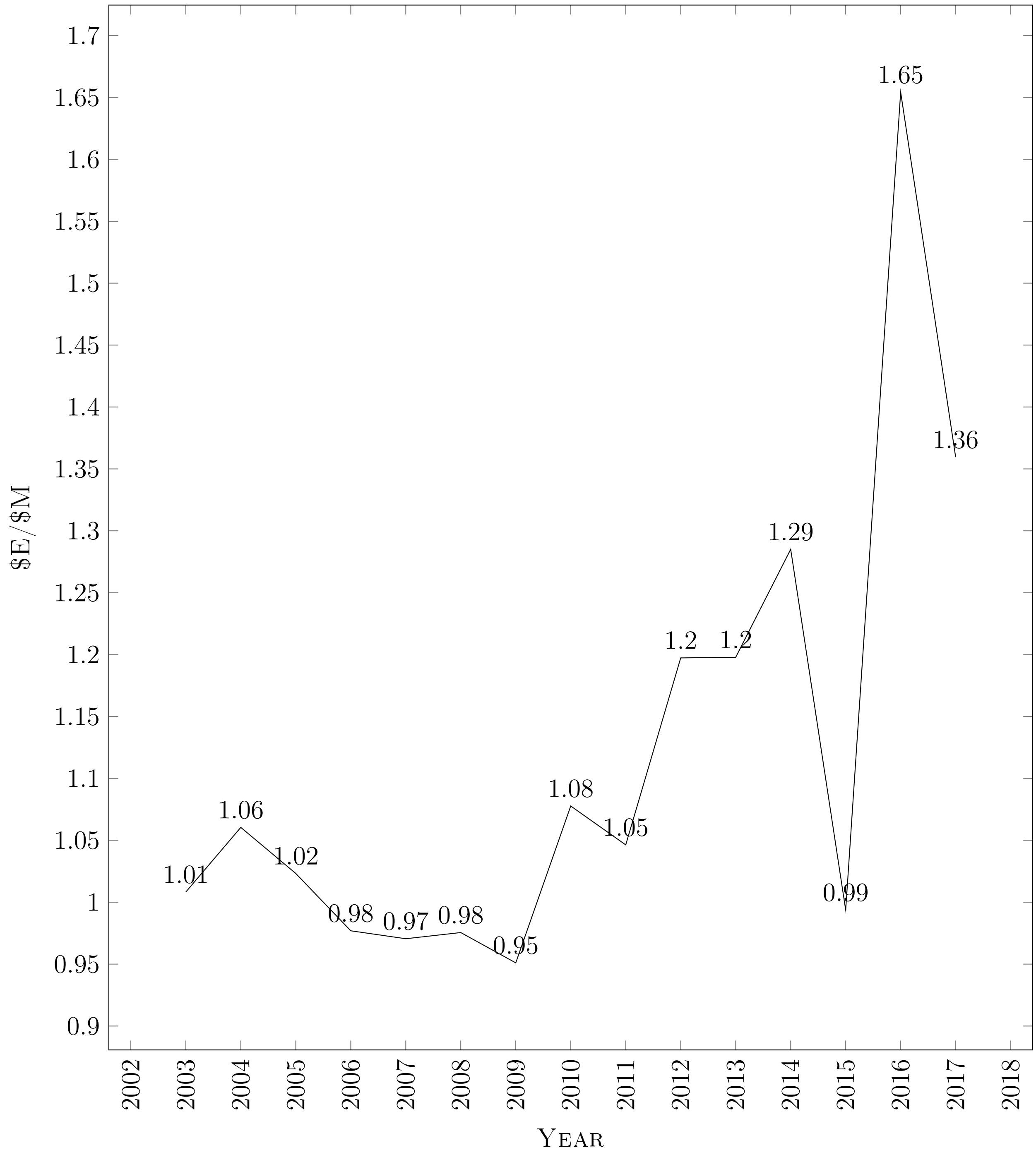

Figure 12: Fraction of evening peak average price ($E) over midday peak average price ($M) per year

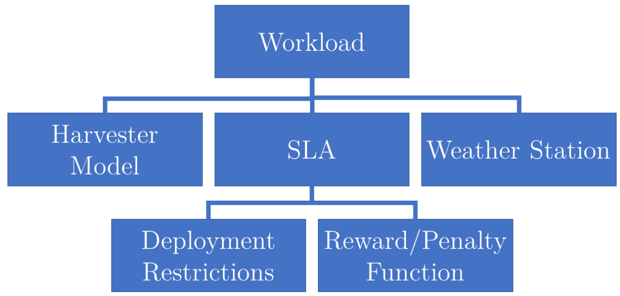

Figure 13: Workload components

into nine years of training data and two years of testing data is given in Section 4.2.0.2.) In Appendix A.4.1 and Appendix A.7.1, we describe how we derived the training and test data from the minute-by-minute Automated Surface Observing System (ASOS) data set DSI 6405 [58]. The sources of the ASOS data are ASOS weather stations, which usually have anemometer heights of approximately 10 or 8 meters; " Typical ASOS wind sensor heights are 33 feet or 27 feet, depending on local site-specific restrictions or requirements" [59]. (Please note that because windspeeds typically increase with altitude[17, p. 668] and wind turbines may be much higher than 10 meters, be careful to not underestimate wind resources at the site of a particular ASOS weather station. For information on assessing wind resources, please visit the National Renewable Energy Laboratory's web page entitled "Wind Resource Assessment" [65].) Where gaps existed, we interpolated the intervening minutes.

- A harvester model (e.g., see Appendix A.3), which defines the harvester's power curve (Section A.3.1), how much energy it takes to deploy, how much energy it takes to retract the harvester, and defines cut-in, cut-out, and re-cut-in windspeeds.

- An electricity-price-vs-time table for use with electricity-cost-dependent complementary metrics (Section 4.1.5). Because negative electricity prices can be counter-intuitive and because we use the sign of electricity prices in the calculation of the MQMP metric, we remind the reader that wholesale electricity prices may become negative. Negative prices do occur at some timestamps in the electricity-price-vs-time table we provide.

4.2.0.1 A function defining rewards/penalties (or incentives) Suppose that an algorithm a has just completed processing workload wrk. The $I(f(a, wrk), z)$ function maps $(f(a, wrk), z)$ to an incentive number $I()$ where $f(a, wrk)$ and z are defined here:

- The function $f(a, wrk)$ scores how well algorithm a processed workload wrk. The range of the function $f(a, wrk)$ is the ordered pair $(x \in \{0, 1\}, y \in \mathbb{R})$ where x indicates whether or not algorithm a violated the OLA and y indicates a score that algorithm a earned.
- The variable $z \in [1, 100]$ represents the value chosen by the community hosting a specific harvester to address the following situations. If algorithm a violated the OLA, then the variable z is the penalty. Otherwise, z specifies the percentage threshold that y must surpass to cause $I()$ to become positive. For example, if the argument passed to the function $I()$ is $(SCNetNorm(a, wrk), z)$, a way for a community near the KPIT weather station (Pittsburgh International Airport) to choose z is to use our test results for KPIT (which are based on KPIT's historical windspeeds) to determine a reasonable goal for $SCNetNorm(a, wrk)$ where OLA 3 is the OLA of workload wrk. Such a reasonable goal is the average of our algorithms' performances listed in Table 3 on the following page where $wrk = \{OLA3, ws = \text{KPIT}, hm = (Towered(Appendix\ A.3))\}$; the average is 0.32 or 32%. We set z to 32 in Equation 4.15 as part of an illustration.

Given the definitions of $f(a, wrk)$ and z immediately above, we define $I(f(a, wrk), z)$ to be

$$I((x, y), z) = \begin{cases} -z, & \text{for } x = 0 \text{ i.e., the OLA was violated} \\ J(y, z) & \text{for } x = 1 \end{cases} \tag{4.11}$$

where

$$J(y, z) = \frac{100}{z} y - 1. \tag{4.12}$$

The variables x, y, z have the following meanings, respectively:

- x is a binary variable indicating whether or not an algorithm a has violated the OLA while processing workload wrk.
- y is the value returned by function $f(a, wrk)$
- z is the threshold that y must surpass in order for $I()$ to be positive

Table 3: Compilation of algorithms' performances of processing KPIT's testing data for OLA 3

Algorithm	Variant*	NetNorm	Source Table
Static	0x0	0.39	29 on page 224
Static	0x2	0.37	35 on page 229
Aging	0x0	0.44	41 on page 234
Aging	0x2	0.30	47 on page 242
Fuzzy	0x0	0.21	53 on page 249
Fuzzy	0x2	0.21	59 on page 256
Average: 0.320			
*variant 0x0 uses current weather only. variant 0x2 uses weather prediction.			

For example, for the case when the function $f(a, wrk)$ is the OLA-compliance version of the NetNorm metric, i.e., $SCNetNorm(a, wrk)$, then $I()$ maps that metric to a reward/penalty number where z defines the threshold NetNorm must surpass in order for a to earn a reward. That is,

$$I(SCNetNorm(a, wrk), z) = \begin{cases} -z, & \text{for } SCNetNorm(a, wrk) = (0, NetNorm(a, wrk)) \\ J(NetNorm(a, wrk), z) & \text{for } (1, NetNorm(a, wrk)) \end{cases} \tag{4.13}$$

where $z \in [1, 100]$ and $SCNetNorm(a, wrk)$ is defined by Equation 4.5 and $J(NetNorm(a, wrk), z)$ is defined as

$$J(NetNorm(a, wrk), z) = \frac{100}{z}(NetNorm(a, wrk)) - 1 \tag{4.14}$$

where $z \in [1, 100]$ and $NetNorm(a, wrk)$ is defined by Equation 4.1.

We illustrate $I()$ with an example. When $z = 32$, if algorithm a's $NetNorm(a, wrk)$ score is better than 32%, then algorithm a earns a reward. If algorithm a scores less than 32%, then it receives a penalty. For example, suppose algorithm a's $NetNorm(a, wrk)$ score is 44% or 0.44. Then, the

$$\begin{aligned} J(NetNorm(a, wrk), z) &= \frac{100}{z}(NetNorm(a, wrk)) - 1, \implies \\ J(NetNorm(a, wrk), 32) &= \frac{100}{32}(0.44) - 1 = 3.125(0.44) - 1 = 1.375 - 1 = 0.375, \end{aligned} \quad (4.15)$$

which is a reward in this case where the $NetNorm(a, wrk)$ score is 44% since $0.375 > 0.0$. Note that $J(NetNorm(a, wrk), 32)$ is 0 if $NetNorm(a, wrk)$ is 32% or 0.32:

$$\begin{aligned} J(NetNorm(a, wrk), z) &= \frac{100}{z}(NetNorm(a, wrk)) - 1, \implies \\ J(NetNorm(a, wrk), 32) &= \frac{100}{32}(0.32) - 1 = 1 - 1 = 0. \end{aligned}$$

4.2.0.2 The training and testing partitions of the benchmarks Context: Suppose that a certain community is considering whether or not to install a retractable-wind-energy harvester at a specific site within a decade. Also suppose that if the community decides to contract a renewable-energy-systems integrator to install and to configure the retractable-harvester energy harvester, the community estimates that the system integrator will commission the completed system up to one year after the community signs the contract with the systems integrator. Thus, if the community were to have a retractable harvester system working before their deadline (i.e., before the decade ends), the community would need to sign the contract within nine years. To prepare to decide whether or not to install a retractable-wind-energy harvester at the specific site, the community places an anemometer. How many years of those nine should the community collect windspeed data to do the following:

- to model the site and,
- to provide retractable-harvester-control-algorithm training data to a renewable-energy-systems integrator (if the community decides to contract the renewable-energy-systems integrator to install and to configure the retractable-harvester energy harvester)?

Local data informed by a consultancy: One answer to how many years of data the community should collect depends on how many years a financial investor requires. In the following example, the answer is one year: "Following the completion of the 12-month measurement campaign. . . , [an international renewable energy consultancy] will provide a full, bank-grade site suitability report and energy yield analysis which will let [a specific energy developer] obtain the necessary lender's funding approval to progress [a 400 MW permanently deployed wind farm] project into construction [in Ethiopia]" [43]. "[The renewable energy consultancy] has developed in-house tools that produce probability of exceedance energy yield values (such as P75 and P90) typically used in project finance" [41].

The consultancy is able to generate a bank-grade report from one year of data about the suitability of a proposed wind farm that is permanently deployed, but is one year of data enough for a consultancy to generate a report about the suitability of a proposed farm that is retractable? The answer seems to be *yes*: Assuming that the one year of data is stored as a time sequence, then it seems that the consultancy would be able to feed that data into a simulator similar to the one that we are making open source (or into HOMER [34] if HOMER were able to model retractable harvesters). Because at this point the consultancy would have access to only one year of testing data and have no training data yet, the simulator would run a control algorithm using default settings, which might not be ideal, but would provide the consultancy with a baseline energy yield analysis. If the baseline yield surpasses a threshold set by a bank, then the proposed retractable wind farm project can move forward.

After the retractable wind farm project is approved by the bank, the consultancy could give the one year of data to the contracted renewable-energy-systems integrator[4]. The systems integrator would use the year's data from the measurement campaign training data in an effort to improve upon the default settings of the control algorithm (which has already surpassed the bank's threshold). After the retractable-harvester system has been installed, the control algorithm can continue to refine its settings while it is controlling a harvester that is producing energy. That is, the control algorithm can continue to improve while it is "on-the-job."

Local trends in windspeeds? Some studies suggest windspeed trends: "Least-squares

[4]The consultancy itself might provide systems integration services.

regression lines fit to the 30 yr time series [1961-1990] show that, on balance, mean monthly maximum winds are increasing within the United States and mean monthly minima are decreasing" [44]. "The two observational data sets both exhibit an overwhelming dominance of trends toward declining values of the 50th and 90th percentile and annual mean wind speeds, which is also the case for simulations conducted using MM5 [the Fifth-Generation National Center for Atmospheric Research / Penn State Mesoscale Model [54]] with (NCEP-2) [National Centers for Environmental Prediction Reanalysis 2 [19] [56]] boundary conditions. However, converse trends are seen in output from the North American Regional Reanalysis, other global reanalyses (NCEP-1 and ERA-40 [a reanalysis of over 40 years of data by the produced by the European Centre for Medium-Range Weather Forecasts (ECMWF) et al. [55] [97]]), and the Regional Spectral Model" [77]. Reanalysis of weather observations involves combining data from various weather-observation instruments to "[help] ensure a level playing field for all instruments throughout the historical record" [57].

How many years[5] of data are required to detect a trend, if any, in our ASOS-derived data, for each station?

About how many samples are required to detect a local trend, the Pacific Northwest National Laboratory offers direction via documentation of the Visual Sample Plan (VSP) software tool[6] entitled "Mann-Kendall Test For Monotonic Trend" [72]. ("The main objective of the Mann-Kendall test is to test the null hypothesis H_0 that there is no trend, against the alternative hypothesis H_1 that there is an upward or downward trend" [94].) However, the Mann-Kendall test assumes that the time-series data being tested is not autocorrelated. But, autocorrelated is often what sequential windspeeds are [7]. Fortunately, a modified Mann-Kendall test for autocorrelated sequences has been developed: "The accuracy of the modified test in terms of its empirical significance level was found to be superior to that of the original Mann-Kendall trend test without any loss of power[8]" [31]. Because of that

[5] A year makes sense to choose as a distance between samples because winds are affected by seasons of the year.

[6] "Visual Sample Plan (VSP) is a software tool for selecting the right number and location of environmental samples so that the results of statistical tests performed on the data collected via the sampling plan have the required confidence for decision making" [73].

[7] "Wind speed time-series data typically exhibit autocorrelation, which can be defined as the degree of dependence on preceding values"[35].

[8] "[T]he power of the test [is] defined as the probability of rejecting [the null hypothesis] when the alter-

superiority, we assume that if the VSP tool were to use the modified Mann-Kendall test, the tool would calculate the same number of samples or less than what the actual VSP tool calculates.

The VSP tool presents parameters on which the number of samples depends. Those parameters are listed directly below; each parameter is followed by a choice enclosed in parentheses. (All numeric choices are the defaults of the VSP tool except for the choice of $x = 2$ units in item 3 below, which is less than the VSP tool's default choice: $x = 10$ units.)

1. What type of trend to detect: upward, downward, or either. (Suppose that the community chooses to detect "either an upward or downward trend".)
2. Whether the expected trend is linear or exponential. (We assume that if a windspeed trend is found, the trend will be linear since Torralba estimated a linear trend involving windspeeds[93].)
3. What the desired confidence level is that a change in x units per year will be detected. (Suppose that the community desires a 90% confidence percentage that a change in 2 units per year will be detected instead of the default 10 units per year because TODO)
4. The desired percentage chance that, if there is no trend, that a trend will be falsely detected. (Suppose that the the community desires that there is no more than a 5% probability that a trend will be falsely detected.)
5. What the estimated standard deviation of the residuals — A residual is the observed value less the predicted value, i.e., $residual = observedValue - predictedValue$ — from the regression line is? (Suppose that the community estimates that the standard deviation from the residuals is 3 units.)

If the choices specified in the parenthetical statements directly above are made, then the Visual Sample Plan tool calculates that 9 sampling periods (i.e., 9 samples spaced one year apart) are needed. "[The Visual Simple Plan software tool] uses a Monte-Carlo simulation to determine the required number of points in time, n, to take a measurement in order to detect a linear trend for specified small probabilities that the [Mann-Kendall] test will make decision errors" [73]. Thus, let us choose 9 years of data to be training data: the first year

native hypothesis is true" [31].

is provided by the consultancy and the remaining 8 are years that the algorithm trains itself on-the job. After that 9 years of training are over, let us benchmark the algorithm.

If we follow a rule of thumb in machine learning that 80% of data be training and 20% be testing, our choosing 9 years to be training data means we should have approximately 2 years of testing data. (For each station, the algorithm using the 9 years of training data can find the Weibull scale and shape parameters of the data to create a validation time-series via a time-series model [26, Section 4] because Weibull distributions are commonly used to describe windspeeds [26] [84]. Our estimates for the Weibull scale and shape parameters for each station's training data is given in Appendix A.17.)

Analyzing station KATL's training data for trends: We summarize here the results of an example in Appendix A.18 where we look for trends in station KATL's training data. Specifically, we looked for trends in yearly estimates of the shape (B) and scale (A) parameters of a Weibull distribution: "[I]ts [probability density function] has the form

$$f(x) = \begin{cases} \frac{B}{A}\left(\frac{x-y}{A}\right)^{B-1} \exp\left[-\frac{(x-\nu)}{A}^{B}\right], & x \geq \nu \\ 0, & \text{otherwise" [6, Equation 5.45, p.185]} \end{cases} \tag{4.16}$$

where ν is the location parameter.

We found that if we set the significance level A to 0.05, then we behave in a manner consistent with our conclusion that no monotonic trend is present in either KATL's scale or shape parameters even though there is a 0.05 possibility that our conclusion is wrong. The significance level of 0.05 is the probability that we commit a Type I error, i.e., that we falsely reject the null hypothesis H_0 : No monotonic trend exists. For details, please see Appendix A.18.

Trends in electricity-price profiles: As described in Section 4.1.5.2, the MQMP metric depends on the sign of the hourly Ontario electricity price (HOEP) in an hour-by-hour file that we provide. Is the number of hours that the HOEP is negative trending upwards? Yes, according to the surface plot shown in Figure 14.

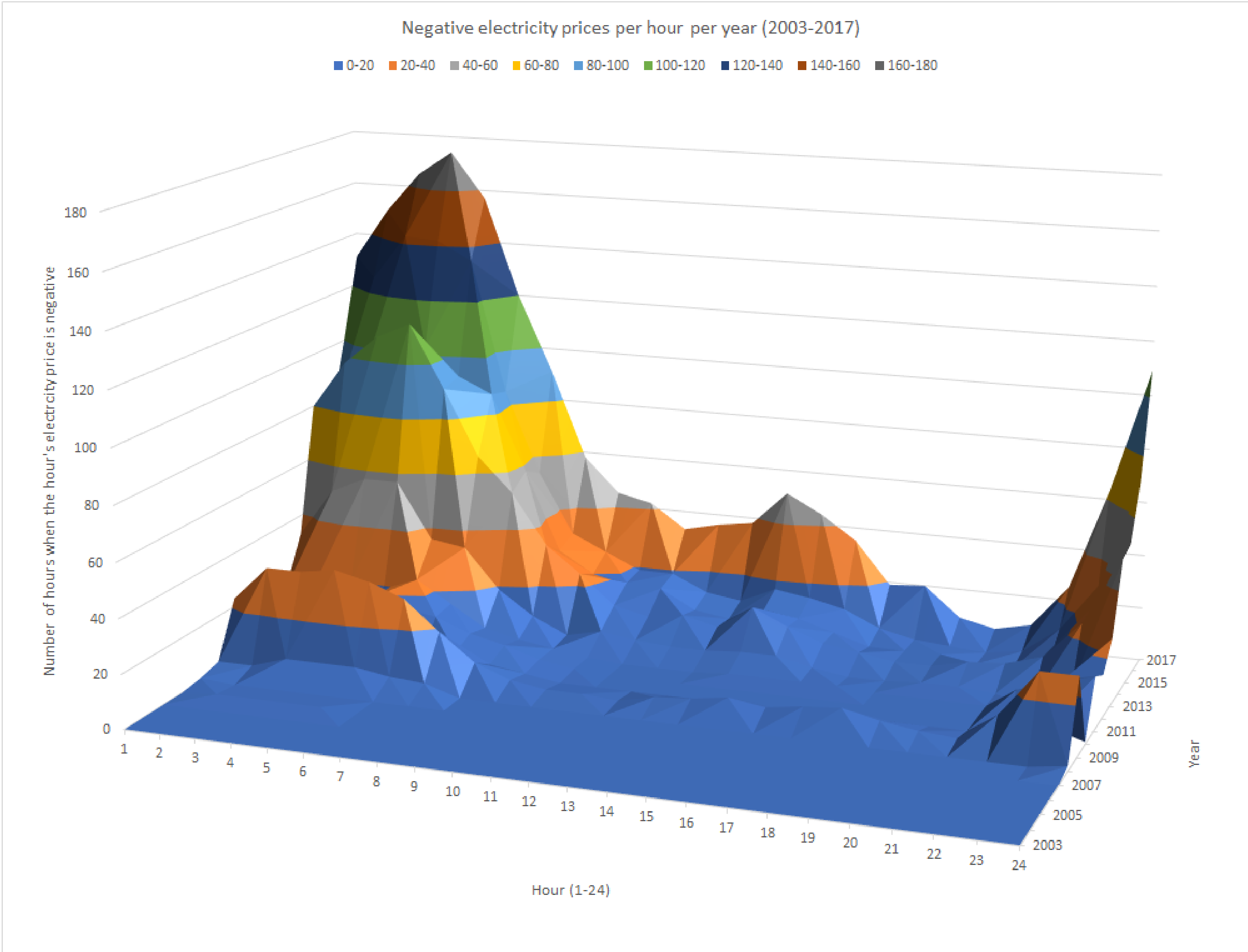

Figure 14: Surface plot of number of hours when electricity prices are negative per hour of day per year

4.2.0.3 Simulated windspeed forecasts Because we do not have the actual weather forecasts associated with our training and testing data, we simulate predicted day-ahead windspeeds. (We supply in the benchmarks simulated predicted day-ahead windspeeds for control algorithms that use predicted windspeeds.) Because predicted day-ahead windspeeds are 24 hours or 1440 minutes into the future, the simulated predicted day-ahead windspeeds are in the column labeled "f1440" in each training and each testing benchmark file (e.g., trainingKPIT2004-2012in.csv and testingKPIT2013-20014in.csv, respectively). In each training and testing benchmark file, for rows having timesteps that are less than one-day away from the end of the file, to indicate that weather prediction is not available, we placed a "-1" in each of those specific rows. (The name "f1440" also suggests that other columns such as "f60" may be added to future versions of this benchmark suite.)

We simulated the predicted day-ahead windspeeds by basing simulated prediction errors on a study of six onshore, windfarms where windspeed prediction errors were found to be Gaussian [47] and that the relative standard deviation (we are assuming that the standard deviation is relative to the mean windspeed) of day-ahead windspeed prediction errors for a certain site was approximately 0.3 [47, Figure 3].

For example, in Appendix A.8, we give the mean windspeed for each station as measured by our benchmark files. The station having the lowest average windspeed is KSAN (San Diego) at 5.13 knots. Thus, using our assumption that the standard deviation of day-ahead windspeed prediction errors for a site is approximately 0.3 times that site's average windspeed, the standard deviation for the day-ahead prediction error for KSAN is 0.3×5.13 knots $= 1.54$ knots.

Let us compare the standard deviation of KSAN's day-ahead prediction error of 1.54 knots to the standard deviation of the prediction error given by Kavasseri [42, Table 1], who studied four sites. Kavasseri reports the following day-prediction variance for one site because the variance is comparable to the variances of the other three sites: a variance of 0.156 mph, which is a standard deviation of 0.39 mph or 0.34 knots. Thus, it seems reasonable to assume that the standard deviation of KSAN's day-ahead prediction error of 1.54 knots (which is 4.5 times Kavasseri's standard deviation) is not overly optimistic. Although, there is a danger is being overly pessimistic, it seems better to underestimate prediction accuracy than to overestimate it for this initial release of the benchmark suite so that algorithm designers do not overly depend on weather prediction data, but assess its accuracy in the field. For example, an algorithm designer can track prediction accuracy in the field and design the algorithm to increase its use of weather predictions if the actual predictions achieve a certain accuracy consistently.

4.3 FINDING "IDEAL" DEPLOYMENT (WAKE) AND RETRACTION (SLEEP) SCHEDULE FOR OLA5, WHICH IS TRANSITION LIMITED

To find the "ideal" deployment (analogous to waking) and retraction (analogous to going to sleep) timestamps (i.e., times at which an algorithm should direct a harvester to deploy or retract) to use to train an algorithm to process a workload (Section 4.1.3) having OLA5, which is limited to two state transitions per month, as its OLA and ws as its weather station, we created an acyclic, weighted, directed graph and then applied Dijkstra's Shortest Path Algorithm. (Recall that *state transition* refers to the harvester's going from being fully deployed to being fully stowed or vice-a-versa.)

4.3.1 Finding the best path through the best monthly instances of deployment patterns

The acyclic directed graph has five nodes for each month (e.g., Figure 15) of the station ws's training data. The five nodes represent five possible deployment patterns that comply with the two-state-transition-per-month maximum limit and represent the best instance of that deployment pattern—How we found each best instance is explained in Section 4.3.2—within the month where *best* is defined as netting the most energy (we inverted the energy value to use the shortest-path algorithm):

1. ONE_RISE_ONE_FALL: The harvester starts the month retracted, deploys and then retracts using exactly its monthly allocation minutes that it may be visible (x minutes where OLA5 defines x as 8760 minutes). The harvester finishes the month retracted.

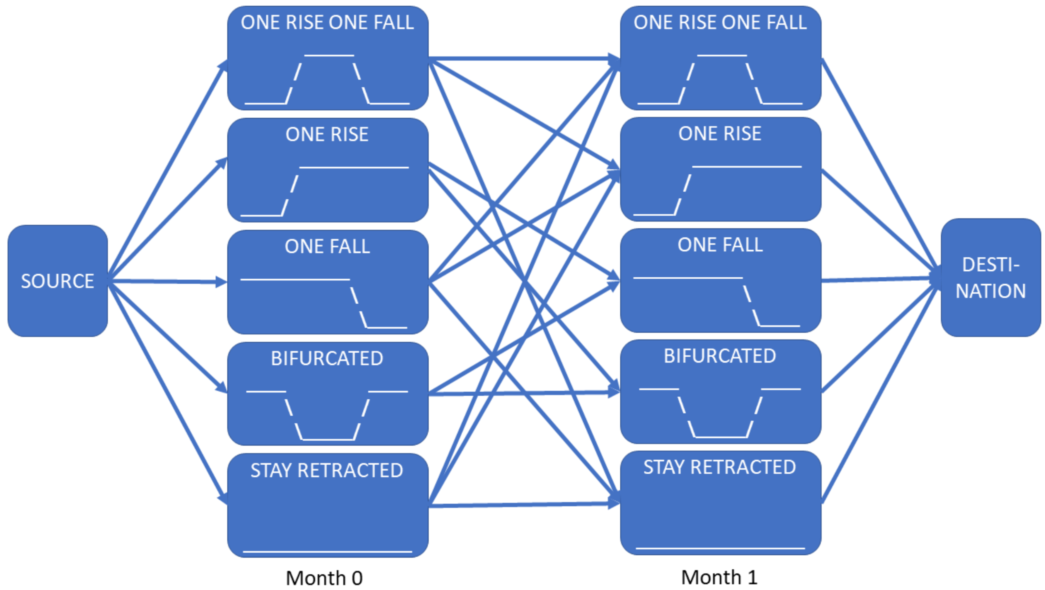

Figure 15: Acyclic directed graph of possible choices for two months. Each node has a weight value (not shown) derived from how much energy the instance that the node represents nets.

2. ONE_RISE: The harvester starts the month retracted, waits until there are x minutes remaining in the month, deploys, and then finishes the month deployed.
3. ONE_FALL: The harvester starts the month deployed. It then waits until it has been visible for x minutes less the time it takes to retract. It then retracts to finish the month retracted.
4. BIFURCATED: The harvester starts and ends the month deployed. Between those two visible periods, the harvester remains stowed.
5. STAY_RETRACTED: The harvester remains stowed for the entire month.

We connected directed edges from each node v of the five nodes of each month to v's compatible nodes of the next month. A node v of a particular month is compatible to each node w_i of the next month if v ends the month in the same state in which w_i begins (where $i \in 1, 2, \ldots, 5$). For example, if v was ONE_RISE_ONE_FALL, then we connected v to w_1, w_2, and w_5, which are ONE_RISE_ONE_FALL, ONE_RISE, and STAY_RETRACTED, respectively.

Our acyclic graph has one start node and one destination node. We found the shortest path from the start node to the destination node through one node of each month to find the "ideal" deployment and retractions timestamps using Dijkstra's shortest path algorithm. The resulting lowest-cost path for each weather station has 9×12 monthly vertices (since the training data is 9 years worth) plus the source and destination vertices. We then traversed the shortest path to create the column labeled "OLA5" in each station's training data file.

We repeated the process described above for each station of the thirty weather stations.

4.3.2 Finding the best instance of each deployment pattern

Above, we described five deployment patterns. Here, we explain how we found the best instance of each deployment pattern. For each month, we created an acyclic graph where each edge has weights reflecting visibility-time used and energy netted. A three-minute example is shown in Figure 16 where

- the deployment time and retraction time (Appendix A.3.5) are one minute each,
- every node labeled 'Rn' and 'Dn' represents the harvester in a fully retracted state or fully deployed state, respectively, at minute n,
- each edge has three weights:
  - energy netted (measured in kilowatt-hours),
  - visibility-time used (measured in minutes), and
  - a cost, which is an inverse of the energy netted.

For each deployment pattern for each month (except the ONE_FALL and BIFIRCATED patterns for the first month of each station's training data because we are assuming that the harvester starts the simulation fully stowed and except the STAY_RETRACTED pattern because we assumed that the harvester consumes no energy when it is stowed), we traversed the monthly acyclic graph to find best instance, i.e., the deployment-pattern instance that nets the most energy within the monthly visibility-time budget.

Traversing each month's acyclic graph (a subset is shown in Figure 16) for each deployment pattern involves examining every possible choice consistent with that pattern and that uses OLA 5's visibility limit of 8760 minutes of each month. For example, examining the

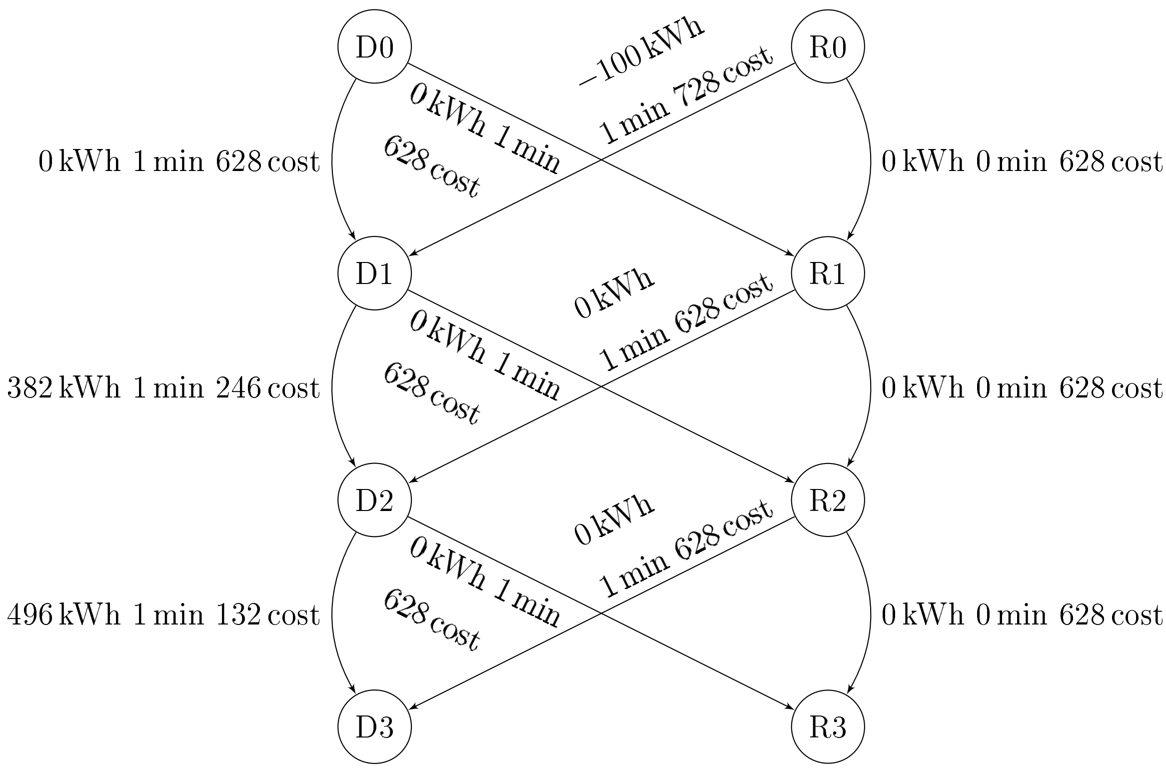

Figure 16: A three-minute example of an acyclic directed graph where each node represents a harvester's state at a certain timestep.

ONE_RISE_ONE_FALL pattern involves calculating the cost when the harvester begins deploying at the first minute of the month ($t_d = 0$) and completes retracting so that it is fully stowed 8760 minutes later ($t_s = t_d + 8760$) and then repeating the calculation of the cost when the harvester begins deploying at the second, third, forth, and so-on minutes of the month until the minute that the harvester completes retracting is the last minute of the month ($t_d = 1, 2, 3, \ldots, x - 8760$ where x is the index of the last minute of the month). We store the deploying timestamp t_d that achieves the lowest cost, the retraction timestamp (t_d less the time the harvester takes to retract), and the cost itself for use the process described in Section 4.3.1. Thus, finding the best instance of ONE_RISE_ONE_FALL for a month takes y iterations where y is 8760 minutes subtracted from the number of minutes in the month (e.g., a 31-day month has 31 times 1440 minutes or 44640 minutes). For a much simpler example, examining the ONE_RISE case each month involves calculating only one instance, where the harvester deploys 8760 minutes before the end of the month.

4.4 LIST OF FILES PROVIDED BY THIS BENCHMARK SUITE

The benchmark suite provides the following files:

1. Thirty training files, each with approximately nine years of minute-by-minute windspeeds and simulated predicted day-ahead windspeeds (Appendix A.4.1 describes the files' fields and naming convention.)
2. Thirty testing files, each with approximately two years of minute-by-minute windspeeds and simulated predicted day-ahead windspeeds (Appendix A.4.1 describes the files' fields and naming convention.)
3. An hourly electricity price file (Appendix A.4.2 provides the file's name and information about its contents.), which as noted in the related work section (Section 2), has eleven years of hour-by-hour electricity prices which may become negative at times.
4. A fuzzy-set-membership-function file describing membership in the set NOT WINDY AT <STATION> for each of the 30 weather stations (The file's naming convention and example content are in Appendix A.2.1.1.)

5. A file containing sunset[9] for the city of St. Louis, Missouri, for all days in the years 2004-2014, inclusive, which are used in the definition of *quiet hours* for St. Louis, the only municipality of the 30 that bases its quiet hours on sunsets (e.g., Pittsburgh's quiet hours run from 10 p.m. to 7 a.m.; Tampa's quiet hours end at 7 a.m. on Monday - Friday., 8 a.m. on Saturday, and 10 a.m. on Sunday, and begin at 6 p.m. everyday. (Appendix A.10).

6. A file delineating when noise is allowed for municipalities corresponding to the 30 weather stations (`Noise_Allowed_Time_Definitions_All_Stations.csv`)

4.5 RECAP OF THE BENCHMARKS

This work provides standard workloads (each of which is comprised of a standard operation limitation agreement (OLA), a weather station, and a harvester model) and metrics to advance the development of retractable-harvester control algorithms including algorithms that use weather prediction, fuzzy logic, and machine learning. The operation limitation agreements were derived largely from a survey of over 300 respondents from across the United States. The windspeed data is minute-by-minute windspeed data that covers approximately all minutes over eleven years for 30 weather stations. (That is, we provide approximately 330 years of minute-by-minute windspeed data.) Also provided is a file of hourly electricity prices for those eleven years.

The minute-by-minute data includes simulated weather forecasts because we assume that algorithms that use forecasts will perform better than those that do not. To facilitate machine learning, we provide "ideal" minute-by-minute deployment and retraction signals in the minute-by-minute training data for all 30 weather stations to comply with a sample operation limitation agreement. For each of the 30 weather stations, we provide a fuzzy-set-membership function to assign degrees of memberships to windspeeds.

[9] Only sunset (and not sunrise times) are provided because St. Louis's quiet hours start at sunset and end at 6 a.m. (and span each Sunday).

4.6 COMPARATIVE AND SENSITIVITY ANALYSIS

4.6.1 Comprehensive results: Comparing performance of each pertinent strategy (i.e., algorithm variant) per each OLA per station

We present the full results of running the algorithms for each OLA for each of 30 weather stations in the appendix. In this section, we present a summary of the results and analyses.

The scores typeset in bold in Table 4 show the highest average scores for each OLA. For OLAs 1 and 2, Revision 1.2 of Aging variants 0x0 and 0x2 outscored the Static and Fuzzy-Crisp categories, earning average NetNorm and MQNetNorm scores of 0.6218 and 0.5722, respectively. For OLAs 3 and 4, Revision 1.2 of Static variant 0x0 scored the highest average NetNorm and MQNetNorm scores: 0.4633 and 0.4224, respectively. For OLAs 5 and 6, Revision 1.1 of Static variants 0x1 and 0x3 were the top scorers, with average NetNorm and MQNetNorm scores of 0.2633 and 0.2477, beating the newer revisions.

Table 4: Explorations 1 & 11: Comparisons of average (μ) performance of all applicable variants for all stations for each OLA of 1 through 6

Purpose of Explor'ns 1 & 11 (Parts 1 & 2): Compare performance of all applicable variants for all stations for OLAs 1 and 2										Results					
								Ranges Training Explored		NetNorm			MQNetNorm		
Exp.	OLA(s)	Algorithm Family	Variant	Rev.	Station(s)	λ	R.T.D. (knots)	deployment threshold of running average windspeed (knots)	window size of moving average windspeed (samples or minutes)	μ	σ	σ/μ	μ	σ	σ/μ
1	1 & 2	Static	0x0	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.6097	0.1307	21%	0.5543	0.1208	22%
1	1 & 2		0x2	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.6090	0.1311	22%	0.5343	0.1200	22%
11	1 & 2	Static	0x0	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.6095	0.1311	22%	0.5545	0.1204	22%
11	1 & 2		0x2	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.6095	0.1311	22%	0.5545	0.1204	22%
								y-intercept (knots)							
1	1 & 2	Aging	0x0	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.6063	0.1291	21%	0.5553	0.1198	22%
1	1 & 2		0x2	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.5950	0.1507	25%	0.5323	0.1183	22%
11	1 & 2	Aging	0x0	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	**0.6218**	0.1310	21%	**0.5722**	0.1211	21%
11	1 & 2		0x2	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	**0.6218**	0.1310	21%	**0.5722**	0.1211	21%
								deployment threshold of membership value in resulting fuzzy set							
1	1 & 2	Fuzzy-Crisp	0x0	1.4	all	0.9	1	0.5	[1, (step 30), 121]	0.5597	0.1204	22%	0.5050	0.1166	23%
1	1 & 2		0x2	1.4	all	0.9	1	0.5	[1, (step 30), 121]	0.5337	0.1129	21%	0.4827	0.1074	22%
11	1 & 2	Fuzzy-Crisp	0x0	1.5	all	0.9	1	0.5	[1, (step 30), 121]	0.5599	0.1203	21%	0.5042	0.1106	22%
11	1 & 2		0x2	1.5	all	0.9	1	0.5	[1, (step 30), 121]	0.5338	0.1126	21%	0.4829	0.1070	22%

Purpose of Explor'ns 1 & 11 (Parts 3 & 4): Compare performance of all applicable variants for all stations for OLAs 3 and 4										Results					
								Ranges Training Explored		NetNorm			MQNetNorm		
Exp.	OLA(s)	Algorithm Family	Variant	Rev.	Station(s)	λ	R.T.D. (knots)	deployment threshold of running average windspeed (knots)	window size of moving average windspeed (samples or minutes)	μ	σ	σ/μ	μ	σ	σ/μ
1	3 & 4	Static	0x0	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.3723	0.0633	17%	0.3343	0.0637	19%
1	3 & 4		0x2	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.3723	0.0633	17%	0.3343	0.0637	19%
11	3 & 4	Static	0x0	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	**0.4633**	0.0819	18%	**0.4224**	0.0735	17%
11	3 & 4		0x2	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.4088	0.0719	18%	0.3751	0.0660	18%
								y-intercept (knots)							
1	3 & 4	Aging	0x0	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.4020	0.0679	17%	0.3710	0.0617	17%
1	3 & 4		0x2	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.3430	0.0748	22%	0.3100	0.0648	21%
11	3 & 4	Aging	0x0	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.3622	0.0692	19%	0.3387	0.0641	19%
11	3 & 4		0x2	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.3852	0.0706	18%	0.3569	0.0652	18%
								deployment threshold of membership value in resulting fuzzy set							
1	3 & 4	Fuzzy-Crisp	0x0	1.4	all	0.9	1	0.5	[1, (step 30), 121]	0.3960	0.0849	21%	0.3610	0.0710	20%

Table 4: (continued)

1	3 & 4		0x2	1.4	all	0.9	1	0.5	[1, (step 30), 121]	0.4520	0.0771	17%	0.4073	0.0732	18%
11	3 & 4	Fuzzy-Crisp	0x0	1.5	all	0.9	1	0.5	[1, (step 30), 121]	0.4333	0.0774	18%	0.4031	0.0718	18%
11	3 & 4		0x2	1.5	all	0.9	1	0.5	[1, (step 30), 121]	0.4106	0.0844	21%	0.3724	0.0769	21%

Purpose of Explor'ns 1 & 11 (Parts 5 & 6): Compare performance of all applicable variants for all stations for OLAs 5 and 6										Results					
								Ranges Training Explored		NetNorm			MQNetNorm		
Exp.	OLA(s)	Algorithm Family	Variant	Rev.	Sta-tion(s)	λ	R.T.D. (knots)	deployment threshold of running average windspeed (knots)	window size of moving average windspeed (samples or minutes)	μ	σ	σ/μ	μ	σ	σ/μ
1	5 & 6	Static	0x1	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	**0.2663**	0.0302	11%	**0.2477**	0.0297	12%
1	5 & 6		0x3	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	**0.2663**	0.0302	11%	**0.2477**	0.0297	12%
11	5 & 6	Static	0x1	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.2479	0.0256	10%	0.2344	0.0267	11%
11	5 & 6		0x3	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.2481	0.0254	10%	0.2345	0.0265	11%
								y-intercept (knots)							
1	5 & 6	Aging	0x1	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.2010	0.0215	11%	0.1867	0.0210	11%
1	5 & 6		0x3	1.1	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.2010	0.0217	11%	0.1860	0.0212	11%
11	5 & 6	Aging	0x1	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.2255	0.0211	9%	0.2122	0.0215	10%
11	5 & 6		0x3	1.2	all	0.9	1	[$ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}$+30]	[1, (step 30), 121]	0.2261	0.0225	10%	0.2115	0.0210	10%
								deployment threshold of membership value in resulting fuzzy set							
1	5 & 6	Fuzzy-Crisp	0x1	1.4	all	0.9	1	0.5	[1, (step 30), 121]	0.2287	0.0321	14%	0.2133	0.0314	15%
1	5 & 6		0x3	1.4	all	0.9	1	0.5	[1, (step 30), 121]	0.2323	0.0353	15%	0.2170	0.0349	16%
11	5 & 6	Fuzzy-Crisp	0x1	1.5	all	0.9	1	0.5	[1, (step 30), 121]	0.2135	0.0277	13%	0.1979	0.0281	14%
11	5 & 6		0x3	1.5	all	0.9	1	0.5	[1, (step 30), 121]	0.2510	0.0314	13%	0.2336	0.0316	14%

4.6.1.1 Weather prediction

A collection of the averages of all the test results are shown in Table 4. A surprising result for these initial revisions of the control algorithms is that their weather-prediction-using variants did not always score higher than then the variants not using weather prediction. For not surpassing the current-weather-only variants, there are at least four possible reasons: 1. The predicted weather was not accurate enough to help, 2. The predicted weather had an unhelpful time horizon (of one day), 3. The algorithms did not use the day-ahead weather predictions effectively, and/or 4. One or more variants had a bug (or unintended feature in some cases).

Let us address reason 4 first. A bug (or unintended feature) was found in the following variants, which have corrected in later revisions. Some "current-weather-only" variants actually used weather prediction. Those variants updated their moving average windspeed twice per timestep. Those variants first updated their moving averages with current weather and then with future weather. The affected "current-weather-only" variants are Static0x1 (Rev. 1.1), Aging0x1 (Rev. 1.1), and Fuzzy-Crisp0x1 (Rev. 1.4). Those variants should not be called "current-weather-only" because they interlace simulated predicted windspeeds in their moving window of windspeeds.

However, that method of interlacing current and future windspeeds is acceptable to be used in the "weather-prediction-using" variants. And that interlacing method, in fact, is used by Static0x3 (Rev. 1.1), Aging0x3 (Rev. 1.1), and Fuzzy-Crisp0x3 (Rev. 1.4).

Note that Revision 1.2 of Static0x3, which does *not* use interlacing, performed worse than Revision 1.1 of Static 0x3, which does use interlacing. Thus, in that case, interlacing can be considered to be an unintended feature. Static0x3 (Rev. 1.1) tied Static0x1 (Rev. 1.1) in the OLA-5-and-6 comparison and are the highest scoring variants for OLAs 5 and 6 (Table 4).

Static0x3 (Rev. 1.1) and Static0x1 (Rev. 1.1) scored exactly the same as each other because the interlacing causes the future-checking conditional in Static0x3 to always be false. Likewise, Aging0x3 (Rev. 1.1) and Aging0x1 (Rev. 1.1) scored exactly the same for the same reason. The weather-prediction-using conditional evaluates to true only when the day-ahead weather is much windier than today. Interlacing causes the algorithm to deem the day-ahead windspeed average as the same as the current one.

Static0x0 (Rev. 1.1), Aging0x0 (Rev. 1.1), and Fuzzy-Crisp0x0 (Rev. 1.4) are unaffected by the unintended interlacing.

Static0x2 (Rev. 1.1) and Aging0x2 (Rev. 1.1) are. Their weather-prediction-using conditionals always evaluate to false.

Because revision 1.2 of Static and Aging do not use interlacing, let us examine possible reason 3 for those revisions. The transition-limited variants (i.e., variant 0x3 of Static, Aging, and Fuzzy-Crisp) require the weather to be windy today and tomorrow before deploying or require the weather to be much windier tomorrow than today before deploying. Thus, if the weather is windy today and calm tomorrow, the harvester will forfeit today's wind. However, in the Static and Agings algorithms, Table 73 on page 290 seems to imply that those conditions are never met because the non-weather-prediction-using and weather-prediction-using variants of the transition-limited Static and Aging algorithms score identically. (Verifying that the conditions are never met by adding detection code to Static and Aging and then re-running the tests for OLAs 5 and 6 are left for future work.) Thus, reason 3 seems to apply to the transition-limited, weather-prediction-using Static and Aging variants.

In the case of Fuzzy-Crisp (Revision 1.4), Table 73 on page 290 shows that, for OLA 5, the weather-prediction-using variant outperformed the non-weather-prediction-using variant by an average NetNorm score of 0.02 points for 12 stations, tied for 11 stations, under-performed by an average NetNorm score of −0.01 points for 7 stations, and for OLA 6, the weather-prediction-using variant outperformed the non-weather-prediction-using variant by an average MQNetNorm score of −0.02 for 13 stations, tied for 11 stations, under-performed by an average MQNetNorm score of −0.01 points for 6 stations. (We leave for future work a similar examination of Revision 1.5.)

The transition-unlimited variants (i.e., variant 0x2) consider tomorrow's weather when the harvester has nearly exhausted its monthly allocated visibility time. In an effort to save visibility time, the harvester will stow if tomorrow is predicted to be much windier than today. Thus, the harvester will consume 20 minutes of allocated visibility time stowing, without harvesting, and then consume 20 minutes to deploy tomorrow, before harvesting. It would seem that the algorithms would use weather prediction more effectively if the algorithms were to consider how much energy is forfeited during those 40 minutes.

A fuller description of how the algorithms use weather prediction is in Appendix 3.6.2.

Possible reason 2 is that the predicted weather had an unhelpful time horizon (of one day). Suppose that the algorithms were using a 20-minute time horizon. When the algorithm saw that the weather would likely be windy in 20 minutes, the algorithm could begin to deploy the harvester. However, OLAs 1-4 require that the harvester be stowed when the weather is not windy. Would it be a violation of OLAs 1-4 if the harvester were to be deploying when the weather seems to be calm? Technically, we defined the weather to be windy when the current rolling average is above a certain threshold, not when the *future* rolling average is. Thus, we recommend that a future revision of this benchmark suite modify OLAs 1-4 to allow the harvester to be visible during the x-minutes it takes the harvester to deploy if the x-minute-ahead future rolling average is above the windiness threshold.

Possible reason 1 is that the predicted weather was not accurate enough to help. Because predicted weather is being used in the field to predict wind farm power output, we know that technology exists to produce useful weather predictions for the wind power industry[10] However, we analyze in Appendix A.14 how much help, in the worst case (KBOS since it has the highest mean windspeed thereby having the highest standard deviation of simulated windspeed prediction errors), our simulated weather predictor provides. The predictor, of course, carries a probability of causing an algorithm to make a wrong prediction about tomorrow's windiness (e.g., 36% if the actual windspeed is 1 knot away from the mean windspeed at KBOS). Thus, let us revisit possible reason 3: The algorithms did not use the day-ahead weather predictions effectively. Future revisions of the algorithms can use weather predictions effectively if the algorithms take into account the error distribution of the weather predictor.

4.6.1.2 Running average window size The results for each station, each OLA, each algorithm, and each variant are given in Appendices A.11 to A.13. The results indicate that the training routine tends to choose a running-average-window size that is higher for the even OLAs than for the odd OLAs. The odd OLAs use NetNorm; the even OLAs use

[10] For example, "[A] system, which came online in September 2009, has already reduced wind energy prediction error by 40% and, in 2010 alone, saved Xcel Energy's ratepayers over $6 million" [79].

MQNetNorm. Higher running-average-window sizes lessen the possibility that the control algorithms start to deploy and then retract without harvesting any energy. Deploying uses energy. The MQNetNorm metric measures how often a harvester uses energy while the grid needs energy. The grid needs energy, as defined by a positive price in the hourly electricity price file we provide in this benchmark suite, 98% of the eleven years of hourly prices that we provide. Thus, if an algorithm starts to deploy and then retracts without harvesting any energy, the algorithm is highly likely to use energy while the grid needs energy. Using energy while the grid needs energy is penalized by the MQNetNorm metric. Therefore, the training algorithm tends to choose running-average-window sizes that are higher for OLAs that use the MQNetNorm metric than for OLAs that use NetNorm.

The tables in Appendix A.11 also show that the running average window size chosen by the trainer to maximize MQNetNorm was sometimes at the upper extreme of our design space (i.e., 121 samples). Thus, we investigated whether future revisions of the algorithms should explore beyond that design space. For example, because the training algorithm had chosen 121 samples to be the window size for KDEN for OLA 2 for Static variant 0x0 (Table 27), we, in a set of side tests, increased the upper limit of our design space to 241 samples and found that training chooses 177 samples. (The side tests involved exploring the following sequences: [1 (step 30) 241], [121 (step 15) 241], [155 (step 5) 211], and [176 (step 1) 196].) However, increasing the upper limit of the design space to an even higher number, 361, did not happen to cause the training algorithm to chose a number higher than 121 minutes for Static variant 0x0's processing the data for all 30 stations in the context of OLAs 3 and 4 (Table 85 on page 333). We compare in Table 5 on page 91 relatively sparse and dense searches for the alphabetically first four stations: KATL, KBOS, KBWI, and KCLE. Note that for each the four stations the densely searching training routine chose a running average window size less than 121 minutes and improved NetNorm scores by 13%, 3%, 10%, and 18% in the context of OLA 3 and improved MQNetNorm scores by 14%, 0%, 11%, and 16%. The average improvements for NetNorm and MQNetNorm scores for all 30 stations are 17% and 15%, respectively (compare Table 85 to Table 29 on page 224). Thus, increasing the granularity of the search of the running average window size and the search of the windspeed deployment threshold might be more effective than only increasing the upper limit of the window size's design space for Static. However, for Fuzzy-Crisp, it make sense to increase the upper limit of the design space beyond 121 minutes since, as noted below in Section 4.6.5.3, training often chose window sizes above 121 minutes during explorations of an extended design space.

A surface plot is shown in Figure 17 on page 97 where Static variant 0x0 densely explored the design space (i.e., the size of the windspeed running (or moving) average window size and the windspeed deployment threshold) to try to find the best settings to use to process station KATL's data within the constraints of OLA 3. That dense exploration found 9 knots and 40 minutes as the deployment threshold and the running average window size, respectively (Table 5). Thus, we show a surface-plot slice where the running average wind size is held at 40 minutes.

The slice bows upward, which is expected since too low deployment thresholds cause the harvester to use visibility time during lower-power wind conditions and too high deployment thresholds cause the harvester to tend to under use visibility time.

For Static and Aging variants (Revision 1.1) and Fuzzy-Crisp variants (Revision 1.4), itemized results per algorithm per station per OLA are given in Appendix A.20. We predict that future revisions of Fuzzy will score higher than these early attempts to write Fuzzy control algorithms in this context. We analyze a design decision we used for this early version of Fuzzy in Section A.22.1. For Static and Aging variants (Revision 1.2) and Fuzzy-Crisp variants (Revision 1.5), itemized results per algorithms per station per OLA are given in the Data Supplement (to be available where the benchmark suite shall be archived).

4.6.2 Exploring why Fuzzy-Crisp variant 0x2 (weather-prediction using, non-transition limited) (Rev. 1.4) usually outperformed variant 0x0 (non-prediction-using, non-transition limited) (Rev. 1.4) for OLA 3

For OLA 3, which does not limit state transitions, we explore why the applicable weather-prediction variant of the Fuzzy-Crisp Hybrid algorithm (variant 0x2) typically outperformed the applicable variant of Fuzzy-Crisp that does not use weather prediction (variant 0x0). We start our analysis by using station KATL's data, where variant 0x2 scored 0.07 points

Table 5: Comparing sparse and dense design-space searches by Static 0x0's processing workloads OLA 3 and 4 for four stations

Static 0x0 Training Revision	OLA	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQBP	MQNetNorm
1.1	3	KATL	7	121	0.39	0.93	0.36
Future	3	KATL	9	40	0.44	0.90	0.40
1.1	4	KATL	7	121	0.39	0.93	0.36
Future	4	KATL	9	101	0.44	0.93	0.41
1.1	3	KBOS	9	91	0.34	0.94	0.32
Future	3	KBOS	10	30	0.35	0.92	0.32
1.1	4	KBOS	9	91	0.34	0.94	0.32
Future	4	KBOS	9	113	0.34	0.94	0.32
1.1	3	KBWI	7	91	0.50	0.92	0.45
Future	3	KBWI	9	32	0.55	0.89	0.49
1.1	4	KBWI	7	61	0.49	0.91	0.45
Future	4	KBWI	9	81	0.54	0.92	0.50
1.1	3	KCLE	8	121	0.34	0.92	0.32
Future	3	KCLE	10	43	0.40	0.90	0.37
1.1	4	KCLE	8	121	0.34	0.92	0.32
Future	4	KCLE	10	71	0.40	0.91	0.37

higher than variant 0x0 (Table 72 on page 288). Let us determine whether or not the training routine for variant 0x2 happened to choose better settings than it chose for variant 0x0. That is, does variant 0x0's score improve 0.07 points if we simply apply to variant 0x0 the settings that training chose for variant 0x2. The running-average window sizes for each month are shown in the Table 6 on page 98 for variants 0x0 and 0x2. (The deployment membership value is 0.5 for both variants for all months.) The bottom row of the table shows the result of applying to variant 0x0 the running-average-window sizes that training chose for variant 0x2.

When we use the variant 0x2 settings for variant 0x0, then variant 0x0 improves from 0.37 to 0.42 for NetNorm, which is only 0.02 points less than what variant 0x2 scored. Thus, we see that, for at least station KATL, a large part of the reason why variant 0x2 scored higher than variant 0x0 is because of the settings that the training procedure chose.

Now let us search for the reason for the 0.02 point difference. We compared variant 0x0's and 0x2's minute-by-minute logs. The logs are identical until 1/2/2013 10:22AM, where variant 0x0 reversed stowing while variant 0x2 continued to stow (Figure 18). (Variant 0x2's action is consistent with its intended design, which is to conserve visibility time.)

Let us now explore how many reversals each variant makes. A *reversal* is defined as a change in direction while in mid-deployment or mid-retraction. Variants 0x0 and 0x2 made 707 and 536 reversals, respectively. which implies variant 0x2 used its time moving more efficiently than variant 0x0.

4.6.3 Examining the relationship between mean windspeed, energy available during quiet hours, and NetNorm

Figure 19 shows that as mean windspeeds increase, NetNorm scores earned by Static variant 0x0 version 1.1 for OLA 1 decrease, roughly. The reason seems to be that the fraction of energy available during quiet hours over the total energy available increases, in general, as mean windspeeds increase (Figure 20).

Notice the almost perfectly linear inverse relationship between the fraction of total energy available during quiet hours vs. the NetNorm scores earned by Static 0x0 v1.1 for OLA 1 shown in Figure 21. As the fraction of total energy available during quiet hours increases, the NetNorm scores decrease. That relationship suggests that a useful metric could be one where NetEnergy is normalized to the energy available when noise is permitted. A name for that new metric could be "NetNorm when Noise Allowed" (NetNormNA). When we divide NetNorm earned by Static 0x0 v1.1 for OLA 1 by the fraction of total energy available during non-quiet hours per station (which is 1 minus the energy available during quiet hours (Table 7)), then the result indicates that Static 0x0 v1.1 harvests an average of 92% of the energy available during non-quiet hours for OLA 1. Static 0x0 v1.1 harvests an average of 62% of the energy available during non-quiet hours for OLA 3. Also for OLA 3, an average of 70% of the energy available during non-quiet hours is harvested by Fuzzy 0x2 v1.4.

4.6.4 A sensitivity analysis of the Retraction-Threshold-Difference setting for Aging (variant 0x0) (Rev. 1.1) for OLAs 3 and 4

For OLAs 3 and 4, Aging (variant 0x0) performed best on average (Table 4), where Aging used the OLA-allowed retraction-threshold-difference (RTD) of 1 knot. Let us examine how Aging (variant 0x0) performs when the RTD is 0, 2, and 3 knots: Aging's average NetNorm and MQNetNorm scores are shown in Table 8 and Figure 22. The results indicate that NetNorm decreases as we increase the RTD, but the MQNetNorm fluctuates. (Table 8 also shows the sizes of the running average window sizes, which tend to increase as expected, except for when RTD = 3. When RTD = 3, the sizes decrease unexpectedly.)

4.6.5 Sensitivity analyses on λ

4.6.5.1 Aging As said in Section 3.1.2.1, we chose λ to be 0.9 because lower values did not perform well in initial tests (during relatively early development). Let us test λ using version 1.1 of Aging variant 0x0 on KPIT's data for OLA 3. Recall that version 1.1 is the version of Aging we compare to Static and Fuzzy-Crisp in Tables 4. And let us test λ using a branch of version 1.1 that uses a denser search space during training than what version 1.1 uses. Recall that Version 1.1's training routine steps through the search space for a y-intercept (where *y-intercept* is introduced in Section 3.4.2 in the context of the Aging algorithms) using a step size of 30 knots (Section 3.7). The branch uses a step size of 1 knot because the 30-knot step size trained relatively poorly when λ equals 0.7 and 0.8, as shown in the results of the two tests (Figure 23). Since the branch produces in unexpected dip at 0.5, let us create yet another branch, where we increase the density and the size of the training routine's search space of the window size of the running average windspeed from the sequence $[1, (\text{step } 5), 121]$ to the sequence $[1, (\text{step } 1), 361]$.

As expected, very low values for λ result in lower NetNorm scores than the higher values of λ. As λ increases, the lowest windspeed deemed to be windy decreases (Appendix A.2.4). Thus, higher values of λ result in lower windspeeds that are deemed to be windy. Because lower windspeeds deemed to be windy can cause Aging to exhaust its visibility time while capturing low-power winds and because higher windspeeds deemed to be windy can cause Aging to forgo harnessing relatively high-power winds, it makes sense that the NetNorm() score rises and then falls as λ increases.

4.6.5.2 Fuzzy-Crisp, OLA 3, transitions unlimited Exploration 4 examines the effect of lambda on Fuzzy-Crisp (variant 0x0) (Rev. 1.4) running KATL's data in the context of OLA 3. Figure 24 shows that as lambda increases, KATL's NetNorm score tends to increase until it peaks when lambda equals 0.7. The curve shows an unexpected dip when lambda equals 0.5 and an unexpected slight rise when lambda equals 0.9. Figure 24 also shows Fuzzy-Crisp's MQNetNorm performance curve, which has the expected shape except at 0.3 where the curve unexpectedly dips. Those unexpected movements could be the result of the training data being different from the testing data. In other words, if we were to plot Fuzzy-Crisp's NetNorm and MQNetNorm performances on the training data, we would probably see smoother curves than we see in Figure 24. A look at how Fuzzy-Crisp performs on the training data when Fuzzy-Crisp's settings are kept constant during each iteration of lambda does in fact reveal a smoother curve (Figure 25), where NetNorm scores increase somewhat logarithmically as lambda increases, relatively leveling off for lambdas equaling 0.8 and 0.9.

Why does the curve have the shape that it does, being relatively level at 0.8 and above? Recall that Fuzzy-Crisp uses Crisp code to retract to ensure that the algorithm meets the

OLA's requirements, which are crisp. Fuzzy-Crisp uses fuzzy code to deploy. That deployment code, which is fuzzy, is not dependent on lambda because lambda is used to create a crisp partition, a lambda-cut. Thus, changing lambda effects only the crisp code. Increasing lambda does not change when Fuzzy-Crisp deploys the harvester, but only when Fuzzy-Crisp retracts it. As lambda increases, the lowest windspeed deemed to be windy decreases. Since OLA 3 requires the harvester to be stowed during calm weather, and increasing lambda decreases the highest windspeed deemed to be calm, increasing lambda allows Fuzzy-Crisp to stay deployed during relatively lighter winds. That is, increasing lambda, lowers the "must-stow" windspeed. We surmise that although Fuzzy-Crisp is consuming valuable visibility time during relatively light winds, Fuzzy-Crisp's keeping the harvester deployed allows the harvester to be ready to capture higher energy winds, which mitigates the relatively invaluable visibility time. We further surmise that the relatively level portion of the curve can is the result of the values of the following two variables probably approaching each other while "must-stow" windspeed thresholds are low:

- the increase of energy gained by decreasing already low "must-stow" windspeed thresholds, and
- the increase of energy forfeited to comply with the visibility time limit.

4.6.5.3 Fuzzy-Crisp, OLA 5, transitions limited For Explorations 5 and 6, we examined how changing lambda affected the scores of Fuzzy-Crisp (variant 0x3) (Rev. 1.4) running KATL's and KBOS's data in the context of OLA 5. Changing lambda had no effect because state-transition-limited variant 0x3 of Fuzzy-Crisp does not use the lowest windspeed deemed to be windy since OLAs 5 and 6 allow the harvester to be visible when the windspeed is 0 knots. Lambda defines the lowest windspeed deemed to be windy. That windspeed threshold does not affect when the harvester must be stowed in the contexts of OLAs 5 and 6. Recall that OLAs 5 and 6 limit the harvester to two state-transitions per month and limit the harvester's visibility time to approximately 20% of the month. OLAs 5 and 6 are concerned with windspeed only when windspeeds approach the mechanical limits of the harvester.

In addition to exploring the effect of lambda on Fuzzy-Crisp, Explorations 5 and 6 examined whether the training routine would chose a running average window size greater than 121 minutes. Results indicate that the training routine often chose values above 121 minutes for OLAs 5 and 6. Thus, for those OLAs, we recommend that values above 121 minutes be explored for the other 28 stations in the contexts of OLAs 5 and 6.

Full results for Explorations 5 and 6, which are for KATL and KBOS, are in Table 86 on page 335 and Table 87 on page 336, respectively.

4.6.6 A sensitivity analysis on the forecasting time horizon

We used the scaled function $y()$ divided by 13.8 ($y()/13.8$) to generate simulated windspeed predictions for KATL's training and testing data in columns entitled "f30", "f60", "f120", "f240", "f480", and "f720", where each number after the 'f' prefix indicates the time horizon in minutes. To do a sensitivity analysis on the time horizon, we created a version of Static variant 0x3 (Rev. 1.1) to use the desired 'f' column to process KATL's augmented data in the context of OLAs 5 and 6.

Static variant 0x3 decides to deploy if the weather is windy now and the weather is very windy at the time horizon. Recall that minute-by-minute windspeed samples are averaged via a moving window, the size of which is determined by the training routine.

Figure 26 shows that NetNorm scores are no further than 0.02 points from each other, as are the MQNetNorm scores. The highest and second highest scoring NetNorm scores are at a 60-minute time horizon and a one-day time horizon, respectively. Similarly, the highest and second highest scoring MQNetNorm scores are at a 120-minute time horizon and a one-day time horizon, respectively.

Full results for Exploration 8 are in Appendix A.26.

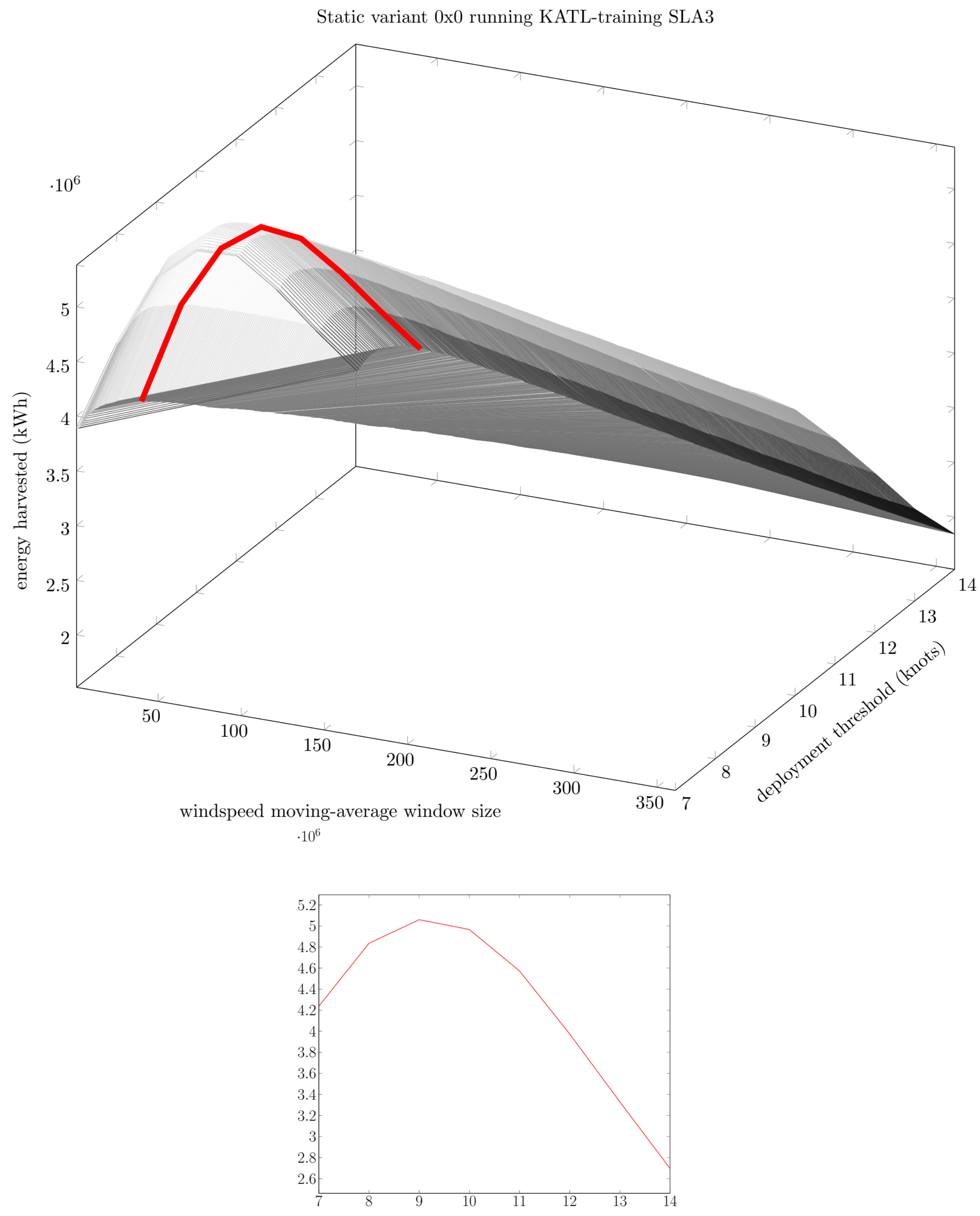

Figure 17: Surface plot showing how running-average-window size and deployment threshold affect Static variant 0x0's running of KATL-training to meet OLA3 (upper plot) and slice when running-average-window size is 40 minutes (lower plot)

Table 6: A comparison of performance of Fuzzy-Crisp variants 0x0 and 0x2 for OLA 3 for station KATL

Fuzzy-Crisp Var.	Running-average-window size per month												Net-Norm	Source Table
	1	2	3	4	5	6	7	8	9	10	11	12		
0x0	1	1	1	1	31	1	1	1	121	1	91	91	0.37	53 on page 249
0x2	1	1	1	1	1	1	1	1	1	1	1	1	0.44	59 on page 256
0x0	1	1	1	1	1	1	1	1	1	1	1	1	0.42	Not. Applic.

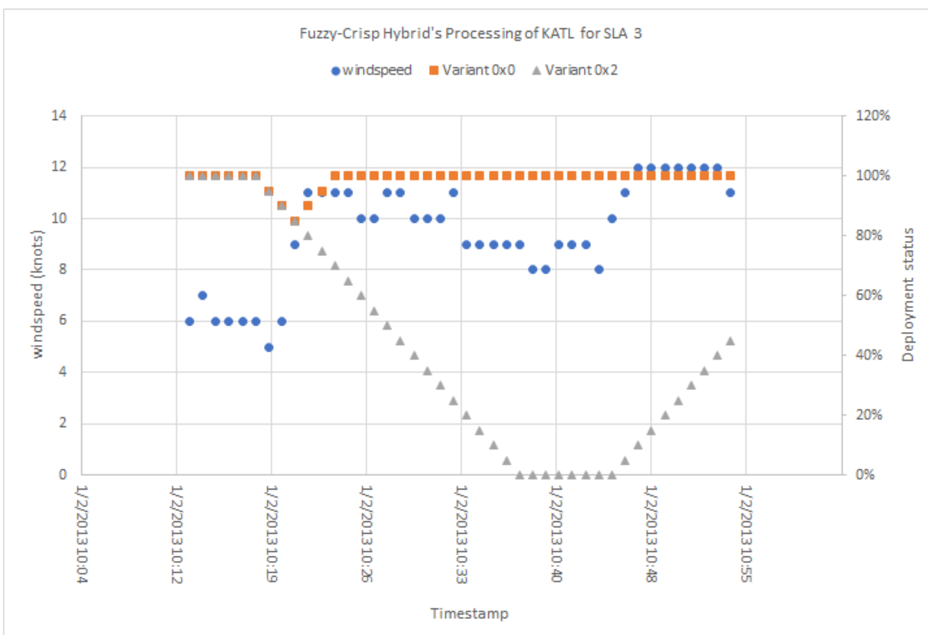

Figure 18: Processing of KATL's data by variants 0x0 and 0x2 of Fuzzy-Crisp Hybrid

Table 7: Fraction of total energy available during quiet hours per station

Station	fEnergyQuiet*	Station	fEnergyQuiet*
KLAX	0.09	KSEA	0.31
KSMX	0.14	KDTW	0.35
KSAC	0.18	KORD	0.36
KPIT	0.21	KCLE	0.36
KIAH	0.23	KATL	0.38
KEUG	0.23	KDEN	0.40
KBWI	0.24	KSAT	0.43
KSFO	0.26	KSTL	0.44
KSAN	0.26	KDCA	0.46
KCLT	0.27	KDFW	0.49
KMCO	0.28	KLAS	0.50
KTPA	0.28	KLGA	0.52
KMCI	0.29	KPHX	0.58
KCVG	0.29	KBOS	0.58
KPHL	0.30	KMSP	0.59
continues above right			

*fraction of total energy available during quiet hours

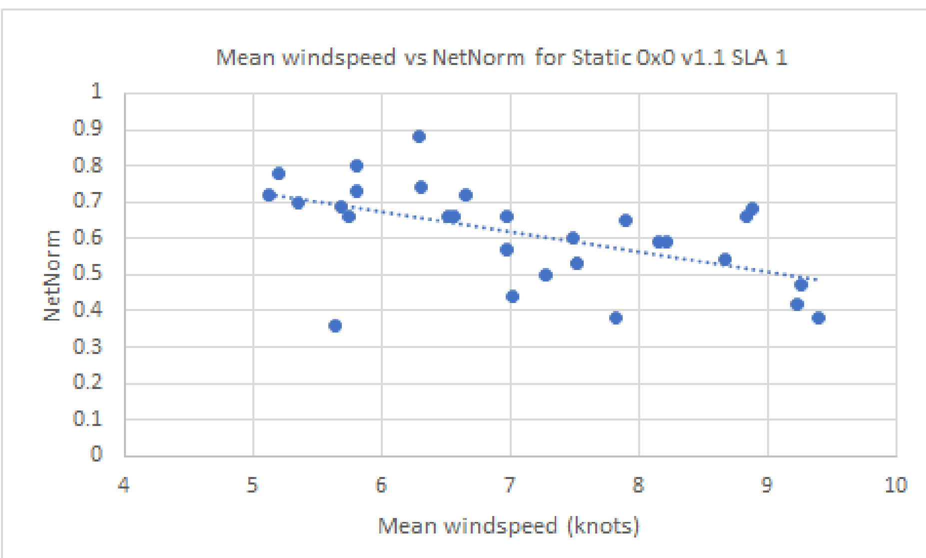

Figure 19: Mean windspeed vs. NetNorm for Static 0x0 v1.1 OLA 1

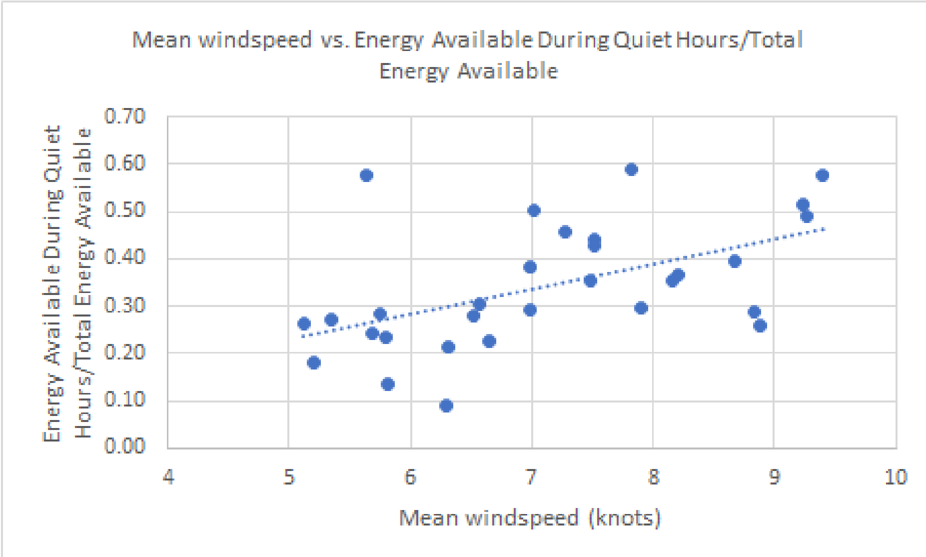

Figure 20: Mean windspeed vs. fraction of total energy available during quiet hours

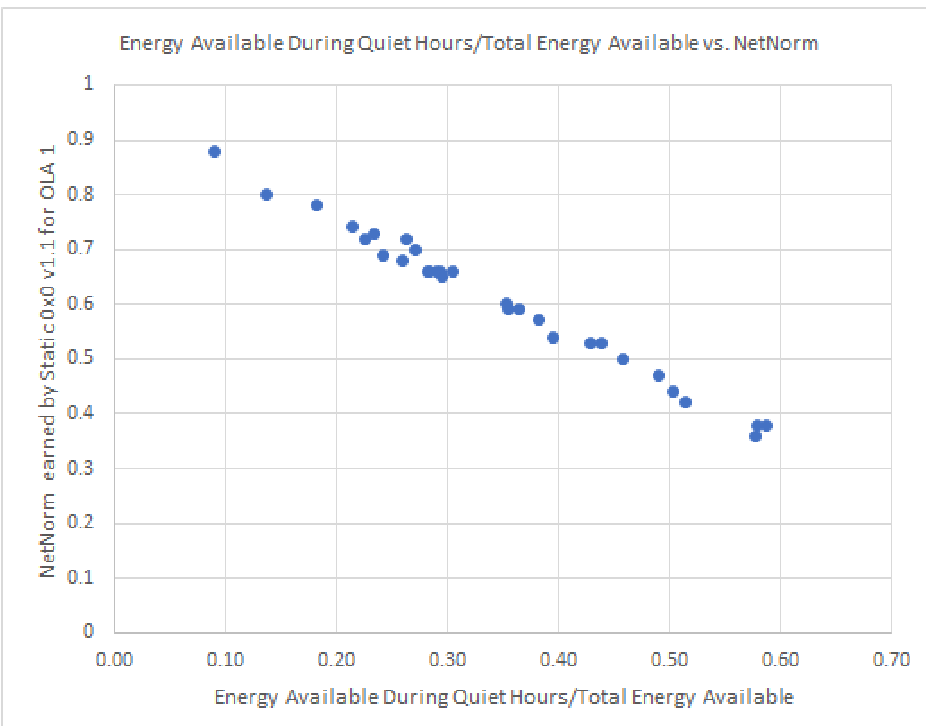

Figure 21: Fraction of total energy available during quiet hours vs. NetNorm earned by Static 0x0 v1.1 for OLA 1

Table 8: Effect of changing the retraction threshold difference on Aging (variant 0x0) for OLAs 3 and 4

RTD (knots)	Net-Norm	MQNet-Norm	Running Avg. Window Size OLA 3	OLA 4	Full Results
0	0.409	0.373	56.00	70.33	Table 82 on page 327
1	0.402	0.371	62.33	82.25	Table 41 on page 234
2	0.398	0.375	69.08	92.83	Table 83 on page 329
3	0.394	0.374	51.00	65.25	Table 84 on page 330

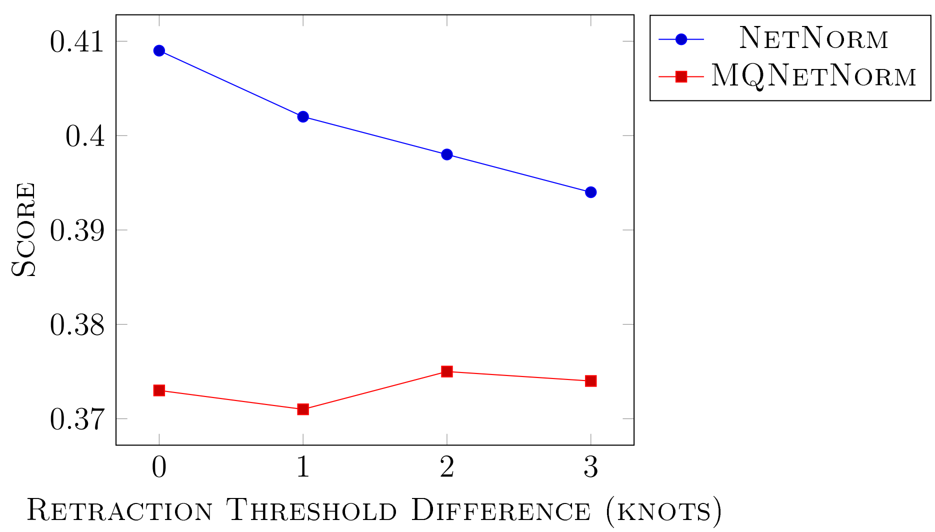

Figure 22: Effect of changing the retraction threshold difference on Aging (variant 0x0) for OLAs 3 and 4

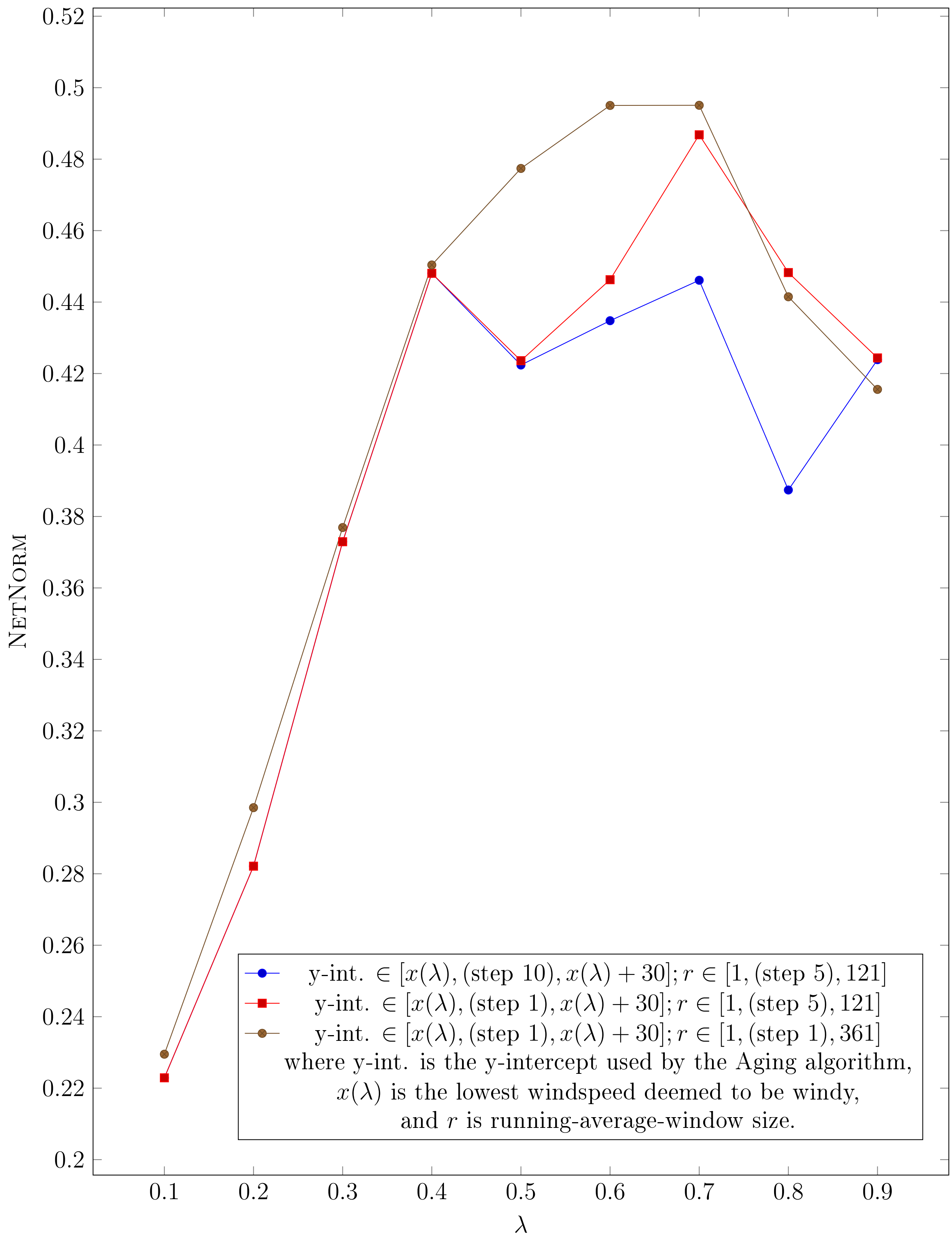

Figure 23: Sensitivity analysis on λ for Aging variant 0x0's processing of KPIT's data for OLA 3

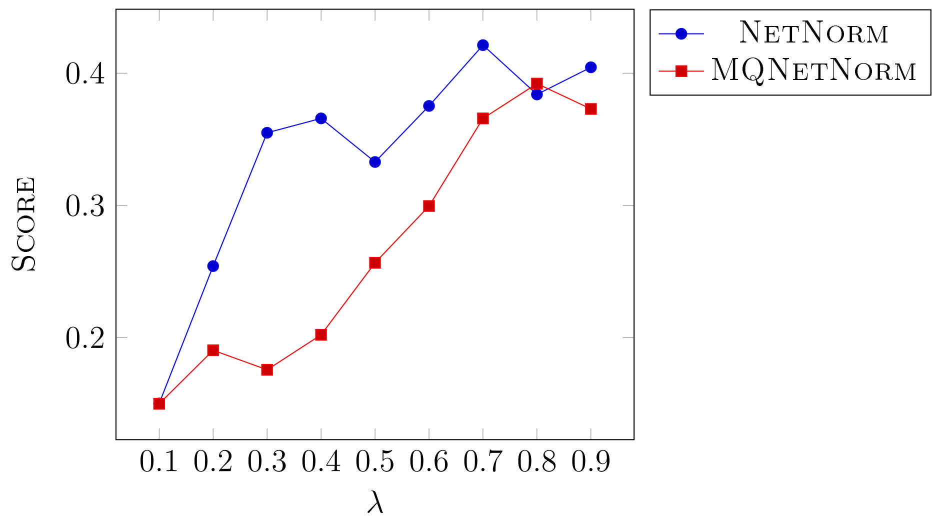

Figure 24: Effect of changing λ on Fuzzy-Crisp (variant 0x0) for OLAs 3 and 4 for KATL

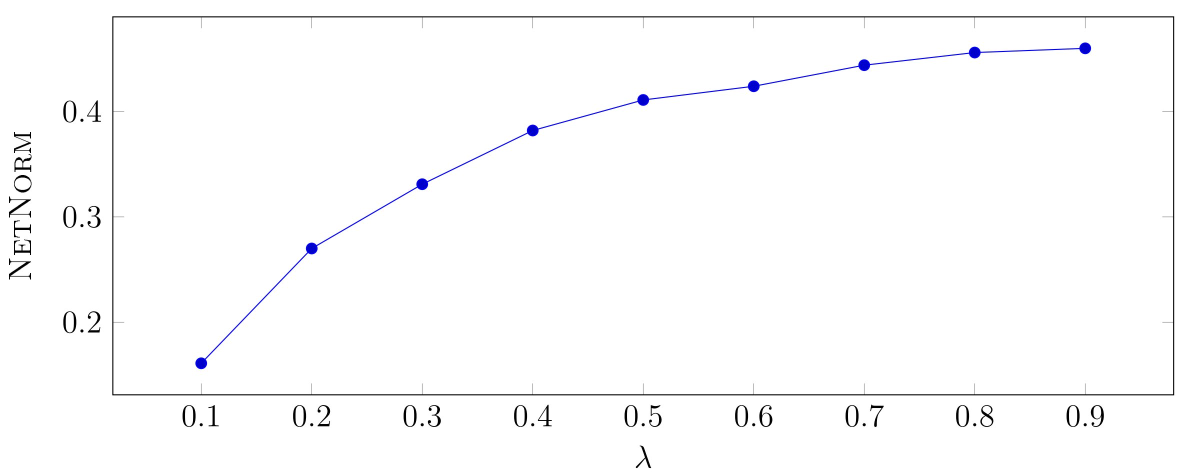

Figure 25: Effect of changing λ on Fuzzy (variant 0x0) for OLAs 3 for KATL's training data where settings are constant during entire run

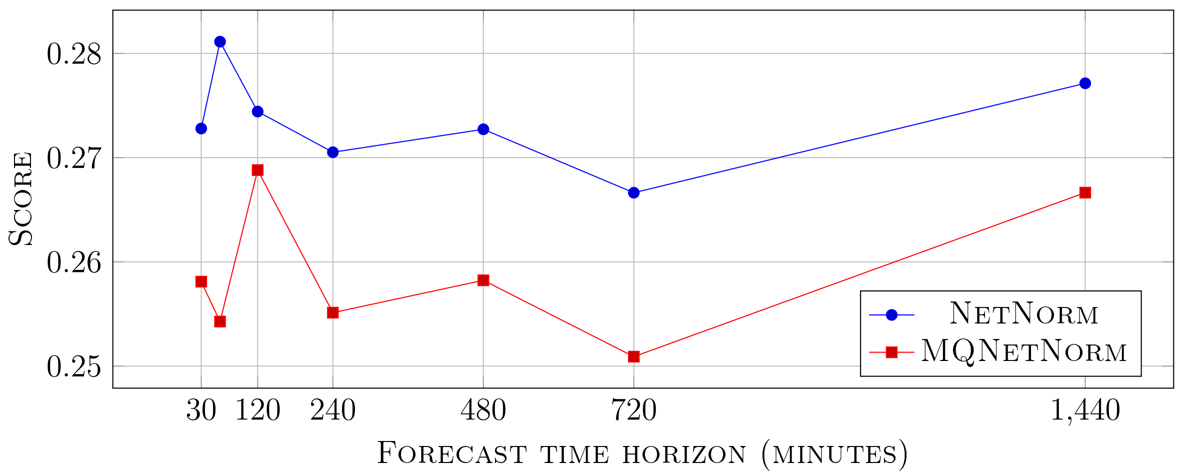

Figure 26: Effect of changing time horizon on Static (variant 0x3) for OLAs 5 and 6 for KATL

4.6.7 Summary of sensitivity analyses and explorations of larger and/or denser design spaces

Table 9 summarizes various sensitivity analyses and explorations of larger and/or denser design spaces.

Table 9: Explorations 2 through 8: Sensitivity analyses and explorations of larger and denser design spaces

Purpose of Explor'n 2: Check how often Training chooses a window size of running average windspeed > 121 minutes

Exp.	OLA(s)	Algorithm Family	Vari-ant	Rev.	Sta-tion(s)	Ranges Training Explored: λ	R.T.D. (knots)	deployment threshold of running average windspeed (knots)	window size of moving average windspeed (samples or minutes)	Results
2	3	Static	0x0	1.1	all	0.9	1	$[ks_{(ws,\lambda)}$, (step 1), $ks_{(ws,\lambda)}+7]$	[1, (step 1), 361]	No window size above 121 minutes was chosen by the training routine

Purpose of Explor'n 3: Sensitivity analysis on λ for Aging 0x0 (Rev. 1.1) for OLA 3

Exp.	OLA(s)	Algorithm Family	Vari-ant	Rev.	Sta-tion(s)	Ranges Training Explored: λ	R.T.D. (knots)	y-intercept (knots)	window size of moving average windspeed (samples or minutes)	Results
										1. As training step sizes decreases, the NetNorm vs. lamda curve becomes smoother. 2. NetNorm vs. lambda curve rapidly increases, plateaus when λ is 0.6 & 0.7, then decreases as λ increases past 0.7
3a	3	Aging	0x0	1.1	KPIT	[0.1, (st. 0.1), 0.9]	1	$[ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}+30]$	[1, (step 5), 121]	
3b	3	Aging	0x0	1.1	KPIT	[0.1, (st. 0.1), 0.9]	1	$[ks_{(ws,\lambda)}$, (step 1), $ks_{(ws,\lambda)}+30]$	[1, (step 5), 121]	
3c	3	Aging	0x0	1.1	KPIT	[0.1, (st. 0.1), 0.9]	1	$[ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}+30]$	[1, (step 1), 121]	

Purpose of Explor'n 4: Sensitivity analysis on λ for Fuzzy-Crisp 0x0 (Rev. 1.4) for OLA 3

Exp.	OLA(s)	Algorithm Family	Vari-ant	Rev.	Sta-tion(s)	Ranges Training Explored: λ	R.T.D. (knots)	deployment threshold of membership value in resulting fuzzy set	window size of moving average windspeed (samples or minutes)	Results
										1. NetNorm vs. lambda curve increases jaggedly when tested with testing data. 2. The curve increases smoothly when tested with TRAINING data.
4	3,4	Fuzzy	0x0	1.4	KATL	[0.1, (st. 0.1), 0.9]	1	0.5	[1, (step 1), 361]	

Purpose of Explor'n 5: Sens. analysis on λ while exploring larger design space for Fuzzy-Crisp 0x3 (Rev. 1.4) for OLAs 5 & 6

Exp.	OLA(s)	Algorithm Family	Vari-ant	Rev.	Sta-tion(s)	Ranges Training Explored: λ	R.T.D. (knots)	deployment threshold of membership value in resulting fuzzy set	window size of moving average windspeed (samples or minutes)	Results
										1. lambda has no effect. 2. The training routine chose a deployment membership value threshold other than 0.5 a number of 22 of 24 times & a window size above 121 a number of 12 out of 24 times.
5	5,6	Fuzzy	0x3	1.4	KATL	[0.1, (st. 0.1), 0.9]	1	[0.1, (st. 0.1), 0.9]	[1, (step 1), 361]	

Purpose of Explor'n 6: Same as Explor'n 5, but for different station

Exp.	OLA(s)	Algorithm Family	Vari-ant	Rev.	Sta-tion(s)	Ranges Training Explored: λ	R.T.D. (knots)	deployment threshold of membership value in resulting fuzzy set	window size of moving average windspeed (samples or minutes)	Results
										1. lambda has no effect. 2. The training routine chose a deployment membership value threshold other than 0.5 a number of 21 of 24 times & a window size above 121 a number of 16 out of 24 times.

Table 9: (continued)

6	5,6	Fuzzy	0x3	1.4	KBOS	[0.1, (st. 0.1), 0.9]	1	[0.1, (st. 0.1), 0.9]	[1, (step 1), 361]	
Purpose of Explor'n 7: Sens. analysis on Retraction Threshold Difference (R.T.D.) on Aging 0x0 (Rev. 1.1)										Results
						Ranges Training Explored				
Exp.	OLA(s)	Algorithm Family	Variant	Rev.	Station(s)	λ	R.T.D. (knots)	y-intercept (knots)	window size of moving average windspeed (samples or minutes)	1. NetNorm vs R.T.D. curve decreases as R.T.D. increases. 2. MQNetNorm vs R.T.D. curve is approximately flat
7	3,4	Aging	0x3	1.1	All	0.9	[1, (st. 1), 4]	$[ks_{(ws,\lambda)}$, (step 10), $ks_{(ws,\lambda)}+30]$	[1, (step 30), 121]	
Purpose of Explor'n 8: Effect of forecast's time horizon t^* on Static 0x3 (Rev. 1.1)										Results
						Ranges Training Explored				
Exp.	OLA(s)	Algorithm Family	Variant	Rev.	Station(s)	λ	R.T.D. (knots)	y-intercept (knots)	window size of moving average windspeed (samples or minutes)	1. NetNorm vs R.T.D. curve decreases as R.T.D. increases. 2. MQNetNorm vs R.T.D. curve is approximately flat
8	5,6	Aging	0x3	1.1	KATL	0.9	1	$[ks_{(ws,\lambda)}$, (step 1), $ks_{(ws,\lambda)}+7]$	[1, (step 1), 121]	

$t \in \{30, 60, 120, 240, 480, 720, 1440$ minutes$\}$

4.6.8 A preliminary comparison of alternative OLA's

Let us define a "visibility event" as the continuous block of time a harvester is visible. Suppose that a community limits both visibility time and "visibility events" to 20% per month and e events per month. How does e affect NetNorm? Preliminary results are shown in Figure 27.

We ran a special variant of Static called "VE" which accepts the maximum number of visibility events as a parameter. Variant VE retracts the harvester when the average windspeed drops below $0.5k_{ws,\lambda}$ where $k_{ws,\lambda}$ is the lowest windspeed deemed to be windy for $ws =$KPIT and $\lambda = 0.7$.

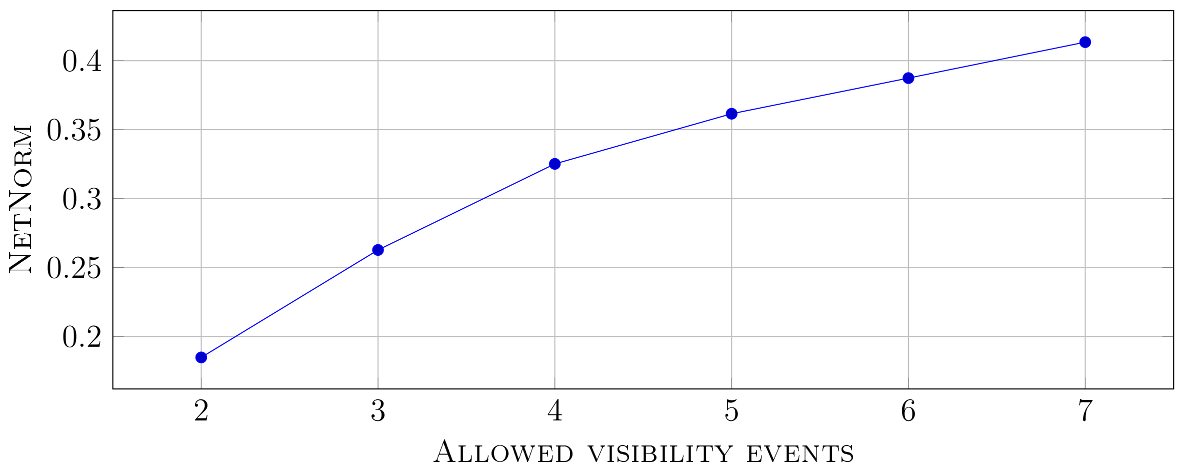

Figure 27: Effect of increasing allowed "visibility events" on Static (variant VE) for OLA 3 for KPIT

5.0 CONCLUSIONS

OLA 1 which does not limit total visibility time, but restricts operation and visibility to "windy" periods outside of quiet hours, saw algorithms harvesting 62% of the available energy. For OLA 3, where the visibility time of the harvester is limited to approximately 20% of the month, Static 0x0 (Rev. 1.2) achieved 46%. For OLA 5, where the visibility time of the harvester is limited to approximately 20% of the month and 2 state-transitions per month (or 3 to stow during high winds), the highest scoring variant netted 27% of the available energy. For each even-numbered OLA, the best MQNetNorm() scores are not more than 8 percentage points less than the best NetNorm() scores for each corresponding odd-numbered OLA (Table 4 on page 84).

A very interesting conclusion relates to the fraction of energy available outside of quiet hours. We noted in Section 4.6.3 that Fuzzy-Crisp netted an average of 70% of the energy available outside of quiet hours for OLA 3. Because OLAs 1-4 limit operation and visibility to non-quiet hours, we suggest that another metric "NetNorm when Noise Allowed" (NetNormNA) would give a truer picture of how well an algorithm uses its allowable time than NetNorm does. However, NetNorm is useful in that it can motivate quiet wind-harnessing technologies that would be permitted to operate during a percentage of quiet hours.

Our final conclusion is that improving the training of the algorithms seems to be a promising way to increase the algorithms' performance. For example, Table 5 on page 91 shows an average NetNorm improvement of 0.04 points (or 11%) by decreasing the step size in the search space for running average window size from 30 minutes to 1 minute.

6.0 FUTURE WORK

The results of running the workloads in this benchmark suite, measuring results using the benchmark's metrics, and analyzing the results led to some of these recommendations for future versions of this benchmark suite:

- Revise OLAs 1-4 to allow the algorithms to begin to deploy before windy periods officially begin. It seems reasonable that retractable-harvester hosting communities would agree to allow harvesters to be visible x minutes before a windy period officially begins where x is the minimum time required for the harvester to deploy. Recall that the definition of "When it's not windy" given in Section 3.1.2.1 implies a windy period begins when the running average windspeed reaches the lowest windspeed deemed to be windy at the pertinent weather station. The running average windspeed depends on window size in effect during the then-current month. If a community were to allow harvesters to begin deploying x minutes before a windy period begins, then that community would need to wait x minutes to determine whether the harvester has complied with the OLA's visibility allowance.
- Thus, because knowing the windspeed x minutes into the future would help a harvester to be fully deployed when a windy period officially begins, it would make sense for future versions of this benchmark suite to provide x-minute-ahead weather prediction. If x equals 20 minutes, which is the deployment time (Section A.3.5) of the harvester hm we used to test our algorithms herein, then we surmise that the 20-minute-ahead weather predictions that would be provided in a future benchmark are more accurate than the 1440-minute-ahead predictions that we currently simulate.
- Require algorithms to log their deployment-and-retraction commands to a standard log-

ging format so that the algorithms' actions can be verified by a "watchdog" program

- Add bird migration data (Section 3.1.2.3).
- For OLAs 5 and 6, explore values for the window size of the windspeed running average above 121 minutes for the remaining 28 stations for Fuzzy-Crisp (Results for two stations are given in Section A.25) and all 30 stations for Static and Aging.

We recommend that MQMPEnergy (Section 4.1.6.2) be considered for addition to future versions of the benchmark suite. We surmise that such a metric would motivate the creation of retractable wind-harnessing technologies tailored for peak-electricity-usage periods.

For future versions of the algorithms, we recommend the following:

- Use the error distribution of weather prediction to better gauge the probability of making a wrong conclusion. If the probability is less than a certain level, then the algorithm would act upon that conclusion. For example, let H_0 be the null hypothesis that tomorrow's running average windspeed will be below the lowest-windspeed-deemed-to-be-windy (Section A.2.4) threshold τ knots at a weather station ws. And let H_1 be the alternative hypothesis that it will be at or above threshold τ. Given the standard deviation for the day-ahead prediction error for ws (Section 4.2.0.3), what is the probability of falsely accepting the null hypothesis (i.e., making a Type I or *alpha* error) or what is the probability of falsely rejecting the null hypothesis (i.e., making a Type II or *beta* error) [53, pp. 5–6]? We calculate the probability of making a Type II error, i.e., concluding that tomorrow will *not* be windy when in actuality tomorrow will be windy, for station KBOS when the day-ahead-predicted windspeed is $\tau + 1$ knots in Appendix A.14.
- Take into account the overhead of deploying and retracting when using weather prediction. The example immediately above does not take into account the length of the windy period, but could be extended to determine the probability of a potential harvesting event to net a positive amount of energy.

We posit areas for exploration:

Our researching 30 municipalities' noise regulations revealed variations in the start and stop times of quiet hours. For example, one municipality (St. Louis) starts quiet hours at sunset. Another municipality (Pittsburgh) starts quiet hours at 10:00 pm every day. It might be worthwhile to find a simple definition of quiet hours that satisfies all 30 municipality's quiet hours to simplify the programming of the algorithms. Currently each algorithm looks up the quiet hours with which it must comply. If the algorithm were to observe universal quiet hours, then algorithm programming could be simplified in that the algorithm would not access a look-up table keyed by weather station and when the weather station is St. Louis's KSTL, the algorithm would not need consult another table for sunset times. Instead, the algorithm would simply start quiet hours before the earliest sunset time of the year, would end quiet hours at the latest hour of all the ending hours of each municipality, and would include weekends and federal holidays. (However, if the algorithm were to observe simple universal quiet hours, then the algorithm would forfeit much allowable harvesting time in some municipalities.)

Another area of possible simplification is to use the same windspeed threshold for all stations to define windy weather instead of basing the definition on the local historical windspeeds. For example, we could define 7 knots, which is the lowest speed of the Beaufort Scale's Gentle Breeze (7 - 10 knots), as the lowest "windy" windspeed for all municipalities. Using 7 knots as the lowest windspeed deemed to be windy for all stations would simplify the benchmark suite because the benchmark suite would supply only one membership function for the fuzzy set "not windy" instead of thirty. It would also simplify the creation of a new metric that measures how much wind energy is available when windspeeds are 7 knots or greater. On the other hand, because 7 knots might seem low to some municipalities, those municipalities might not agree that 7 knots is windy. Thus, exploring each municipality's perception of windiness is an area of future work.

Using its own windspeed sensor, each algorithm can use numerical methods on series of windspeeds to predict weather in addition to accessing external weather forecasts. For example, Kulkarni et al. claim, "It has been found that wind speed can be predicted with a reasonable degree of accuracy using two methods, viz., extrapolation using periodic curve fitting and [Artificial Neural Networks]" [46]. Adding internal weather-forecasting to each algorithm might give each algorithm useful flexibility to choose its own forecasting time horizon if statistically significant.

Others have researched how to tune algorithms; It might be worthwhile to use advanced techniques (e.g., a procedure called CALIBRA [2]) to tune them.

Data mining the 30 sets of historical data could provide insight into how to group the sets (e.g., grouped by Weibull shape parameters) to recommend an OLA for a particular group.

Because some survey respondents did not seem to understand wind power (e.g., “they all still run on oil so what is the point”) or retractable harvesters (“i [sic] just don’t understand their purpose”) and some survey companies allow a video to be shown or accessible as part of the survey, the inclusion of instructional videos in future surveys might help to solve the problem of some respondents’ lack of retractable-wind-harvester knowledge.

Consider adding an OLA or revising one or more of the existing standard OLAs to address “shadow flicker,” which are “shadows on the ground and surrounding structures that may emanate from the rotating blades of a wind turbine” [71].

7.0 BIBLIOGRAPHY

[1] *A win for us: Judge says 'No' to Dan's Mountain wind farm. Editorial.* times-news.com. January 27, 2017. URL: http://www.times-news.com/opinion/editorials/a-win-for-us/article_127f1189-ccf3-5953-ba74-4024ee1e1428.html.

[2] Belarmino Adenso-Diaz and Manuel Laguna. "Fine-tuning of algorithms using fractional experimental designs and local search". In: *Operations research* 54.1 (2006), pp. 99–114.

[3] C.C. Aggarwal. *Outlier Analysis*. Springer International Publishing, 2016. ISBN: 9783319475783. URL: https://books.google.com/books?id=KyG1DQAAQBAJ.

[4] Altaeros Energies. *Technology*. URL: www.altaerosenergies.com/technology.html.

[5] Astronomical Applications Department. *Sun or Moon Rise/Set Table for One Year*. U.S. Naval Observatory. 1June 2016. URL: http://aa.usno.navy.mil/data/docs/RS_OneYear.php.

[6] J. Banks. *Discrete-event System Simulation*. Discrete-event System Simulation. Prentice Hall, 2001. ISBN: 9780130887023. URL: https://books.google.com/books?id=NV1RAAAAMAAJ.

[7] Christian Bienia et al. "The PARSEC benchmark suite: Characterization and architectural implications". In: *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*. ACM. 2008, pp. 72–81.

[8] Glen Brown, Roy Haggard, and Brook Norton. "Inflatable structures for deployable wings". In: *16th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, p. 2068.

[9] M.H. Brown et al. *Electricity Transmission: A Primer*. National Council on Electric[ity] Policy, 2004. ISBN: 9781580243520. URL: https://books.google.com/books?id=yQwKAQAAMAAJ.

[10] Central Plains Cranes. *Self-erecting Tower Cranes*. URL: http://www.centralplainscranes.com/Self-erecting_Cranes.html.

[11] Debrup Chakraborty and Nikhil R Pal. "A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification". In: *IEEE Transactions on Neural Networks* 15.1 (2004), pp. 110–123.

[12] Antonello Cherubini et al. "Airborne Wind Energy Systems: A review of the technologies". In: *Renewable and Sustainable Energy Reviews* 51 (2015), pp. 1461–1476.

[13] Commercial and Residential Reference Building Models. *Commercial and Residential Hourly Load Profiles for all TMY3 Locations in the United States*. Maintainer: nathan.clark@nrel.gov. OpenEI. URL: http://en.openei.org/datasets/files/961/pub/RESIDENTIAL_LOAD_DATA_E_PLUS_OUTPUT/.

[14] National Research Council et al. *Terrorism and the electric power delivery system*. National Academies Press, 2012. URL: https://www.nap.edu/catalog/12050/terrorism-and-the-electric-power-delivery-system.

[15] Department of Statistics. *S3̇ Hypothesis Testing*. The Pennsylvania State University. URL: https://onlinecourses.science.psu.edu/statprogram/reviews/statistical-concepts/hypothesis-testing.

[16] M. Diehl. "Airborne Wind Energy: Basic Concepts and Physical Foundations". In: *Airborne Wind Energy*. Ed. by U. Ahrens, M. Diehl, and R. Schmehl. Green Energy and Technology. Springer Berlin Heidelberg, 2013. Chap. 1. ISBN: 9783642399657. URL: https://books.google.com/books?id=807ABAAAQBAJ.

[17] I. Dincer. *Comprehensive Energy Systems*. Elsevier Science, 2018. ISBN: 9780128149256. URL: https://books.google.com/books?id=foxODwAAQBAJ.

[18] Paul Dvorak. *Tiny turbines are looking for work*. 2014. URL: http://www.windpowerengineering.com/featured/business-news-projects/tiny-turbines-looking-work.

[19] Earth System Research Laboratory. *NCEP-DOE Reanalysis 2: Summary*. National Oceanic and Atmospheric Administration. URL: https://www.esrl.noaa.gov/psd/data/gridded/data.ncep.reanalysis2.html.

[20] Office of Energy Efficiency & Renewable Energy. *EnergyPlus Energy Simulation Software*. URL: http://apps1.eere.energy.gov/buildings/energyplus/.

[21] Energy Information Administration. *Electricity Explained, Factors Affecting Electricity Prices*. May 2018. URL: https://www.eia.gov/energyexplained/index.php?page=electricity_factors_affecting_prices.

[22] Energy Information Administration. *Glossary*. URL: https://www.eia.gov/tools/glossary/.

[23] Energy Information Administration. *Negative prices in wholesale electricity markets indicate supply inflexibilities*. February 23, 2012. URL: https://www.eia.gov/todayinenergy/detail.php?id=5110.

[24] Federal Energy Regulatory Commission. *Electromagnetic Pulse: Effects on the U.S. Power Grid, Executive Summary*. URL: https://www.ferc.gov/industries/electric/indus-act/reliability/cybersecurity/ferc_executive_summary.pdf.

[25] Awi Federgruen and Nan Yang. "Infinite horizon strategies for replenishment systems with a general pool of suppliers". In: *Operations research* 62.1 (2014), pp. 141–159.

[26] Andrés Feijóo and Daniel Villanueva. "Assessing wind speed simulation methods". In: *Renewable and Sustainable Energy Reviews* 56 (2016), pp. 473–483.

[27] Jeremy Firestone et al. "Reconsidering barriers to wind power projects: community engagement, developer transparency and place". In: *Journal of environmental policy & planning* 20.3 (2018), pp. 370–386.

[28] David Friedman, Yael M Cycowicz, and Helen Gaeta. "The novelty P3: an event-related brain potential (ERP) sign of the brain's evaluation of novelty". In: *Neuroscience & Biobehavioral Reviews* 25.4 (2001), pp. 355–373.

[29] Sidney A Gauthreaux, Carroll G Belser, and Donald Van Blaricom. "Using a network of WSR-88D weather surveillance radars to define patterns of bird migration at large spatial scales". In: *Avian migration*. Springer, 2003, pp. 335–346.

[30] P. Gipe. *Wind Power: Renewable Energy for Home, Farm, and Business*. Chelsea Green Publishing Company, 2004. ISBN: 9781931498142. URL: https://books.google.com/books?id=RNvHAgAAQBAJ.

[31] Khaled Hamed and A Ramachandra Rao. "A Modified Mann-Kendall Trend Test for Autocorrelated Data". In: *Journal of Hydrology* 204 (Jan. 1998), pp. 182–196. DOI: 10.1016/S0022-1694(97)00125-X.

[32] Pavithra Harsha and Munther Dahleh. "Optimal management and sizing of energy storage under dynamic pricing for the efficient integration of renewable energy". In: *IEEE Transactions on Power Systems* 30.3 (2014), pp. 1164–1181.

[33] Ben Hoen et al. "Do Wind Turbines Make Good Neighbors? Results from a Nationwide Survey of Residents Living Near U.S. Wind Power Projects". In: (2018). A link to the webinar recorded on January 30, 2018, can be found at https://emp.lbl.gov/webinar/overall-analysis-attitudes-across-1700. This project is part of a broader set of projects under the National Survey of Attitudes of Wind Power Project Neighbors which are summarized at https://emp.lbl.gov/projects/wind-neighbor-survey.

[34] HOMER Energy. *About HOMER Energy LLC*. HOMER Energy. URL: https://www.homerenergy.com/company/index.html.

[35] HOMER Energy LLC. *Autocorrelation*. URL: https://www.homerenergy.com/products/pro/docs/3.11/autocorrelation.html.

[36] HOMER Energy Support. *Getting Wind data from Other Sources into HOMER Pro*. HOMER Energy. URL: http://usersupport.homerenergy.com/customer/en/portal/articles/2187118-getting-wind-data-from-other-sources-into-homer-pro.

[37] Independent Electricity System Operator. *HOEP [Hourly Ontario Energy Price], 2002-2017*. URL: http://ieso.ca/-/media/files/ieso/power-data/data-directory/hoep_2002-2017.csv?la=enviahttp://ieso.ca/en/power-data/data-directory/.

[38] D. Jager and A. Andreas. *NREL Report No. DA-5500-56489*. NREL National Wind Technology Center (NWTC): M2 Tower; Boulder, Colorado (Data). URL: http://dx.doi.org/10.5439/1052222.

[39] J-SR Jang and Chuen-Tsai Sun. "Neuro-fuzzy modeling and control". In: *Proceedings of the IEEE* 83.3 (1995), pp. 378–406.

[40] Daniel R Jiang et al. "A comparison of approximate dynamic programming techniques on benchmark energy storage problems: Does anything work?" In: *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE. 2014, pp. 1–8.

[41] John Wood Group PLC. *Energy yield prediction*. URL: https://www.woodgroup.com/what-we-do/view-by-products-and-services/clean-energy/services/analysis/energy-yield-prediction.

[42] Rajesh G Kavasseri and Krithika Seetharaman. "Day-ahead wind speed forecasting using f-ARIMA models". In: *Renewable Energy* 34.5 (2009), pp. 1388–1393.

[43] Rajesh G Kavasseri and Krithika Seetharaman. "SgurrEnergy provides wind analysis for phase 1 of 400MW Ethiopian wind project". In: *Windpower Engineering & Development* (August 1, 2013). URL: https://www.windpowerengineering.com/projects/test-measurement/sgurrenergy-provides-wind-analysis-for-phase-1-of-400mw-ethiopian-wind-project.

[44] Katherine Klink. "Trends in mean monthly maximum and minimum surface wind speeds in the coterminous United States, 1961 to 1990". In: *Climate Research* 13.3 (1999), pp. 193–205.

[45] Andreas Kroll and Horst Schulte. "Benchmark problems for nonlinear system identification and control using soft computing methods: Need and overview". In: *Applied Soft Computing* 25 (2014), pp. 496–513.

[46] Makarand A Kulkarni et al. "Wind speed prediction using statistical regression and neural network". In: *Journal of Earth System Science* 117.4 (2008), pp. 457–463.

[47] Matthias Lange. "On the uncertainty of wind power predictions—Analysis of the forecast accuracy and statistical distribution of errors". In: *Journal of Solar Energy Engineering* 127.2 (2005), pp. 177–184.

[48] Makani. *Energy Kites*. URL: http://www.google.com/makani/.

[49] Ryan McAleer. *Application for Dooish wind farm rejected by planners*. The Ulster Herald. January 30, 2016. URL: http://ulsterherald.com/2016/01/30/application-for-dooish-wind-farm-rejected-by-planners.

[50] William McKinney and James DeLaurier. "Wingmill: an oscillating-wing windmill". In: *Journal of energy* 5.2 (1981), pp. 109–115.

[51] Daniel Mossé and Guy Gadola. "Controlling wind harvesting with wireless sensor networks". In: *Green Computing Conference (IGCC), 2012 International*. IEEE. 2012, pp. 1–6.

[52] Eduard Muljadi and Charles P Butterfield. "Pitch-controlled variable-speed wind turbine generation". In: *IEEE transactions on Industry Applications* 37.1 (2001), pp. 240–246.

[53] K.R. Murphy, B. Myors, and A. Wolach. *Statistical Power Analysis: A Simple and General Model for Traditional and Modern Hypothesis Tests, Fourth Edition*. Routledge, 2014. ISBN: 9781317680574. URL: https://ebookcentral.proquest.com/lib/pitt-ebooks/detail.action?docID=1754641.

[54] National Center for Atmospheric Research. *MM5 Modeling System Overview*. University Center for Atmospheric Research. URL: http://www2.mmm.ucar.edu/mm5/overview.html.

[55] National Center for Atmospheric Research Staff (Eds). *The Climate Data Guide: ERA40*. University Center for Atmospheric Research. URL: https://climatedataguide.ucar.edu/climate-data/era40.

[56] National Center for Atmospheric Research Staff (Eds). *The Climate Data Guide: NCEP Reanalysis (R2)*. University Center for Atmospheric Research. URL: https://climatedataguide.ucar.edu/climate-data/ncep-reanalysis-r2.

[57] National Centers for Environmental Information. *Reanalysis*. NOAA. URL: https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/reanalysis.

[58] National Climatic Data Center. *DATA DOCUMENTATION FOR DATA SET 6405 (DSI-6405), ASOS SURFACE 1-MINUTE, PAGE 1 DATA*. Tech. rep. National Cli-

matic Data Center, 151 Patton Ave., Asheville, NC 28801-5001 USA, July 2006. URL: http://www1.ncdc.noaa.gov/pub/data/documentlibrary/tddoc/td6405.pdf.

[59] National Oceanic and Atmospheric Administration et al. *Automated Surface Observing System (ASOS) User's Guide*. Tech. rep. Mar. 1998. URL: https://www.weather.gov/media/asos/aum-toc.pdf.

[60] National Weather Service. *National Weather Service Glossary*. URL: https://w1.weather.gov/glossary/.

[61] NCEP Central Operations. *Meteorological Assimilation Data Ingest System*. National Oceanic and Atmospheric Administration. URL: https://madis.ncep.noaa.gov/.

[62] NOAA. *Beaufort Wind Scale*. URL: https://www.spc.noaa.gov/faq/tornado/beaufort.html.

[63] North American Reliability Corporation. *Understanding the Grid*. Aug. 2013. URL: https://www.nerc.com/AboutNERC/Documents/Understanding%20the%20Grid%20AUG13.pdf.

[64] NREL. *Energy Systems Analysis Tools*. National Renewable Energy Laboratory (NREL). URL: https://www.nrel.gov/analysis/energy-systems-tools.html.

[65] NREL. *Wind Resource Assessment*. National Renewable Energy Laboratory (NREL). URL: https://www.nrel.gov/wind/resource-assessment.html.

[66] National Renewable Energy Laboratory (NREL). *National Solar Radiation Data Base, 1991-2010 Update*. URL: http://rredc.nrel.gov/solar/old_data/nsrdb/1991-2010.

[67] NREL WISDEM Team. *WISDEM®*. National Renewable Energy Laboratory (NREL). URL: https://nwtc.nrel.gov/WISDEM.

[68] NREL WISDEM Team: K. Dykes, S. A. Ning, P. Graf, G. Scott, Y. Guo, R. King, T. Parsons, R. Damiani, P. Fleming. *Documentation for WISDEM*. National Renewable Energy Laboratory (NREL). URL: http://wisdem.github.io/WISDEM/documentation.html.

[69] Office of Electricity Delivery and Energy Reliability. *The role of microgrids in helping to advance the nation's energy system*. URL: http://energy.gov/oe/services/technology-development/smart-grid/role-microgrids-helping-advance-nation-s-energy-system.

[70] Office of Energy Efficiency & Renewable Energy. *How Do Wind Turbines Survive Severe Storms?* U.S. Department of Energy. URL: https://www.energy.gov/eere/articles/how-do-wind-turbines-survive-severe-storms.

[71] F Oteri. *Overview of Existing Wind Energy Ordinances*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2008.

[72] Pacific Northwest National Laboratory. *Mann-Kendall Test For Monotonic Trend*. URL: https://vsp.pnnl.gov/help/vsample/design_trend_mann_kendall.htm.

[73] Pacific Northwest National Laboratory. *What is VSP*. URL: https://vsp.pnnl.gov/help/Vsample/What_is_VSP.htm.

[74] John Pitre. *Folding blade turbine*. US Patent App. 12/461,716. Aug. 2009.

[75] James A Potter. *Retractable wind machine*. US Patent 4,342,539. Aug. 1982.

[76] Warren B Powell. "A unified framework for stochastic optimization". In: *European Journal of Operational Research* (2018).

[77] SC Pryor et al. "Wind speed trends over the contiguous United States". In: *Journal of Geophysical Research: Atmospheres* 114.D14 (2009).

[78] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

[79] Research Applications Laboratory. *Bringing the wind to the grid*. National Center for Atmospheric Research. URL: https://ral.ucar.edu/solutions/bringing-the-wind-to-the-grid.

[80] T.J. Ross. *Fuzzy Logic with Engineering Applications*. Wiley, 2016. ISBN: 9781119235842. URL: https://books.google.com/books?id=ZFAeDQAAQBAJ.

[81] Daniel F Salas and Warren B Powell. "Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Grid-Level Energy Storage". In: *INFORMS Journal on Computing* 30.1 (2017), pp. 106–123.

[82] P. Scheckel. *The Homeowner's Energy Handbook: Your Guide to Getting Off the Grid*. Storey Publishing, LLC, 2013. ISBN: 9781603428477. URL: https://books.google.com/books?id=owPp4RMxw8IC.

[83] Norbert Schwarz. "Self-reports: how the questions shape the answers." In: *American psychologist* 54.2 (1999), p. 93.

[84] JV Seguro and TW Lambert. "Modern estimation of the parameters of the Weibull wind speed distribution for wind energy analysis". In: *Journal of wind engineering and industrial aerodynamics* 85.1 (2000), pp. 75–84.

[85] Jeremy Ryan Shipley, Jeffrey F Kelly, and Winifred F Frick. "Toward integrating citizen science and radar data for migrant bird conservation". In: *Remote Sensing in Ecology and Conservation* (2017).

[86] Olivia J Smith, Natashia Boland, and Hamish Waterer. "Solving shortest path problems with a weight constraint and replenishment arcs". In: *Computers & Operations Research* 39.5 (2012), pp. 964–984.

[87] Staff Reporter. *Planners say no to wind turbine at Donegal business park*. Donegal Democrat. January 14, 2017. URL: http://www.donegaldemocrat.ie/news/news/230908/planners-say-no-to-wind-turbine-at-business-park.html.

[88] Iain Staffell. "Wind turbine power curves". In: *Imperial College London* (2012), p. 2. URL: http://www.academia.edu/1489838/Wind_Turbine_Power_Curves.

[89] Swanson School of Engineering. *Daniel Jiang, Daniel Jiang*. University of Pittsburgh. URL: https://www.engineering.pitt.edu/People/Faculty/Profiles/Daniel-Jiang/.

[90] Jeffrey Swofford and Michael Slattery. "Public attitudes of wind energy in Texas: Local communities in close proximity to wind farms and their effect on decision-making". In: *Energy policy* 38.5 (2010), pp. 2508–2519.

[91] Time and Date AS. *Past Weather in Orlando, Florida, USA — October 2010*. URL: https://www.timeanddate.com/weather/usa/orlando/historic?month=10&year=2010.

[92] Jeff Tollefson. *US electrical grid on the edge of failure, Network analysis suggests geography makes grid inherently unstable*. URL: http://www.nature.com/news/us-electrical-grid-on-the-edge-of-failure-1.13598.

[93] Verónica Torralba, Francisco J Doblas-Reyes, and Nube Gonzalez-Reviriego. "Uncertainty in recent near-surface wind speed trends: a global reanalysis intercomparison". In: *Environmental Research Letters* 12.11 (2017), p. 114019.

[94] Kishor Shridharbhai Trivedi and ProQuest (Firm). *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. Wiley, 2016. ISBN: 9781119314202. URL: https://books.google.com/books?id=_yOXDAAAQBAJ.

[95] Tom Tschida. *Inflatable Wing Deployment Sequence*. Dryden Flight Research Center. 2001. URL: https://www.dfrc.nasa.gov/Gallery/Photo/InflatableWing/HTML/ED01-0273-1.html.

[96] Tux Turkel. *Why Maine's renewable power remains a pipe dream*. Portland Press Herald. December 19, 2016. URL: http://www.pressherald.com/2016/12/19/why-maines-renewable-power-remains-a-pipe-dream/.

[97] Sakari M Uppala et al. "The ERA-40 re-analysis". In: *Quarterly Journal of the royal meteorological society* 131.612 (2005), pp. 2961–3012.

[98] U.S. Census Bureau. *2010 Census, 2010 Census Summary File 1, Selected Age Groups: 2010 - United States – Metropolitan Statistical Area; and for Puerto Rico, Table PCT12*. generated by Guy Gadola; using American FactFinder; (4 April 2018). URL: http://factfinder.census.gov.

[99] U.S. Energy Information Administration. *Electricity Explained, How Electricity Is Delivered To Consumers*. URL: http://www.eia.gov/energyexplained/index.cfm?page=electricity_delivery.

[100] U.S. Energy Information Administration. *What is the electric power grid and what are some challenges it faces?* Sept. 2014. URL: https://web.archive.org/web/20150224173513/http://www.eia.gov/energy_in_brief/article/power_grid.cfm.

[101] Vestas Wind Systems A/S. *General Specification for V90 - 3.0 MW*. Mar. 2004. URL: https://web.archive.org/web/20180403132610/http://www.gov.pe.ca/photos/sites/envengfor/file/950010R1_V90-GeneralSpecification.pdf.

[102] Weather Underground. *About Our Data*. URL: https://www.wunderground.com/about/data.

[103] S. Wilcox and W. Marion. *Users Manual for TMY3 Data Sets*. Tech. rep. NREL/TP-581-43156. National Renewable Energy Laboratory (NREL), Revised May 2008. URL: http://www.nrel.gov/docs/fy08osti/43156.pdf.

[104] Wind Energy - The Facts (WindFacts). *Wake Effect*. A Consortium led by the European Wind Energy Association. URL: https://www.wind-energy-the-facts.org/wake-effect.html.

A.1 PSEUDO-CODE FOR THE ALGORITHMS

A.1.1 Most recent revision of each variant of each category

Algorithm 1 The Static algorithm (variant 0x0, i.e., current-weather only, transitions open)

Revision 1.2

procedure STATIC0X0(out, S, m, d, Q, hm, c)
▷ Output: out has either the value **deploy** which means "deploy or remain deployed" or **retract** which means "retract or remain retracted" (out is a static; Its value at its first call is **retract**)
▷ Input: S = set of minute-by-minute windspeed samples
▷ Input: m = window size (in samples) of moving average windspeed
▷ Input: d = deployment threshold (in knots)
▷ Input: Q = set of timestamps in quiet hours
▷ Input: hm = harvester model
▷ Input: c = allocated visibility minutes per month

    $r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

    FRACTION_VISIBLE_TIME_THRESHOLD $\leftarrow$ 0.99
    for all $s \in S$ **do** ▷ For each raw windspeed sample
        $w.avg \leftarrow$ updateMovingAverage($s.raw, m$)
        ▷ m-sized window includes latest raw windspeed

        if $(w.avg < r)$ or ▷ Avg. windspeed is less than retraction threshold
$(s.timestamp \in Q)$ or ▷ In quiet hours
$((s.timestamp + hm.retraction\_time) \in Q)$ or ▷ Allow time to stow
$((hm.bInCutOutState$(s.raw) $\in Q)$ or ▷ Boolean: Harvester in cut-out
$(hm.getFractionVisbilePlusTimeToRetractMonthly(c) >$
FRACTION_VISIBLE_TIME_THRESHOLD) **then**
            $out \leftarrow$ **retract**
        else if $w.avg \geq d$ **then**
            $out \leftarrow$ **deploy**
        end if
    end for
end procedure

Algorithm 2 The Static algorithm (variant 0x1, i.e., current-weather only, transitions limited) Revision 1.2

procedure STATIC0X1(out, S, m, d, Q, hm, c)
▷ Output: out has either the value **deploy** which means "deploy or remain deployed" or **retract** which means "retract or remain retracted" (out is a static; Its value at its first call is **retract**)
▷ S = set of minute-by-minute windspeed samples
▷ m = window size (in samples) of moving average windspeed
▷ d = deployment threshold (in knots)
▷ Q = set of timestamps in quiet hours
▷ hm = harvester model
▷ c = allocated visibility minutes per month
  $r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

  FRACTION_VISIBLE_TIME_THRESHOLD $\leftarrow$ 0.99
  for all $s \in S$ **do** ▷ For each raw windspeed sample
    $w.avg \leftarrow$ updateMovingAverage($s.raw, m$)
▷ m-sized window includes latest raw windspeed

    if ($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out
(($hm.getFractionVisbilePlusTimeToRetractMonthly(c) >$
FRACTION_VISIBLE_TIME_THRESHOLD)
then
      $out \leftarrow$ **retract**
    else if ($hm.getFractionVisbilePlusTimeToRetractMonthly(c) < 1$) and
(getMinutesInMonthRemaining($s.date$) $< c$) or
▷ harvester has not yet been visible this month
(($w.avg > d$) ▷ Note: $\geq$ is permitted
) **then**

      $out \leftarrow$ **deploy**
    end if
  end for
end procedure

Algorithm 3 The Static algorithm (variant 0x2, i.e., weather prediction, transitions open)

Revision 1.2

```
procedure Static0x2(out, S, m, d, Q, hm, c)
        ▷ Output: out has either the value deploy which means "deploy or remain deployed"
    or retract which means "retract or remain retracted" (out is a static; Its value at its first call
    is retract)
                                        ▷ S = set of minute-by-minute windspeed samples
                              ▷ m = window size (in samples) of moving average windspeed
                                                  ▷ d = deployment threshold (in knots)
                                                   ▷ Q = set of timestamps in quiet hours
                                                                ▷ hm = harvester model
                                             ▷ c = allocated visibility minutes per month
    r ← d − 1                                    ▷ Calc. retraction threshold (in knots)

    FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99
    MUCH_WINDIER ← 1.25
    FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED = 0.64;
    for all s ∈ S do                                 ▷ For each raw windspeed sample
        w.avg ← updateMovingAverage(s.raw, m)
                                          ▷ m-sized window includes latest raw windspeed

        f.avg ← updateMovingAverageFuture(s.f1440, m)
                             ▷ m-sized window includes latest simulated predicted windspeed

        if (w.avg < r) or                 ▷ Avg. windspeed is less than retraction threshold
(s.timestamp ∈ Q) or                                                   ▷ In quiet hours
((s.timestamp + hm.retraction_time) ∈ Q) or                           ▷ Allow time to stow
(hm.bInCutOutState(s.raw) or                             ▷ Boolean: Harvester in cut-out
((hm.getFractionVisbilePlusTimeToRetractMonthly(c) >
FRACTION_VISIBLE_TIME_THRESHOLD) or
( (hm.getFractionVisbilePlusTimeToRetractMonthly(c) >
FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED) and
(f.avg > (w.avg∗MUCH_WINDIER)) )
        then
            out ← retract
        else if w.avg ≥ d then
            out ← deploy
        end if
    end for
end procedure
```

Algorithm 4 The Static algorithm (variant 0x3, i.e., weather prediction, transitions limited)

Revision 1.2

```
procedure Static0x3(out, S, m, d, Q, hm, c)
        ▷ Output: out has either the value deploy which means "deploy or remain deployed"
          or retract which means "retract or remain retracted" (out is a static; Its value at its first call
          is retract)
                                                  ▷ S = set of minute-by-minute windspeed samples
                                    ▷ m = window size (in samples) of moving average windspeed
                                                        ▷ d = deployment threshold (in knots)
                                                       ▷ Q = set of timestamps in quiet hours
                                                                  ▷ hm = harvester model
                                                   ▷ c = allocated visibility minutes per month
    r ← d − 1                                         ▷ Calc. retraction threshold (in knots)

    FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99
    for all s ∈ S do                                       ▷ For each raw windspeed sample
        w.avg ← updateMovingAverage(s.raw, m)
                                              ▷ m-sized window includes latest raw windspeed

        f.avg ← updateMovingAverageFuture(s.f1440, m)
                                  ▷ m-sized window includes latest simulated predicted windspeed

        if (hm.bInCutOutState(s.raw) or                      ▷ Boolean: Harvester in cut-out
((hm.getFractionVisbilePlusTimeToRetractMonthly(c) >
FRACTION_VISIBLE_TIME_THRESHOLD)
 then
            out ← retract
        else if (hm.getFractionVisbilePlusTimeToRetractMonthly(c) < 1) and
(getMinutesInMonthRemaining(s.date) < c) or
                                             ▷ harvester has not yet been visible this month
( (w.avg > d) and
                                                                  ▷ Note: ≥ is permitted
(!((f.avg > (w.avg*MUCH_WINDIER))                 ▷ much windier tomorrow ) ) )  then

            out ← deploy
        end if
    end for
end procedure
```

Algorithm 5 The Aging algorithm (variant 0x0, i.e., current-weather only, transitions open)

Revision 1.2

procedure AGING0X0(out, S, m, d, Q, hm, c)
▷ Output: out has either the value **deploy** which means "deploy or remain deployed" or **retract** which means "retract or remain retracted" (out is a static; Its value at its first call is **retract**)
▷ S = set of minute-by-minute windspeed samples
▷ $m[1:12]$ = array of monthly window sizes (in samples) of moving average windspeed
▷ $b[1:12]$ = array of monthly y-intercepts (in knots)
▷ Q = set of timestamps in quiet hours
▷ hm = harvester model
▷ l = lowest windspeed deemed to be windy
▷ c = allocated visibility minutes per month

FRACTION_VISIBLE_TIME_THRESHOLD $\leftarrow$ 0.99
for all $s \in S$ **do** ▷ For each raw windspeed sample
$d \leftarrow$ getDeploymentThreshold($s.timestamp$, $b[s.month]$, l)
▷ $d = -mx + b$ where $m = (l - b[s.month])/$(minutes in 31 days) and x is number of minutes remaining in month
▷ Slope is negated since x decreases as month ages
$r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

$w.avg \leftarrow$ updateMovingAverage($s.raw, m$)
▷ m-sized window includes latest raw windspeed

if ($w.avg < r$) or ▷ Avg. windspeed is less than retraction threshold
($s.timestamp \in Q$) or ▷ In quiet hours
(($s.timestamp + hm.retraction\_time) \in Q$) or ▷ Allow time to stow
($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out
(($hm.getFractionVisbilePlusTimeToRetractMonthly(c) >$
FRACTION_VISIBLE_TIME_THRESHOLD)
then
$out \leftarrow$ **retract**
else if $w.avg \geq d$ **then**
$out \leftarrow$ **deploy**
end if
end for
end procedure

Algorithm 6 The Aging algorithm (variant 0x1, i.e., current-weather only, transitions limited) Revision 1.2

procedure AGING0X1(out, S, m, d, Q, hm, c)
▷ Output: out has either the value **deploy** which means "deploy or remain deployed" or **retract** which means "retract or remain retracted" (out is a static; Its value at its first call is **retract**)
▷ S = set of minute-by-minute windspeed samples
▷ $m[1:12]$ = array of monthly window sizes (in samples) of moving average windspeed
▷ $b[1:12]$ = array of monthly y-intercepts (in knots)
▷ Q = set of timestamps in quiet hours
▷ hm = harvester model
▷ l = lowest windspeed deemed to be windy
▷ c = allocated visibility minutes per month

FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99
for all $s \in S$ **do** ▷ For each raw windspeed sample
$d \leftarrow$ getDeploymentThreshold($s.timestamp$, $b[s.month]$, l)
▷ $d = -mx + b$ where $m = (l - b[s.month])/(\text{minutes in 31 days})$ and x is number of minutes remaining in month
▷ Slope is negated since x decreases as month ages
$r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

$w.avg \leftarrow$ updateMovingAverage$(s.raw, m)$
▷ m-sized window includes latest raw windspeed

if ($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out
(($hm.getFractionVisbilePlusTimeToRetractMonthly(c)$ >
FRACTION_VISIBLE_TIME_THRESHOLD)
then
$out \leftarrow$ **retract**
else if ($hm.getFractionVisbilePlusTimeToRetractMonthly(c) < 1$) and
(getMinutesInMonthRemaining($s.date$) $< c$) or
▷ harvester has not yet been visible this month
(($w.avg > d$) ▷ Note: $\geq$ is permitted
) **then**

$out \leftarrow$ **deploy**
end if
end for
end procedure

Algorithm 7 The Aging algorithm (variant 0x2, i.e., weather prediction, transitions open)

Revision 1.2

procedure AGING0X2(out, S, m, d, Q, hm, c)

▷ Output: out has either the value **deploy** which means “deploy or remain deployed” or **retract** which means “retract or remain retracted” (out is a static; Its value at its first call is **retract**)

▷ S = set of minute-by-minute windspeed samples

▷ $m[1:12]$ = array of monthly window sizes (in samples) of moving average windspeed

▷ $b[1:12]$ = array of monthly y-intercepts (in knots)

▷ Q = set of timestamps in quiet hours

▷ hm = harvester model

▷ l = lowest windspeed deemed to be windy

▷ c = allocated visibility minutes per month

    for all $s \in S$ **do** ▷ For each raw windspeed sample

        $d \leftarrow$ getDeploymentThreshold($s.timestamp$, $b[s.month]$, l)

▷ $d = mx + b$ where $m = (l - b[s.month])/$(minutes in 31 days) and x is number of minutes remaining in month

▷ Bug in Revision 1.2: m’s sign is wrong

        $r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

        $w.avg \leftarrow$ updateMovingAverage($s.raw, m$)

▷ m-sized window includes latest raw windspeed

        $f.avg \leftarrow$ updateMovingAverageFuture($s.f1440, m$)

▷ m-sized window includes latest simulated predicted windspeed

        MUCH_WINDIER = 1.25;

        FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED = 0.64;

        if ($w.avg < r$) or ▷ Avg. windspeed is less than retraction threshold

($s.timestamp \in Q$) or ▷ In quiet hours

(($s.timestamp + hm.retraction\_time) \in Q$) or ▷ Allow time to stow

($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out

(($hm.getFractionVisbilePlusTimeToRetractMonthly(c) >$

FRACTION_VISIBLE_TIME_THRESHOLD) or

(($hm.getFractionVisbilePlusTimeToRetractMonthly(c) >$

FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED) and

($f.avg > (w.avg *$MUCH_WINDIER)))

        then

            $out \leftarrow$ **retract**

        else if $w.avg \geq d$ **then**

            $out \leftarrow$ **deploy**

        end if

    end for

end procedure

Algorithm 8 The Aging algorithm (variant 0x3, i.e., weather prediction, transitions limited)

Revision 1.2

procedure AGING0X3(out, S, m, d, Q, hm, c)
▷ Output: out has either the value **deploy** which means "deploy or remain deployed" or **retract** which means "retract or remain retracted" (out is a static; Its value at its first call is **retract**)
▷ S = set of minute-by-minute windspeed samples
▷ $m[1:12]$ = array of monthly window sizes (in samples) of moving average windspeed
▷ $b[1:12]$ = array of monthly y-intercepts (in knots)
▷ Q = set of timestamps in quiet hours
▷ hm = harvester model
▷ l = lowest windspeed deemed to be windy
▷ c = allocated visibility minutes per month

FRACTION_VISIBLE_TIME_THRESHOLD $\leftarrow$ 0.99
MUCH_WINDIER $\leftarrow$ 1.25
for all $s \in S$ **do** ▷ For each raw windspeed sample
$d \leftarrow$ getDeploymentThreshold($s.timestamp$, $b[s.month]$, l)
▷ $d = -mx + b$ where $m = (l - b[s.month])/(\text{minutes in 31 days})$ and x is number of minutes remaining in month
▷ Slope is negated since x decreases as month ages
$r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

$w.avg \leftarrow$ updateMovingAverage($s.raw, m$)
▷ m-sized window includes latest raw windspeed

$f.avg \leftarrow$ updateMovingAverageFuture($s.f1440, m$)
▷ m-sized window includes latest simulated predicted windspeed

if ($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out
(($hm.getFractionVisibilePlusTimeToRetractMonthly(c)$ >
FRACTION_VISIBLE_TIME_THRESHOLD)
then
$out \leftarrow$ **retract**
else if ($hm.getFractionVisibilePlusTimeToRetractMonthly(c) < 1$) and
(getMinutesInMonthRemaining($s.date$) $< c$) or
▷ harvester has not yet been visible this month
(($w.avg > d$) and
▷ Note: $\geq$ is permitted
(!(($f.avg > (w.avg*$MUCH_WINDIER)) ▷ much windier tomorrow))) **then**

$out \leftarrow$ **deploy**
end if
end for
end procedure

Algorithm 9 The Fuzzy-Crisp algorithm (variant 0x0, i.e., current-weather only, transitions open) Revision 1.5

```
procedure FuzzyCrisp0x0(out, S, m, d, Q, hm, c)
        ▷ Output: out has either the value deploy which means "deploy or remain deployed"
          or retract which means "retract or remain retracted" (out is a static; Its value at its first call
          is retract)
        ▷ S = set of minute-by-minute windspeed samples
        ▷ m[1 : 12] = array of monthly window sizes (in samples) of moving average windspeed
        ▷ v[1 : 12] = array of monthly membership values in combined fuzzy set
        ▷ Q = set of timestamps in quiet hours
        ▷ hm = harvester model
        ▷ l = lowest windspeed deemed to be windy

    r ← l − 1                                           ▷ Calc. retraction threshold (in knots)

    for all s ∈ S do                                    ▷ For each raw windspeed sample
        w.avg ← updateMovingAverage(s.raw, m)
                                                        ▷ m-sized window includes latest raw windspeed

        if (w.avg < r) or                               ▷ Avg. windspeed is less than retraction threshold
(s.timestamp ∈ Q) or                                    ▷ In quiet hours
((s.timestamp + hm.retraction_time) ∈ Q) or             ▷ Allow time to stow
hm.bInCutOutState(s.raw)                                ▷ Boolean: Harvester in cut-out
 then
            out ← retract
                                                        ▷ Ensure that algorithm meets agreement, which is crisp

        else                                            ▷ Use fuzzy-code to deploy
            if (windy and
(if not approaching quiet hours or
if fraction of time spent stowed is low))
≥ v[s.month] then
                out ← deploy
            end if
        end if
    end for
end procedure
```

Algorithm 10 The Fuzzy-Crisp algorithm (variant 0x1, i.e., current-weather only, transitions limited) (Transition-Limited) Revision 1.5

```
procedure FuzzyCrisp0x1(out, S, m, d, Q, hm, c)
        ▷ Output: out has either the value deploy which means "deploy or remain deployed"
        or retract which means "retract or remain retracted" (out is a static; Its value at its first call
        is retract)
                                        ▷ S = set of minute-by-minute windspeed samples
        ▷ m[1 : 12] = array of monthly window sizes (in samples) of moving average windspeed
                        ▷ v[1 : 12] = array of monthly membership values in combined fuzzy set
                                                   ▷ Q = set of timestamps in quiet hours
                                                                  ▷ hm = harvester model
                                              ▷ l = lowest windspeed deemed to be windy
                                              ▷ c = allocated visibility minutes per month
    r ← l − 1                                  ▷ Calc. retraction threshold (in knots)

    for all s ∈ S do                                ▷ For each raw windspeed sample
        w.avg ← updateMovingAverage(s.raw, m)
                                      ▷ m-sized window includes latest raw windspeed

        FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99

        if (hm.bInCutOutState(s.raw) or                 ▷ Boolean: Harvester in cut-out
(hm.getFractionVisbilePlusTimeToRetractMonthly(c) >
FRACTION_VISIBLE_TIME_THRESHOLD) or
(getMinutesInMonthRemaining(s.date) ≤ hm.TIME_TO_RETRACT_MINUTES)
                                              ▷ End the month retracted (not required)
 then
            out ← retract

        else if (hm.getFractionVisbilePlusTimeToRetractMonthly(c) < 1) then
                                                           ▷ Use fuzzy-code to deploy
            if (windy or
ApproachingUseItOrLoseItTimePoint) then
                out ← deploy
            end if
        end if
    end for
end procedure
```

Algorithm 11 The Fuzzy-Crisp algorithm (variant 0x2, i.e., weather prediction, transitions open) (Weather-prediction-using) Revision 1.5

```
procedure FuzzyCrisp0x2(out, S, m, d, Q, hm, c)
    ▷ Output: out has either the value deploy which means "deploy or remain deployed" or retract which means "retract or remain retracted" (out is a static; Its value at its first call is retract)
                                                            ▷ S = set of minute-by-minute windspeed samples
                                                            ▷ m[1 : 12] = array of monthly window sizes (in samples) of moving average windspeed
                                                            ▷ v[1 : 12] = array of monthly membership values in combined fuzzy set
                                                            ▷ Q = set of timestamps in quiet hours
                                                            ▷ hm = harvester model
                                                            ▷ l = lowest windspeed deemed to be windy

    r ← l − 1                                               ▷ Calc. retraction threshold (in knots)

    for all s ∈ S do                                        ▷ For each raw windspeed sample
        w.avg ← updateMovingAverage(s.raw, m)
                                                            ▷ m-sized window includes latest raw windspeed

        f.avg ← updateMovingAverageFuture(s.f1440, m)
                                                            ▷ m-sized window includes latest day-ahead predicted windspeed

        bFutureWindspeedUnavailable = (s.f1440 < 0)         ▷ Boolean
        if (w.avg < r) or                                   ▷ Avg. windspeed is less than retraction threshold
(s.timestamp ∈ Q) or                                        ▷ In quiet hours
((s.timestamp + hm.retraction_time) ∈ Q) or                 ▷ Allow time to stow
hm.bInCutOutState(s.raw)                                    ▷ Boolean: Harvester in cut-out
 then
            out ← retract
                                                            ▷ Ensure that algorithm meets agreement, which is crisp

        else                                                ▷ Use fuzzy-code to deploy
            if bFutureWindspeedUnavailable then
                if (windy and
(if not approaching quiet hours or
if fraction of time spent stowed is low))
 then
                    out ← deploy
                else
                    if ((very windy tomorrow and
running out of time) or
not very windy tomorrow or
not running out of time) and
(windy and
(if not approaching quiet hours or
if fraction of time spent stowed is low)) then
                        out ← deploy
                    end if
                end if
            end if
        end if
    end for
end procedure
```

Algorithm 12 The Fuzzy-Crisp algorithm (variant 0x3, i.e., weather prediction, transitions limited) (Transition-limited, Weather-prediction-using) Revision 1.5

```
procedure FuzzyCrisp0x3(out, S, m, d, Q, hm, c)
        ▷ Output: out has either the value deploy which means "deploy or remain deployed"
    or retract which means "retract or remain retracted" (out is a static; Its value at its first call
    is retract)
                                        ▷ S = set of minute-by-minute windspeed samples
        ▷ m[1 : 12] = array of monthly window sizes (in samples) of moving average windspeed
                        ▷ v[1 : 12] = array of monthly membership values in combined fuzzy set
                                                ▷ Q = set of timestamps in quiet hours
                                                            ▷ hm = harvester model
                                            ▷ l = lowest windspeed deemed to be windy
                                            ▷ c = allocated visibility minutes per month
    r ← l − 1                                   ▷ Calc. retraction threshold (in knots)

    for all s ∈ S do                                ▷ For each raw windspeed sample
        w.avg ← updateMovingAverage(s.raw, m)
                                        ▷ m-sized window includes latest raw windspeed

        f.avg ← updateMovingAverageFuture(s.f1440, m)
                            ▷ m-sized window includes latest simulated predicted windspeed

        bFutureWindspeedUnavailable = (s.f1440 < 0)                         ▷ Boolean
        FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99
        if (hm.bInCutOutState(s.raw) or                 ▷ Boolean: Harvester in cut-out
(hm.getFractionVisbilePlusTimeToRetractMonthly(c) >
FRACTION_VISIBLE_TIME_THRESHOLD) or
(getMinutesInMonthRemaining(s.date) ≤ hm.TIME_TO_RETRACT_MINUTES)
                                            ▷ End the month retracted (not required)
 then
            out ← retract

        else if (hm.getFractionVisbilePlusTimeToRetractMonthly(c) < 1) then
                                                            ▷ Use fuzzy-code to deploy
            if bFutureWindspeedUnavailable then
                if (windy or
ApproachingUseItOrLoseItTimePoint)  then
                    out ← deploy
                end if
            else
                if ((windy today and tomorrow) or
ApproachingUseItOrLoseItTimePoint)  then
                    out ← deploy
                end if
            end if
        end if
    end for
end procedure
```

A.1.2 Previous revision of each variant of each category

Algorithm 13 The Static algorithm (variant 0x0) Revision 1.1

procedure STATIC0X0(S, m, d, Q, hm, c)
▷ S = set of minute-by-minute windspeed samples
▷ m = window size (in samples) of moving average windspeed
▷ d = deployment threshold (in knots)
▷ Q = set of timestamps in quiet hours
▷ hm = harvester model
▷ c = allocated visibility minutes per month
  $r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

  FRACTION_VISIBLE_TIME_THRESHOLD $\leftarrow 0.99$
  for all $s \in S$ **do** ▷ For each raw windspeed sample
    $w.avg \leftarrow$ updateMovingAverage$(s.raw, m)$
▷ m-sized window includes latest raw windspeed

    if $(w.avg < r)$ or ▷ Avg. windspeed is less than retraction threshold
$(s.timestamp \in Q)$ or ▷ In quiet hours
$((s.timestamp + hm.retraction\_time) \in Q)$ or ▷ Allow time to stow
$((hm.bInCutOutState$(s.raw) $\in Q)$ or ▷ Boolean: Harvester in cut-out
$(hm.getFractionVisbilePlusTimeToRetractMonthly(c) >$
FRACTION_VISIBLE_TIME_THRESHOLD) **then**
      RetractOrRemainRetracted
    else if $w.avg \geq d$ **then**
      DeployOrRemainDeployed
    end if
  end for
end procedure

Algorithm 14 The Static algorithm (variant 0x1) (Transition-limited) Revision 1.1

```
procedure Static0x1(S, m, d, Q, hm, c)
                                         ▷ S = set of minute-by-minute windspeed samples
                                         ▷ m = window size (in samples) of moving average windspeed
                                         ▷ d = deployment threshold (in knots)
                                         ▷ Q = set of timestamps in quiet hours
                                         ▷ hm = harvester model
                                         ▷ c = allocated visibility minutes per month
    r ← d − 1                            ▷ Calc. retraction threshold (in knots)

    FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99
    for all s ∈ S do                     ▷ For each raw windspeed sample
        w.avg ← updateMovingAverage(s.raw, m)
                                         ▷ m-sized window includes latest raw windspeed

        updateMovingAverage(s.f1440, m)
▷ Bug in Revision 1.1: MovingAverage instead of MovingAverageFuture is being updated with weather prediction

        if (hm.bInCutOutState(s.raw) or  ▷ Boolean: Harvester in cut-out
((hm.getFractionVisbilePlusTimeToRetractMonthly(c) >
FRACTION_VISIBLE_TIME_THRESHOLD)
 then
            RetractOrRemainRetracted
        else if (hm.getFractionVisbilePlusTimeToRetractMonthly(c) < 1) and
(getMinutesInMonthRemaining(s.date) < c) or
                                         ▷ harvester has not yet been visible this month
( (w.avg > d)                            ▷ Note: ≥ is permitted
)  then

            DeployOrRemainDeployed
        end if
    end for
end procedure
```

Algorithm 15 The Static algorithm (variant 0x2) (Weather-prediction-using) Revision 1.1

procedure STATIC0X2(S, m, d, Q, hm, c)
  ▷ S = set of minute-by-minute windspeed samples
  ▷ m = window size (in samples) of moving average windspeed
  ▷ d = deployment threshold (in knots)
  ▷ Q = set of timestamps in quiet hours
  ▷ hm = harvester model
  ▷ c = allocated visibility minutes per month
  $r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

  FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99
  MUCH_WINDIER ← 1.25
  FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED = 0.64;
  for all $s \in S$ **do** ▷ For each raw windspeed sample
    $w.avg \leftarrow$ updateMovingAverage($s.raw, m$)
    ▷ m-sized window includes latest raw windspeed

    $f.avg \leftarrow$ updateMovingAverage($s.f1440, m$)
    ▷ Bug (or unintended feature) in Revision 1.1: MovingAverage instead of MovingAverageFuture is being updated with weather prediction
    if ($w.avg < r$) or ▷ Avg. windspeed is less than retraction threshold
($s.timestamp \in Q$) or ▷ In quiet hours
(($s.timestamp + hm.retraction\_time$) $\in Q$) or ▷ Allow time to stow
($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out
(($hm.getFractionVisbilePlusTimeToRetractMonthly(c) >$
FRACTION_VISIBLE_TIME_THRESHOLD) or
(($hm.getFractionVisbilePlusTimeToRetractMonthly(c) >$
FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED) and
($f.avg >$ ($w.avg*$MUCH_WINDIER)))
    then
      RetractOrRemainRetracted
    else if $w.avg \geq d$ **then**
      DeployOrRemainDeployed
    end if
  end for
end procedure

Algorithm 16 The Static algorithm (variant 0x3) (Transition-limited, Weather-prediction-using) Revision 1.1

procedure STATIC0X3(S, m, d, Q, hm, c)
  ▷ S = set of minute-by-minute windspeed samples
  ▷ m = window size (in samples) of moving average windspeed
  ▷ d = deployment threshold (in knots)
  ▷ Q = set of timestamps in quiet hours
  ▷ hm = harvester model
  ▷ c = allocated visibility minutes per month
  $r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

  FRACTION_VISIBLE_TIME_THRESHOLD $\leftarrow$ 0.99
  for all $s \in S$ **do** ▷ For each raw windspeed sample
    $w.avg \leftarrow$ updateMovingAverage($s.raw, m$)
    ▷ m-sized window includes latest raw windspeed

    $f.avg \leftarrow$ updateMovingAverage($s.f1440, m$)
    ▷ Bug (or unintended feature) in Revision 1.1: MovingAverage instead of MovingAverageFuture is being updated with weather prediction

    if ($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out
  (($hm.getFractionVisbilePlusTimeToRetractMonthly(c) >$
  FRACTION_VISIBLE_TIME_THRESHOLD)
  then
      RetractOrRemainRetracted
    else if ($hm.getFractionVisbilePlusTimeToRetractMonthly(c) < 1$) and
  (getMinutesInMonthRemaining($s.date$) $< c$) or
  ▷ harvester has not yet been visible this month
  (($w.avg > d$) and
  ▷ Note: $\geq$ is permitted
  (!(($f.avg > (w.avg *$MUCH_WINDIER)) ▷ much windier tomorrow))) **then**

      DeployOrRemainDeployed
    end if
  end for
end procedure

Algorithm 17 The Aging algorithm (variant 0x0) Revision 1.1

procedure AGING0X0(S, $m[1:12]$, $b[1:12]$, Q, hm, l, c)
▷ S = set of minute-by-minute windspeed samples
▷ $m[1:12]$ = array of monthly window sizes (in samples) of moving average windspeed
▷ $b[1:12]$ = array of monthly y-intercepts (in knots)
▷ Q = set of timestamps in quiet hours
▷ hm = harvester model
▷ l = lowest windspeed deemed to be windy
▷ c = allocated visibility minutes per month

FRACTION_VISIBLE_TIME_THRESHOLD $\leftarrow$ 0.99
for all $s \in S$ **do** ▷ For each raw windspeed sample
$d \leftarrow$ getDeploymentThreshold($s.timestamp$, $b[s.month]$, l)
▷ $d = mx + b$ where $m = (l - b[s.month])/$(minutes in 31 days) and x is number of minutes remaining in month
▷ Bug in Revision 1.1: m's sign is wrong
$r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

$w.avg \leftarrow$ updateMovingAverage($s.raw, m$)
▷ m-sized window includes latest raw windspeed

if ($w.avg < r$) or ▷ Avg. windspeed is less than retraction threshold
($s.timestamp \in Q$) or ▷ In quiet hours
(($s.timestamp + hm.retraction\_time) \in Q$) or ▷ Allow time to stow
($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out
(($hm.getFractionVisbilePlusTimeToRetractMonthly(c) >$
FRACTION_VISIBLE_TIME_THRESHOLD)
then
RetractOrRemainRetracted
else if $w.avg \geq d$ **then**
DeployOrRemainDeployed
end if
end for
end procedure

Algorithm 18 The Aging algorithm (variant 0x1) (Transition-limited) Revision 1.1

procedure AGING0X1(S, $m[1:12]$, $b[1:12]$, Q, hm, l, c)
▷ S = set of minute-by-minute windspeed samples
▷ $m[1:12]$ = array of monthly window sizes (in samples) of moving average windspeed
▷ $b[1:12]$ = array of monthly y-intercepts (in knots)
▷ Q = set of timestamps in quiet hours
▷ hm = harvester model
▷ l = lowest windspeed deemed to be windy
▷ c = allocated visibility minutes per month

  FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99
  for all $s \in S$ **do** ▷ For each raw windspeed sample
    $d \leftarrow$ getDeploymentThreshold($s.timestamp$, $b[s.month]$, l)
    ▷ $d = mx + b$ where $m = (l - b[s.month])/$(minutes in 31 days) and x is number of minutes remaining in month
    ▷ Bug in Revision 1.1: m's sign is wrong
    $r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

    $w.avg \leftarrow$ updateMovingAverage($s.raw, m$)
    ▷ m-sized window includes latest raw windspeed

    updateMovingAverage($s.f1440, m$)
    ▷ Bug in Revision 1.1: MovingAverage instead of MovingAverageFuture is being updated with weather prediction

    if ($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out
    (($hm.getFractionVisbilePlusTimeToRetractMonthly(c)$ >
    FRACTION_VISIBLE_TIME_THRESHOLD)
    then
      RetractOrRemainRetracted
    else if ($hm.getFractionVisbilePlusTimeToRetractMonthly(c)$ < 1) and
    (getMinutesInMonthRemaining($s.date$) < c) or
    ▷ harvester has not yet been visible this month
    (($w.avg > d$) ▷ Note: $\geq$ is permitted
) **then**

      DeployOrRemainDeployed
    end if
  end for
end procedure

Algorithm 19 The Aging algorithm (variant 0x2) (Weather-prediction-using) Revision 1.1

procedure AGING0X2(S, $m[1:12]$, $b[1:12]$, Q, hm, l, c)
▷ S = set of minute-by-minute windspeed samples
▷ $m[1:12]$ = array of monthly window sizes (in samples) of moving average windspeed
▷ $b[1:12]$ = array of monthly y-intercepts (in knots)
▷ Q = set of timestamps in quiet hours
▷ hm = harvester model
▷ l = lowest windspeed deemed to be windy
▷ c = allocated visibility minutes per month

    for all $s \in S$ **do** ▷ For each raw windspeed sample
        $d \leftarrow$ getDeploymentThreshold($s.timestamp$, $b[s.month]$, l)
▷ $d = mx + b$ where $m = (l - b[s.month])/$(minutes in 31 days) and x is number of minutes remaining in month
▷ Bug in Revision 1.1: m's sign is wrong
        $r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

        $w.avg \leftarrow$ updateMovingAverage$(s.raw, m)$
▷ m-sized window includes latest raw windspeed

        $f.avg \leftarrow$ updateMovingAverage$(s.f1440, m)$
▷ Bug (or unintended feature) in Revision 1.1: MovingAverage instead of MovingAverageFuture is being updated with weather prediction

        MUCH_WINDIER = 1.25;
        FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED = 0.64;
        if ($w.avg < r$) or ▷ Avg. windspeed is less than retraction threshold
($s.timestamp \in Q$) or ▷ In quiet hours
(($s.timestamp + hm.retraction\_time$) $\in Q$) or ▷ Allow time to stow
($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out
(($hm.getFractionVisbilePlusTimeToRetractMonthly(c)$ >
FRACTION_VISIBLE_TIME_THRESHOLD) or
(($hm.getFractionVisbilePlusTimeToRetractMonthly(c)$ >
FRACTION_VISIBLE_TIME_THRESHOLD_SOMEWHAT_EXHAUSTED) and
($f.avg >$ ($w.avg*$MUCH_WINDIER)))
        then
            RetractOrRemainRetracted
        else if $w.avg \geq d$ **then**
            DeployOrRemainDeployed
        end if
    end for
end procedure

Algorithm 20 The Aging algorithm (variant 0x3) (Transition-limited, Weather-prediction-using) Revision 1.1

procedure AGING0X3(S, $m[1:12]$, $b[1:12]$, Q, hm, l, c)
▷ S = set of minute-by-minute windspeed samples
▷ $m[1:12]$ = array of monthly window sizes (in samples) of moving average windspeed
▷ $b[1:12]$ = array of monthly y-intercepts (in knots)
▷ Q = set of timestamps in quiet hours
▷ hm = harvester model
▷ l = lowest windspeed deemed to be windy
▷ c = allocated visibility minutes per month

FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99
MUCH_WINDIER ← 1.25
for all $s \in S$ **do** ▷ For each raw windspeed sample
$d \leftarrow$ getDeploymentThreshold($s.timestamp$, $b[s.month]$, l)
▷ $d = mx + b$ where $m = (l - b[s.month])/$(minutes in 31 days) and x is number of minutes remaining in month
▷ Bug in Revision 1.1: m's sign is wrong
$r \leftarrow d - 1$ ▷ Calc. retraction threshold (in knots)

$w.avg \leftarrow$ updateMovingAverage$(s.raw, m)$
▷ m-sized window includes latest raw windspeed

$f.avg \leftarrow$ updateMovingAverage$(s.f1440, m)$
▷ Bug (or unintended feature) in Revision 1.1: MovingAverage instead of MovingAverageFuture is being updated with weather prediction

if ($hm.bInCutOutState$(s.raw) or ▷ Boolean: Harvester in cut-out
(($hm.getFractionVisbilePlusTimeToRetractMonthly(c)$ >
FRACTION_VISIBLE_TIME_THRESHOLD)
then
RetractOrRemainRetracted
else if ($hm.getFractionVisbilePlusTimeToRetractMonthly(c) < 1$) and
(getMinutesInMonthRemaining($s.date$) $< c$) or
▷ harvester has not yet been visible this month
(($w.avg > d$) and
▷ Note: $\geq$ is permitted
(!(($f.avg > (w.avg*$MUCH_WINDIER)) ▷ much windier tomorrow))) **then**

DeployOrRemainDeployed
end if
end for
end procedure

Algorithm 21 The Fuzzy-Crisp algorithm (variant 0x0) Revision 1.4

procedure FuzzyCrisp0x0(S, $m[1:12]$, $v[1:12]$, Q, hm, l)
▷ S = set of minute-by-minute windspeed samples
▷ $m[1:12]$ = array of monthly window sizes (in samples) of moving average windspeed
▷ $v[1:12]$ = array of monthly membership values in combined fuzzy set
▷ Q = set of timestamps in quiet hours
▷ hm = harvester model
▷ l = lowest windspeed deemed to be windy

    $r \leftarrow l - 1$ ▷ Calc. retraction threshold (in knots)

    for all $s \in S$ **do** ▷ For each raw windspeed sample
        $w.avg \leftarrow$ updateMovingAverage$(s.raw, m)$
▷ m-sized window includes latest raw windspeed

        if $(w.avg < r)$ or ▷ Avg. windspeed is less than retraction threshold
$(s.timestamp \in Q)$ or ▷ In quiet hours
$((s.timestamp + hm.retraction\_time) \in Q)$ or ▷ Allow time to stow
$hm.bInCutOutState(s.raw)$ ▷ Boolean: Harvester in cut-out
then
            RetractOrRemainRetracted
▷ Ensure that algorithm meets agreement, which is crisp

        else ▷ Use fuzzy-code to deploy
            if (windy and
(if not approaching quiet hours or
if fraction of time spent stowed is low))
$\geq v[s.month]$ **then**
                DeployOrRemainDeployed
            end if
        end if
    end for
end procedure

Algorithm 22 The Fuzzy-Crisp algorithm (variant 0x1) (Transition-Limited) Revision 1.4

```
procedure FuzzyCrisp0x1(S, m[1 : 12], v[1 : 12], Q, hm, l, c)
                                              ▷ S = set of minute-by-minute windspeed samples
        ▷ m[1 : 12] = array of monthly window sizes (in samples) of moving average windspeed
                        ▷ v[1 : 12] = array of monthly membership values in combined fuzzy set
                                                       ▷ Q = set of timestamps in quiet hours
                                                                   ▷ hm = harvester model
                                                 ▷ l = lowest windspeed deemed to be windy
                                                ▷ c = allocated visibility minutes per month
    r ← l − 1                                          ▷ Calc. retraction threshold (in knots)

    for all s ∈ S do                                       ▷ For each raw windspeed sample
        w.avg ← updateMovingAverage(s.raw, m)
                                           ▷ m-sized window includes latest raw windspeed

        updateMovingAverage(s.f1440, m)      ▷ Bug in Revision 1.4: MovingAverage instead of
    MovingAverageFuture is being updated with weather prediction
        FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99

        if (hm.bInCutOutState(s.raw) or                     ▷ Boolean: Harvester in cut-out
(hm.getFractionVisbilePlusTimeToRetractMonthly(c) >
FRACTION_VISIBLE_TIME_THRESHOLD) or
(getMinutesInMonthRemaining(s.date) ≤ hm.TIME_TO_RETRACT_MINUTES)
                                                    ▷ End the month retracted (not required)
 then
            RetractOrRemainRetracted

        else if (hm.getFractionVisbilePlusTimeToRetractMonthly(c) < 1) then
                                                               ▷ Use fuzzy-code to deploy
            if (windy or
ApproachingUseItOrLoseItTimePoint)  then
                Deploy
            end if
        end if
    end for
end procedure
```

Algorithm 23 The Fuzzy-Crisp algorithm (variant 0x2) (Weather-prediction-using) Revision 1.4

```
procedure FuzzyCrisp0x2(S, m[1 : 12], v[1 : 12], Q, hm, l)
                                    ▷ S = set of minute-by-minute windspeed samples
                                    ▷ m[1 : 12] = array of monthly window sizes (in samples) of moving average windspeed
                                    ▷ v[1 : 12] = array of monthly membership values in combined fuzzy set
                                    ▷ Q = set of timestamps in quiet hours
                                    ▷ hm = harvester model
                                    ▷ l = lowest windspeed deemed to be windy
   r ← l − 1                        ▷ Calc. retraction threshold (in knots)
    for all s ∈ S do                ▷ For each raw windspeed sample
        w.avg ← updateMovingAverage(s.raw, m)
                                    ▷ m-sized window includes latest raw windspeed
        f.avg ← updateMovingAverageFuture(s.f1440, m)
                                    ▷ m-sized window includes latest day-ahead predicted windspeed
        bFutureWindspeedUnavailable = (s.f1440 < 0)          ▷ Boolean
        if (w.avg < r) or           ▷ Avg. windspeed is less than retraction threshold
(s.timestamp ∈ Q) or                ▷ In quiet hours
((s.timestamp + hm.retraction_time) ∈ Q) or                  ▷ Allow time to stow
hm.bInCutOutState(s.raw)            ▷ Boolean: Harvester in cut-out
 then
            RetractOrRemainRetracted
                                    ▷ Ensure that algorithm meets agreement, which is crisp
        else                        ▷ Use fuzzy-code to deploy
            if bFutureWindspeedUnavailable then
                if (windy and
(if not approaching quiet hours or
if fraction of time spent stowed is low))
 then
                    DeployOrRemainDeployed
                else
                    if ((very windy tomorrow and
running out of time) or
not very windy tomorrow or
not running out of time) and
(windy and
(if not approaching quiet hours or
if fraction of time spent stowed is low)) then
                        DeployOrRemainDeployed
                    end if
                end if
            end if
        end if
    end for
end procedure
```

Algorithm 24 The Fuzzy-Crisp algorithm (variant 0x3) (Transition-limited, Weather-prediction-using) Revision 1.4

```
procedure FuzzyCrisp0x3(S, m[1 : 12], v[1 : 12], Q, hm, l, c)
                                        ▷ S = set of minute-by-minute windspeed samples
        ▷ m[1 : 12] = array of monthly window sizes (in samples) of moving average windspeed
                  ▷ v[1 : 12] = array of monthly membership values in combined fuzzy set
                                                   ▷ Q = set of timestamps in quiet hours
                                                               ▷ hm = harvester model
                                                ▷ l = lowest windspeed deemed to be windy
                                                ▷ c = allocated visibility minutes per month
    r ← l − 1                                       ▷ Calc. retraction threshold (in knots)

    for all s ∈ S do                                      ▷ For each raw windspeed sample
        w.avg ← updateMovingAverage(s.raw, m)
                                           ▷ m-sized window includes latest raw windspeed

        f.avg ← updateMovingAverage(s.f1440, m)    ▷ Bug (or unintended feature) in Revision
  1.4: MovingAverage instead of MovingAverageFuture is being updated with weather prediction
        bFutureWindspeedUnavailable = (s.f1440 < 0)                               ▷ Boolean
        FRACTION_VISIBLE_TIME_THRESHOLD ← 0.99
        if (hm.bInCutOutState(s.raw) or                  ▷ Boolean: Harvester in cut-out
(hm.getFractionVisbilePlusTimeToRetractMonthly(c) >
FRACTION_VISIBLE_TIME_THRESHOLD) or
(getMinutesInMonthRemaining(s.date) ≤ hm.TIME_TO_RETRACT_MINUTES)
                                                  ▷ End the month retracted (not required)
 then
            RetractOrRemainRetracted

        else if (hm.getFractionVisbilePlusTimeToRetractMonthly(c) < 1) then
                                                               ▷ Use fuzzy-code to deploy
            if bFutureWindspeedUnavailable then
                if (windy or
ApproachingUseItOrLoseItTimePoint)  then
                    Deploy
                end if
            else
                if ((windy today and tomorrow) or
ApproachingUseItOrLoseItTimePoint)  then
                    Deploy
                end if
            end if
        end if
    end for
end procedure
```

A.2 CREATING AND USING FUZZY-SET MEMBERSHIP FUNCTIONS

A.2.1 Example of how to make a membership function for the fuzzy set NOT WINDY AT KBOS

We define the fuzzy term *not windy enough* or *not windy* to mean *not powerful enough* or *not powerful*, which is another fuzzy term, but nonetheless our equating *not windy* with *not powerful* is the reason why we used a power calculation in the following procedure:

1. Load the training data for the KBOS family of benchmarks into R. “R is a free software environment for statistical computing and graphics” [78].

   The following R command creates the data frame *df*:

   The data frame *df* has five columns as shown here. As stated in Section 4.2.0.3, the column name “f1440” indicates that the column’s values are simulated predicted wind-speeds 1440 minutes in advance of the current time. The name suggests that additional columns with predicted windspeeds at different time horizons may be added in future benchmarks suite revisions.

```
> names(df)
[1] "timestamp"
[2] "wind_knots"
[3] "actual.vs..interpolated"
[4] "f1440"
[5] "OLA5"
```

2. From data frame *df*, extract the wind speed data, which is the *wind_knots* column:

3. Create a frequency distribution of all the windspeeds in the KBOS training data (which includes interpolated windspeeds for the missing minutes, as mentioned in Section 4.2).

```
> # Generate sequence 0,1,2,...,125
> breaks = seq(0, 125, by=1)
> # Label each value in wind_knots with the interval, e.g., [0,1)
    ,
> #   to which it belongs
> wind_knots.cut = cut(wind_knots, breaks, right=FALSE)
> # Build a table of counts of each applied label, i.e., interval
```

```
> wind_knots.freq = table(wind_knots.cut)
```

4. For each "bucket" of windspeeds, we calculate the power that that bucket generated via the following the formula [30]:

$$P = nw^3$$

where w is the label, i.e., a windspeed, on the bucket and n is the number of instances of that windspeed in that bucket.

5. Because we are defining *not windy* to mean not powerful, we convert the windspeed to a wind power. Per the power equation above, as windspeed increases, wind power increases cubically. Thus, we cube `wind_knots`:

```
> # Combine sequence 0,1,...,124 into vector
> bucket_label_wind_knots = c(0:124)
> # Multiply each cell by cube of its windspeed
> wind_power = wind_knots.freq * (bucket_label_wind_knots ^ 3)
> # Cumulatively sum the vector wind_power
> wind_power.cumsum = cumsum(wind_power)
> # Divide by 125th cumulative sum
> wind_power.relativecumsum = wind_power.cumsum / wind_power.
  cumsum[125]
```

6. Now that we have created

```
> x <- cbind(wind_power.relativecumsum/.8)
```

7. Because dividing numbers greater than .8 results in quotients greater than 1, we cap the quotients at 1 (and invert by subtracting from 1 because we want a membership function for the set NOT WINDY):

```
> membership_function_windy_not <- 1 - apply(x, 1, function(x) if
  (x>1) 1 else x)
```

8. To create the plot of the membership function (Figure 28), use the following commands, where `xlab` and `ylab` define the x-axis and y-axis titles, respectively:

```
> for_plot = cbind(seq(0,124), membership_function_windy_not)
> png()
> plot(for_plot[,1], for_plot[,2], xlim=c(0,20), ylab=
+ "Degree of Membership in 'Not Windy at KBOS'", xlab='Windspeed
  (knots)',
+ main="Membership Function for Fuzzy Set 'Not Windy at KBOS'")
```

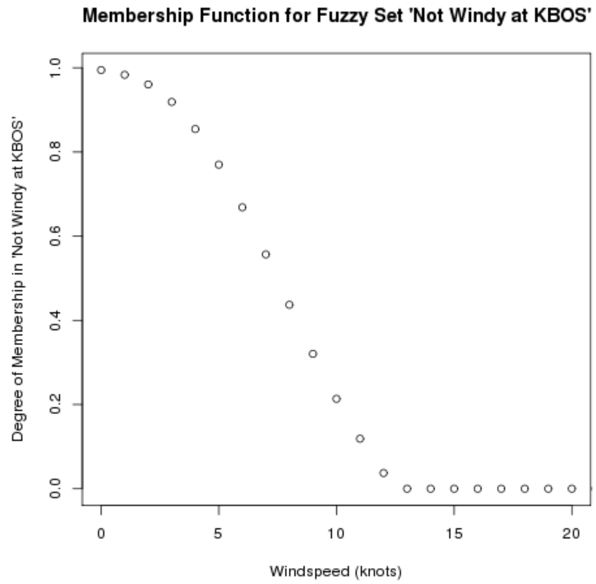

Figure 28: Membership function for fuzzy set 'Not Windy at KBOS' where KBOS refers to weather station KBOS

```
6  > dev.off()
```

The plot of the membership function (Figure 28) shows how the function maps each windspeed (measured in knots) in its domain to degrees of membership in the set NOT WINDY AT KBOS. The membership function can be used by fuzzy algorithms to determine how windy the weather measured by a windspeed is.

A.2.1.1 Files describing fuzzy-set membership in the set NOT WINDY AT <STATION>

We integrated the R code essentially the same as that directly above into an R script to create a NOT WINDY membership function for each training set (i.e., we created a membership function for each of the 30 weather stations). We are including each NOT WINDY membership function in the benchmark suite. Each NOT WINDY membership function is in its own file named "training<Station>2004-

2012imembershipFunctionNotWindy.out" where <Station> is the weather station's four-character ICAO code (e.g, KPIT). An example of a version of "trainingKPIT2004-2012imembershipFunctionNotWindy.out" is shown in Listing 1. The left-hand column is a set of ranges of windspeeds (measured in knots). Each element of that set is mapped to a membership value. For example, the range [0,1) knots is mapped to the membership value of 1.000. Thus, the range [0,1) is fully a member of the set NOT WINDY AT KPIT. For another example, the range [15,16) knots is mapped to the membership value of 0.167; That range's degree of membership in NOT WINDY AT KPIT is 0.167. Because the degree of membership in the set NOT WINDY AT KPIT is 0.00 for all windspeeds 18 knots and greater, rows in the range [23,123] knots have been omitted from Listing 1:

Listing 1: trainingKPIT2004-2012imembershipFunctionNotWindy.out

```
             membership_function_windy_not
[0,1)                                1.000
[1,2)                                1.000
[2,3)                                0.999
[3,4)                                0.995
[4,5)                                0.982
[5,6)                                0.959
[6,7)                                0.921
[7,8)                                0.866
[8,9)                                0.796
[9,10)                               0.712
[10,11)                              0.619
[11,12)                              0.523
[12,13)                              0.427
[13,14)                              0.334
[14,15)                              0.247
[15,16)                              0.167
[16,17)                              0.095
[17,18)                              0.032
[18,19)                              0.000
[19,20)                              0.000
[20,21)                              0.000
[21,22)                              0.000
...
[124,125)                            0.000
```

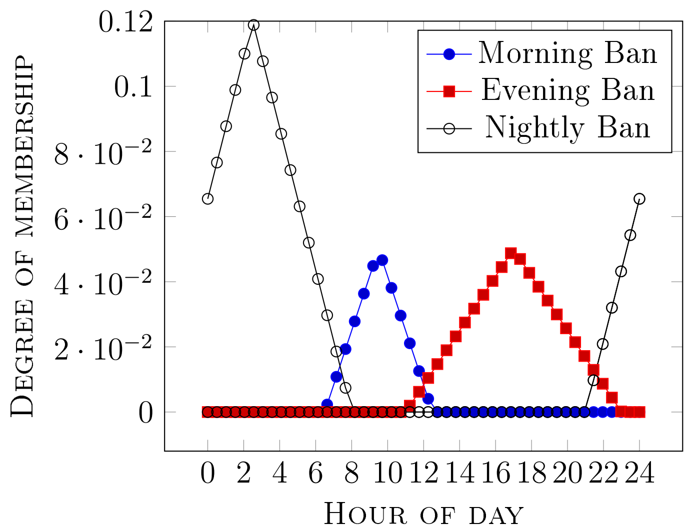

Figure 29: Example membership functions for the fuzzy sets representing morning, evening, and nightly visibility bans

A.2.2 Example of how to make fuzzy-set membership function for set DAILY VISIBILITY BAN [IN EFFECT]

Let T be the set of all minutes in a day. Thus, the elements 0:00 and 23:59 are in set T. Let T have the following fuzzy subsets:

- MORNING BAN
- AFTERNOON BAN
- NIGHTLY BAN

Let us make the membership functions from a daily repeating triangular function where the height of the triangle is dependent on the survey results. An example of such a daily repeating triangular function is shown here:

$$\begin{aligned} &\mathit{dailyTriangle}(\mathit{hour}, \mathit{scale}, \mathit{halfbase}, \mathit{midpoint}) = \\ &\quad \mathit{scale} \times \max(0, (\mathit{normalizer}(\mathit{halfbase})) \times (2 \times \min(\mathit{dailyPattern}(\mathit{hour}, \mathit{midpoint}), \\ &\qquad\qquad 2 - \mathit{dailyPattern}(\mathit{hour}, \mathit{midpoint} - \mathit{voffset}(\mathit{halfbase}))); \quad (1) \end{aligned}$$

where

$$dailyPattern(hour, midpoint) = \mod (hour - (midpoint - 12), 24)/12; \tag{2}$$

$$interimHeight(halfbase) = halfbase/6; \tag{3}$$

$$normalizer(halfbase) = 1/(interimHeight(halfbase)); \tag{4}$$

$$voffset(halfbase) = 2 - interimHeight(halfbase); \tag{5}$$

Let us continue this example by deriving membership functions from our actual survey data. The survey results indicate that 245 respondents "[s]upport retractable wind turbines under certain conditions." The following numbers of those 245 respondents chose to ban harvester visibility in the morning, afternoon, and night: 13, 13, and 29 or 5%, 5%, and 12%, respectively. Let the morning, afternoon, and night hours be 07:00-12:00, 12:00-22:00, and 22:00-07:00, respectively. Because the survey did not explicitly define the intervals, we are using the common understanding that noon divides morning and afternoon, and for night hours we are using the quiet hours in Pittsburgh (Table A.10 on page 214). The midpoints and durations of each range (e.g. the midpoint and duration of 22:00-07:00 are 02:30 and 9 hours, respectively) used to derive the midpoint and width of each triangle, where the width is one hour greater than the duration to allow the following membership functions to overlap slightly (Figure 29) for this example:

$$mf_{MORNING\_BAN} = dailyTriangle(hour, scale = 0.05, halfbase = 3, midpoint = 9.5); \tag{6}$$

$$mf_{AFTERNOON\_BAN} = dailyTriangle(hour, scale = 0.05, halfbase = 6, midpoint = 17); \tag{7}$$

$$mf_{NIGHTLY\_BAN} = dailyTriangle(hour, scale = 0.12, halfbase = 5.5, midpoint = 2.5); \tag{8}$$

For example, suppose that the time of day is 19:00 (i.e., 7 p.m.). The degree of membership of 19:00 in the fuzzy set DAILY BAN is

$$\begin{aligned}
mf_{DAILY_BAN}(x) &= mf_{MORNING_BAN}(x) \vee mf_{AFTERNOON_BAN}(x) \vee \\
&\quad mf_{NIGHTLY_BAN}(x) \\
&= mf_{MORNING_BAN}(19{:}00) \vee mf_{AFTERNOON_BAN}(19{:}00) \vee \\
&\quad mf_{NIGHTLY_BAN}(19{:}00) \\
&= \max(mf_{MORNING_BAN}(19{:}00), mf_{AFTERNOON_BAN}(19{:}00), \\
&\quad mf_{NIGHTLY_BAN}(19{:}00)) \\
&= \max(0, 0.03, 0) \\
&= 0.03
\end{aligned}$$

Let us continue this example in the following section by using $mf_{DAILY\_VISIBILITY\_BAN}$ as part of a larger expresion.

A.2.3 Translating membership values to a retraction speed

Suppose that a harvester has a maximum retraction speed M. We are assuming that the maximum retraction speed is the same as the deployment speed for the following example. (Deployment speed is the inverse of deployment time, which is defined in Appendix A.3.5.) And suppose that the harvester is able to be deployed at any speed in the range [M/2, M]. Then, we can create a function that maps membership values to a retraction speed in the range [M/2, M]:

$$\begin{aligned}
y = 0.5M((&mf_{DAILY_VISIBILITY_BAN}(t) \vee mf_{NOT_WINDY}(x) \vee \\
&mf_{TOO_WINDY}(x) \vee mf_{USED_ALL_ITS_ALLOCATED_VISIBILITY}(h)) + 1) \qquad (9)
\end{aligned}$$

where x is the windspeed, t is the time of day, h is how much time the harvester has been visible, the symbol $\vee$ is the maximum operator, and y is the resulting deployment speed.

Let us continue the example from the previous section where $mf_{DAILY\_VISIBILITY\_BAN}$ equals 0.03. Suppose that a wind speed x_0 has a degree of membership of 0.25 in the set

NOT_WINDY, a degree of membership of 0 in the set TOO_WINDY, and 0.75 in the fuzzy set USED_ALL_ITS_ALLOCATED_VISIBILITY. Then Equation 9 becomes

$$
\begin{aligned}
y &= 0.5M((0.03 \vee 0.25 \vee 0 \vee 0.75) + 1) && (10)\\
&= 0.5M(\max(0.03, 0.25, 0, 0.75) + 1) && (11)\\
&= 0.5M(0.75 + 1) && (12)\\
&= 0.5M(1.75) && (13)\\
&= 0.875M. && (14)
\end{aligned}
$$

which translates the membership values to 87.5% of the maximum retraction speed.

For Equation 9, as the highest membership value increases, the retraction speed increases, which is by design, because each fuzzy set in Equation 9 is a reason to retract the harvester.

A problem with using fuzzy logic to retract the harvester is that the controlling authority would need to agree that the harvester is complying with operational restrictions if the algorithm is using an agreed-upon fuzzy equation, which might be more difficult to understand than a crisp operational restriction. We avoided that problem by using crisp code to retract in the Fuzzy-Crisp hybrid algorithm (Section 3.4.3 on page 35).

A.2.4 Windspeeds deemed "windy" for crisp applications

Let us describe how to create a lambda-cut set from the NOT_WINDY membership function (Appendix A.2.1) of each weather station ws. (Recall that a general definition of a lambda-cut set can be found in Ross [80] and in this case, the lambda-cut set is the set of all windspeeds in the fuzzy set NOT WINDY having membership values of λ or higher.) Each station's NOT_WINDY membership function $\mu_{(NOT\_WINDY,ws)}()$ maps a windspeed s to a membership value $\mu_{(NOT\_WINDY,ws)}(s) \in M = \{x \in \mathbb{R} | x \in [0,1]\}$. Choose a $\lambda \in M$. The lambda-cut set of the fuzzy set NOT_WINDY is made up of all windspeeds that have membership values equal to or greater than λ. That is, $(NOT\_WINDY, ws)_\lambda = \{y | \mu_{(NOT\_WINDY,ws)}(y) \geq \lambda\}$.

For each station ws, we deem windspeeds (that are in the domain of $\mu_{(NOT\_WINDY,ws)}()$) to be "windy" if those windspeeds are not in $(NOT\_WINDY, ws)_\lambda$, where $\lambda = 0.9$. For each station ws, the least windspeed that we deem to be "windy" in the set of natural numbers is given in Table 11. Each "Lowest Windspeed Deemed to Be Windy" happens either be a Light Breeze (4-6 knots) or a Gentle Breeze (7-10 knots) as defined by the Beaufort Wind Scale (Table 10 on the next page [62]) when $\lambda = 0.9$.

Table 10: Beaufort Wind Scale

Wind (knots)	WMO Classification
Less than 1	Calm
1-3	Light Air
4-6	Light Breeze
7-10	Gentle Breeze
11-16	Moderate Breeze
17-21	Fresh Breeze
22-27	Strong Breeze
28-33	Near Gale
34-40	Gale
41-47	Strong Gale
48-55	Storm
56-63	Violent Storm
64+	Hurricane

Before choosing 0.9, we also explored other values for λ, which did not perform well in our initial testing of an Aging algorithm.

Table 11: Lowest windspeed deemed to be "windy" for each station for six values of the parameter λ.

λ	0.4	0.5	0.6	0.7	0.8	0.9
station	Lowest Windspeed Deemed to Be Windy (knots)					
KATL	12	11	10	9	8	7
KBOS	15	14	13	12	10	9
KBWI	13	12	10	9	8	7
KCLE	14	13	12	11	10	8
KCLT	10	9	8	8	7	5
KCVG	14	13	12	10	9	7
KDCA	13	12	11	10	9	7
KDEN	18	15	14	12	10	9
KDFW	16	14	13	12	11	9
KDTW	14	13	12	10	9	8
KEUG	11	10	10	9	8	6
KIAH	13	12	11	10	9	7
KLAS	14	13	12	11	9	7
KLAX	12	11	10	9	8	7
KLGA	15	14	13	12	10	9
KMCI	14	13	12	11	10	8
KMCO	13	12	11	10	9	7
KMSP	13	12	11	10	9	8
KORD	13	12	11	10	9	8
KPHL	14	13	12	11	9	8

Table 11: (continued)

KPHX	12	11	10	9	7	6
KPIT	13	12	11	10	8	7
KSAC	12	11	10	9	7	6
KSAN	10	9	8	8	7	6
KSAT	12	11	11	10	9	7
KSEA	11	11	10	9	8	6
KSFO	17	16	15	14	12	10
KSMX	15	13	12	11	10	8
KSTL	13	12	11	10	9	7
KTPA	10	9	8	8	7	6
Max.	18	16	15	14	12	10
Min.	10	9	8	8	7	5

A.3 MODEL OF HARVESTER THAT IS TOWERED

This version of the benchmark suite has one (for simplicity) wind harvester model: a towered harvester. We chose to derive the model of the towered retracted harvester from the Vestas V90-3.0 MW.[1]

The wind harvester model comprises the following items, each explained in a subsection below:

- a power curve (A.3.1), i.e., a power output function, which maps windspeed to power output,
- a deployment energy (A.3.3) function, which maps windspeed to the energy required to deploy the harvester,

[1] The Vestas V90-3.0 MW has the median rotor diameter, 90 m, of the five diameters of the five Vestas turbines (V34, V80, V90, V112, and V164) listed in Staffell [88].

- a retraction energy (A.3.4) function, which maps windspeed to the energy required to retract the harvester, and
- deployment speed (A.3.5) and retraction speed parameters.

A.3.1 Power curve

The power outputs of the V90-3.0 MW are given in Table 12 [88] for a range of windspeeds. The power output function uses the data shown in Table 12 to find the power that the V90 outputs given a windspeed. We assume that the turbine is available 100% of the time. A plot of Table 12 is shown in Figure 30 on page 163. That is, we assume that scheduled or unscheduled events (e.g., maintenance) will not interfere with the harvester's energy harvesting. Note that Table 12 indicates that the cut-out speed is 49 knots, at and above which the turbine outputs 0 kW. Cutting-out power output to 0 is an effort to protect certain components. However, that effort might not be sufficient if the windspeeds become extremely high. The actual V90's design parameter is 82.6 knots or $42.5\,\mathrm{m\,s^{-1}}$ for *maximum average wind* defined as "10 min., 50 years' mean wind speed" [101, Section 4.2] if winds are within other limits such as turbulence [101, Section 4.2]. (We assume that the envisioned retractable version of the V90 turbine can survive winds up to at least 80 knots.)

Table 12: Power output of the V90-3.0 MW vs. windspeed [88]

Windspeed (knots)	Power (kW)	Windspeed (knots)	Power (kW)
0	0	30	3000
1	0	31	3000
2	0	32	3000
3	0	33	3000
4	0	34	3000
5	0	35	3000
6	0	36	3000
7	38	37	3000
8	87	38	3000
9	141	39	3000
10	207	40	3000
11	286	41	3000
12	382	42	3000
13	496	43	3000
14	628	44	3000
15	781	45	3000
16	957	46	3000
17	1153	47	3000
18	1366	48	3000
19	1589	49	0
20	1814	50	0
21	2035	51	0
22	2250	52	0
23	2452	53	0
24	2634	54	0
25	2784	55	0
26	2888	56	0
27	2949	57	0
28	2980	58	0
29	2993	59	0
continues above right		60 [and more]	0

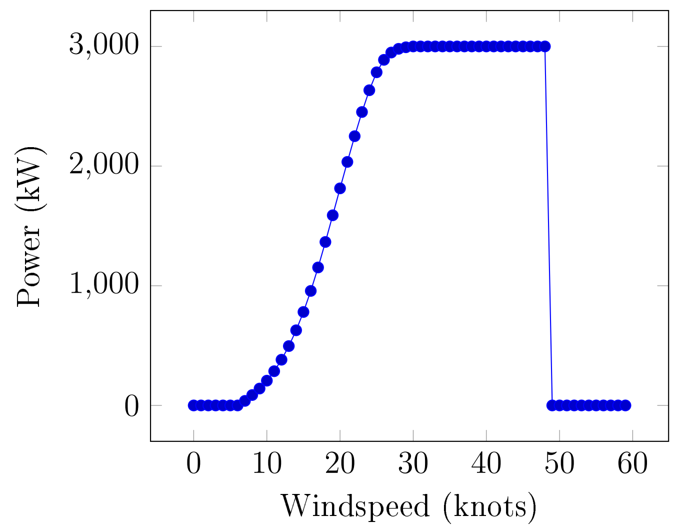

Figure 30: Power curve of harvester model

A.3.2 Too windy

We are interpreting *too windy* to mean too windy to harvest. Being too windy to harvest is defined by the following constants in a V90-3MW specification [101, p. 22], where *cut-out speed* is the speed at which and above the harvester outputs no power (because turbines are designed to try to avoid damage from high winds by feathering the blades, i.e., increasing the pitch angle of each blade to reduce the blade surface area facing the wind [70] [52, Page 5]) and *recut-in speed* is the speed to which the wind must decrease before the harvester resumes outputting power after a cut-out:

- CUT_OUT_SPEED_KNOTS = 49 (25 m/s) and
- RECUT_IN_SPEED_KNOTS = 39 (20 m/s).

The V90's cut-out speed of 49 knots can be clearly seen in Table 12. Note that Table 12 does not show the recut-in speed; Table 12 does not apply to situations where the harvester is in cut-out mode and windspeeds are between 39 and 49, exclusive.[2]

The highest windspeed is the benchmark suite is 67 knots, which in the file

[2]Those "pre-recut-in" situations were ignored when we found the "ideal" deployment and retraction times for OLA5, as noted in a footnote referenced from Appendix 4.3.

trainingKMCO2004-2012in.csv and less than the 80-knot survivability windspeed of the modeled harvester, as stated in Appendix A.3.1. Recall that cutting the power output is an effort to survive high winds, but extremely high winds could damage a wind turbine. The cut-out speed of the V90 (49 knots) [88] is much less than the V90's designed survivability windspeed (82.6 knots) [101, Section 4.2].

A.3.3 Deployment energy

It takes a certain amount of energy to deploy a retractable harvester. Harvesters can conceivably be helped to be lifted by airfoils. Thus, our model calculates deployment energy as inversely proportional to windspeed. That is, as windspeeds increase, deployment energy decreases.

Our envisioned retractable-harvester relies solely on airfoils to deploy when the windspeed is 20 knots or higher. That is, when the windspeed is 20 knots or higher, all energy to lift the harvester is provided by airfoils' lifting force. We represent the 20-knot energy in Listing 2 on the next page by cubing the value 20 since wind power is cubically proportional to windspeed [30]. If the windspeed is less than 20 knots, then the harvester in our model consumes energy from the grid. In the code listing below, we refer to that required grid energy as `energy_required_to_lift_kwh_per_event` where each event is a deployment. The required grid energy is the maximum of 0 kWh and the product of a scaling factor (explained in the next paragraph) times the difference of 20 cubed less the current windspeed cubed. We normalize the event's required grid energy by dividing it by the number of minutes the harvester takes to go from being fully stowed to being fully deployed (i.e., the deployment time (Section A.3.5)), which is assumed to be 20 minutes.

The scaling factor ensures that the required grid energy is equal to the energy that the harvester captures per minute at a windspeed slightly above the harvester's cut-in windspeed. The cut-in windspeed of the V90 is 7 knots (Section A.3.1). At which, the V90 harnesses 38 kilowatts of power. If the 7-knot windspeed persists for an hour, the V90 harvests 38 kilowatt-hours of energy (Section A.3.1). Thus, each minute at 7 knots the harvester captures 38 kilowatt-hours/60 minutes = 0.63 kilowatt-hours/minute. At 8 knots, the harvester captures

1.45 kilowatt-hours/minute.

If the harvester is not fully deployed at 8 knots, then it is not converting wind energy to electricity. The required grid energy to *lift* the harvester at 8 knots is 1.50 kilowatt-hours/minute, at 9 knots is 1.45 kilowatt-hours/minute, at 10 knots is 1.40 kilowatt-hours/minute for 20 minutes, and continues to decrease (Table 13). The required energy is decreasing as windspeeds are increasing because the envisioned retractable harvester is using airfoils to help lift itself. Thus, although the harvester is not transforming wind power into electrical power as the harvester is deploying, the harvester may be directly transforming wind power into mechanical lifting power. If the aerodynamic lift provided by the airfoils is not sufficient, the harvester obtains the balance of lifting energy from the grid. The balance of lifting energy is coming from the grid, instead of from another source, in order to be consistent with the context of the MQMP metric (Section 4.1.5.2). The required grid energy of 1.5 kilowatt-hours/minute to lift the harvester is approximately equal to the 1.45 kilowatt-hours/minute that the harvester would have captured if it were fully deployed at 8 knots, which is approximately where the per-minute power curve crosses the plot of required lifting energy from the grid in Figure 31. Energy harvested if the harvester were *fully deployed* and the required lifting energy from the grid if the harvester is *being deployed* is listed in Table 13.

Listing 2: Code calculating how much energy is required to lift harvester

```
public int getDeploymentEnergyUsedPerMinuteKwh(int time_to_deploy_minutes, int windspeed_knots) {

  // Assuming all energy to lift harvester can be provided by airfoils
  // when wind speed is 20 knots or higher
  final int DEPLOYMENT_ENERGY_CONSUMPTION_THRESHOLD_KWH = (int) Math.pow(20, 3);
  final double DEPLOYMENT_ENERGY_SCALING_FACTOR = .004;

  double energy_required_to_lift_kwh_per_event = (DEPLOYMENT_ENERGY_CONSUMPTION_THRESHOLD_KWH
    - Math.pow(windspeed_knots, 3)) * DEPLOYMENT_ENERGY_SCALING_FACTOR;

  if (energy_required_to_lift_kwh_per_event < 0)
```

```
    energy_required_to_lift_kwh_per_event = 0;

  return ((int) energy_required_to_lift_kwh_per_event /
      time_to_deploy_minutes);

}
```

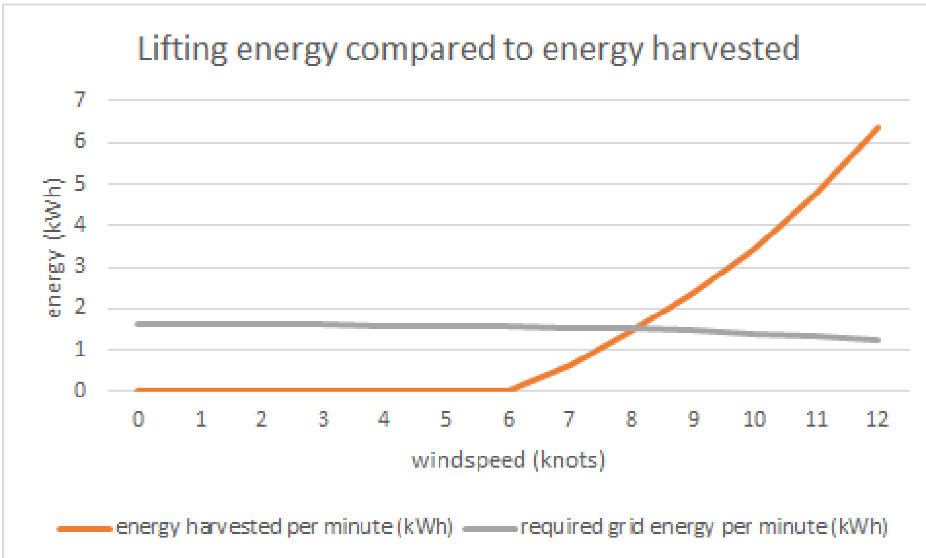

Figure 31: Comparing energy harvested when fully deployed to portion of lifting energy required from grid

A.3.4 Retraction energy

Retracting a retractable harvester could consume energy in some cases. However, our model assumes that gravity will retract the harvester without using any energy.

A.3.5 Deployment time

The *deployment time* is the inverse of deployment speed. A retractable harvester moves from its stowed state to its fully deployed state at a certain speed. That speed can be specified in terms of deployment time. For example, a deployment time of 20 minutes means that the speed at which the harvester moves from being fully stowed to being fully deployed is such

Table 13: Comparing energy harvested when fully deployed to portion of lifting energy required from grid

windspeed (knots)	power harvested (kW)	energy harvested per minute (kWh)	required grid energy per minute (kWh)
0	0	0.00	1.60
1	0	0.00	1.60
2	0	0.00	1.60
3	0	0.00	1.59
4	0	0.00	1.59
5	0	0.00	1.58
6	0	0.00	1.56
7	38	0.63	1.53
8	87	1.45	1.50
9	141	2.35	1.45
10	207	3.45	1.40
11	286	4.77	1.33
12	382	6.37	1.25
13	496	8.27	1.16
14	628	10.47	1.05
15	781	13.02	0.93
16	957	15.95	0.78
17	1153	19.22	0.62
18	1366	22.77	0.43
19	1589	26.48	0.23
20	1814	30.23	0.00
21	2035	33.92	0.00
22	2250	37.50	0.00
23	2452	40.87	0.00
24	2634	43.90	0.00
25	2784	46.40	0.00
26	2888	48.13	0.00
27	2949	49.15	0.00
28	2980	49.67	0.00
29	2993	49.88	0.00
30	3000	50.00	0.00
31	3000	50.00	0.00

that it takes 20 minutes for the harvester to make that rising transition. (A rising transition is defined as going from being fully stowed to being fully deployed.) Thus, the harvester's deployment speed in that case is 1 rising transition per 20 minutes.

For the harvester model specified by the OLAs of this revision of the benchmarks, we are assuming that the deployment speed is equal to the retraction speed. Such an assumption simplifies the modeling.

For this revision of the model, we chose the deployment and retraction time to each be 20 minutes, which falls into the 15-to-30-minute range of how much time it typically takes to erect a self-erecting crane [10]. " 'Self-erectors' approximate working ranges include jib radii (which defines 'reach') from 80 to 160 feet [24 to 49 meters], hook heights from 55 to 120 feet [17 to 37 meters]" [10]. Adding the approximate reach and hook height of 49 and 37 meters, respectively, results in 86 meters, which is more than the 80-meter tower of the V90-3MW wind turbine, the harvester on which our model is based; The V90-3MW has tower heights of 65 and 80 meters [101, Section 7.17].

A.4 WINDSPEED AND HOURLY ELECTRICITY PRICE DATA SOURCES

A.4.1 Windspeed data from qualifying weather stations in text

The benchmarks were partly derived from the Automated Surface Observing System (ASOS) data set DSI 6405 windspeed data[3] for the years 2004-2014 for 30 weather stations meeting the following criteria to facilitate *future work involving energy usage and solar energy harvesting*, each station has the following:

- A low energy usage file x and a high energy usage file y, where the files x and y are each a dataset file of hourly energy usage data[4] apparently (since the data is in a directory

[3] ASOS data set DSI 6405 does not include energy usage data.

[4] Outdoor weather affects how much energy many buildings use. The hourly energy usage data the cited directory [13] were derived from typical meteorological year version 3 (TMY3) data sets. "A typical meteorological year (TMY) data set... holds hourly meteorological values that typify conditions at a specific location over a longer period of time, such as 30 years. TMY data sets are widely used... for modeling

Table 14: Data for this work (wind data) and for future work (energy usage and solar data)

icao	maslib	Energy-usage profile filename root	Wind data filename e.g.	Solar data filename e.g.
KPHX	722780	USA_AZ_Phoenix-Sky.Harbor.Intl.AP.722780_TMY3	64050KPHX201401.dat	722780_1991_solar.csv
KLAX	722950	USA_CA_Los.Angeles.Intl.AP.722950_TMY3	64050KLAX201401.dat	722950_1991_solar.csv
KSAC	724830	USA_CA_Sacramento.Exec.AP.724830_TMY3	64050KSAC201401.dat	724830_1991_solar.csv
KSAN	722900	USA_CA_San.Diego-Lindbergh.Field.722900_TMY3	64050KSAN201401.dat	722900_1991_solar.csv
KSFO	724940	USA_CA_San.Francisco.Intl.AP.724940_TMY3	64050KSFO201401.dat	724940_1991_solar.csv
KSMX	723940	USA_CA_Santa.Maria.Public.AP.723940_TMY3	64050KSMX201401.dat	723940_1991_solar.csv
KDEN	725650	USA_CO_Denver.Intl.AP.725650_TMY3	64050KDEN201401.dat	725650_1991_solar.csv
KMCO	722050	USA_FL_Orlando.Intl.AP.722050_TMY3	64050KMCO201401.dat	722050_1991_solar.csv
KTPA	722110	USA_FL_Tampa.Intl.AP.722110_TMY3	64050KTPA201401.dat	722110_1991_solar.csv
KATL	722190	USA_GA_Atlanta-Hartsfield-Jackson.Intl.AP.722190_ TMY3	64050KATL201401.dat	722190_1991_solar.csv
KORD	725300	USA_IL_Chicago-OHare.Intl.AP.725300_TMY3	64050KORD201401.dat	725300_1991_solar.csv
KCVG	724210	USA_KY_Cincinnati-Northern.Kentucky.AP.724210_TMY3	64050KCVG201401.dat	724210_1991_solar.csv
KBOS	725090	USA_MA_Boston-Logan.Intl.AP.725090_TMY3	64050KBOS201401.dat	725090_1991_solar.csv
KBWI	724060	USA_MD_Baltimore-Washington.Intl.AP.724060_TMY3	64050KBWI201401.dat	724060_1991_solar.csv
KDTW	725370	USA_MI_Detroit.Metro.AP.725370_TMY3	64050KDTW201401.dat	725370_1991_solar.csv
KMSP	726580	USA_MN_Minneapolis-St.Paul.Intl.AP.726580_TMY3	64050KMSP201401.dat	726580_1991_solar.csv
KMCI	724460	USA_MO_Kansas.City.Intl.AP.724460_TMY3	64050KMCI201401.dat	724460_1991_solar.csv
KSTL	724340	USA_MO_St.Louis-Lambert.Intl.AP.724340_TMY3	64050KSTL201401.dat	724340_1991_solar.csv
KCLT	723140	USA_NC_Charlotte-Douglas.Intl.AP.723140_TMY3	64050KCLT201401.dat	723140_1991_solar.csv
KLAS	723860	USA_NV_Las.Vegas-McCarran.Intl.AP.723860_TMY3	64050KLAS201401.dat	723860_1991_solar.csv
KLGA	725030	USA_NY_New.York-LaGuardia.AP.725030_TMY3	64050KLGA201401.dat	725030_1991_solar.csv
KCLE	725240	USA_OH_Cleveland-Hopkins.Intl.AP.725240_TMY3	64050KCLE201401.dat	725240_1991_solar.csv
KEUG	726930	USA_OR_Eugene-Mahlon.Sweet.AP.726930_TMY3	64050KEUG201401.dat	726930_1991_solar.csv
KPHL	724080	USA_PA_Philadelphia.Intl.AP.724080_TMY3	64050KPHL201401.dat	724080_1991_solar.csv
KPIT	725200	USA_PA_Pittsburgh.Intl.AP.725200_TMY3	64050KPIT201401.dat	725200_1991_solar.csv
KDFW	722590	USA_TX_Dallas-Fort.Worth.Intl.AP.722590_TMY3	64050KDFW201401.dat	722590_1991_solar.csv
KIAH	722430	USA_TX_Houston-Bush.Intercontinental.AP.722430_TMY3	64050KIAH201401.dat	722430_1991_solar.csv
KSAT	722530	USA_TX_San.Antonio.Intl.AP.722530_TMY3	64050KSAT201401.dat	722530_1991_solar.csv
KDCA	724050	USA_VA_Arlington-Ronald.Reagan.Washington.Natl.AP. 724050_TMY3	64050KDCA201401.dat	724050_1991_solar.csv
KSEA	727930	USA_WA_Seattle-Tacoma.Intl.AP.727930_TMY3	64050KSEA201401.dat	727930_1991_solar.csv

named "RESIDENTIAL_LOAD_DATA_E_PLUS_OUTPUT" [13]) generated by the EnergyPlus building energy-consumption simulator [20] over one year in the cited Open Energy Information directory [13] and

- Solar energy data z from the years 1991-2010 where the solar data z is the station's hourly solar data from the updated 1991-2010 National Solar Radiation Database (NSRDB) [66].

Filename information of source data-files for this work (using wind data) and for future work (using energy usage and solar data) is shown in Table 14.

Nearly every minute, the ASOS data set DSI 6405 provides an average of windspeeds of the previous two minutes [58, p. 3]. We call each provided two-minute average a *sample* to help differentiate the two-minute average from the running average we calculate.

Because some records are missing or mis-formatted in the original ASOS data, we first filtered the original data to produce consistently formatted datasets. Details of that filtering are described in Appendix A.7.1.

The consistently formatted datasets have some missing minutes. Minutes are missing either because we filtered-out mis-formatted records or because some samples were not in the original data. From those formatted datasets with missing minutes, we produced a set of datasets having no missing minutes by interpolating windspeeds.

Each *training* record has five fields, which are comma delimited:

1. timestamp (e.g., `2009010100000600`) The timestamp field has the following subdivisions:
   - The first twelve characters are the local standard time in the format YYYYMMDDHHMM. The local standard time does not adjust for daylight saving time.
   - The next four characters are the Universal Coordinated Time (UTC) in the the format HHMM. The UTC time can be used to verify that the local timestamp is standard time and not daylight saving time.
2. wind_knots (e.g., 7) is the average windspeed over the previous two minutes measured in knots.

---

renewable energy conversion systems. Although not designed to provide meteorological extremes, TMY data have natural diurnal and seasonal variations and represent a year of typical climatic conditions for a location. *The TMY should not be used to predict weather for a particular period of time, nor is it an appropriate basis for evaluating real-time energy production or efficiencies for building design applications or solar conversion systems*" [103, emphasis in original].

3. actual vs. interpolated (e.g., a) is always a single letter: an 'a' indicates actual and an 'i' indicates interpolated windspeed, where the actual windspeed is the actual value found in the corresponding ASOS record.
4. f1440 is a column of simulated forecasted[5] day-ahead (which is 1440-minutes-ahead) windspeeds in knots. A negative one (-1) in this field of a specific row means that the forecast is not available for that row. We are not able to forecast beyond the final day of the windspeed data because we simulate day-ahead forecasts by looking at tomorrow's windspeeds and adjusting them using a Gaussian probability distribution, as described in Section 4.2.0.3. Thus, each record in the final day of each station's testing data has a -1 in it's f1440 column.
5. OLA5 is the column of "ideal" algorithm outputs to meet or exceed the requirements of OLA5 and OLA6. A 'd' and an 'r' indicate the harvester should deploy (or remain deployed) and should retract (or remain retracted), respectively. Column OLA5 serves as approximate "ground truth." The data in OLA5 is not necessarily optimal when the harvester experiences a cut-out because column OLA5 was generated via a weighted directed graph that does not model recut-in speeds that are less than the cut-out speed. Note: The OLA5 column is provided only in the training files because the OLA5 column is for algorithm training only and not for algorithm testing.

   Example records of a *training* file are shown here:

```
timestamp,wind_knots,actual vs. interpolated,f1440,OLA5
20040101221103 11,3,a,7,r
20040101221203 12,4,a,7,r
20040101221303 13,4,a,5,r
20040101221403 14,4,a,6,r
20040101221503 15,4,a,6,r
20040101221603 16,3,a,6,r
20040101221703 17,3,a,5,r
20040101221803 18,3,a,8,r
20040101221903 19,3,a,3,r
20040101222003 20,3,a,4,r
```

   Example records of a *testing* file follow:

```
timestamp,wind_knots,actual vs. interpolated,f1440
```

[5] We explain how we simulated the forecasts in Sec 4.2.0.3 on page 76.

```
2013010100000500,4,i,2
2013010100010501,4,i,6
2013010100020502,4,i,2
2013010100030503,4,i,6
2013010100040504,4,i,3
2013010100050505,4,i,3
2013010100060506,4,i,3
2013010100070507,4,i,4
2013010100080508,4,i,3
```

Each final windspeed file has the following naming convention. The prefix indicates whether the file is testing or training data via the full word *testing* or *training*. The next four characters indicate the weather station that measured the original data (e.g. KATL). The next nine characters indicate the date range of the file's contents (e.g., 2009-2014). (The benchmark suite might not have windspeed data for January 1, 2004, 00:00 and immediately subsequent minutes for some stations or for December 31, 2014, 23:59 and immediately preceding minutes because that data was not in the original ASOS data.) All local timestamps are not in daylight saving time, but standard time. The extension .csv identifies that the each record is comma separated. Two example filenames are given here:

`trainingKCLE2004-2012in.csv`

`testingKATL2013-2014in.csv`

A.4.2 Hourly electricity price data

We supply an electricity price for each hour in the years 2004 to 2014, inclusive. The prices are in the file named "HOEP(Hourly Ontario Energy Price)_2004-2014(CADperKWH).csv". The price of electricity (in kWh) is in Canadian dollars because we derived the file from data available from Ontario's Independent Electricity System Operator (IESO) [37]. The IESO website making the data available via http://www.ieso.ca/en/Power-Data/Data-Directory has terms of use requiring that the following notice be included with any information we use or reproduce from the IESO site: "Copyright ©2017 Independent Electricity System Operator, all rights reserved. This information is subject to the general terms of use set out in the IESO's website (www.ieso.ca)." Thus, the hourly electricity price data file that we derived from the IESO data has a two-line header followed by the data, for example:

```
"Copyright © 2017 Independent Electricity System Operator,....",,
Date,Hour (0-23),HOEP (C$/kWh)
1-Jan-04,0,0.0309
1-Jan-04,1,0.02713
1-Jan-04,2,0.02523
1-Jan-04,3,0.02429
1-Jan-04,4,0.02442
1-Jan-04,5,0.02623
1-Jan-04,6,0.02644
1-Jan-04,7,0.0207
```

The price data can be used by algorithms that measure their performance using the electricity-price dependent metrics, e.g., MQNetNorm (Equation 4.8) and SCMQNetNorm (Equation 4.6).

A.5 COMMUNITY-SPECIFIED OPERATING BOUNDARIES

Specific values for the operating limits described in this section are in some of the OLAs (Section 3.1.3). For example, OLA 1 has noise and view policies, but does not restrict the maximum time visible; OLA 5 does not have noise and view policies, but does specify a maximum time visible.

A.5.1 Community-specified noise and view policies

Some communities might specify that harvesters be stowed during certain times (e.g., tourist seasons, sunrises, sunsets, sleeping hours). We call those specifications *noise and view policies.*

A.5.2 Community-specified maximum time visible

Similar to noise and view polices, some communities might specify that harvesters be stowed after the harvesters are visible a certain percentage of a period of time. For example, if a community specifies that a harvester must be stowed at least 40% of each month, then the community is permitting the harvester to be visible up to 60% of the month. In other words, the community is specifying that the harvester's maximum time visible is 60% of each month.

A.5.3 Community-specified maximum deployment speed

A community-specified maximum deployment speed is a speed limit on the rate at which a harvester can move from being fully stowed to being fully deployed. For example, a community might specify that a harvester move from being fully stowed to being fully deployed in no less than 20 minutes, which is the minimum deployment time. Deployment speed is the inverse of deployment time (defined in Section A.3.5).

Communities might specify a speed limit to do the following:

- Share space and reduce risk
  - Pilots of aircraft might appreciate the extra time to vacate the airspace near a retractable harvester
  - A retractable harvester might have pinch points where a maintenance worker might become pinched if a harvester moves too quickly. For example, if a maintenance worker is near a large open hinge and the hinge begins to close quickly, the rapidly closing hinge might clamp the worker.
- Maintain tranquility of a landscape: Quickly deploying retractable harvesters might disturb the visual serenity of a view. They might also make more noise than slowly moving retractable harvesters. (We assume that noise would be governed by each hosting community's noise regulations.)

We are assuming that a community would not specify the opposite (a minimum deployment speed or maximum deployment time) because faster deployment speeds reduce unproductive visibility time.

A.6 RETRACTION-SUITABLE WIND ENERGY HARVESTING TECHNOLOGIES

Although development of retractable harvesters is outside the scope of this work, we include examples of envisioned and actual harvesters in this section.

A.6.1 Retractable land/sea-supported wind energy systems

Retractable land/sea-supported wind energy systems include wind energy systems that are mechanically supported by the land or sea.

A.6.1.1 Wind-power-harvesting fabric (envisioned) A micro-wind turbine has been developed that has a 1.8 mm rotor diameter [18]. It is conceivable that thousands of the micro turbines could be embedded into a fabric. The resulting wind-power-harvesting fabric could be supported by two parallel wires suspended between two vertical poles as shown in Figure 32. Also shown in that figure is a wire that a motor reels onto a spool to pull the fabric from the fabric's storage location. When the algorithm determines it is time to deploy the wind harvesting fabric, the motor pulls the fabric so that the fabric fills the rectangular plane formed by the two parallel supporting wires and the two vertical poles (Figure 33 (not drawn to scale)). Two separate perpendicular planes can be used to help ensure that the wind is not parallel to at least one plane (Figure 34) since wind parallel to a plane causes all the plane's micro-turbines except those at the windward edge of the fabric to be in the wakes[6] of the other turbines.

Fabric configured as a folding fan: Instead of using a pair of vertical poles and wires, the fabric could be formed into a folding fan. The folded/closed fan could be stored in a single vertical pole. An envisioned wind harvester having a folding fan is shown in the deployed and stowed states in Figure 35. The fan is kept facing the wind by a tail fin, which is not shown in the figure. Moveable ribs supporting the folding fan are supported by a retractable tower, which is retracted as needed.

[6]"Wind turbines extract energy from the wind and downstream there is a wake from the wind turbine, where wind speed is reduced" [104].

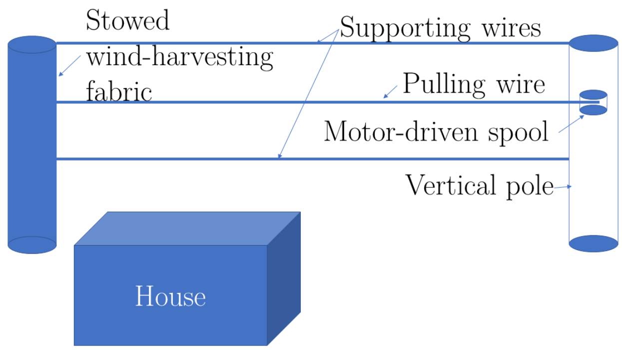

Figure 32: Rolled fabric stowed in a vertical pole (image is not drawn to scale)

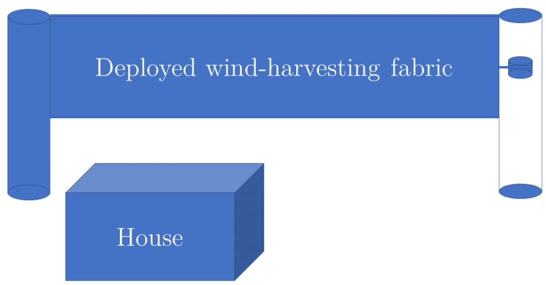

Figure 33: Fabric deployed between supporting wires (image is not drawn to scale)

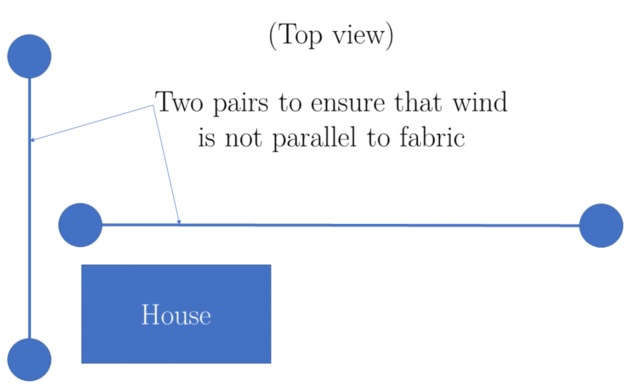

Figure 34: Top view of fabric planes perpendicularly arranged to each other ensuring that wind direction is not parallel to at least one fabric plane (image is not drawn to scale)

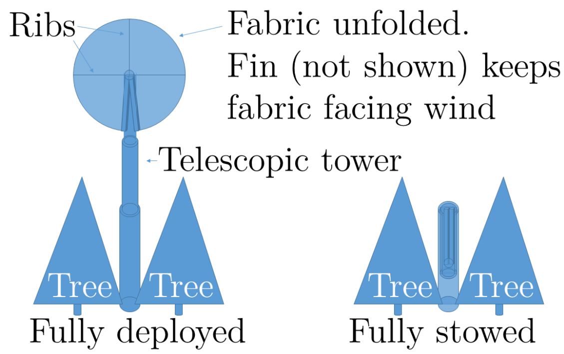

Figure 35: Folding-fan-style wind harvester using fabric shown in deployed and stowed states

A.6.1.2 Harvesters having telescopic supports Two categories follow:

Non-expandable blades (patent granted, expired): For example, a granted patent claims a wind machine having a "telescopic means adapted to permit said airfoil support means to be lowered into a storage mode in which at least some of the airfoils are stored in a zone protected from some of the effects of a windstorm" [75].

Expandable blades (envisioned): Figure 36 shows an envisioned wind turbine having a telescopic tower and inflatable blades. The envisioned inflatable blades are modeled after the inflatable wings with which NASA and others have experimented [95] [8].

Instead of inflatable wind turbine blades, foldable wind turbine blades [74] (Figure 37) could potentially be used to stow and deploy wind turbines. The design shown in Figure 37 could be potentially be modified such that when the blades are folded, the nacelle could be rotated toward the sky such that the folded blades point vertically to avoid interference with trees and to better blend into the scenery or then retract with a telescopic tower (Figure 38).

A.6.1.3 Oscillating-wing windmill (concept published, embodiment envisioned) An oscillating-wing windmill to which Mckinney and DeLaurier [50] refer has only one wing. The wing is arranged horizontally. It flutters (e.g., pitches and plunges as shown in states "c" and "d" of Figure 39 on page 180) to harvest wind energy. The wing seems to have the

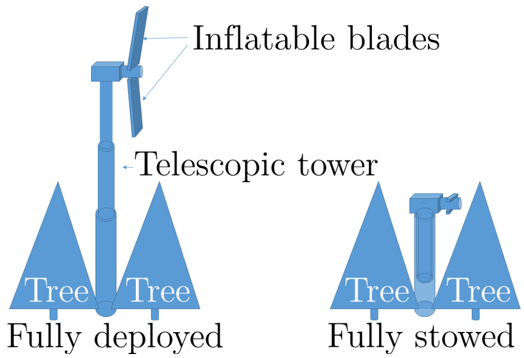

Figure 36: Deployed and stowed turbine having a telescopic tower and inflatable blades

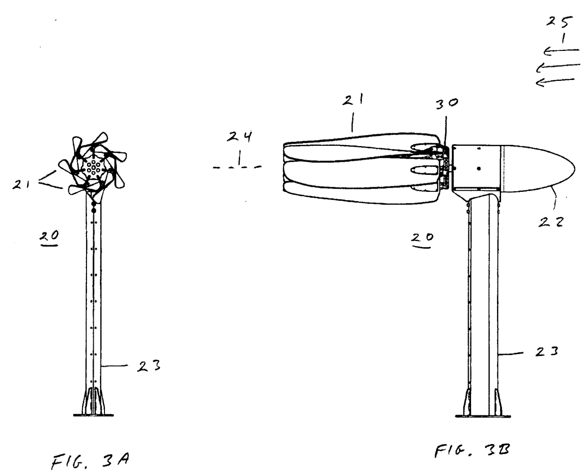

Figure 37: Patent drawing showing blades folded [74]

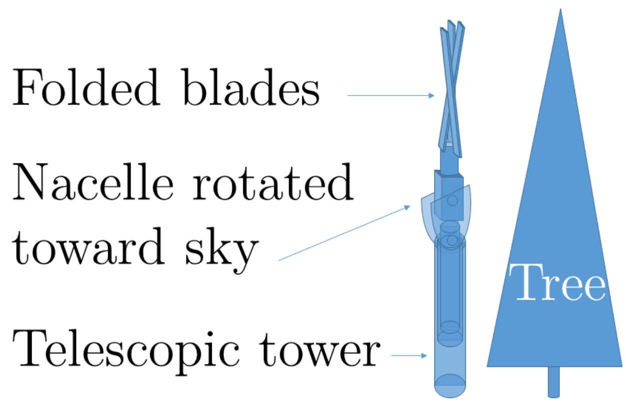

Figure 38: Stowed turbine; folded blades pointing upward

potential to be relatively easily blended into the surrounding architecture, e.g., a roof as shown in state "a" of Figure 39, when not in use.

A.6.2 Airborne wind energy systems (actual)

Airborne wind energy systems use airborne harvesters to convert wind to usable energy and comprises both lighter-than-air and heavier-than-air systems [16] [12].

The category of airborne, lighter-than-air wind energy systems includes harvesters integrated with tethered aerostats (e.g., Altaeros Energies' "autonomous tethered airborne platforms" [4]).

Heavier-than-air wind energy systems include kite systems such as the Makani energy kite [48]. The Makani system uses on-board propellers to lift the kite into the wind and to keep it flying when windspeeds dip.

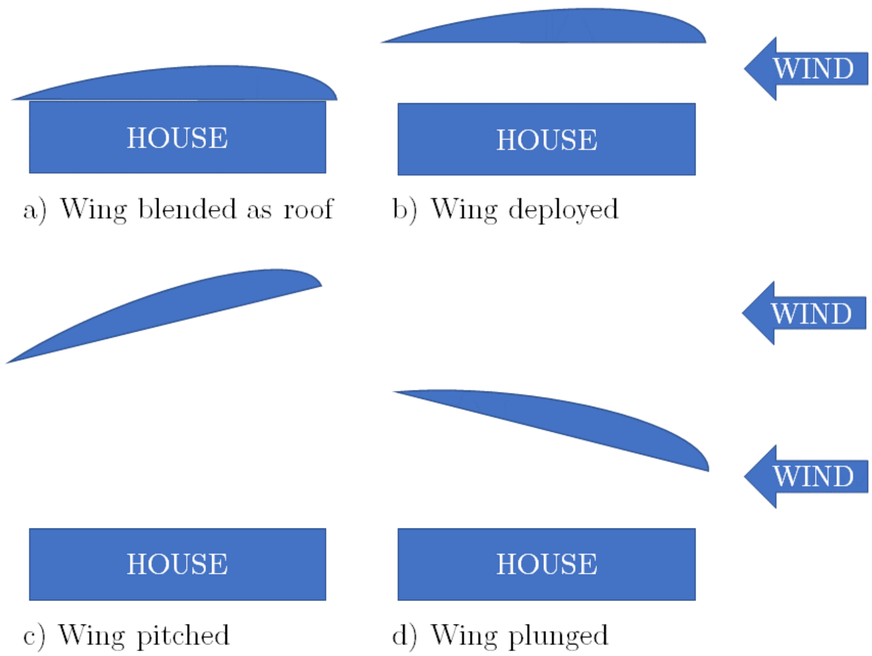

Figure 39: Oscillating-wing windmill shown in four states

A.7 PREPARING THE WINDSPEED DATA

A.7.1 Filtering

We are using files in the Automated Surface Observing System (ASOS) data set DSI 6405 [59]. Some lines in those files contain non-windspeed-data where we expect windspeeds. To ignore those lines, we filtered out each line that

1. Does not have a number in its windspeed field,
2. Has a number or other character that is not a space in the column that immediately follows its windspeed field,
3. Has an identical local timestamp to its previous line,
4. Does not have a number in the least significant place of its wind heading field, or
5. Does not have a space that precedes its windspeed field.

The results of our filtering are shown in Table 16. The table shows that in all cases less than 3% of lines were removed by our filters.

A.7.2 Shifting fields

We found that the windspeed and wind heading fields changed locations between years 2004 and 2014 for most weather stations. These are the field locations we encountered for windspeed and for wind heading respectively:

Speed1: Windspeed in columns 74, 75, and 76.

Speed2: Windspeed in columns 76, 77, and 78.

Heading1: Wind heading in columns 68, 69, and 70.

Heading2: Wind heading in columns 71, 72, and 73.

Heading3: Wind heading in columns 67, 68, and 69.

We extracted the windspeeds from the shifting field locations in ASOS data to create comma separated files that would eventually become this benchmark's wind-data .csv files. The shifting field locations in the ASOS data are shown in Table 15 as a function of timestamp ranges. For example, station KPHX's data has windspeed values in the location we call "Speed2" above for all records having timestamps in the range [PHX2013053007231423, end], which is from 2013-05-30 07:23, 14:23 UTC, to the end of KPHX's 2014 data, inclusive. All of KPHX's records outside that range (i.e., the balance of the records) have windspeed values in the location we call "Speed1" above.

A.7.3 Removing anachronistic records

We omitted anachronistic records (i.e., in the sequence of the ASOS data, records having timestamps earlier than the timestamps of records occurring earlier in the sequence). Anachronistic records were found in only six of the thirty weather stations, KBWI, KEUG, KIAH, KLGA, KORD, and KPIT, and comprised a minuscule percentage (approximately 0.0007%, which equals the approximately 1000 anachronistic records shown in Table 17 divided by the approximately 150,000,000 samples, which comprise the eleven years of minute-

Table 15: Ranges of timestamps where fields are in certain columns in original ASOS data

icao	Spd.1	Speed2	Hdg.1	Heading2	Heading3[a]
KPHX	bal.[b]	[PHX2013053007231423, end[c]]	bal.	[PHX2013053007231423, end]	N/A
KLAX	all	none	bal.	none	[LAX2004010117230123, LAX2006013007391539)
KSAC	all	none	bal.	none	[SAC2004010118370237, SAC2004050500070807)
KSAN	all	none	bal.	none	[SAN2004010118060206, SAN2004031809381738)
KSFO	all	none	all	none	none
KSMX	bal.	[SMX2014082706071407, end]	bal.	[SMX2014082706071407, end]	[SMX2004010100000800, SMX2004030905111311)
KDEN	bal.	[DEN2013121113272027, end]	bal.	[DEN2013121113272027, end]	[DEN2004010100000700, DEN2004082506391339)
KMCO	bal.	[MCO2014022410051505, end]	bal.	[MCO2014022410051505, end]	[MCO2004010100000500, MCO2006092000330533)
KATL	all	none	bal.	none	[ATL2004010122110311, ATL2004021802330733)
KORD	bal.	[ORD2013121213101910, end]	bal.	[ORD2013121213101910, end]	[ORD2004010100350635, ORD2004060706491249)
KCVG	all	none	bal.	none	[CVG2004010119390039, CVG2004010709381438)
KBOS	all	none	all	none	none
KBWI	all	none	bal.	none	[BWI2004010120190119, BWI2004010601170617)
KDTW	all	none	bal.	none	[DTW2004010119490049, DTW2004071405401040)
KMSP	all	none	all	none	none
KMCI	bal.	[MCI2013121208581458, end]	bal.	[MCI2013121208581458, end]	[MCI2004010100000600, MCI2004070707581358)
KSTL	all	none	bal.	none	[STL2004010100000600, STL2005042502190819)
KCLT	all	none	bal.	none	[CLT2004010119360036, CLT2004012201480648)
KLAS	bal.	[LAS2014040909481748, end]	bal.	[LAS2014040909481748, end]	[LAS2004010117140114, LAS2004031002341034)
KLGA	all	none	bal.	none	[LGA2004010100200520, LGA2004030113571857)
KCLE	bal.	[CLE2013011510351535, end]	bal.	[CLE2013011510351535, end]	none
KEUG	bal.	[EUG2013121910281828, end]	bal.	[EUG2013121910281828, end]	[EUG2004010100000800, EUG2004030904571257)
KPHL	bal.	[PHL2014022011001600, end]	bal.	[PHL2014022011001600, end]	[PHL2004010123030403, PHL2004020204580958)
KPIT	bal.	[PIT2014030710371537, end]	bal.	[PIT2014030710371537, end]	[PIT2004010100000500, PIT2005012809041404)
KDFW	bal.	[DFW2013012411311731, end]	bal.	[DFW2013012411311731, end]	none
KIAH	all	none	bal.	none	[IAH2004010100000600, IAH2004031823150515)
KSAT	bal.	[SAT2014022709261526, end]	bal.	[SAT2014022709261526, end]	[SAT2004010120240224, SAT2005082308021402)
KDCA	all	none	all	none	none
KSEA	all	none	bal.	none	[SEA2004010100000800, SEA2006031402341034)

[a]The range could include the last line in each Heading3 range (e.g., LAX2006013007391539), but our filter does not include the last line in each range. We seem to have enough data that the last line in each Heading3 range is not needed for most simulations. Thus, we did not re-filter the data to include the last line in each Heading3 range.

[b]balance of records (i.e., all records having timestamps outside the range(s) given in this row for this field

[c]i.e., end of 2014 data

by-minute data for 30 stations) of the ASOS data after we had filtered it according to the method described immediately above. As shown in Table 17, of six stations having anachronistic records, KIAH had the highest percentage of anachronistic records, which was 0.006%.

A.7.4 Handling identically timestamped records

We discarded all records where timestamps were repeated in the ASOS data. The percentage of runs of identically timestamped samples compared to the number of uniquely timestamped

Table 16: Percentage of lines in ASOS data that the filters listed in Section A.7.1 discarded) ordered descendingly

icao	lines discarded (%)	icao	lines discarded (%)	icao	lines discarded (%)
KORD	2.45%	KIAH	0.37%	KMCI	0.23%
KTPA	1.17%	KMCO	0.36%	KSEA	0.22%
KSMX	0.93%	KLAX	0.33%	KSFO	0.2%
KDEN	0.69%	KPHL	0.33%	KBWI	0.17%
KPIT	0.58%	KSAC	0.31%	KCLT	0.14%
KDCA	0.52%	KPHX	0.3%	KCVG	0.13%
KEUG	0.48%	KSAT	0.28%	KBOS	0.11%
KCLE	0.43%	KATL	0.26%	KSAN	0.11%
KSTL	0.41%	KDFW	0.26%	KLAS	0.08%
KLGA	0.4%	KDTW	0.23%	KMSP	0.07%
continues above right		continues above right			

Table 17: Statistics on anachronistic records

station	No. of anachronistic records	Max. run size	No. of samples	Anachronistic records as percentage of no. of samples
KBWI	190	190	5419390	0.004%
KEUG	199	110	5205945	0.004%
KIAH	337	337	5198385	0.006%
KLGA	21	21	5185102	0.000%
KORD	246	218	5147727	0.005%
KPIT	23	23	5306440	0.000%

samples is less than one tenth of a percent (i.e., 0.09%) and, on a station-by-station basis, a maximum of less than three tenths of a percent (0.29% for KORD).

Table 18: Statistics about windspeeds between uniquely timestamped records and between uniquely timestamped records and the first records of each run of identically timestamped records

station	see legend*	mean	std. dev.	kur-tosis	skew-ness	n	mean(B)/ mean(A)
*Legend: A = statistics about normalized difference of windspeeds** in samples having unique timestamps. B = statistics about normalized difference of windspeeds** between first sample of run of samples sharing timestamps and the most recent sample having a unique timestamp B/A = quotient of group B's mean divided group's A mean **The normalized difference of windspeeds $\delta_{\text{knotsPerMinute}}$ equals the difference of wind-speeds δ_{knots} over the difference between timestamps δ_{minute} (to account for timestamps that differ by more than one minute); i.e., $\delta_{\text{knotsPerMinute}} = \delta_{\text{knots}}/\delta_{\text{minute}}$.							
KATL	A	1.37E-04	0.87	3.95	0.17	5436757	
	B	7.27E-01	3.66	1.45	-0.39	11	
	B/A	5.29E+03					5.29E+03
KBOS	A	2.92E-05	0.90	2.45	0.09	5.46E+06	
	B	5.00E-01	3.87	4.57	1.62	10	
	B/A	1.71E+04					1.71E+04
KBWI	A	-2.55E-05	0.86	4.57	0.17	5394105	
	B	2.89E-01	3.49	9.10	2.40	38	
	B/A	-1.14E+04					-1.14E+04
KCLE	A	-1.16E-05	0.88	3.62	0.15	5393670	
	B	3.04E-01	1.58	8.56	1.94	121	

Table 18: (continued)

	B/A	-2.62E+04					-2.62E+04
KCLT	A	-4.16E-06	0.79	2.64	0.14	5487847	
	B	-8.33E-02	1.38	-1.00	0.18	12	
	B/A	2.00E+04					2.00E+04
KCVG	A	4.06E-06	0.84	14.43	0.13	5347487	
	B	2.50E-01	1.99	18.96	3.50	43	
	B/A	6.16E+04					6.16E+04
KDCA	A	3.87E-05	0.87	9.86	0.15	4736355	
	B	1.18E-01	4.38	267.99	14.65	397	
	B/A	3.06E+03					3.06E+03
KDEN	A	-2.12E-04	0.88	10.86	0.24	5338217	
	B	-2.25E-02	1.75	16.98	-2.74	89	
	B/A	1.06E+02					1.06E+02
KDFW	A	-3.74E-04	0.90	2.28	0.17	5300401	
	B	-1.52E-01	2.71	17.76	-2.98	79	
	B/A	4.06E+02					4.06E+02
KDTW	A	9.32E-05	0.85	2.90	0.15	5424015	
	B	1.54E+00	3.13	4.01	1.92	13	
	B/A	1.65E+04					1.65E+04
KEUG	A	-4.56E-05	0.70	12.06	0.19	5195953	
	B	-5.00E-02	1.39	8.43	-2.36	20	
	B/A	1.10E+03					1.10E+03
KIAH	A	-1.06E-05	0.83	2.49	0.20	5193286	
	B	6.09E-02	1.10	9.69	-0.56	115	
	B/A	-5.76E+03					-5.76E+03
KLAS	A	-2.33E-04	0.90	2.88	0.16	5433173	
	B	-2.50E-01	0.71	-0.23	0.40	8	

Table 18: (continued)

	B/A	1.07E+03					1.07E+03
KLAX	A	4.89E-05	0.71	6.46	0.08	5332533	
	B	-3.41E-01	3.08	35.96	-5.82	41	
	B/A	-6.99E+03					-6.99E+03
KLGA	A	1.28E-04	0.95	1.91	0.11	5174896	
	B	-5.56E-02	0.86	0.21	0.13	30	
	B/A	-4.33E+02					-4.33E+02
KMCI	A	-2.92E-05	0.85	3.15	0.15	5494553	
	B	-4.73E-01	1.85	12.01	-2.97	55	
	B/A	1.62E+04					1.62E+04
KMCO	A	-1.11E-04	0.80	216.34	1.62	5383438	
	B	1.39E-01	1.55	11.66	1.99	36	
	B/A	-1.25E+03					-1.25E+03
KMSP	A	-8.57E-05	0.88	3.84	0.11	5497700	
	B	-1.80E+00	4.29	1.21	-1.33	10	
	B/A	2.10E+04					2.10E+04
KORD	A	1.99E-05	0.91	2.16	0.13	5117591	
	B	7.93E-02	1.47	12.35	-2.16	74	
	B/A	3.98E+03					3.98E+03
KPHL	A	7.14E-05	0.86	2.48	0.16	5223332	
	B	-5.56E-01	1.81	5.46	-2.16	9	
	B/A	-7.78E+03					-7.78E+03
KPHX	A	-9.70E-04	0.84	4.68	0.22	5275447	
	B	2.73E-01	1.79	2.93	-1.13	11	
	B/A	-2.81E+02					-2.81E+02
KPIT	A	-1.50E-04	0.84	3.22	0.19	5298517	
	B	-1.79E-01	1.95	3.16	-0.65	13	

Table 18: (continued)

	B/A	1.19E+03					1.19E+03
KSAC	A	1.63E-04	0.81	40.27	0.07	5402902	
	B	-1.10E+00	2.85	3.93	-2.05	371	
	B/A	-6.72E+03					-6.72E+03
KSAN	A	-4.12E-05	0.69	10.00	-0.04	5389450	
	B	-8.16E-02	0.81	0.30	-0.09	98	
	B/A	1.98E+03					1.98E+03
KSAT	A	-2.34E-04	0.93	3.13	0.20	5444969	
	B	1.61E-02	0.91	1.12	0.05	620	
	B/A	-6.90E+01					-6.90E+01
KSEA	A	9.68E-05	0.75	9.46	0.18	5375319	
	B	4.88E-01	3.47	16.76	3.36	41	
	B/A	5.04E+03					5.04E+03
KSFO	A	4.77E-05	0.79	2.40	0.09	5344491	
	B	2.52E-02	1.25	12.16	0.51	278	
	B/A	5.28E+02					5.28E+02
KSMX	A	5.15E-04	0.75	26.05	0.09	5371896	
	B	1.51E-01	2.13	33.83	4.56	93	
	B/A	2.92E+02					2.92E+02
KSTL	A	1.96E-05	0.87	2.54	0.16	5524684	
	B	1.17E-01	1.54	27.07	3.45	94	
	B/A	5.96E+03					5.96E+03
KTPA	A	2.76E-05	0.79	7.61	0.14	5148249	
	B	0.00E+00	0.80	0.30	-0.22	57	
	B/A	0.00E+00					0.00E+00
						Average:	3.85E+03

A.7.5 Processed outliers

After the data cleaning described above, we identified outliers using two methods. The first method is to use the *adjacent windspeed difference per minute* defined as the quotient of the difference in windspeeds of two consecutive records divided by the number of minutes separating those two windspeeds:

$$\delta_{tb,ta} = \frac{s_{tb}^{speed} - s_{ta}^{speed}}{tb - ta}$$

where

- ta and tb are the timestamps of two consecutive samples,
- s_{ta}^{speed} is the windspeed in knots of sample s having timestamp ta,
- $tb - ta$ is the difference of timestamps tb and ta in minutes, and
- $tb > ta$ (i.e., timestamp tb is later than ta) so that $\delta_{tb,ta} < 0$ only when $s_{tb}^{speed} < s_{ta}^{speed}$.

For each weather station ws in our list of 30 stations, we examined the distribution of non-zero $\delta_{tb,ta}$'s by first finding the standard deviation of $\delta_{tb,ta}$ (to which we refer as $\sigma_{\delta,ws}$) listed in Table 19. Second, we created a frequency distribution table of the product $(\delta)(\sigma_{\delta,ws})$ for each weather station ws, as listed in Table 20. For each range of the 30 frequency distributions in Table 20, we summed the count to produce the combined frequency distribution table shown as Table 21, which is plotted as a histogram in Figure 40 on page 196.

Figure 40 shows that an extreme-value test [3] may be appropriately applied to find outlying windspeeds. The figure shows that approximately 99% of the non-negative δ's are less than 3 standard deviations from 0. However, we deemed extreme-values of δ to be those values that are greater than 20 times $\sigma_{\delta,ws}$. Thus, let's refer to 20 times $\sigma_{\delta,ws}$. as *thresh*, hereafter.

When a $\delta_{tb,ta}$ exceeded *thresh*, we discarded the sample s_{tb}, which is the sample having timestamp tb, and all samples s_t following s_{tb} until $delta_{t,ta} \leq thresh$. All samples that passed that filter were subject to another outlier filter: the strong-gale filter.

The strong-gale filter uses the fact that winds faster than strong-gales (i.e., winds above 47 knots) rarely occur on land [62]. We compared any winds faster than strong gales to Weather Underground's archive, which includes ASOS data in addition to over 250,000 personal weather stations and over 26,000 Meteorological-Assimilation-Data-Ingest-System stations [102]. If our above-strong-gale windspeed for a certain day d exceeded the maximum windspeed for day d archived at Weather Underground, we discarded our windspeed; otherwise, we kept the windspeed in all cases except for one case. Weather Underground mistakenly records 1,000 mph as the maximum windspeed for 2010-10-19 at station KMCO. Since 1,000 mph is obviously false, we turned to another another source [91], which showed less than 5 mph for that date.

Table 19: Descriptive statistics of δ for each station

station	mean	standard deviation	kurtosis	skewness
KATL	0.00	0.87	3.95	0.17
KBOS	0.00	0.90	2.45	0.09
KBWI	0.00	0.86	4.57	0.17
KCLE	0.00	0.88	3.62	0.15
KCLT	0.00	0.79	2.64	0.14
KCVG	0.00	0.84	14.43	0.13
KDCA	0.00	0.87	9.86	0.15
KDEN	0.00	0.88	10.86	0.24
KDFW	0.00	0.90	2.28	0.17
KDTW	0.00	0.85	2.90	0.15
KEUG	0.00	0.70	12.06	0.19
KIAH	0.00	0.83	2.49	0.20
KLAS	0.00	0.90	2.88	0.16
KLAX	0.00	0.71	6.46	0.08
KLGA	0.00	0.95	1.91	0.11
KMCI	0.00	0.85	3.15	0.15
KMCO	0.00	0.80	216.34	1.62
KMSP	0.00	0.88	3.84	0.11
KORD	0.00	0.91	2.16	0.13
KPHL	0.00	0.86	2.48	0.16
KPHX	0.00	0.84	4.68	0.22
KPIT	0.00	0.84	3.22	0.19
KSAC	0.00	0.81	40.27	0.07
KSAN	0.00	0.69	10.06	-0.04
KSAT	0.00	0.93	3.13	0.20
KSEA	0.00	0.75	9.46	0.18
KSFO	0.00	0.79	2.40	0.09
KSMX	0.00	0.75	26.06	0.09
KSTL	0.00	0.87	2.54	0.16
KTPA	0.00	0.79	7.61	0.14

Table 20: For each weather station ws, the frequency distributions of $(\delta \geq 0)(\sigma_{\delta,ws})$

station	[0,1)	[1,2)	[2,3)	[3,4)	[4,5)	[5,6)	[6,7)	[7,8)	[8,9)	[9,10)	[10,11)	[11,12)	[12,13)	[13,14)	[14,15)	[15,16)	[16,17)	[17,18)	[18,19)	[19,20)
KATL	3997537	169853	33031	5926	1079	277	148	32	8	10	4	1	4	3	0	2	0	1	2	0
KBOS	3937380	180312	34554	6558	1166	245	51	11	11	5	1	0	0	1						
KBWI	4024500	164549	34700	7099	1495	351	118	6	6	2	2	0	3	0	0	0	0	0	0	0
KCLE	3947912	172187	32377	6105	1217	356	114	77	19	4	3	5	1	1	1	2	2	0	0	2
KCLT	4196470	137101	22047	3947	90	31	21	7	3	1										
KCVG	4009064	153373	29400	5448	1102	368	57	25	16	6	9	2	1	0	0	0	0	0	0	0
KDCA	3525021	134720	29315	6721	1582	404	156	29	16	7	6	4	4	0	0	2	0	1	0	0
KDEN	3952564	154902	36839	8972	2315	688	222	157	30	13	4	6	4	4	4	0	3	0	1	1
KDFW	3817708	193430	38617	6833	1105	231	86	33	26	14	4	0	2	1	1					
KDTW	4004092	162864	29205	5052	953	334	49	23	26	14	8	6	1	0	0	1				
KEUG	4169716	88396	15575	315	99	8	6	8	2	1	2	0	1	1	0	0	1	0	0	0
KIAH	3870370	153100	27766	4548	790	247	38	21	9	4	3	0	1							
KLAS	3981235	175039	41368	9271	1851	380	81	20	22	5	1									
KLAX	4171954	78622	7055	157	44	2	2	0	0	0	0	0	1	1	0	0	0	0	0	1
KLGA	3579399	208577	40568	7598	1327	307	77	24	13	6	4	2	1	2	0	1	0	0	0	1
KMCI	4102080	167817	31459	5445	1012	302	28	21	12	13	10	2	1	2	0	0	0	1	0	0
KMCO	4136849	134191	23207	4564	244	102	45	33	6	6	5	6	1	1	4	0	0	0	0	0
KMSP	3985184	181355	33295	5818	995	191	63	37	8	4	4	1	2	0	1	0	0	0	0	0
KORD	3652652	185014	34176	6043	1113	245	81	38	13	17	3	1	1	2						
KPHL	3836098	154173	28853	5480	1050	272	97	18	3	9	2	1	4							
KPHX	3977796	143135	33588	7720	1620	438	40	20	9	10	5	2	0	0	1	1				
KPIT	3979226	151656	30269	6090	1231	451	49	36	13	4	4	2	2							
KSAC	4098961	144062	21219	2784	527	38	37	19	29	20	9	6	15	4	4	3	2	5	8	1
KSAN	4273561	76449	6127	77	23	5	0	1	5	1	0	1	0	2	0	1	0	0	1	
KSAT	3842823	210265	42621	7844	1398	263	84	32	27	13	9	3	2	0	0	0	0	0	0	0
KSEA	4165960	105754	13541	2265	112	27	31	8	3	5	1	0	0	0	0	0	0	0	0	0
KSFO	4080242	126901	18026	3288	137	38	18	16	3	2										
KSMX	4226634	102326	13901	2810	310	203	212	47	50	61	18	10	12	3	1	14	3	2	5	0
KSTL	4036294	176657	31860	5542	1008	241	161	27	21	15	7	3	1	2						
KTPA	3889128	116711	14283	2349	139	47	37	66	38	21	11	11	2	2	2	3	2	1	0	1

station	[20,21)	[21,22)	[22,23)	[23,24)	[24,25)	[25,26)	[26,27)	[27,28)	[28,29)	[29,30)	[30,31)	[31,32)	[32,33)	[33,34)	[34,35)	[35,36)	[36,37)	[37,38)	[38,39)	[39,40)
KATL	0	0	0	0	1															
KBOS																				
KBWI	1	1	0	0	1	1														
KCLE	0	0	1																	
KCLT																				
KCVG	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
KDCA	2	0	2	0	0	1	0	0	0	1	1	0	1							
KDEN	0	1	1	0	2	0	0	0	0	1	1	0	0	1	0	0	1			
KDFW																				
KDTW																				
KEUG	0	0	0	0	0	1	0	0	0	0	0	1								
KIAH																				
KLAS																				
KLAX	0	1	1																	
KLGA																				
KMCI	0	1																		

Table 20: (continued)

KMCO	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
KMSP	1																			
KORD																				
KPHL																				
KPHX																				
KPIT																				
KSAC	2	3	0	0	2	3	0	0	1	1	3	1	0	2	2	0	0	1		
KSAN																				
KSAT	0	1	0	1	1	0	0	1												
KSEA	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1				
KSFO																				
KSMX	2	2	2	1	1															
KSTL																				
KTPA	0	1	0	0	0	0	0	0	0	0	0	0	0	1						

station	[40,41)	[41,42)	[42,43)	[43,44)	[44,45)	[45,46)	[46,47)	[47,48)	[48,49)	[49,50)	[50,51)	[51,52)	[52,53)	[53,54)	[54,55)	[55,56)	[56,57)	[57,58)	[58,59)	[59,60)
KATL																				
KBOS																				
KBWI																				
KCLE																				
KCLT																				
KCVG																				
KDCA																				
KDEN																				
KDFW																				
KDTW																				
KEUG																				
KIAH																				
KLAS																				
KLAX																				
KLGA																				
KMCI																				
KMCO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KMSP																				
KORD																				
KPHL																				
KPHX																				
KPIT																				
KSAC																				
KSAN																				
KSAT																				
KSEA																				
KSFO																				
KSMX																				
KSTL																				
KTPA																				

station	[60,61)	[61,62)	[62,63)	[63,64)	[64,65)	[65,66)	[66,67)	[67,68)	[68,69)	[69,70)	[70,71)	[71,72)	[72,73)	[73,74)	[74,75)	[75,76)	[76,77)	[77,78)	[78,79)	[79,80)
KATL																				

Table 20: (continued)

KBOS																				
KBWI																				
KCLE																				
KCLT																				
KCVG																				
KDCA																				
KDEN																				
KDFW																				
KDTW																				
KEUG																				
KIAH																				
KLAS																				
KLAX																				
KLGA																				
KMCI																				
KMCO	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0
KMSP																				
KORD																				
KPHL																				
KPHX																				
KPIT																				
KSAC																				
KSAN																				
KSAT																				
KSEA																				
KSFO																				
KSMX																				
KSTL																				
KTPA																				

station	[80,81)	[81,82)	[82,83)	[83,84)	[84,85)	[85,86)	[86,87)
KATL							
KBOS							
KBWI							
KCLE							
KCLT							
KCVG							
KDCA							
KDEN							
KDFW							
KDTW							
KEUG							
KIAH							
KLAS							
KLAX							
KLGA							
KMCI							
KMCO	0	0	0	0	0	0	1
KMSP							

Table 20: (continued)

KORD								
KPHL								
KPHX								
KPIT								
KSAC								
KSAN								
KSAT								
KSEA								
KSFO								
KSMX								
KSTL								
KTPA								

A.7.6 Interpolating

Gaps in the training and testing files were linearly interpolated. A gap is one or more missing minute-by-minute records. For example, the following sequence of windspeed records (where each record has a windspeed field preceded by a timestamp field having the format `YYYYMMDDHHmmhhmm` where `YYYY` is the year, `MM` is the month, `DD` is the day, `HH` is the local standard-time hour, `mm` is the minute, and `hh` is the UTC hour) has two gaps (a one-record or two-minute gap between UTC 05:02 and UTC 05:04 and a three-record or four-minute gap between UTC 05:05 and UTC 05:09):

```
2004010100010501,5
2004010100020502,5
2004010100040504,5
2004010100050505,5
2004010100090509,6
```

After interpolation, the sequence is

```
timestamp,wind_knots,actual vs. interpolated
2004010100010501,5,a
2004010100020502,5,a
2004010100030503,5,i
2004010100040504,5,a
2004010100050505,5,a
2004010100060506,5,i
2004010100070507,5,i
2004010100080508,5,i
2004010100090509,6,a
```

The interpolation routine (a portion of which is shown immediately below in Java code) truncates fractional values instead of rounding to the nearest integer:

```java
double dWindSpeedKnotsInterval = (iWindSpeedKnotsNext - iWindSpeedKnots)/ (double) diffInMinutes;
double dWindSpeedInterpolated = (double) iWindSpeedKnots;
for (int j=1; j<diffInMinutes; j++) {
  calLocal.add(Calendar.MINUTE, 1);
  calUtc.add(Calendar.MINUTE, 1);

  dWindSpeedInterpolated += dWindSpeedKnotsInterval;

  // build field
  String formattedLocalTimestamp = sdf.format(calLocal.getTime());
```

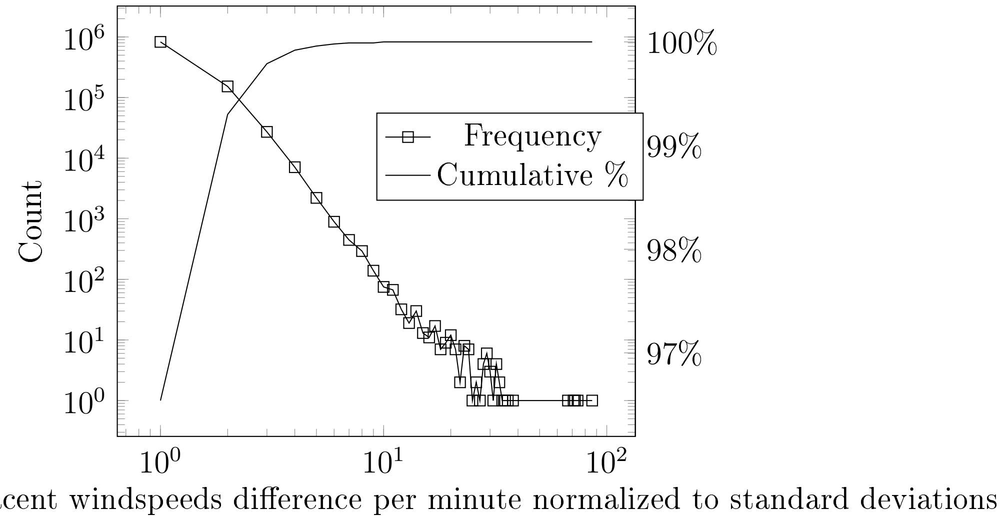

Figure 40: Histogram of Table 21 sans frequencies of 0: Frequency distribution table summarizing the 30 stations' frequency distributions of $\sigma_{\delta,ws}$ for all $\delta > 0$

```
    String formattedUtcHourAndMinutes = sdfUtc.format(calUtc.getTime
       ());
    String interpolatedLine = formattedLocalTimestamp +
      formattedUtcHourAndMinutes +
      "," + (int) dWindSpeedInterpolated;
      writeLine(interpolatedLine + ",i\n", writer);
}
```

A.8 AVERAGE WINDSPEEDS FOR EACH STATION

The average windspeed in knots is given in ascending order in Table 22 for each station's 11 years of minute-by-minute windspeed data in the `wind_knots` fields of the station's training and testing benchmark files. As stated elsewhere, to derive that windspeed data, we cleaned and interpolated ASOS data.

Table 21: Frequency distribution table summarizing the 30 stations' frequency distributions of $\sigma_{\delta,ws}$ for all $\delta \geq 0$

$\sigma_{\delta,ws} \forall \delta \geq 0$	Frequency	$\sigma_{\delta,ws} \forall \delta \geq 0$	Frequency	$\sigma_{\delta,ws} \forall \delta \geq 0$	Frequency
[0,1)	4503491	[29,30)	6	[58,59)	0
[1,2)	828842	[30,31)	3	[59,60)	0
[2,3)	152669	[31,32)	1	[60,61)	0
[3,4)	27134	[32,33)	4	[61,62)	0
[4,5)	7092	[33,34)	2	[62,63)	0
[5,6)	2209	[34,35)	1	[63,64)	0
[6,7)	892	[35,36)	1	[64,65)	0
[7,8)	447	[36,37)	1	[65,66)	0
[8,9)	293	[37,38)	0	[66,67)	0
[9,10)	139	[38,39)	1	[67,68)	1
[10,11)	75	[39,40)	0	[68,69)	0
[11,12)	67	[40,41)	0	[69,70)	0
[12,13)	32	[41,42)	0	[70,71)	0
[13,14)	19	[42,43)	0	[71,72)	1
[14,15)	30	[43,44)	0	[72,73)	1
[15,16)	13	[44,45)	0	[73,74)	0
[16,17)	11	[45,46)	0	[74,75)	1
[17,18)	17	[46,47)	0	[75,76)	0
[18,19)	7	[47,48)	0	[76,77)	0
[19,20)	9	[48,49)	0	[77,78)	0
[20,21)	12	[49,50)	0	[78,79)	0
[21,22)	7	[50,51)	0	[79,80)	0
[22,23)	2	[51,52)	0	[80,81)	0
[23,24)	8	[52,53)	0	[81,82)	0
[24,25)	7	[53,54)	0	[82,83)	0
[25,26)	1	[54,55)	0	[83,84)	0
[26,27)	2	[55,56)	0	[84,85)	0
[27,28)	1	[56,57)	0	[85,86)	0
[28,29)	4	[57,58)	0	[86,87)	1
cont. above right		cont. above right		[87,∞]	0

A.9 SURVEY DATA

Please see Tables 23 on the next page, 24 on page 205, and 25 on page 206.

Table 22: For each station, the average windspeed in knots over the 11 years of the benchmark's minute-by-minute windspeeds (after cleaning and interpolating ASOS data)

station	knots	station	knots
KSAN	5.12	KLAS	7.02
KSAC	5.20	KDCA	7.27
KCLT	5.35	KDTW	7.49
KPHX	5.64	KSTL	7.52
KBWI	5.68	KSAT	7.52
KTPA	5.75	KMSP	7.82
KEUG	5.80	KPHL	7.90
KSMX	5.81	KORD	8.15
KLAX	6.29	KCLE	8.21
KPIT	6.31	KDEN	8.67
KMCO	6.52	KMCI	8.83
KSEA	6.56	KSFO	8.88
KIAH	6.65	KLGA	9.23
KCVG	6.98	KDFW	9.26
KATL	6.98	KBOS	9.40

Table 23: Interview and respondent information

No.	Survata Interview ID	Date (UTC)	Ctry.	St.	Metro Area	Gender	Age	O. S.	Browser
1	b5746a41-4017-4ef4-9ddc-ffebb7ba78dd	2017-11-15 20:08:52.0	US	OH	CINCINNATI	f	35 to 44	windows	ie
2	b7e45552-335c-48fe-96b6-88d7edb8201b	2017-11-15 20:10:26.0	US	PA		m	55 to 64	windows	ie
3	9d353b4b-6c9e-47ad-88f9-6b54e6f36631	2017-11-15 20:10:07.0	US	NY	NEW YORK	m	25 to 34	windows	chrome
4	cd1d142e-950a-49e7-baa1-6f208997328b	2017-11-15 20:11:24.0	US	NV	SALT LAKE CITY	f	18 to 24	windows	chrome
5	de6f0fc9-2bfe-48a1-ba69-c88df30d93fd	2017-11-15 20:11:57.0	US	FL	ORLANDO-DAYTONA BCH-MELBRN	m	18 to 24	android	none
6	8802fa41-00e9-45e8-96c9-ffcc6e29c508	2017-11-15 20:11:40.0	US	OR	PORTLAND, OR	m	25 to 34	windows	chrome
7	f8974fbc-a2c7-427b-bc0d-8625e662e698	2017-11-15 20:12:50.0	US	MD	BALTIMORE	f	25 to 34	windows	chrome
8	fdba5fa8-d07b-442b-8fe7-84e65ea7be6c	2017-11-15 20:13:29.0	US	PA	HARRISBURG-LNCSTR-LEB-YORK	m	35 to 44	windows	ie
9	d7c26bb8-3244-4e99-9bf6-594f548c9d90	2017-11-15 20:14:07.0	US	NV	LAS VEGAS	f	55 to 64	windows	other
10	1609a0be-ca14-4e4e-945a-98627f333602	2017-11-15 20:14:42.0	US	MO	KANSAS CITY	f	25 to 34	android	none
11	082f8fd6-c0d8-467a-b0be-909afdba9a8c	2017-11-15 20:16:33.0	US	OH	CINCINNATI	f	45 to 54	windows	chrome
12	a857d98f-6a00-4286-a358-aa8e208c1ba7	2017-11-15 20:19:50.0	US	VA	ROANOKE-LYNCHBURG	f	45 to 54	windows	chrome
13	a554cfcb-6d0f-4d63-a9aa-fcd453fc64ee	2017-11-15 20:19:18.0	US	AZ	PHOENIX (PRESCOTT)	m	25 to 34	macos	chrome
14	dca7b497-4a47-4a02-9de6-72013f3d2702	2017-11-15 20:17:48.0	US	CA	SANTABARBRA-SANMAR-SANLUOB	f	18 to 24	other	chrome
15	4270177a-49ac-47b8-8cd3-3251c7948a07	2017-11-15 20:18:25.0	US	MO	ST. LOUIS	f	55 to 64	other	chrome
16	fddb292d-4a35-458a-80f1-2b7ca6e6d8ff	2017-11-15 20:19:15.0	US	CT	HARTFORD & NEW HAVEN	m	18 to 24	android	chrome
17	7c77fe7a-4fd6-473e-8b59-0090d6d4005e	2017-11-15 20:20:41.0	US	NY	UTICA	f	65 and over	windows	chrome
18	a3304740-918d-4cb7-a45b-372bce1bbf42	2017-11-15 20:20:41.0	US	IN	INDIANAPOLIS	f	18 to 24	windows	chrome
19	b37c7a75-f507-4966-b958-18ba896830e5	2017-11-15 20:21:05.0	US	VT	BURLINGTON-PLATTSBURGH	m	18 to 24	windows	chrome
20	3c24fda4-757a-423b-ae72-4e1b700463df	2017-11-15 20:21:22.0	US	UT	SALT LAKE CITY	f	45 to 54	windows	firefox
21	a2076bc5-87b9-4c5f-85af-bbfa3ef338b0	2017-11-15 20:20:46.0	US	NV	LAS VEGAS	m	18 to 24	macos	chrome
22	fe88f393-f992-4cbc-b081-4fb8f1aa63f2	2017-11-15 20:23:21.0	US	MO	ST. LOUIS	m	25 to 34	windows	chrome
23	b625d466-78da-4852-9239-7c72d70a9f0b	2017-11-15 20:22:38.0	US	IN	LAFAYETTE, IN	f	45 to 54	windows	other
24	a7a52944-4b19-486f-a052-3060230066ef	2017-11-15 20:23:19.0	US	WA	SEATTLE-TACOMA	m	25 to 34	android	chrome
25	26dbafc7-2878-4ebc-a67f-22d0d7486188	2017-11-15 20:22:24.0	US	NY	NEW YORK	m	25 to 34	windows	chrome
26	89d44838-06ff-4b9c-9b2c-d602f8104488	2017-11-15 20:24:44.0	US	LA	BATON ROUGE	f	45 to 54	windows	other
27	601a12d5-0b2e-48b4-844e-0eadca5071f4	2017-11-15 20:23:26.0	US	NY	ROCHESTER, NY	f	25 to 34	other	chrome
28	0836c184-82fa-4913-b3ef-d8dacacf2dc8	2017-11-15 20:26:26.0	US	FL	ORLANDO-DAYTONA BCH-MELBRN	f	65 and over	windows	chrome
29	34631f62-5f2e-45e7-ac12-425bbfffb5aa	2017-11-15 20:25:27.0	US	NY	NEW YORK	f	25 to 34	windows	firefox
30	7ea67efe-39eb-440b-b40f-55fd1124711e	2017-11-15 20:25:44.0	US	SC	GREENVLL-SPART-ASHEVLL-AND	f	55 to 64	windows	other
31	07176b31-40a0-4419-863b-b43056786648	2017-11-15 20:25:35.0	US	PA		m	25 to 34	windows	chrome
32	b33d414e-7758-4258-bb4b-0c188af8e150	2017-11-15 20:25:54.0	US	CA	CHICO-REDDING	m	25 to 34	windows	chrome
33	d56766cc-15dc-4b08-8fcd-50ea3d47f5ef	2017-11-15 20:28:13.0	US	TX	WACO-TEMPLE-BRYAN	m	18 to 24	macos	chrome
34	b43265d1-3e08-49d4-9f14-2f1ba7e22424	2017-11-15 20:27:15.0	US	FL	ORLANDO-DAYTONA BCH-MELBRN	f	25 to 34	android	chrome
35	60092ef6-af86-4743-b92c-5f6968f8cf81	2017-11-15 20:28:45.0	US	WV	CHARLESTON-HUNTINGTON	f	55 to 64	windows	chrome
36	c437c1e6-433d-4fa4-87eb-542b01eab6b8	2017-11-15 20:29:25.0	US	FL	ORLANDO-DAYTONA BCH-MELBRN	m	25 to 34	windows	chrome
37	6cd62600-47c2-4639-a036-c0ba1e1449fa	2017-11-15 20:30:00.0	US	NY	BINGHAMTON	f	35 to 44	windows	chrome
38	3482ea61-0c30-493d-9046-833d0d937658	2017-11-15 20:29:22.0	US	MN	MINNEAPOLIS-ST. PAUL	m	25 to 34	android	none
39	4b44b3ee-ba21-47c7-9c4e-a1aff217f934	2017-11-15 20:30:15.0	US	OH	CLEVELAND-AKRON (CANTON)	f	25 to 34	windows	chrome
40	10e8a8a4-e59e-477e-add2-2020118b6852	2017-11-15 20:29:52.0	US	MI	DETROIT	f	25 to 34	android	chrome
41	7c9ea4cd-d8cd-4f69-a36e-ebdd9cc121a0	2017-11-15 20:31:32.0	US	TX		f	18 to 24	android	none
42	d53b40ac-ae10-4100-8e45-b5486b1f3146	2017-11-15 20:33:39.0	US	FL	TAMPA-ST. PETE (SARASOTA)	f	55 to 64	windows	chrome
43	bbf0c0f0-0464-4c1c-8971-c5c7e92ada32	2017-11-15 20:33:32.0	US	PA	PHILADELPHIA	m	25 to 34	macos	chrome
44	decec206-744b-4323-ace0-3c4b1a6f00ea	2017-11-15 20:36:44.0	US	MI	TRAVERSE CITY-CADILLAC	m	35 to 44	windows	firefox
45	1c22b08d-93fe-4b67-a3c3-abce20e8f7e7	2017-11-15 20:33:57.0	US	IN	SOUTH BEND-ELKHART	m	18 to 24	macos	chrome
46	cce91ce2-3708-4a24-bc3f-64ffa82da968	2017-11-15 20:33:41.0	US	OH	TOLEDO	m	18 to 24	windows	chrome
47	d029393a-5a4f-434c-9e4a-2c3612da1bea	2017-11-15 20:34:26.0	US	OH	CINCINNATI	m	35 to 44	windows	chrome
48	580b3ee3-cd74-4cb5-911f-e5bf7c4a3706	2017-11-15 20:35:20.0	US	NC	CHARLOTTE	m	18 to 24	macos	safari
49	419282e5-8192-43b6-8a54-704228b5f88a	2017-11-15 20:35:51.0	US	VA	HARRISONBURG	f	18 to 24	windows	chrome

Continued on next page

Table 23: (continued)

50	4d5101c1-aa81-4279-af0a-5e7845b43078	2017-11-15 20:44:53.0	US	LA	MONROE-EL DORADO	f	65 and over	windows	other
51	8a31dc14-c04d-4c20-a39d-38c0403b9cd7	2017-11-15 20:37:04.0	US	NY	BUFFALO	f	55 to 64	windows	other
52	17b20999-876c-45c0-b81f-f9105907aba9	2017-11-15 20:37:14.0	US	MI	GRAND RAPIDS-KALMZOO-B.CRK	f	55 to 64	windows	chrome
53	fd1f868e-c288-495c-a1bb-acdce74e5d66	2017-11-15 20:37:03.0	US	MD	BALTIMORE	m	25 to 34	windows	ie
54	68b422dd-4ed6-4b5f-bbfa-499485c0a5ca	2017-11-15 20:37:05.0	US	PA	PHILADELPHIA	m	25 to 34	windows	chrome
55	8c08ceb4-3865-44d0-91a6-ebd02388e976	2017-11-15 20:40:52.0	US	FL	FT. MYERS-NAPLES	f	55 to 64	windows	firefox
56	bdc70d07-f2af-48b1-8230-5b64f31fc568	2017-11-15 20:41:26.0	US	CA	SAN FRANCISCO-OAK-SAN JOSE	m	25 to 34	windows	firefox
57	beaeac49-4aa9-4b48-93e9-9f26175c01b5	2017-11-15 20:42:14.0	US	OH	COLUMBUS, OH	m	18 to 24	ios	safari
58	b679bf6d-792b-4004-8be5-7b4e615c339c	2017-11-15 20:41:56.0	US	AL	HUNTSVILLE-DECATUR (FLOR)	f	25 to 34	ios	other
59	ce2428bd-eed1-4f0e-b294-641e4cec2a37	2017-11-15 20:42:21.0	US	SC		f	25 to 34	android	chrome
60	dcd8638b-ff4a-4571-97ba-0cd23a055285	2017-11-15 20:43:20.0	US	PA	PITTSBURGH	f	25 to 34	windows	chrome
61	dc2f6b40-7184-48bd-9cff-8d48cfb76b46	2017-11-15 20:43:40.0	US	CO	DENVER	m	35 to 44	windows	chrome
62	ea678201-f82d-4b80-a7df-cabbc54b653e	2017-11-15 20:43:50.0	US	TX	AUSTIN	f	25 to 34	ios	safari
63	731786b2-6759-4ff2-8847-e5c05883691a	2017-11-15 20:46:31.0	US	NC	GREENVILLE-N.BERN-WASHNGTN	f	25 to 34	macos	safari
64	cc5f53b4-f737-4904-adc9-7931d5dcbb98	2017-11-15 20:45:27.0	US	FL	FT. MYERS-NAPLES	f	55 to 64	windows	chrome
65	b63afe56-b2f6-48a3-a025-1e70d4d26539	2017-11-15 20:46:35.0	US	LA	BATON ROUGE	f	55 to 64	windows	firefox
66	e00abb46-7a76-4005-bff7-2a65b8cce226	2017-11-15 20:46:16.0	US	AL	BIRMINGHAM (ANN AND TUSC)	f	18 to 24	ios	safari
67	e76eec51-0800-44b2-bf53-ed8c4b4935e8	2017-11-15 20:46:48.0	US	MI	DETROIT	m	18 to 24	macos	chrome
68	697d2ff6-bac8-4ca0-8dac-8a1a965433b9	2017-11-15 20:48:02.0	US	CT	HARTFORD & NEW HAVEN	f	55 to 64	windows	firefox
69	714e0023-19e7-4276-93a9-c0aea08da3b1	2017-11-15 20:52:15.0	US	RI	PROVIDENCE-NEW BEDFORD	f	55 to 64	windows	other
70	56c2f830-cbd5-42d1-aa3d-3df283b696c9	2017-11-15 20:49:30.0	US	CA	FRESNO-VISALIA	f	18 to 24	ios	safari
71	877dc8a1-fdc1-4d2b-9af5-fe71e4db4ed5	2017-11-15 20:58:15.0	US	TX	SAN ANTONIO	f	45 to 54	windows	chrome
72	26ce17ed-d751-42ce-b24e-10721e6ea5c3	2017-11-15 20:53:41.0	US	KS	WICHITA-HUTCHINSON PLUS	f	65 and over	windows	firefox
73	1b505dd2-33bb-4a73-a557-bd4981156da9	2017-11-15 20:52:36.0	US	IL	CHAMPAIGN&SPRNGFLD-DECATUR	f	25 to 34	android	chrome
74	e8564a5b-10f2-4750-9126-7cbfe976f7a3	2017-11-15 20:52:57.0	US	KY	LEXINGTON	m	18 to 24	android	chrome
75	05b07aef-7bcc-4035-8389-73410c83c5ca	2017-11-15 20:52:51.0	US	MI	GRAND RAPIDS-KALMZOO-B.CRK	f	25 to 34	windows	chrome
76	bd30b2e7-ed0f-441f-bc5f-07d1a048a7a4	2017-11-15 20:53:05.0	US	CA	SAN DIEGO	f	35 to 44	ios	other
77	baae8722-424b-48e1-bfa3-9a9372b6c1d6	2017-11-15 20:53:28.0	US	SC	GREENVLL-SPART-ASHEVLL-AND	m	25 to 34	windows	chrome
78	475030fc-3ed4-4819-9c14-9c6083451142	2017-11-15 20:53:12.0	US	OH	CINCINNATI	m	35 to 44	macos	chrome
79	3057d8fc-668f-42b3-a09f-e4ab6b1201d2	2017-11-15 20:54:11.0	US	AL	BIRMINGHAM (ANN AND TUSC)	f	25 to 34	android	chrome
80	f9e73f22-b45c-4567-a5af-b237096ba16e	2017-11-15 20:56:50.0	US	AR	FT. SMITH-FAY-SPRNGDL-RGRS	f	65 and over	windows	chrome
81	d7fbf27c-f499-4704-91d3-59281440a407	2017-11-15 20:54:41.0	US	WY	CHEYENNE-SCOTTSBLUFF	m	35 to 44	windows	other
82	54248ff0-09d0-4904-af31-7f184ca009c5	2017-11-15 20:54:51.0	US	KY	BOWLING GREEN	f	35 to 44	windows	other
83	759bedcc-6fc3-42e5-a8a1-a6986efbd320	2017-11-15 20:55:59.0	US	OH	CLEVELAND-AKRON (CANTON)	f	18 to 24	macos	safari
84	8a7555fd-21c6-475e-8ef8-d6abe9f0ab79	2017-11-15 20:57:30.0	US	KY	LOUISVILLE	f	25 to 34	android	chrome
85	ff0f4dfe-f62a-4b20-b5c4-f0fc2ff4d91d	2017-11-15 20:56:08.0	US	NY	NEW YORK	m	25 to 34	macos	safari
86	e572f549-c87d-4bfa-886f-2d9a961d540f	2017-11-15 20:57:24.0	US	GA	ALBANY, GA	f	25 to 34	android	chrome
87	cb5a85b9-38a9-4853-8864-c1afae4653b9	2017-11-15 20:57:58.0	US	RI	PROVIDENCE-NEW BEDFORD	f	55 to 64	windows	chrome
88	2c152776-a208-4bfc-ac92-f3a2ef5b8a3f	2017-11-15 20:58:52.0	US	FL	TAMPA-ST. PETE (SARASOTA)	f	55 to 64	windows	chrome
89	d7d1f0c1-ddee-443e-9fd7-cf4724bb77b3	2017-11-15 20:59:42.0	US	NY	ALBANY-SCHENECTADY-TROY	f	25 to 34	windows	firefox
90	00eb4b2e-a7ca-4a5c-900e-94189b3a32cd	2017-11-15 21:00:06.0	US	CA	CHICO-REDDING	f	55 to 64	windows	chrome
91	45f4df4d-3eaf-4d73-9339-ce9a01467a06	2017-11-15 20:59:00.0	US	PA	PHILADELPHIA	m	45 to 54	windows	ie
92	06a9b1d9-da03-447b-ac29-14c4c797464f	2017-11-15 21:02:07.0	US	CA	SACRAMNTO-STKTON-MODESTO	f	65 and over	android	chrome
93	f46ef5bf-9173-4787-ad63-19be3e303817	2017-11-15 21:02:52.0	US	TN	JACKSON, TN	f	55 to 64	windows	chrome
94	98c03171-754f-4efd-b21d-15e5b3e93db1	2017-11-15 21:01:35.0	US	UT	SALT LAKE CITY	f	45 to 54	windows	other
95	45dea764-24ba-42fd-b617-23fc0f88c0ec	2017-11-15 21:01:41.0	US	PA	WILKES BARRE-SCRANTON-HZTN	m	55 to 64	windows	chrome
96	7298438a-4a7a-470c-a4dc-2a0282d231db	2017-11-15 21:00:54.0	US	MI	GRAND RAPIDS-KALMZOO-B.CRK	f	55 to 64	windows	other
97	e556c93a-063b-4539-9700-5235eefddd25	2017-11-15 21:01:55.0	US	SC	MYRTLE BEACH-FLORENCE	f	45 to 54	windows	chrome
98	2d053200-8553-4009-a6f2-a11a727bbe7e	2017-11-15 21:01:14.0	US	IL	TERRE HAUTE	f	65 and over	windows	chrome
99	6c5c203a-75bb-41d1-8353-970116687955	2017-11-15 21:02:18.0	US	CT	HARTFORD & NEW HAVEN	m	45 to 54	windows	chrome

Continued on next page

Table 23: (continued)

100	9d95e695-3073-45b5-a0fd-f9ee7105ec19	2017-11-15 21:01:30.0	US	TN	TRI-CITIES, TN-VA	f	55 to 64	windows	chrome
101	eeab2ec7-50b9-4361-89cf-2c71afdd15d0	2017-11-15 21:01:43.0	US	WI	GREEN BAY-APPLETON	m	35 to 44	windows	chrome
102	04b47051-a14e-478b-bb11-d7bd5eadb881	2017-11-15 21:02:49.0	US	MN	MINNEAPOLIS-ST. PAUL	m	45 to 54	windows	firefox
103	a8fa4bf7-fcae-4125-a560-82a57ee7125a	2017-11-15 21:01:14.0	US	OR	BOISE	m	25 to 34	android	none
104	cade6994-a478-4d15-8950-716d3d8d7a0d	2017-11-15 21:02:19.0	US	MI	DETROIT	f	55 to 64	windows	chrome
105	27e97099-0d86-48cc-aeb5-0c06e1460dc6	2017-11-15 21:03:09.0	US	WA	SEATTLE-TACOMA	f	55 to 64	windows	chrome
106	1ae4e168-ddd5-4f5c-ba4f-ef35d489992d	2017-11-15 21:03:04.0	US	NY	BUFFALO	f	65 and over	windows	chrome
107	91ee71ac-bb24-4e6e-9762-199a2935859a	2017-11-15 21:03:44.0	US	HI	HONOLULU	m	18 to 24	windows	other
108	3545d20c-d6fa-4bc6-a731-3293910eef7a	2017-11-15 21:03:15.0	US	GA	ATLANTA	f	65 and over	windows	chrome
109	fcd3646b-10e9-4c4e-bf4c-4c65161ff47d	2017-11-15 21:02:19.0	US	OH	CINCINNATI	f	45 to 54	windows	ie
110	93b7025c-d036-43f8-ab05-354338c79178	2017-11-15 21:03:08.0	US	NV	RENO	m	18 to 24	windows	chrome
111	4653d275-d630-43e3-9e28-ede5f8642c5c	2017-11-15 21:03:36.0	US	WA	SEATTLE-TACOMA	m	55 to 64	macos	safari
112	2476eb71-85e8-40f0-a7c8-f463e67bcf00	2017-11-15 21:03:11.0	US	MA	BOSTON (MANCHESTER)	f	55 to 64	ios	safari
113	03dfd799-04f7-48ed-ad87-381eae736502	2017-11-15 21:03:38.0	US	MO	ST. LOUIS	f	55 to 64	windows	chrome
114	9ca8a410-a938-4cc6-8348-0adbee0a32f5	2017-11-15 21:02:38.0	US	TN	NASHVILLE	m	35 to 44	ios	safari
115	40d44f99-dd07-4c20-a4f5-83b227f78fa1	2017-11-15 21:03:51.0	US	SC	GREENVLL-SPART-ASHEVLL-AND	m	35 to 44	windows	chrome
116	95eabfa5-b0ed-4068-90a4-d86d0869c7e8	2017-11-15 21:03:28.0	US	MI	DETROIT	f	25 to 34	windows	chrome
117	bb2702f4-770e-4667-9615-a04010fe4414	2017-11-15 21:13:09.0	US	CO	DENVER	f	18 to 24	android	chrome
118	df9a5da4-4830-42e7-a570-fcab6dc6c4c6	2017-11-15 21:05:14.0	US	PA	PHILADELPHIA	f	65 and over	windows	chrome
119	aa4f7bdb-07c5-4ec8-880e-e44441251b0a	2017-11-15 21:05:18.0	US	CA	CHICO-REDDING	f	25 to 34	android	none
120	509b690e-1b7c-494e-8412-79ca61253c03	2017-11-15 21:10:03.0	US	MT	HELENA	f	65 and over	windows	other
121	3719f6c9-13e8-4d0e-a9f7-2a7cc8d3c9a9	2017-11-15 21:07:17.0	US	GA		f	45 to 54	windows	ie
122	2127bea3-5f08-48a3-8895-90569c374896	2017-11-15 21:04:59.0	US	TN	KNOXVILLE	f	25 to 34	windows	other
123	fb8c1247-0cdd-495b-9293-2ec4d069ec9c	2017-11-15 21:07:00.0	US	WI	MADISON	f	45 to 54	windows	other
124	679eb30a-2dcb-494a-b48e-aba7287e84e2	2017-11-15 21:07:06.0	US	ND	MINOT-BSMRCK-DCKNSN(WLSTN)	m	35 to 44	windows	chrome
125	e8737392-f19f-42ca-9662-f114eb21e0e9	2017-11-15 21:07:53.0	US	PA	WILKES BARRE-SCRANTON-HZTN	m	65 and over	windows	chrome
126	802fc9f1-cdbc-4a79-8b33-62cd6c2866e3	2017-11-15 21:07:16.0	US	WY	SALT LAKE CITY	m	35 to 44	android	none
127	5375edbe-4c10-44e2-8736-50f7732052f7	2017-11-15 21:07:33.0	US	MO	KANSAS CITY	f	55 to 64	windows	other
128	153577a8-e163-438f-9b99-87fcbefef183	2017-11-15 21:07:29.0	US	TX	CORPUS CHRISTI	f	55 to 64	windows	other
129	8eb15225-f670-437b-a52c-57b375d93e68	2017-11-15 21:07:12.0	US	TN	CHATTANOOGA	f	65 and over	macos	chrome
130	806ffb20-108a-4d91-af44-a1248669a42b	2017-11-15 21:07:22.0	US	NY	NEW YORK	m	55 to 64	windows	chrome
131	b54f66af-1168-4130-8233-5f0212d5ab50	2017-11-15 21:08:38.0	US	OK	TULSA	m	55 to 64	windows	other
132	938af9d1-6695-4aec-bce8-331a4f32055d	2017-11-15 21:08:34.0	US	CA	LOS ANGELES	m	25 to 34	windows	chrome
133	aabf855d-0208-4df8-a31b-933e5a8f95e0	2017-11-15 21:08:41.0	US	NC	CHARLOTTE	f	25 to 34	windows	chrome
134	19870bbf-7b2c-43a2-8e67-ddc47d3a0a51	2017-11-15 21:11:29.0	US	OH	CLEVELAND-AKRON (CANTON)	f	35 to 44	android	chrome
135	a3ffdaf8-f7d8-48a4-98e3-0b3b3fda7fb1	2017-11-15 21:09:42.0	US	TN	NASHVILLE	m	55 to 64	windows	chrome
136	4f87edaa-ec99-485e-bb3d-d2fc8ab6bed8	2017-11-15 21:09:42.0	US	FL	MIAMI-FT. LAUDERDALE	m	25 to 34	ios	safari
137	31022702-0814-4b6b-b3d2-62cbad174728	2017-11-15 21:10:47.0	US	IL	ROCKFORD	f	55 to 64	windows	firefox
138	a2566c97-f10f-4a57-9c2c-f69c13796ae9	2017-11-15 21:10:03.0	US	NJ	PHILADELPHIA	f	45 to 54	windows	chrome
139	d49b15fb-295e-473a-b59e-d46c892b12d5	2017-11-15 21:12:40.0	US	WV	WASHINGTON, DC (HAGRSTWN)	f	65 and over	windows	chrome
140	82aa3005-ef1a-4f66-bc9a-bb96c553e70b	2017-11-15 21:11:44.0	US	WA	SEATTLE-TACOMA	f	45 to 54	ios	safari
141	c53adb09-5e8c-494e-825c-65c37d395cb1	2017-11-15 21:09:57.0	US	OH	COLUMBUS, OH	f	55 to 64	windows	other
142	8245a8b8-320e-46b6-ae88-a771c2a0ee47	2017-11-15 21:09:28.0	US	NY	NEW YORK	f	65 and over	windows	chrome
143	47545c45-052f-4a5a-9e19-c1e1363509f1	2017-11-15 21:11:37.0	US	FL	TAMPA-ST. PETE (SARASOTA)	f	65 and over	windows	other
144	c1df249e-c466-4f48-8b8a-504a6855a934	2017-11-15 21:10:37.0	US	TX	BEAUMONT-PORT ARTHUR	f	35 to 44	windows	chrome
145	9b382fea-d574-4121-b9a7-92160cdfd67d	2017-11-15 21:11:12.0	US	MO	ST. LOUIS	f	65 and over	macos	safari
146	6ef40f78-c84e-449e-b0f0-f0bd1ec31ce2	2017-11-15 21:11:38.0	US	MO	ST. LOUIS	m	55 to 64	windows	chrome
147	89599c53-e490-416f-9c78-cc1b34c75094	2017-11-15 21:11:14.0	US	WV	WASHINGTON, DC (HAGRSTWN)	f	55 to 64	windows	chrome
148	7f76ebee-3601-409e-8472-5e1b3173f85b	2017-11-15 21:10:44.0	US	TX	DALLAS-FT. WORTH	m	25 to 34	ios	safari
149	a2c79658-4e26-496d-9288-959300b54ba4	2017-11-15 21:11:05.0	US	OH	TOLEDO	m	35 to 44	windows	chrome

Continued on next page

Table 23: (continued)

150	f5c81caf-22b8-43fa-9e6f-4e777ded3d96	2017-11-15 21:11:27.0	US	MN	MINNEAPOLIS-ST. PAUL	m	18 to 24	ios	none
151	37122911-462f-4f0a-98cc-dd589a67d063	2017-11-15 21:11:40.0	US	SC	COLUMBIA, SC	m	35 to 44	android	none
152	07494c08-6243-455e-bc28-8b007d612d43	2017-11-15 21:13:03.0	US	TN	CHATTANOOGA	f	65 and over	windows	chrome
153	c202b28e-0b01-4816-b10a-20ac0e223f78	2017-11-15 21:11:44.0	US	WI	MADISON	f	18 to 24	windows	chrome
154	bd9f9063-20ee-4a9f-ab13-20cc9ab0bd07	2017-11-15 21:12:02.0	US	OH	COLUMBUS, OH	m	35 to 44	windows	chrome
155	ec61f018-f19a-4f12-89f7-aa2d2dfaece7	2017-11-15 21:12:40.0	US	NC	RALEIGH-DURHAM (FAYETVLLE)	f	45 to 54	windows	chrome
156	f0e9e248-e85e-4b6d-b8c0-855fedde455c	2017-11-15 21:12:51.0	US	PA	PITTSBURGH	f	45 to 54	windows	chrome
157	73fd9187-4f3b-4e47-8c98-8e58e2fa67ba	2017-11-15 21:13:46.0	US	AR	LITTLE ROCK-PINE BLUFF	f	45 to 54	windows	firefox
158	1b592da4-be10-42dc-aa08-621a8b1306ed	2017-11-15 21:13:27.0	US	PA	WILKES BARRE-SCRANTON-HZTN	f	25 to 34	windows	other
159	ab20b704-1056-4fba-a7d9-52e227ae045c	2017-11-15 21:15:02.0	US	MI	GRAND RAPIDS-KALMZOO-B.CRK	f	65 and over	windows	chrome
160	ee62d076-abe5-4f05-8925-afcf2950bcaf	2017-11-15 21:14:00.0	US	NJ	PHILADELPHIA	m	25 to 34	windows	chrome
161	a528f7ad-677f-4e1e-b7a7-1751ea753bb3	2017-11-15 21:13:49.0	US	PA	PHILADELPHIA	f	65 and over	macos	safari
162	9f2a0a89-442c-408c-855b-a08961b2e13d	2017-11-15 21:14:07.0	US	CT	HARTFORD & NEW HAVEN	m	18 to 24	ios	other
163	269af535-a403-4a3e-9ff3-464e362e7bfc	2017-11-15 21:14:33.0	US	GA	ATLANTA	f	45 to 54	windows	chrome
164	235c7b32-f79f-4456-a1c6-bbaeb6c5f015	2017-11-15 21:14:31.0	US	OH	CLEVELAND-AKRON (CANTON)	m	18 to 24	windows	chrome
165	c401589a-132c-4563-91f0-55dd8503b901	2017-11-15 21:14:24.0	US	FL	ORLANDO-DAYTONA BCH-MELBRN	m	65 and over	ios	other
166	ce787bff-805b-4d29-b767-fcfa94d6a3bd	2017-11-15 21:15:51.0	US	NM		m	18 to 24	android	chrome
167	ac51bf69-f6ff-4c69-b0b3-393dbb005a81	2017-11-15 21:15:32.0	US	OH	CINCINNATI	m	25 to 34	windows	chrome
168	a64ef346-55e9-49b1-931c-1f2b9eb7833f	2017-11-15 21:16:37.0	US	MI	GRAND RAPIDS-KALMZOO-B.CRK	m	35 to 44	windows	chrome
169	ace2391c-11eb-43af-87d2-a3bea5c52d99	2017-11-15 21:17:21.0	US	OK	OKLAHOMA CITY	m	45 to 54	windows	chrome
170	4efc00e1-794b-480e-a24d-2de00a89a61e	2017-11-15 21:16:06.0	US	IL	CHAMPAIGN&SPRNGFLD-DECATUR	m	18 to 24	macos	chrome
171	948c619f-a74b-41dd-ac1f-322ebfb14fde	2017-11-15 21:17:37.0	US	NY	NEW YORK	f	35 to 44	windows	chrome
172	ee698965-a7a4-4cce-90cc-21e88f2eed4f	2017-11-15 21:17:04.0	US	NY	NEW YORK	m	55 to 64	windows	chrome
173	f44e8cb1-e431-4eeb-8020-ff04d568b99e	2017-11-15 21:17:36.0	US	CA	LOS ANGELES	f	55 to 64	windows	other
174	ee81fd6a-3e56-46cf-8627-60c7db936516	2017-11-15 21:17:53.0	US	AK	ANCHORAGE	m	55 to 64	windows	chrome
175	70a8a0d6-65d8-4e14-80e9-caa075077f57	2017-11-15 21:19:28.0	US	NC	GREENSBORO-H.POINT-W.SALEM	f	65 and over	windows	chrome
176	433b8bf3-578b-48b4-a746-deac2b8f2f22	2017-11-15 21:19:07.0	US	NH	BOSTON (MANCHESTER)	f	65 and over	windows	chrome
177	36da39ee-d118-4b67-966a-ac6a3ba292db	2017-11-15 21:19:10.0	US	TX	HOUSTON	f	55 to 64	windows	other
178	46c9c3db-c3e9-494b-97e8-b295167533bf	2017-11-15 21:18:24.0	US	OK	TULSA	f	25 to 34	windows	ie
179	7ef0f0f1-55d3-43cf-af0a-6e259a49e297	2017-11-15 21:20:42.0	US	TX	HOUSTON	f	45 to 54	windows	chrome
180	e113b635-44e0-4af9-8cf8-7e3aedeea753	2017-11-15 21:18:45.0	US	OH	COLUMBUS, OH	f	35 to 44	windows	chrome
181	ea8fdd76-3a10-4f22-9b7a-50fa69507ad3	2017-11-15 21:21:43.0	US	FL	TAMPA-ST. PETE (SARASOTA)	m	55 to 64	android	chrome
182	d57e5b83-c510-426b-8b5e-481caac64991	2017-11-15 21:20:29.0	US	MN	MANKATO	f	65 and over	windows	other
183	727243f1-2f5d-4584-9cda-a15a7e4e6d05	2017-11-15 21:20:18.0	US	AZ	TUCSON (SIERRA VISTA)	f	65 and over	other	firefox
184	bf995cd2-ba8b-4289-924a-ba97325f229b	2017-11-15 21:19:44.0	US	TN	KNOXVILLE	f	35 to 44	windows	chrome
185	2ededd37-69a8-4457-a149-f3cc09a5bb82	2017-11-15 21:21:37.0	US	IN	INDIANAPOLIS	f	65 and over	windows	chrome
186	906bfc40-84fe-4d6b-8a7a-1241f302cfbb	2017-11-15 21:21:24.0	US	UT	SALT LAKE CITY	f	55 to 64	windows	other
187	d3a962de-4553-41f1-b5b5-d7257524ca64	2017-11-15 21:19:53.0	US	NY	ROCHESTER, NY	m	25 to 34	windows	other
188	6c0a9811-ad12-4e8f-8807-8bcdbb62ec30	2017-11-15 21:23:45.0	US	IA	CEDAR RAPIDS-WTRLO-IWC&DUB	m	55 to 64	android	none
189	7e910679-a794-4d8c-814e-1d62e50aa266	2017-11-15 21:20:12.0	US	PA	PHILADELPHIA	m	65 and over	windows	chrome
190	669132d2-99ad-4183-b067-5354fc736060	2017-11-15 21:21:12.0	US	WI	GREEN BAY-APPLETON	m	65 and over	windows	chrome
191	17eb9728-7a3f-4865-a404-8a75d492e21d	2017-11-15 21:21:40.0	US	AZ	PHOENIX (PRESCOTT)	f	55 to 64	windows	chrome
192	5432438d-2fa5-461f-829e-cbf229440c03	2017-11-15 21:22:54.0	US	PA	WILKES BARRE-SCRANTON-HZTN	m	65 and over	windows	other
193	6343ebe1-191f-4c31-83c3-30cabbd7ccae	2017-11-15 21:21:53.0	US	LA	NEW ORLEANS	m	18 to 24	macos	chrome
194	0009218e-e03a-40b2-a7f0-4f503e1cb564	2017-11-15 21:23:02.0	US	WA	SEATTLE-TACOMA	f	65 and over	windows	other
195	ddd3deda-f944-4d9f-8bd4-280aedb1b7ad	2017-11-15 21:22:56.0	US	ID	BOISE	m	35 to 44	macos	safari
196	71860a7d-4e34-43c3-898a-47c079e9dfbf	2017-11-15 21:23:14.0	US	CT	HARTFORD & NEW HAVEN	f	45 to 54	windows	chrome
197	ab47e859-26a9-43b5-86bb-90b1dd183f93	2017-11-15 21:23:58.0	US	FL	TAMPA-ST. PETE (SARASOTA)	m	55 to 64	android	chrome
198	ace42aa5-d910-45fe-9c68-5c0ac84b9ffa	2017-11-15 21:29:48.0	US	OH	CLEVELAND-AKRON (CANTON)	f	45 to 54	windows	firefox
199	64491b3d-a2bb-4940-8b8a-d0f50f1700a1	2017-11-15 21:23:54.0	US	TX	CORPUS CHRISTI	f	45 to 54	windows	firefox

Continued on next page

Table 23: (continued)

200	3a85b7d2-b90f-4286-bbb0-9302b6d86863	2017-11-15 21:26:14.0	US	CA	LOS ANGELES	f	55 to 64	windows	chrome
201	18cd00c0-9c38-42cd-974b-fb6217fa78c4	2017-11-15 21:25:02.0	US	TX		m	35 to 44	android	chrome
202	31689ca7-d33f-460e-8312-75e84ad09854	2017-11-15 21:24:35.0	US	OK		f	25 to 34	android	none
203	44f37260-9da9-494e-af94-ae095e3ea3d9	2017-11-15 21:24:32.0	US	MO	KANSAS CITY	m	55 to 64	windows	other
204	5a0816c8-7fce-41f4-a440-1e9a32ac9a30	2017-11-15 21:24:35.0	US	WA	SEATTLE-TACOMA	f	25 to 34	macos	firefox
205	77798684-ceee-4811-8db6-2f9ec9296f30	2017-11-15 21:26:06.0	US	AZ	PHOENIX (PRESCOTT)	f	55 to 64	windows	firefox
206	a620e823-be61-4c11-968d-6dc4aa2fd53c	2017-11-15 21:25:31.0	US	WA	PORTLAND, OR	f	65 and over	windows	other
207	7c89c5ac-b96a-4026-b6fe-b05155950394	2017-11-15 21:26:01.0	US	MI	DETROIT	f	65 and over	windows	chrome
208	139d1fb6-32f1-45e2-a825-d61046e85d2c	2017-11-15 21:25:10.0	US	PA	WILKES BARRE-SCRANTON-HZTN	f	55 to 64	windows	other
209	cf155ec0-9d6a-47a4-b3e8-752d82bb4bae	2017-11-15 21:26:55.0	US	MI	FLINT-SAGINAW-BAY CITY	f	55 to 64	windows	other
210	8dd6ede2-eddd-486b-9879-4248e57c5edd	2017-11-15 21:26:53.0	US	MO	ST. LOUIS	m	45 to 54	android	none
211	8c3f3ade-b4c3-46bf-9930-75974e9d5cad	2017-11-15 21:26:53.0	US	WI	WAUSAU-RHINELANDER	m	45 to 54	windows	chrome
212	8b1771db-ce01-41f9-9ac4-b41bdec9575a	2017-11-15 21:27:58.0	US	PA	PITTSBURGH	f	55 to 64	windows	chrome
213	4f1c887c-3941-425f-8aa4-ca6d8eedb414	2017-11-15 21:28:36.0	US	RI	PROVIDENCE-NEW BEDFORD	f	55 to 64	windows	chrome
214	4e835bd1-72a9-4866-abf2-211e03b4a576	2017-11-15 21:28:55.0	US	IL	CHICAGO	m	45 to 54	windows	firefox
215	63396bd9-3672-4493-8eef-235c6dcf9d99	2017-11-15 21:28:38.0	US	WI	WAUSAU-RHINELANDER	m	18 to 24	android	none
216	00370c04-9675-45e2-96d7-b203bb71acfe	2017-11-15 21:29:41.0	US	VA	TRI-CITIES, TN-VA	f	55 to 64	windows	other
217	fd7e8902-3add-457d-b2e3-f23a89c2caf9	2017-11-15 21:31:05.0	US	PA	WILKES BARRE-SCRANTON-HZTN	f	25 to 34	macos	safari
218	4ffafb2b-3e65-4eee-9f28-263622a322fa	2017-11-15 21:29:55.0	US	IL	CHICAGO	m	35 to 44	windows	chrome
219	861c8de3-338b-4c28-a8f6-061f43617934	2017-11-15 21:28:53.0	US	IA	DAVENPORT-R.IOLAND-MOLINE	f	25 to 34	windows	chrome
220	d1989baa-ec06-4514-88c2-2768840b64c4	2017-11-15 21:28:52.0	US	PA	PITTSBURGH	m	55 to 64	windows	other
221	9143ff4a-1e2a-44ad-a322-30f0a15e2557	2017-11-15 21:28:57.0	US	OH	CINCINNATI	m	45 to 54	windows	chrome
222	52e4b3b6-4371-4011-ba06-7fa3ba031542	2017-11-15 21:28:56.0	US	PA	PHILADELPHIA	m	25 to 34	ios	safari
223	dd27d4c8-b2c6-478a-8e17-46285391db5a	2017-11-15 21:29:26.0	US	OH	CINCINNATI	m	45 to 54	windows	chrome
224	cc1bf1c6-782f-4c01-97cb-5e296ec0b50e	2017-11-15 21:29:54.0	US	OH	DAYTON	f	45 to 54	windows	other
225	b39d02af-399e-4b7b-a5cc-0b35efa2b581	2017-11-15 21:30:05.0	US	FL	WEST PALM BEACH-FT. PIERCE	f	65 and over	windows	chrome
226	0c5576f9-1ebe-4550-9068-c846d0fb6b64	2017-11-15 21:28:54.0	US	VA	HARRISONBURG	m	18 to 24	windows	chrome
227	15c67607-176b-4881-abf4-0afa59797905	2017-11-15 21:30:15.0	US	PA	ERIE	m	55 to 64	windows	other
228	d245e3dd-5f4b-4642-ae09-dd09b8317861	2017-11-15 21:30:24.0	US	FL	TAMPA-ST. PETE (SARASOTA)	m	55 to 64	windows	other
229	9af14e2e-3082-4127-a61b-4455f01bb3d5	2017-11-15 21:30:34.0	US	PA	PHILADELPHIA	f	65 and over	ios	safari
230	03f3a0ed-2e20-4975-ac8c-583b5175cfc9	2017-11-15 21:33:57.0	US	ND	MINOT-BSMRCK-DCKNSN(WLSTN)	f	65 and over	windows	other
231	6a3f7653-5004-41a7-bca3-af0c758b577e	2017-11-15 21:30:59.0	US	PA	PHILADELPHIA	m	25 to 34	windows	chrome
232	7922e36c-df47-48da-bd9a-763b71c00e5d	2017-11-15 21:32:27.0	US	WI	LA CROSSE-EAU CLAIRE	f	25 to 34	windows	chrome
233	497b4b1c-f79f-4144-9dd9-b9c66ce6ec63	2017-11-15 21:31:35.0	US	KY	BOWLING GREEN	f	65 and over	windows	chrome
234	2b20ca92-813f-4512-8da6-63ea095fac06	2017-11-15 21:31:10.0	US	MA	BOSTON (MANCHESTER)	m	25 to 34	android	chrome
235	77881dff-7bd0-4e8d-b3c5-29d77baad9ae	2017-11-15 21:32:47.0	US	FL	ORLANDO-DAYTONA BCH-MELBRN	f	65 and over	windows	firefox
236	82969bc3-2025-41ef-89e0-ded14c2febd9	2017-11-15 21:33:03.0	US	NE	NORTH PLATTE	m	65 and over	windows	chrome
237	3074b367-800d-4129-a91a-fbff76815623	2017-11-15 21:31:51.0	US	OH	CINCINNATI	m	45 to 54	windows	chrome
238	ef743387-7ec3-4dd8-ad45-f46671632d4e	2017-11-15 21:33:21.0	US	TX	DALLAS-FT. WORTH	f	55 to 64	android	none
239	29c1bccc-5f1b-4575-b231-e69e39e291ea	2017-11-15 21:32:16.0	US	FL	ORLANDO-DAYTONA BCH-MELBRN	m	45 to 54	windows	other
240	23169c9e-1d27-46d3-bb28-f88a42184a3e	2017-11-15 21:32:47.0	US	PA	PHILADELPHIA	m	35 to 44	ios	safari
241	3fd2c8f7-d076-411b-a021-b38164802405	2017-11-15 21:34:27.0	US	MI	DETROIT	f	55 to 64	windows	other
242	dfdf976c-8814-493e-aed0-0e3a04611b41	2017-11-15 21:33:58.0	US	SC	CHARLESTON, SC	f	65 and over	windows	other
243	983f81d9-c1a8-4ce4-a55c-575c6bd3ef37	2017-11-15 21:35:44.0	US	TX	HOUSTON	f	65 and over	windows	other
244	e5e2fa86-2777-4062-a5f2-a2e540817de6	2017-11-15 21:33:28.0	US	NY	WATERTOWN	m	55 to 64	windows	firefox
245	3328b95a-51bc-4c16-9572-4b6076af8746	2017-11-15 21:34:00.0	US	NY	NEW YORK	m	65 and over	ios	safari
246	eb0ca344-bf0a-43e1-a463-9ea1f3408730	2017-11-15 21:34:27.0	US	TX	CORPUS CHRISTI	m	55 to 64	windows	other
247	0f367444-81f5-42da-ae36-b636f3e13231	2017-11-15 21:34:19.0	US	MI	DETROIT	m	25 to 34	android	none
248	7dad9754-aa87-492d-8ebd-5dce62ca9f0f	2017-11-15 21:34:56.0	US	CO	DENVER	f	25 to 34	other	chrome
249	c70cca09-4fe6-4a75-8ecb-ba72643c8c01	2017-11-15 21:33:51.0	US	PA	PHILADELPHIA	m	45 to 54	windows	chrome

Continued on next page

Table 23: (continued)

250	3e7df673-0595-4303-ad4b-7975e804f859	2017-11-15 21:34:39.0	US	OK	TULSA	m	35 to 44	windows	chrome
251	9eafb719-b73a-4b70-8efb-3f8805b55bbc	2017-11-15 21:35:33.0	US	CA	SAN DIEGO	f	55 to 64	windows	chrome
252	1f211d9d-5b95-47da-9da0-e332ba1b4fe6	2017-11-15 21:35:32.0	US	PA	PHILADELPHIA	m	25 to 34	windows	firefox
253	f92e54f0-628a-4c7b-8831-3f42a4f3d03c	2017-11-15 21:34:37.0	US	IL	ST. LOUIS	m	65 and over	windows	chrome
254	d225d8e5-a87f-454d-b8dc-f3f0a7ecbad8	2017-11-15 21:36:04.0	US	OH	CINCINNATI	m	45 to 54	windows	other
255	d90cb592-75cc-4890-b0a4-1dffcc002355	2017-11-15 21:35:29.0	US	PA	PHILADELPHIA	m	25 to 34	windows	chrome
256	e24a885e-6c60-4920-b715-bdd1889465ad	2017-11-15 21:35:24.0	US	OR	PORTLAND, OR	m	18 to 24	android	none
257	a83d16cd-21e5-45ce-8b76-2e43e5a779ae	2017-11-15 21:35:10.0	US	OH	COLUMBUS, OH	f	45 to 54	windows	chrome
258	ba3faf88-8dd7-4fd1-b6f8-327f9e68f488	2017-11-15 21:35:56.0	US	CA	SACRAMNTO-STKTON-MODESTO	m	25 to 34	windows	chrome
259	dfc749d1-6287-4b0f-8809-18532f5473c3	2017-11-15 21:39:07.0	US	IN	EVANSVILLE	f	65 and over	windows	chrome
260	5a99af06-4d85-4017-b025-8bce3ff1e09a	2017-11-15 21:36:34.0	US	PA	PHILADELPHIA	m	35 to 44	windows	ie
261	d172c513-64c0-4710-a96f-ebf4c6f96871	2017-11-15 21:37:58.0	US	MI	FLINT-SAGINAW-BAY CITY	f	65 and over	windows	chrome
262	e301e6ac-3816-45b0-84ba-465f2a2020b3	2017-11-15 21:38:14.0	US	AR	SPRINGFIELD, MO	f	25 to 34	windows	chrome
263	8bbdd380-9989-4fba-85d4-b73fa462a214	2017-11-15 21:38:00.0	US	IN	INDIANAPOLIS	m	65 and over	windows	chrome
264	d0f399e0-fc58-4c7d-8bfb-b92658591c41	2017-11-15 21:40:33.0	US	WI	MILWAUKEE	f	25 to 34	android	none
265	bb1a18e9-c6bf-4231-9600-bfb92c948888	2017-11-15 21:40:25.0	US	NC	RALEIGH-DURHAM (FAYETVLLE)	f	55 to 64	ios	other
266	34f0dde1-5d44-4b34-833a-716da57bdce2	2017-11-15 21:38:21.0	US	NC	GREENVILLE-N.BERN-WASHNGTN	m	18 to 24	windows	chrome
267	6714a573-a86d-4fab-8596-64ecb7aa0607	2017-11-15 21:38:04.0	US	IL	CHICAGO	m	25 to 34	other	chrome
268	69390f6e-7b31-459c-9f44-4e8b7eb47ffe	2017-11-15 21:39:09.0	US	WI	GREEN BAY-APPLETON	f	55 to 64	windows	chrome
269	6719b180-1748-4b83-8c83-c5abfea6e863	2017-11-15 21:38:27.0	US	AZ	PHOENIX (PRESCOTT)	f	25 to 34	android	none
270	4547d0b2-9253-41cc-99d4-fcb99743580a	2017-11-15 21:39:09.0	US	MN	MINNEAPOLIS-ST. PAUL	m	18 to 24	windows	chrome
271	13058db8-d5e5-44b0-be6a-0b89049b3556	2017-11-15 21:38:19.0	US	MO	KANSAS CITY	m	55 to 64	windows	chrome
272	63592ffe-1456-49c1-b5d9-ccd0c5334246	2017-11-15 21:38:55.0	US	DC	WASHINGTON, DC (HAGRSTWN)	m	25 to 34	windows	chrome
273	0448df71-83f3-425b-87da-15e442de058c	2017-11-15 21:40:27.0	US	ME	BANGOR	f	55 to 64	windows	chrome
274	95163040-d77e-444f-95a1-2b3e7ee80349	2017-11-15 21:45:44.0	US	FL	TAMPA-ST. PETE (SARASOTA)	m	25 to 34	windows	chrome
275	7b9dc13e-5757-4e95-9a6e-f97d19395b13	2017-11-15 21:41:26.0	US	FL	WEST PALM BEACH-FT. PIERCE	m	55 to 64	windows	chrome
276	bb818bc4-dffb-4c85-93ce-d8e1bc828f3f	2017-11-15 21:40:08.0	US	AL	HUNTSVILLE-DECATUR (FLOR)	f	45 to 54	windows	chrome
277	1f8d5e8b-5687-42dd-be02-7b4f5540b7b4	2017-11-15 21:40:32.0	US	CT	NEW YORK	m	55 to 64	windows	chrome
278	9c53530e-ed19-48a6-9beb-78531610a1cb	2017-11-15 21:40:19.0	US	MO	COLUMBIA-JEFFERSON CITY	f	25 to 34	windows	ie
279	dc09cd64-9239-46a7-b1bb-8f8879b33c12	2017-11-15 21:42:26.0	US	NJ	NEW YORK	m	55 to 64	android	none
280	25f45ed9-1fcb-471c-9ffb-eeab3001a540	2017-11-15 21:42:30.0	US	NE	LINCOLN & HASTINGS-KRNY	m	65 and over	windows	chrome
281	cc675f3d-f48b-4e98-8d95-a7870009712f	2017-11-15 21:41:29.0	US	NH	BOSTON (MANCHESTER)	f	25 to 34	ios	safari
282	bce3571d-abca-47c8-8e9a-4a705b65f75a	2017-11-15 21:43:23.0	US	CA	LOS ANGELES	f	65 and over	windows	other
283	bf65f93c-570b-4d17-851e-1dd39dbe80fe	2017-11-15 21:41:46.0	US	NH	BOSTON (MANCHESTER)	f	35 to 44	windows	other
284	b04fd7a9-92d1-4a3f-8ca5-1f8b399e78d4	2017-11-15 21:41:35.0	US	NJ	PHILADELPHIA	m	25 to 34	windows	chrome
285	b2cd958e-2214-4103-912f-3791564b21c9	2017-11-15 21:42:35.0	US	KS	KANSAS CITY	f	65 and over	windows	other
286	6c8111b6-8149-4df9-8aaf-97bcf828e3f1	2017-11-15 21:42:07.0	US	AL	HUNTSVILLE-DECATUR (FLOR)	m	18 to 24	android	chrome
287	7e982155-70d5-4a77-8ad4-5d30a25122b6	2017-11-15 21:41:30.0	US	CA		m	25 to 34	windows	ie
288	1bfdca17-8d1d-4df7-8004-69ecf6ae29d7	2017-11-15 21:41:27.0	US	CA	YUMA-EL CENTRO	f	18 to 24	other	chrome
289	bf60171d-ca9f-4d33-a84e-a05d4f6f39a2	2017-11-15 21:42:32.0	US	NC	RALEIGH-DURHAM (FAYETVLLE)	f	25 to 34	windows	chrome
290	36b1dddb-a69b-41d8-9ae0-ac3e8e957a93	2017-11-15 21:44:47.0	US	MI	LANSING	f	45 to 54	windows	chrome
291	fb7ac5e5-3d9c-405a-a2bc-d0ba872bb6bf	2017-11-15 21:42:04.0	US	NM	ALBUQUERQUE-SANTA FE	f	18 to 24	android	chrome
292	41febb94-28c9-468d-808f-2dcd40e36fa8	2017-11-15 21:41:58.0	US	NY	BUFFALO	f	25 to 34	ios	other
293	47414a7f-c367-4635-8034-37b028014edc	2017-11-15 21:42:31.0	US	OK	TULSA	m	18 to 24	android	chrome
294	b5cb80bc-9b1c-42c3-b6a2-7e4d9e832252	2017-11-15 21:41:55.0	US	PA	HARRISBURG-LNCSTR-LEB-YORK	m	25 to 34	windows	chrome
295	6c756144-5fd7-454b-bd88-2ff400c63ef1	2017-11-15 21:42:18.0	US	OH	CINCINNATI	f	25 to 34	windows	chrome
296	99526bd4-9b29-4f8d-947a-b798d0ebff39	2017-11-15 21:42:52.0	US	KY	KNOXVILLE	f	25 to 34	ios	safari
297	8a3dd9de-d36f-46f6-9d27-840c60aad9f5	2017-11-15 21:43:07.0	US	FL	TAMPA-ST. PETE (SARASOTA)	f	45 to 54	android	chrome
298	bd2298e6-5cae-490a-b8fe-6ea2ec1ab563	2017-11-15 21:42:16.0	US	AL	MOBILE-PENSACOLA (FT WALT)	m	25 to 34	ios	safari
299	3c5296f4-04cb-4d3f-bd84-dfd42429693b	2017-11-15 21:44:08.0	US	TX	ODESSA-MIDLAND	f	18 to 24	android	chrome

Continued on next page

Table 23: (continued)

300	ff538b55-a043-465f-b933-655c7a69c887	2017-11-15 21:44:59.0	US	TN	KNOXVILLE	m	35 to 44	windows	chrome
301	afb2d653-4d81-4337-92f3-8082c2a30e4f	2017-11-15 21:45:17.0	US	FL	FT. MYERS-NAPLES	f	65 and over	windows	other
302	86758ee0-fa73-4d15-ac28-8a85ebeb6a4f	2017-11-15 21:45:03.0	US	OH	LIMA	f	18 to 24	android	chrome
303	20883daf-8511-46ad-8d0d-f08dfcfb1e31	2017-11-15 21:46:09.0	US	NC	GREENVLL-SPART-ASHEVLL-AND	f	18 to 24	android	chrome
304	48f4fddf-b004-43cb-bf5f-e55b07115828	2017-11-15 21:45:54.0	US	OH	CINCINNATI	f	35 to 44	windows	chrome

Table 24: Key of abbreviations used in header of Table 25

Abbrev.	**Question**
Q1	Would you support or oppose a wind turbine project if you could always see the installed turbines from where you live?
Q2	Would you support or oppose a wind turbine project viewable from where you live that uses only retractable wind turbines? Retractable wind turbines are able to be deployed and retracted when specified. E.g., deploy when windy and retract when calm.
S (Q2.Choice1)	Support retractable wind turbines under certain conditions
O (Q2.Choice2)	Oppose retractable wind turbines
Q3	When should the retractable wind turbines be retracted and hidden? Check all that apply.
3C1 (Q3.Choice1)	When it's not windy
3C2 (Q3.Choice2)	When the month is not March
3C3 (Q3.Choice3)	During every morning
3C4 (Q3.Choice4)	During every afternoon
3C5 (Q3.Choice5)	During every night
3C6 (Q3.Choice6)	When birds are migrating
3C7 (Q3.Choice7)	During every weekend
3C8 (Q3.Choice8)	After it has been visible a certain proportion the month
C9 (Q3.Choice9)	Other
Q3.Choice9 [value]	Other [value]
Q4	After what percentage of the month that the wind turbine has been visible should the turbine be hidden?
Q5	If you have any comments, please share them here. (Especially, if you oppose all types of wind turbines including retractable wind turbines, why?)

Table 25: Interview responses

No.	Q1	Q2	C1	C2	C3	C4	C5	C6	C7	C8	C9	Q3.Choice9 [value]	Q4	Q5
2	Support	S	0	0	0	0	0	1	0	0			10	no comments
3	Support	S	1	0	1	0	0	1	0	1			50	none
4	Support	S	1	1	0	0	1	0	1	1			78	i dont
5	Support	S	0	0	0	0	0	1	0	0			60	No comment
6	Support	O												If they are retractable, they will kill fewer birds. We need to kill all the birds.
7	Support	S	0	0	0	0	0	1	0	0			50	No
8	Support	S	1	0	0	0	0	1	0	0			20	None
9	Support	S	1	0	0	0	0	0	0	0			10	do not oppose at all
10	Support	S	0	0	0	0	0	1	0	0	1	Windy days storms	20	No comments
11	Support	S	1	0	0	0	0	0	0	0			10	no comment
12	Support	S	1	0	0	0	0	0	0	0			80	no comments...
13	Support	S	0	0	0	0	0	0	0	1			0	no
14	Support	S	0	0	0	0	0	1	0	0			0	No
15	Oppose	O												none
16	Support	S	1	1	1	0	1	0	0	0			5	none
17	Oppose	O												I just don't like them being a part of our beautiful landscape.
18	Support	S	1	0	0	0	0	0	0	0			20	nope
19	Support	S	1	0	0	0	0	1	0	0			1	NO
20	Support	S	1	0	0	0	0	0	0	0			7	no
21	Support	O												no comment
22	Support	S	1	0	0	0	0	1	0	0			10	I support the use of wind turbines.
23	Support	S	0	0	0	0	0	1	0	0			5	I dont
24	Support	S	0	0	0	0	0	1	0	0			100	I do not oppose
25	Support	S	0	0	0	0	0	1	0	0			90	I don't have any comments
26	Support	S	0	0	0	0	1	0	0	0			5	no comment
27	Support	S	1	0	0	0	0	1	0	0			90	no comments
28	Oppose	S	0	0	0	0	0	1	1	0			0	no comment
29	Support	S	0	0	0	0	0	1	0	0			90	pro wind energy
30	Support	S	1	0	0	0	0	0	0	0			90	I think they are a great idea
31	Support	S	0	0	0	0	0	0	0	0	1	Never	0	None
32	Support	S	1	0	0	0	0	1	0	0			11	none i can think of
33	Support	S	1	0	0	0	0	0	0	0			99	Alternative energy sources including wind turbines are an important advancement in our environment to limit fossil fuel and coal usage for energy.
34	Support	S	1	0	0	0	0	0	0	0			25	I don't oppose
35	Support	S	0	0	0	0	0	0	0	1			5	Great idea
36	Support	S	0	0	1	0	0	0	0	0			40	because of cost to have them
37	Support	S	0	0	0	0	0	0	0	0	1	Doesn't matter	80	None at this time
38	Support	S	1	0	0	0	0	0	0	0			10	They good plan

Continued on next page

Table 25: (continued)

No.	Q1	Q2	C1	C2	C3	C4	C5	C6	C7	C8	C9	Q3.Choice9 [value]	Q4	Q5
39	Oppose	O												it would save on money
40	Support	S	1	0	0	0	0	0	0	0			50	None
41	Oppose	S	0	0	0	0	1	0	0	0			20	None
42	Support	S	1	0	0	0	0	0	0	0			50	none
43	Support	S	1	0	0	0	0	1	0	0			10	go turbines
44	Oppose	S	0	0	0	0	0	1	0	0			20	I think wind turbines should mainly be used away from residential areas: offshore, mountains, plains.
45	Support	S	1	0	0	0	0	0	0	0			0	none
46	Support	O												no
47	Support	S	0	0	0	0	1	0	0	0			10	No opposition.
48	Support	S	1	0	0	0	0	1	0	0			2	None
49	Support	S	0	0	0	0	0	0	0	1			20	no
50	Support	S	0	0	1	0	1	0	0	0			100	none
51	Support	S	0	0	0	1	0	0	0	0			10	FOR IT
52	Support	S	1	0	0	0	0	0	0	0			1	no comment
53	Support	O												I dont
54	Support	O												no
55	Support	S	1	0	0	0	0	1	0	0			40	better than burning coal
56	Support	S	1	0	0	0	0	0	0	0			40	NO COMMENT
57	Oppose	S	0	0	0	1	0	0	0	0			55	Why
58	Support	S	0	0	0	1	0	0	0	0			27	Support
59	Support	S	1	0	0	0	0	0	0	0			20	None
60	Support	S	0	0	0	0	1	1	0	0			41	Cold weather and wind
61	Support	S	0	0	1	1	1	0	1	0			100	i would not mind viewing a turbine during any part of the day
62	Support	S	0	0	0	0	1	0	0	0			25	I love them
63	Support	S	0	0	0	0	0	0	0	0	1	Doing a hurricane	20	Only during a hurricane, it would be scary seeing them move fast because of the strong winds.
64	Support	S	1	0	0	0	0	0	0	0			0	we should be using then
65	Support	S	1	0	0	0	0	1	0	0			50	I worry about birds being killed from the turbines
66	Support	S	1	0	0	0	0	1	0	0			75	I really have no opposition. I think it's a great reusable energy source.
67	Support	S	0	0	0	0	0	0	1	0			40	none
68	Support	S	0	0	0	0	0	1	0	0			100	Don't oppose
69	Oppose	O												they all still run on oil so what is the point
70	Oppose	S	0	0	0	0	0	0	0	1			40	I have no comment
71	Support	S	1	0	0	0	0	1	0	0			1	no opposition
72	Oppose	S	0	0	0	0	1	0	0	0			20	no opion
73	Support	S	0	0	1	0	0	0	0	0			10	None
74	Support	S	1	0	0	0	0	1	0	0			0.8	Nope

Continued on next page

Table 25: (continued)

No.	Q1	Q2	C1	C2	C3	C4	C5	C6	C7	C8	C9	Q3.Choice9 [value]	Q4	Q5
75	Support	S	1	0	0	0	0	0	0	0			50	go you
76	Support	S	1	0	0	0	0	0	0	0			5	None
77	Support	S	1	0	0	0	0	0	0	0			50	I have none.
78	Support	O												none
79	Oppose	O												I don't know
80	Support	S	0	1	0	0	0	1	0	1			89	no comment
81	Support	O												guess I don't
82	Support	S	0	0	0	1	0	0	0	0			50	none
83	Support	S	0	1	0	0	0	0	0	0			0	i love them good
84	Support	S	1	0	0	0	0	1	0	1			85	I'm not super familiar with facts on wind turbines, but if it brings about a safer, more efficient/environmentally friendly energy process, I'm all for it.
85	Oppose	S	0	0	0	0	1	0	0	0			10	no
86	Oppose	O												I opose because I am in aware of what it is and I don't agree to nothing without proper knowledge
87	Support	S	1	0	0	0	0	0	0	0			1	none
88	Support	S	1	0	0	0	0	1	0	0			20	None
89	Support	O												why have them up spend the time money & effort if retractable?
90	Support	S	1	0	0	0	0	1	0	0			100	We need all the alternative power we can get.
91	Oppose	O												no
92	Support	S	1	0	0	0	0	1	0	0			5	I like them, I think they're attractive and environment friendly.
93	Support	S	1	0	0	0	0	0	0	0			3	If they would save on electric bills I'm for it.
94	Support	S	1	0	0	0	0	1	0	0	1	During a storm	0	no comment
95	Oppose	O												I worked in the environmental field and i have cleaned uo oil that have leaked from them.
96	Support	S	1	0	0	0	0	0	0	0			0	no comments
97	Support	S	0	1	0	0	0	0	0	0			1	no
98	Support	S	0	0	0	0	1	0	0	0			60	no comment
99	Support	S	1	0	0	0	0	1	0	0			99	I support renewable energy sources... wind, solar, tidal, geothermal.
100	Support	S	1	0	0	0	0	1	0	0			15	no, I am cool
101	Support	S	1	0	0	0	0	1	0	0			80	None.
102	Support	S	1	0	0	0	0	0	0	0			0	Support
103	Support	O												Nope

Continued on next page

Table 25: (continued)

No.	Q1	Q2	C1	C2	C3	C4	C5	C6	C7	C8	C9	Q3.Choice9 [value]	Q4	Q5
104	Support	S	1	0	0	0	0	0	0	0			1	none at this time thank you
105	Support	S	1	0	0	0	0	0	0	0			0	I do not oppose.
106	Support	S	0	1	0	0	0	1	0	0			1	I do not oppose
107	Support	S	0	0	0	0	0	0	0	0	1	No need to retract them, we need wind turbines 24/7	100	I have no opposition against any sort of wind turbines. We should use them en masse every day.
108	Oppose	O												no comment
109	Oppose	O												nothing to share
110	Support	S	1	0	0	0	0	1	0	0			50	I do not oppose wind turbines
111	Support	S	1	0	0	0	0	0	0	0			20	None
112	Support	S	0	0	0	0	0	0	0	0	1	No idea	10	Don't know much about turbines
113	Support	S	0	0	0	0	0	0	0	0	1	never	0	we should always use the turbines, they should never be hidden
114	Oppose	O												Wind sucks
115	Support	S	1	0	0	0	0	0	0	0			10	I'm for reusable energy
116	Support	S	1	0	0	0	0	1	0	0			15	I live in an urban area so this isn't really possible for me.
117	Oppose	S	0	0	0	0	1	0	0	0			20	No comment
118	Support	S	1	0	0	0	0	0	0	0			30	I support wind turbines
119	Support	S	1	0	0	0	0	0	0	0			70	Sounds great!
120	Support	S	0	0	0	0	0	0	0	0	1	not sure	15	no comment
121	Support	S	0	0	0	0	0	0	0	0	1	whenever the turbine doesn't need to be in use	20	Don't feel one way or the other. If turbines create energy at a low cost, wouldn't matter if I could see it or not.
122	Support	O												no
123	Support	S	1	0	0	0	0	0	0	0			20	I don't oppose them
124	Oppose	O												I don't support one industry destroying another. I do think we always need a backup to any power source. If we put all our eggs in one basket, we are setting ourselves up to be destroyed.
125	Support	S	1	0	0	0	0	1	0	0			10	no comment.
126	Support	S	1	0	0	0	0	0	0	0			15	I love turbines
127	Support	S	0	0	0	0	0	1	0	0			0	none
128	Support	S	1	0	0	0	0	1	0	0			0	no comments
129	Oppose	O												Solar is better
130	Support	S	0	0	0	0	0	1	0	0			0	no
131	Support	S	1	0	0	0	0	0	0	0			1	no comment

Continued on next page

Table 25: (continued)

No.	Q1	Q2	C1	C2	C3	C4	C5	C6	C7	C8	C9	Q3.Choice9 [value]	Q4	Q5
132	Support	S	1	0	0	0	0	1	0	0			60	I don't really have a problem with visible turbines, since I'm a proponent of renewable energy.
133	Support	S	1	0	0	0	0	1	0	0			45	wind turbines lower pollution
134	Support	S	1	0	0	0	0	0	0	0			100	I feel they should be every where because they are beneficial.
135	Support	S	1	0	0	0	0	1	0	0			50	No comment
136	Support	S	0	0	1	0	0	1	0	0			50	No
137	Support	S	0	0	0	0	0	0	0	0	1	weather	1	do not have one
138	Support	S	0	0	0	0	1	1	0	0			50	no comments
139	Support	S	1	0	0	0	0	1	0	0			1	satisfied with my answers
140	Support	S	0	0	0	0	0	0	0	1			50	None
141	Oppose	O												no myy concern
142	Support	O												no
143	Support	S	0	0	0	0	0	1	0	0			30	no comments
144	Support	S	1	0	0	0	0	0	0	0			20	No Comments
145	Support	S	0	0	0	0	0	1	0	0			10	n0ne
146	Support	S	0	0	0	0	0	1	0	0			3	no comments
147	Support	S	0	0	0	0	0	1	0	0			1	none
148	Oppose	S	0	1	0	0	0	0	0	0			7	No
149	Support	S	1	0	0	0	0	0	0	0			0	no comment
150	Support	S	0	0	0	0	0	1	0	0			50	I don't know
151	Support	S	0	0	0	0	1	0	0	0			83	nah
152	Support	S	1	0	0	0	1	0	0	0			3	no comments
153	Support	S	0	0	0	0	0	0	0	0	1	who cares, turbines are good	100	nope
154	Support	S	1	0	0	0	0	0	0	0			100	all renewable is great
155	Support	S	1	0	0	0	0	0	0	0			10	none
156	Support	O												None.
157	Support	S	1	0	0	0	1	1	0	0			60	none
158	Oppose	S	0	0	0	0	0	1	0	0			20	too noisy
159	Support	S	1	0	0	0	0	1	0	0			70	don't oppose
160	Support	S	1	0	0	0	0	1	0	0			75	none
161	Oppose	O												don't like them
162	Support	S	1	0	0	0	0	1	0	0			75	Don't mind them
163	Oppose	S	1	0	0	0	0	0	0	0			2	HAVE NONE
164	Support	S	1	0	0	0	0	0	0	0			10	None
165	Support	O												No
166	Support	S	0	0	0	0	0	0	1	0			80	No
167	Support	O												no comment
168	Support	S	1	0	0	0	0	0	0	0			85	no
169	Support	S	0	0	0	0	0	1	0	0			10	none
170	Support	S	0	0	0	0	0	1	0	0			10	NO

Continued on next page

Table 25: (continued)

No.	Q1	Q2	C1	C2	C3	C4	C5	C6	C7	C8	C9	Q3.Choice9 [value]	Q4	Q5
171	Support	S	1	0	0	0	0	0	0	0			50	no
172	Support	S	1	0	0	0	0	0	0	0			90	no
173	Support	S	0	0	0	0	0	1	0	0			50	none
174	Oppose	O												eyesore
175	Support	S	0	0	0	0	0	1	0	0			5	NONE
176	Support	S	1	0	0	0	0	0	0	0			75	no comment
177	Support	S	1	0	0	0	0	0	0	0			0	No comment.
178	Support	S	0	0	0	0	0	1	0	0			23	nope
179	Support	S	1	0	0	0	0	0	0	0			50	none
180	Support	S	1	0	0	0	0	0	0	0			100	none
181	Support	S	1	0	0	0	0	1	0	0			75	It does not matter to me
182	Support	S	0	0	0	0	0	1	0	0			89	no comment
183	Support	O												I DO NOT oppose
184	Support	S	1	0	0	0	0	0	0	0			12	no
185	Support	S	1	0	0	0	0	0	0	0			0	im not for or against anythig as long as the electric company dont charge an arm and a leg for basicaly free evergy
186	Support	S	1	0	0	0	0	0	0	0			30	just think that they are a eye sore
187	Oppose	O												Because they impact wildlife severely (in my area)
188	Support	S	0	0	0	0	0	1	0	0			85	not really
189	Support	O												none
190	Oppose	O												Unsightly
191	Oppose	O												thy make the area look bad
192	Oppose	S	1	0	0	0	0	1	0	0			50	none
193	Support	S	0	0	0	0	0	0	0	1			95	none
194	Oppose	S	1	0	1	1	0	0	0	0			35	no commentent
195	Support	S	1	0	0	0	0	1	0	0			75	No comment
196	Support	S	1	0	0	0	0	0	0	0			5	none
197	Support	S	1	0	0	0	0	0	0	0			0	None
198	Support	S	1	0	0	0	0	0	0	0			0	I have no comments.
199	Support	S	1	0	0	0	0	1	0	0			25	None
200	Support	S	1	0	0	0	0	1	0	0			10	none
201	Support	S	1	0	0	0	0	0	0	0			100	None
202	Support	S	1	0	0	0	0	1	0	0			15	Dont oppose. We need to move forward with other types of energy.
203	Support	S	1	0	0	0	0	0	0	0			9	no
204	Support	S	1	0	0	0	1	1	0	0			20	I'm all for green energy
205	Oppose	S	1	0	0	0	0	0	0	0			2	no comments
206	Support	S	1	0	0	0	0	0	0	0			0	no comment
207	Oppose	S	1	0	0	0	0	1	0	0			50	no additional comment
208	Oppose	O												None
209	Support	S	1	0	0	0	0	0	0	0			1	no comments

Continued on next page

Table 25: (continued)

No.	Q1	Q2	C1	C2	C3	C4	C5	C6	C7	C8	C9	Q3.Choice9 [value]	Q4	Q5
210	Support	S	1	0	0	0	0	1	0	0			99	More turbines!
211	Support	S	0	0	0	0	0	0	0	0	1	don't know	75	none
212	Support	S	0	0	0	0	0	0	0	1			50	no
213	Support	S	0	0	0	0	0	1	0	0			1	none
214	Support	S	1	0	0	0	0	1	0	0			25	They're okay, but not the answer to energy needs
215	Oppose	O												Wind turbines are not as affective as they should be.
216	Support	S	0	0	0	0	0	1	0	0			10	nq comment
217	Support	S	1	0	0	0	0	0	0	0			0	i have no opposition
218	Support	S	0	0	0	0	1	0	0	0			25	none
219	Support	S	0	0	0	0	1	0	0	0			2	no comment
220	Support	S	0	0	0	0	1	0	0	0			50	none
221	Support	S	1	0	0	0	0	0	0	0			1	none
222	Support	S	0	0	1	0	0	0	0	0			1	Go Birds
223	Support	S	1	0	0	0	0	0	0	0			21	none
224	Support	S	0	0	1	0	0	0	0	0			75	help farmers
225	Support	S	0	0	0	0	0	1	1	0			50	nope
226	Support	S	0	0	0	0	0	1	0	0			34	no
227	Support	S	0	0	0	0	1	0	0	0			40	none
228	Support	S	0	1	0	0	0	0	0	0			6	no0
229	Support	O												No
230	Support	S	0	0	0	0	0	1	1	1			55	no
231	Support	S	0	0	0	0	0	1	0	0			60	no
232	Support	S	1	0	0	0	0	0	0	0			0	none
233	Oppose	O												none
234	Oppose	O												No
235	Support	S	1	0	0	0	0	1	0	0			5	I have no opposition to wind turbines.
236	Support	S	0	0	0	1	0	0	0	0			10	dont know
237	Oppose	O												Better options
238	Support	S	1	0	0	0	0	0	0	0			0	I think they look awsome
239	Oppose	O												no comment
240	Oppose	S	0	0	1	0	0	0	0	0			75	I don't care
241	Support	S	1	0	0	0	0	0	0	0			1	I like the look of windmills.
242	Oppose	O												none
243	Oppose	S	0	0	0	0	0	0	0	1			10	none
244	Oppose	O												noo comment
245	Support	S	0	0	0	0	0	1	0	0			40	No
246	Support	S	1	0	0	0	0	1	0	0			100	We need wind turbines.
247	Support	S	1	0	0	0	0	0	0	0			80	I don't mind turbines... they remind me of renewable energy
248	Support	S	0	0	0	0	0	1	0	0			75	i don't know much about this but coooool
249	Oppose	O												none
250	Support	S	0	0	0	0	0	1	0	0			100	none

Continued on next page

Table 25: (continued)

No.	Q1	Q2	C1	C2	C3	C4	C5	C6	C7	C8	C9	Q3.Choice9 [value]	Q4	Q5
251	Support	S	1	0	0	0	0	0	0	0			15	I feel they are a good way to get energy
252	Oppose	S	0	0	0	0	1	1	0	0			5	save the environment you pricks
253	Support	O												NONE
254	Oppose	O												NONE
255	Support	S	1	0	0	0	0	1	0	0			100	Wind is good, install turbines!
256	Oppose	S	1	0	0	0	0	0	0	0			20	Nah
257	Oppose	S	0	0	0	0	0	0	1	0			2	domt knoe
258	Support	S	0	0	0	0	0	0	0	1			7	no comment
259	Oppose	O												i just don't understand their purpose .
260	Support	S	0	0	0	0	1	0	0	0			25	no
261	Support	S	1	0	0	0	1	0	0	0			23	none.
262	Support	S	0	0	0	0	0	1	0	0			24	no
263	Support	S	0	0	0	0	1	0	0	0			100	none
264	Support	O												To close to home
265	Support	S	1	0	0	0	0	0	0	0			10	No comment
266	Support	S	1	0	0	0	0	0	0	0			100	None
267	Oppose	O												I have no comment.
268	Support	S	1	0	0	0	0	1	0	0			3	similar to windmills could be a good thing
269	Support	S	1	0	0	0	0	1	0	0			25	None
270	Support	S	0	0	0	0	0	1	0	0			0	I do not oppose wind turbines.
271	Support	O												interference
272	Support	S	0	0	0	1	0	0	0	0			52	none
273	Support	S	0	0	0	1	0	1	1	0			30	I do not oppose wind turbines.
274	Support	S	0	0	0	0	0	0	0	0	1	it doesn't bother me either way	100	clean energy is the future....it has to be
275	Support	S	1	0	0	0	0	0	0	0			0	I Love Wind Turbines
276	Oppose	O												i just do
277	Oppose	S	1	0	0	0	0	0	0	0			50	I have worked OEM wind projects
278	Support	S	1	0	0	0	0	1	0	0			25	no
279	Support	S	1	0	0	0	0	0	1	0			50	No coments
280	Support	S	1	0	0	0	0	0	0	0			100	none
281	Support	S	1	0	0	0	0	0	0	0			5	I do not oppose wind turbines
282	Support	S	1	0	0	0	0	0	0	0			0	NONE
283	Support	S	0	0	0	0	0	1	0	0			10	Don't have any
284	Support	S	1	0	0	0	0	1	0	0			0	If it saves money, then I have no problem with it.
285	Support	S	1	0	0	0	0	0	0	0			1	nope
286	Support	S	1	0	0	0	0	0	0	0			90	None
287	Oppose	O												none
288	Oppose	O												because money

Continued on next page

Table 25: (continued)

No.	Q1	Q2	C1	C2	C3	C4	C5	C6	C7	C8	C9	Q3.Choice9 [value]	Q4	Q5
289	Oppose	O												They produce low-frequency wave sound that makes people depressed if they are exposed over a longer period.
290	Support	S	0	0	0	0	0	0	0	0	1	doesn't matter	100	I don't oppose any forms of turbines
291	Oppose	S	1	0	0	0	0	0	1	0			15	No
292	Support	O												Do it
293	Oppose	S	0	0	0	1	1	0	1	0			50	no
294	Oppose	S	0	0	1	1	1	0	0	0			80	none
295	Support	S	1	0	0	0	0	0	0	0			20	no comment
296	Support	S	0	0	0	1	0	1	0	0			60	I think it would be a good use of power, even though we don't have much wind here
297	Oppose	O												No comment
298	Oppose	O												no
299	Support	S	1	0	1	1	1	0	0	0			25	None
300	Oppose	O												ugle
301	Support	S	1	0	0	0	0	0	0	0			10	no comment
302	Support	S	1	0	0	0	0	1	0	0			75	The turbines are a good idea.
303	Support	S	1	0	0	0	0	1	0	0			75	I don't know why people would oppose such a wonderful tool for resources.
304	Oppose	S	1	0	0	0	0	0	0	0			1	no

A.10 DERIVED QUIET HOURS FOR EACH MUNICIPALITY

With each of the thirty weather stations, we associated a municipality. Each of those municipalities have legal codes that prohibit certain noise-making activities during certain hours. From those prohibitions, we derived quiet hours for each municipality to be used by control algorithms satisfying the standard OLAs derived herein that prohibit wind-energy harvesting at night. Those derived quiet hours are shown in Table 26.

Table 26: Derived Quiet Hours

Station	Municipal Code Text	Derived Quiet Hours	Reference	URL
KATL	"Section 74-134. - Construction noise. "Construction noise that does not otherwise qualify under any of the exclusions listed in 74-132 shall comply with the following: between the hours of 7:00 p.m. and 7:00 a.m. the following day on a weekday or between 7:00 p.m. and 9:00 a.m. on a weekend day or legal holiday, construction noise of any type including but not limited to, noise caused by the erection (including excavation), demolition, alteration, or repair of any building, as well as the operation of any earth-moving equipment, crane, saw, drill, pile driver, steam shovel, pneumatic hammer, hoist, automatic nailer or stapler, or any similar equipment, shall not be plainly audible within any residential zoning district more than 100 feet beyond the property boundary of the property from which the noise emanates; provided that between the hours of 7:00 a.m. (or 9:00 a.m. on a weekend day or legal holiday) and 7:00 p.m., the prohibitions of this article shall not apply to construction noise."	7 p.m. and 7 a.m. the following day on a weekday or between 7 p.m. and 9 a.m. on a weekend day or federal holiday[7]	Atlanta, GA, Municipal Code §74-134	https://library.municode.com/ga/atlanta/codes/code_of_ordinances
KBOS	"No erection, demolition, alteration, or repair of any building and excavation in regard thereto, except between the hours of 7:00 a.m. and 6:00 p.m., on weekdays or except in the interest of public safety or welfare, upon the issuance of and pursuant to an Off Hours Permit from the Commissioner, Inspectional Services Department, which may be renewed for one or more periods of not exceeding one week each. Any person violating this section hereof shall be punished by a fine of Three Hundred Dollars ($300.00) for each offense. All fines issued under this section may be recovered by the noncriminal disposition procedures promulgated in G.L. c. 40, s. 21D. Each day shall constitute a new offense."	6 p.m. to 7 a.m. and Saturday and Sunday	Boston, MA, Municipal Code §16-26.4	http://library.amlegal.com/nxt/gateway.dll/Massachusetts/boston/chapterxviprohibitionspenaltiesandpermit?f=templates$fn=default.htm$3.0$vid=amlegal:boston_ma$anc=JD_16-26
KBWI	"(b) Home activities —vehicle repairs; power tools. "Noncommercial vehicular repairs and the use of home workshops, power tools, and power garden equipment are allowed: "(1) between the hours of 7 a.m. and 9 p.m. on weekdays; and "(2) between the hours of 10 a.m. and 10 p.m. on weekends and legal holidays."	Nighttime starts: 9 p.m. weekdays and 10 p.m. on weekends and federal holidays.[7] Nighttime ends: 7 a.m. weekdays and 10 a.m. weekends and federal holidays.[7]	Baltimore (MD) City Revised Code §9-208	http://ca.baltimorecity.gov/codes/Art%2000%20-%20Health.pdf
KCLE	"The making of any unnecessary noise by a person or by his or her operation of any instrument, device, agency or vehicle, and/or the performance of any construction or demolition activity or the operation of any mechanical, electrical, pneumatically, hydraulically powered or battery operated apparatus used in connection with any construction or demolition activity between the hours of 7:00 p.m. and 7:00 a.m."	7 p.m. to 7 a.m.	Cleveland, OH, Municipal Code §605.10	http://library.amlegal.com/nxt/gateway.dll/Ohio/cleveland_oh/partsixoffensesandbusinessactivitiescode/titleigeneraloffenses/chapter605-disorderlyconductandactivity?f=templates$fn=default.htm$3.0$vid=amlegal:cleveland_oh$anc=JD_605.10

Table 26: (continued)

KCLT	"(a) It shall be unlawful to carry on the following activities in any residentially zoned area of the city or within 300 feet of any residentially occupied structure in any zone of the city: . . . "(2) The operation of construction machinery between the hours of 9:00 p.m. and 7:00 a.m."	9 p.m. to 7 a.m.	Charlotte, NC, Municipal Code §15-63	charlottenc.gov/CMPD/Documents/Ordinances/Noise_Ordinance.pdf
KCVG	"No person other than in the event of emergency, shall, between the hours from 9:00 p.m. to 7:00 a.m. the following day, engage in or undertake any construction or demolition activity or the operation of any mechanical, electrical or battery-operated apparatus which produces loud sound which disturbs the peace and quiet of the neighborhood within 500 feet of places of residence, hospitals or other residential institutions, without first obtaining a special permit from the director of buildings and inspections or the city engineer for such nighttime construction. For purposes of this section, construction shall include every operation regulated by the Cincinnati - Ohio Basic Building Code."	9 p.m. to 7 a.m.	Cincinatii, OH, Municipal Code §909-7	https://library.municode.com/oh/cincinnati/codes/code_of_ordinances?nodeId=TITIXMI_CH909CONO_S909-3LONO
KDCA	"2803.2 No noise from construction, excluding minor home repairs, shall be permitted within a residential, special purpose, or waterfront zone on any Sunday or legal holiday, or after 7:00 p.m. and before 7:00 a.m. on any weekday."	All day Sunday or federal holiday.[7] Other days: 7 p.m. to 7 a.m.	Washington, DC, Municipal Code §2803.2	https://dcregs.dc.gov
KDEN	"(5) Construction equipment and activities. "a. No person shall operate, or cause to be operated, any construction equipment nor conduct any construction activities, including, without limitation, preparation activities, job site deliveries, and job site pick-ups, on weekdays between the hours of 9:00 p.m. and 7:00 a.m. in a manner that exceeds the sound pressure limits of Table A. "b. No person shall operate, or cause to be operated, any construction equipment nor conduct any construction activities, including, without limitation, preparation activity, job-site deliveries, and job-site pick-ups, on weekends between the hours of 5:00 p.m. and 8:00 a.m. in a manner that exceeds the sound pressure limits of Table A."	Nighttime starts weekdays: 9 p.m. Nighttime starts weekends: 5 p.m. Nightime ends weekdays: 7 a.m. Nighttime ends weekends: 8 a.m.	Denver, CO, Municipal Code §36-7	https://www.denvergov.org/content/dam/denvergov/Portals/771/documents/PHI/HFHH/Chapter-36-Noise-Control.pdf
KDFW	"(8) Any construction activity related to the erection, excavation, demolition, alteration, or repair of any building on or adjacent to a residential use, as defined in the Dallas Development Code, other than between the hours of 7:00 a.m. and 7:00 p.m., Monday through Friday, and between the hours of 8:00 a.m. and 7:00 p.m. on Saturdays and legal holidays, except that the director of public works may issue a written permit to exceed these hours in the case of urgent necessity in the interest of public safety or for other reasons determined by the director of public works to be necessary for the public health, safety, or welfare. For purposes of this paragraph, 'legal holidays' include New Year's Day (January 1), Memorial Day (observed date), Fourth of July (July 4), Labor Day (observed date), Thanksgiving Day (observed date), and Christmas Day (December 25)."	Nighttime starts 7 p.m. Nighttime ends 8 a.m. on Saturdays and federal holidays[7] and ends 7 a.m. on other days.	Dallas, TX, Municipal Code §30-2	http://dallas-tx.elaws.us/code/coor_appsid836964_ch30_sec30-2
KDTW	"Truck driving schools shall not be open or operated on Sunday, or between the hours of 9:00 p.m. and 7:00 a.m. on any other day"	All day Sunday. 9 p.m. to 7 a.m. other days.	Detroit, MI, Municipal Code §36-1-4	https://library.municode.com/mi/detroit/codes/code_of_ordinances?nodeId=PTIIICICO_CH36NO_S36-1-4REHOTRDRSC

Table 26: (continued)

KEUG	"d. Construction or repair of buildings, streets, etc. Constructing (including excavating), demolishing, altering, or repairing any building, street, sidewalk, driveway, sewer or utility line between the hours of 7 p.m. and 7 a.m. the following day."	7 p.m. to 7 a.m.	Eugene, OR, Municipal Code §6.750-d	https://www.eugene-or.gov/DocumentCenter/View/2697/Chapter-6-Environment-and-Health
KIAH	"(5) "The sound was produced by the erection, excavation, construction, or demolition of any building or structure, including the use of any necessary tools or equipment, conducted between the hours of 7 a.m. and 8 p.m., which activity did not produce a sound exceeding 85 dB(A) when measured from the property line of the residential property where the sound is being received. "	8 p.m. to 7 a.m.	Houston, TX, Municipal Code §Section 30-16-5	https://library.municode.com/tx/houston/codes/code_of_ordinances?nodeId=COOR_CH30NOSOLERE
KLAS	"(H) The erection, including the excavation, demolition, alteration or repair of any building in any new or existing residential district, or the excavation, construction or repair of any right-of-way improvements in any new or existing residential district other than between the hours of seven a.m. and six p.m., except in the case of urgent necessity in the interest of public health and safety and then only with a permit from the designated official."	6 p.m. to 7 a.m.	Las Vegas, Nevada, Municipal Code §9.16.030-H	https://library.municode.com/nv/las_vegas/codes/code_of_ordinances?nodeId=TIT9HESA_CH9.16NO_9.16.030ENPRNO
KLAX	"(a) No person shall, between the hours of 9:00 P.M. and 7:00 A.M. of the following day, perform any construction or repair work of any kind upon, or any excavating for, any building or structure, where any of the foregoing entails the use of any power driven drill, riveting machine excavator or any other machine, tool, device or equipment which makes loud noises to the disturbance of persons occupying sleeping quarters in any dwelling hotel or apartment or other place of residence."	9 p.m. to 7 a.m.	Los Angeles, CA, Municipal Code, Section 41.40-(a)	http://library.amlegal.com/nxt/gateway.dll/California/lamc/municipalcode?f=templates$fn=default.htm$3.0$vid=amlegal:losangeles_ca_mc
KLGA	"Except as otherwise provided in this subchapter, it shall be unlawful to engage in or to cause or permit any person to engage in construction work other than on weekdays between the hours of 7 a.m. and 6 p.m. A person may however perform construction work in connection with the alteration or repair of an existing one or two family owner-occupied dwelling classified in occupancy group J-3 or a convent or rectory on Saturdays and Sundays between the hours of 10 a.m. and 4 p.m. provided that such dwelling is located more than 300 feet from a house of worship."	Weekdays: Nighttime starts at 6 p.m and nighttime ends at 7 p.m. Weekends: Nighttime ends 10 a.m. and starts 4 p.m.	New York City Administrative Code §24-222	http://library.amlegal.com/nxt/gateway.dll/New%20York/admin/newyorkcityadministrativecode?f=templates$fn=default.htm$3.0$vid=amlegal:newyork_ny

Table 26: (continued)

KMCI	“Section 46-5. - Specific prohibitions. “Domestic power tools. No person shall operate or permit the operation of any mechanically or electrically-powered saw, drill, sander, grinder, lawn or garden tool, or similar device used in a residential district between the hours of 10:00 p.m. and 7:00 a.m. on a residential property or within 250 feet of a residential real property boundary (when operated on commercial or light-industrial property), unless such activities do not exceed the limits set forth in Table I. ” “Section 46-26. - Exceptions to the noise code. “The provisions of the noise code shall not apply to: “(1) The emission of sound for the purpose of alerting persons to the existence of an emergency; “(2) The emission of sound in the performance of emergency work; “(3) Construction work, as defined in 46-3;”	10 p.m. to 7 a.m.	Kansas City, MO, Municipal Code §46-5	https://library.municode.com/mo/kansas_city/codes/code_of_ordinances?nodeId=COORKAMIVOII_CH46NOCO_ARTIINGE_S46-3DE
KMCO	“The following types of noise are hereby made exempt from the provisions of this Chapter: “(1) Noises from temporary construction and maintenance activities between 7 AM and 9 PM except Sundays and federal holidays.”	9 p.m. to 7 a.m. and Sundays and federal holidays.[7]	Orlando, FL, Municipal Code, §42.06	https://library.municode.com/fl/orlando/codes/code_of_ordinances?nodeId=TITIICICO_CH42NO
KMSP	“(a) Operation of construction equipment without a permit is allowed only on Monday through Friday from 7:00 a.m. to 6:00 p.m., not including federal holidays. Operation of construction equipment outside of these hours and days without a permit from the Minneapolis Health Department and/or their designee is prohibited.”	6 p.m. to 7 a.m. and Saturdays and Sundays and federal holidays[7]	Minneapolis, MN, Municipal Code §59.30	https://library.municode.com/mn/minneapolis/codes/code_of_ordinances?nodeId=COOR_TIT3AIPOENPR_CH59COAC_59.30AFHOWOPERE
KORD	“(a) No person shall use or cause the use of any mechanical equipment or tool operated by fuel or electric power in building, construction, repair or demolition operations between the hours of 8:00 P.M. and 8:00 A.M. within 600 feet of any residential building or hospital.”	8 p.m. to 8 a.m.	Chicago, IL, Municipal Code §8-32-140	library.amlegal.com/nxt/gateway.dll/Illinois/chicago_il/municipalcodeofchicago?f=templates$fn=default.htm$3.0$vid=amlegal:chicago_il
KPHL	“(11) Regular Construction. Construction between the hours of 7 a.m. and 8 p.m., Monday through Friday, or between the hours of 8 a.m. and 8 p.m. on weekends and legal, national or state holidays.”	Weekdays: Nighttime starts at 8 p.m. Nighttime ends at 7 a.m. Weekends and federal holidays[7]: Nighttime ends 8 a.m. Nighttime starts 8 p.m.	Philadelphia, Pennsylvania, The Philadelphia Code §10-402	http://library.amlegal.com/nxt/gateway.dll/Pennsylvania/philadelphia_pa/thephiladelphiacode?f=templates$fn=default.htm$3.0$vid=amlegal:philadelphia_pa

[7]For the purposes of this benchmark suite, we are defining *federal holidays* as “New Year’s Day (January 1), [Martin Luther King Jr. Day (observed date)], Memorial Day (observed date), Fourth of July (July 4), Labor Day (observed date), Thanksgiving Day (observed date), and Christmas Day (December 25)” [Dallas, TX, Municipal Code, §30-2-8].

Table 26: (continued)

KPHX	“(h) Building construction. Construction including erection, excavation, demolition, alteration or repair of any building within 500 feet of any inhabited structure, other than between the hours of 6:00 a.m. and 7:00 p.m. from May 1 to and including September 30 and between the hours of 7:00 a.m. and 7:00 p.m. beginning October 1 to and including April 30 on non-holiday weekdays. Except that the Planning and Development Director may grant a permit to conduct such operations outside listed hours, on weekends, or on holidays:”	7 p.m. to 7 a.m. and weekends and federal holidays[7]	Phoenix, Arizona, Municipal Code §23-14	https://www.codepublishing.com/AZ/Phoenix/
KPIT	“(f) Exemptions . A person shall be exempt from this section provided that:” . . . “(4) The noise emanates from lawn care and other household maintenance equipment and tools used between 7:00 a.m. and 10:00 p.m.; or “(5) The noise is being generated by construction activities properly permitted in the City of Pittsburgh; or”	10 p.m. to 7 a.m.	Pittsburgh, Pennsylvania, Municipal Code §601.04	https://library.municode.com/pa/pittsburgh/codes/code_of_ordinances?nodeId=COOR_TITSIXCO_ARTIRERIAC_CH601PUOR_S601.04NOCO
KSAC	“Notwithstanding any other provision of the chapter to the contrary, the following acts, among others, are declared to be loud, disturbing, and unnecessary noises in violation of this chapter, but such enumeration shall not be deemed to be exclusive, namely: . . . “ E. Tools. The use or operation between the hours of ten p.m. and seven a.m. of any power saw, power planer, or other powered tool or appliance or saw or hammer, or other tool, so as to disturb the quiet, comfort, or repose of persons in any dwelling, hotel, motel, apartment, or other type of residence, or of any person in the vicinity.”	10 p.m. to 7 a.m.	Sacramento, California, Muncipal Code §8.68.200	http://www.qcode.us/codes/sacramento/
KSAN	“(a) It shall be unlawful for any person, between the hours of 7:00 p.m. of any day and 7:00 a.m. of the following day, or on legal holidays as specified in Section 21.04 of the San Diego Municipal Code, with exception of Columbus Day and Washington’s Birthday, or on Sundays, to erect, construct, demolish, excavate for, alter or repair any building or structure in such a manner as to create disturbing, excessive or offensive noise unless a permit has been applied for and granted beforehand by the Noise Abatement and Control Administrator. . . . .”	7 p.m. to 7 a.m. and Sundays and federal holidays[7]	San Diego, CA, Municipal Code §59.5.0404	http://docs.sandiego.gov/municode/MuniCodeChapter05/Ch05Art9.5Division04.pdf

Table 26: (continued)

KSAT	"Section 21-51. - Definitions and standards. "*Daytime/evening* shall mean the hours between six o'clock a.m. and ten o'clock p.m., Sunday through Thursday and six o'clock a.m. and eleven o'clock p.m. Friday and Saturday. "Section 21-52. - Noise nuisance enumeration. "(a) The following acts, among others not hereinafter enumerated, are declared to be 'noise nuisances,' and are unlawful and in violation of the provisions of this division when such acts are done or accomplished or carried on in such a manner, or with such volume, intensity, or with continued duration, so as to annoy, to distress, or to disturb the quiet, comfort, or repose of a person of reasonable nervous sensibilities, within the vicinity or hearing thereof, or so as to endanger or injure the safety or health of humans or animals, or so as to interfere with the physical well being of humans or animals, or so as to endanger or injure personal or real property: . . . "6) The erection, including construction, excavation, demolition, alteration, or repair work, or the permitting or causing thereof, of any building or other structure, or the operation or the permitting or causing the operation of any tools or equipment used in construction, excavation, drilling, demolition, alteration or repair work: "a. Other than during the daytime on week days;"	10 p.m to 6 a.m. on Monday to Thursday and to 11 p.m. on Friday. Quiet hours include all-day Saturday and Sunday	San Antonio, Texas, Municipal Code, Article III (§21-51 and §21-52)	https://library.municode.com/tx/san_antonio/codes/code_of_ordinances
KSEA	"A. The exterior sound level limits established by Sections 25.08.410 and 25.08.420, as measured from the property line of the real property of another person or at a distance of 50 feet from the construction or maintenance equipment making the sound, whichever is greater, may be exceeded during the following times by the sound levels specified in subsection 25.08.425.B for the types of equipment listed in that subsection 25.05.425.B. "1. Within Lowrise, Midrise, Highrise, Residential-Commercial, and Neighborhood Commercial zones, between 7 a.m. and 7 p.m. on weekdays and between 9 a.m. and 7 p.m. on weekends and legal holidays, except that for parking lot maintenance or if the equipment is being used for a public project, then between 7 a.m. and 10 p.m. on weekdays and between the hours of 9 a.m. and 10 p.m. on weekends and legal holidays. "2. Within all other zones, between 7 a.m. and 10 p.m. on weekdays and between 9 a.m. and 10 p.m. on weekends and legal holidays.	Nighttime starts: 10 p.m. Nighttime begins: 7 a.m. on weekdays and 9 a.m. on weekends and federal holidays.[7]	Seattle, WA, Municipal Code §25.08.425	https://library.municode.com/wa/seattle/codes/municipal_code?nodeId=TIT25ENPRHIPR_CH25.08NOCO_SUBCHAPTER_IIDE_25.08.280PUNUNO
KSFO	"It shall be unlawful for any person, between the hours of 8:00 p.m. of any day and 7:00 a.m. of the following day to erect, construct, demolish, excavate for, alter or repair any building or structure if the noise level created thereby is in excess of the ambient noise level by 5 dBA at the nearest property plane, unless a special permit therefor has been applied for and granted by the Director of Public Works or the Director of Building Inspection."	8 p.m. to 7 a.m.	San Francisco, CA, Police Code, §2908	library.amlegal.com/nxt/gateway.dll?f=templates&fn=default.htm&vid=amlegal:sanfrancisco_ca

Table 26: (continued)

KSMX	"(e) Noise of construction caused by hand tools, power tools or equipment, when the noise occurs at a time other than: "(1) between the hours of 7:00 a.m. and 6:00 p.m., Monday through Friday; or "(2) between the hours of 8:00 a.m. and 5:00 p.m., Saturday through Sunday; or" "(3) allowed by permit issued by the Noise Control Officer. ..."	Nighttime begins: 6 p.m. Monday through Friday and 5 p.m. on Saturday and Sunday. Nighttime ends: 7 a.m. Monday through Friday and 8 a.m. on Saturday and Sunday.	Santa Maria, CA, Municipal Code, §5-5.06	http://www.qcode.us/codes/santamaria/
KSTL	"Construction, demolition and excavation within one thousand (1,000) feet of a residential property, including excavation, demolition, alteration or repair of any building, land clearing, land grading or road and utility construction within one thousand (1,000) feet of a residential property is prohibited before 6:00 a.m. and after dusk, Monday through Saturday, except in case of an urgent necessity in the interest of public safety for a period of three (3) days. After three (3) days the urgent necessity will be deemed to have elapsed unless a permit has been obtained from the Building Commissioner which allows specific action during any of the hours between 6:00 a.m. and dusk, Monday through Saturday."	sunset[8] (which is before dusk) to 6 a.m. and Sunday	St. Louis, MO, Municipal Code §15.50.081	https://library.municode.com/mo/st._louis/codes/code_of_ordinances?nodeId=TIT15PUPEMOWE_DIVIVOFAGPUPE_CH15.50NO
KTPA	"5-301.2.1 The generation of any avoidable or unreasonably loud, disturbing or unnecessary noise by construction activity on private property, other than between the hours of: (1) 7:00 a.m. and 6:00 p.m. Monday through Friday; (2) 8:00 a.m. and 6:00 p.m. on Saturday; or (3) 10:00 a.m. and 6:00 p.m. on Sunday is prohibited if such construction activity is within one thousand five hundred (1,500) feet of any building or portion thereof which is actually occupied and used either a single family or multi-family residence."	Nighttime Ends: 7 a.m. Monday - Friday. 8 a.m. Saturday. 10 a.m. Sunday. Nighttime Begins: 6 p.m. everyday.	Tampa, FL, Municipal Code §5-301.2.1	https://www.tampagov.net/sites/default/files/construction-services/files/Forms/I026ConstructionNoise.pdf

[8] With the benchmark suite described herein, we are supplying a .csv file having sunset times [5] for St. Louis from 2004 to 2014: STL_Sunset_Times.csv. All times in STL_Sunset_Times.csv are Central Standard Time.

A.11 RESULTS OF THE ALGORITHMS STATIC

A.11.1 Using current weather only

A.11.1.1 OLAs 1 & 2 Please see Tables 27 below and 28 on the next page.

Table 27: Results of the processing of OLAs 1 and 2 by Static (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.1

OLA	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm
1	KATL	7	31	0.57	0.91	0.51	KMCI	8	1	0.66	0.89	0.59
2	KATL	7	61	0.56	0.92	0.52	KMCI	8	31	0.66	0.93	0.62
1	KBOS	9	31	0.38	0.92	0.35	KMCO	7	1	0.66	0.82	0.55
2	KBOS	9	61	0.38	0.93	0.35	KMCO	7	61	0.65	0.92	0.60
1	KBWI	7	31	0.69	0.89	0.62	KMSP	8	31	0.38	0.92	0.35
2	KBWI	7	61	0.69	0.91	0.63	KMSP	8	91	0.37	0.93	0.35
1	KCLE	8	31	0.59	0.92	0.54	KORD	8	31	0.59	0.92	0.54
2	KCLE	8	61	0.59	0.93	0.54	KORD	8	91	0.58	0.94	0.55
1	KCLT	5	1	0.70	0.76	0.53	KPHL	8	31	0.65	0.92	0.60
2	KCLT	5	91	0.69	0.91	0.63	KPHL	8	61	0.65	0.93	0.60
1	KCVG	7	31	0.66	0.91	0.60	KPHX	6	31	0.36	0.80	0.29
2	KCVG	7	61	0.66	0.92	0.60	KPHX	6	121	0.35	0.87	0.30
1	KDCA	7	1	0.50	0.86	0.43	KPIT	7	1	0.74	0.81	0.60
2	KDCA	7	61	0.50	0.93	0.46	KPIT	7	61	0.73	0.92	0.67
1	KDEN	8	1	0.54	0.78	0.42	KSAC	6	1	0.78	0.78	0.61
2	KDEN	8	121	0.51	0.92	0.47	KSAC	6	91	0.77	0.93	0.72
1	KDFW	9	31	0.47	0.93	0.43	KSAN	5	1	0.72	0.87	0.63
2	KDFW	9	31	0.47	0.93	0.43	KSAN	5	61	0.71	0.92	0.66
1	KDTW	8	31	0.60	0.92	0.55	KSAT	7	1	0.53	0.83	0.44
2	KDTW	8	91	0.59	0.94	0.55	KSAT	7	121	0.52	0.94	0.49
1	KEUG	6	1	0.73	0.83	0.61	KSEA	6	1	0.66	0.81	0.53
2	KEUG	6	91	0.71	0.92	0.66	KSEA	6	121	0.65	0.93	0.60
1	KIAH	7	1	0.72	0.83	0.60	KSFO	10	1	0.68	0.90	0.61
2	KIAH	7	61	0.71	0.93	0.66	KSFO	10	61	0.66	0.94	0.63
1	KLAS	8	1	0.44	0.77	0.34	KSMX	8	1	0.80	0.86	0.69
2	KLAS	8	31	0.44	0.89	0.39	KSMX	8	61	0.78	0.92	0.71
1	KLAX	7	1	0.88	0.90	0.79	KSTL	7	1	0.53	0.84	0.44
2	KLAX	7	61	0.85	0.94	0.80	KSTL	7	91	0.52	0.93	0.48
1	KLGA	9	31	0.42	0.90	0.38	KTPA	6	1	0.66	0.77	0.51
2	KLGA	9	61	0.42	0.92	0.38	KTPA	6	91	0.64	0.91	0.58

Table 28: Average performance of the processing of OLAs 1 and 2 over all 30 weather stations by Static (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.1

OLA	Average	Metric
OLA 1	0.610	NetNorm
OLA 2	0.555	MQNetNorm

A.11.1.2 OLAs 3 & 4 Please see Table 29 below and Table 30 on the next page.

Table 29: Results of the processing of OLAs 3 and 4 by Static (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.1

OLA	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm
3	KATL	7	121	0.39	0.93	0.36	KMCI	8	61	0.32	0.94	0.30
4	KATL	7	121	0.39	0.93	0.36	KMCI	8	61	0.32	0.94	0.30
3	KBOS	9	91	0.34	0.94	0.32	KMCO	7	91	0.45	0.92	0.42
4	KBOS	9	91	0.34	0.94	0.32	KMCO	7	91	0.45	0.92	0.42
3	KBWI	7	91	0.50	0.92	0.45	KMSP	8	61	0.35	0.93	0.33
4	KBWI	7	61	0.49	0.91	0.45	KMSP	8	91	0.35	0.93	0.33
3	KCLE	8	121	0.34	0.92	0.32	KORD	8	121	0.32	0.93	0.30
4	KCLE	8	121	0.34	0.92	0.32	KORD	8	121	0.32	0.93	0.30
3	KCLT	5	121	0.38	0.91	0.35	KPHL	8	121	0.40	0.93	0.37
4	KCLT	5	121	0.38	0.91	0.35	KPHL	8	121	0.40	0.93	0.37
3	KCVG	7	121	0.34	0.92	0.31	KPHX	6	31	0.34	0.80	0.27
4	KCVG	7	121	0.34	0.92	0.31	KPHX	6	121	0.33	0.87	0.29
3	KDCA	7	121	0.37	0.94	0.35	KPIT	7	91	0.39	0.91	0.36
4	KDCA	7	121	0.37	0.94	0.35	KPIT	7	91	0.39	0.91	0.36
3	KDEN	8	61	0.34	0.90	0.31	KSAC	6	121	0.51	0.93	0.48
4	KDEN	8	121	0.34	0.91	0.31	KSAC	6	121	0.51	0.93	0.48
3	KDFW	9	61	0.32	0.93	0.30	KSAN	5	121	0.43	0.92	0.40
4	KDFW	9	91	0.32	0.93	0.30	KSAN	5	91	0.43	0.92	0.40
3	KDTW	8	91	0.41	0.93	0.38	KSAT	7	121	0.32	0.94	0.30
4	KDTW	8	91	0.41	0.93	0.38	KSAT	7	121	0.32	0.94	0.30
3	KEUG	6	121	0.49	0.92	0.45	KSEA	6	121	0.38	0.93	0.35
4	KEUG	6	121	0.49	0.92	0.45	KSEA	6	121	0.38	0.93	0.35
3	KIAH	7	121	0.41	0.93	0.38	KSFO	10	121	0.40	0.94	0.38
4	KIAH	7	121	0.41	0.93	0.38	KSFO	10	121	0.40	0.94	0.38
3	KLAS	8	31	0.40	0.89	0.35	KSMX	8	61	0.62	0.91	0.57
4	KLAS	8	91	0.40	0.92	0.37	KSMX	8	121	0.64	0.92	0.59
3	KLAX	7	121	0.45	0.93	0.42	KSTL	7	121	0.36	0.92	0.33
4	KLAX	7	121	0.45	0.93	0.42	KSTL	7	121	0.36	0.92	0.33
3	KLGA	9	91	0.35	0.92	0.33	KTPA	6	121	0.47	0.90	0.42
4	KLGA	9	121	0.36	0.92	0.33	KTPA	6	121	0.47	0.90	0.42

Table 30: Average performance of the processing of OLAs 3 and 4 over all 30 weather stations by Static (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.1

OLA	Average	Metric
OLA 3	0.397	NetNorm
OLA 4	0.367	MQNetNorm

A.11.1.3 OLAs 5 & 6 Please see Table 31 below and Table 32 below.

Table 31: Results of the processing of OLAs 5 and 6 by Static (variant 0x1, i.e., current weather only, transitions limited) Rev. 1.1

OLA	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm
5	KATL	17	31	0.27	0.95	0.25	KMCI	18	61	0.24	0.93	0.22
6	KATL	17	31	0.27	0.95	0.25	KMCI	18	61	0.24	0.93	0.22
5	KBOS	19	121	0.28	0.95	0.27	KMCO	17	31	0.26	0.98	0.26
6	KBOS	19	121	0.28	0.95	0.27	KMCO	17	31	0.26	0.98	0.26
5	KBWI	17	31	0.23	0.97	0.23	KMSP	18	31	0.25	0.90	0.23
6	KBWI	17	31	0.22	0.98	0.21	KMSP	18	31	0.25	0.90	0.23
5	KCLE	18	31	0.26	0.86	0.22	KORD	18	31	0.29	0.90	0.26
6	KCLE	18	31	0.26	0.86	0.22	KORD	18	31	0.27	0.90	0.24
5	KCLT	15	31	0.20	0.97	0.20	KPHL	18	31	0.28	0.95	0.27
6	KCLT	15	31	0.20	0.97	0.20	KPHL	18	31	0.28	0.95	0.27
5	KCVG	17	31	0.24	0.93	0.22	KPHX	16	31	0.27	0.96	0.26
6	KCVG	17	31	0.24	0.93	0.22	KPHX	16	31	0.27	0.96	0.26
5	KDCA	17	31	0.28	0.95	0.27	KPIT	17	31	0.22	0.95	0.21
6	KDCA	17	31	0.28	0.95	0.27	KPIT	17	31	0.20	0.96	0.19
5	KDEN	18	61	0.24	0.94	0.23	KSAC	16	31	0.30	0.96	0.29
6	KDEN	18	61	0.24	0.94	0.23	KSAC	16	31	0.30	0.96	0.28
5	KDFW	19	61	0.27	0.93	0.25	KSAN	15	1	0.27	0.97	0.26
6	KDFW	19	61	0.27	0.93	0.25	KSAN	15	1	0.27	0.97	0.26
5	KDTW	18	31	0.30	0.89	0.27	KSAT	17	91	0.26	0.95	0.24
6	KDTW	18	31	0.28	0.89	0.25	KSAT	17	91	0.26	0.95	0.24
5	KEUG	16	31	0.26	0.95	0.25	KSEA	16	31	0.36	0.95	0.34
6	KEUG	16	31	0.26	0.95	0.25	KSEA	16	31	0.36	0.95	0.34
5	KIAH	17	31	0.26	0.98	0.25	KSFO	20	91	0.24	0.93	0.22
6	KIAH	17	31	0.25	0.98	0.24	KSFO	20	91	0.23	0.93	0.21
5	KLAS	18	31	0.31	0.92	0.28	KSMX	18	61	0.30	0.98	0.30
6	KLAS	18	31	0.31	0.92	0.28	KSMX	18	61	0.30	0.98	0.29
5	KLAX	17	61	0.26	0.97	0.25	KSTL	17	31	0.27	0.93	0.25
6	KLAX	17	61	0.24	0.97	0.24	KSTL	17	31	0.27	0.93	0.25
5	KLGA	19	61	0.28	0.95	0.27	KTPA	16	31	0.24	0.98	0.24
6	KLGA	19	61	0.28	0.95	0.27	KTPA	16	31	0.24	0.98	0.24

Table 32: Average performance of the processing of OLAs 5 and 6 over all 30 weather stations by Static (variant 0x1, i.e., current weather only, transitions limited) Rev. 1.1

OLA	Average	Metric
OLA 5	0.267	NetNorm
OLA 6	0.247	MQNetNorm

A.11.2 Using weather prediction

A.11.2.1 OLAs 1 & 2 Please see Table 33 below and Table 34 on the next page.

Table 33: Results of the processing of OLAs 1 and 2 by Static (variant 0x2, i.e., weather prediction, transitions unlimited) Rev. 1.1

OLA	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm
1	KATL	7	1	0.57	0.80	0.45	KMCI	8	1	0.66	0.89	0.59
2	KATL	7	121	0.54	0.92	0.50	KMCI	8	1	0.66	0.89	0.59
1	KBOS	9	1	0.38	0.86	0.32	KMCO	7	1	0.66	0.82	0.55
2	KBOS	9	91	0.36	0.93	0.34	KMCO	7	61	0.63	0.91	0.57
1	KBWI	7	1	0.69	0.74	0.52	KMSP	8	1	0.38	0.84	0.32
2	KBWI	7	121	0.64	0.92	0.59	KMSP	8	121	0.35	0.93	0.33
1	KCLE	8	1	0.59	0.86	0.51	KORD	8	1	0.59	0.86	0.50
2	KCLE	8	121	0.57	0.93	0.53	KORD	8	121	0.57	0.93	0.54
1	KCLT	5	1	0.70	0.76	0.53	KPHL	8	1	0.65	0.85	0.55
2	KCLT	5	121	0.68	0.91	0.62	KPHL	8	121	0.62	0.93	0.57
1	KCVG	7	1	0.66	0.81	0.54	KPHX	6	1	0.36	0.54	0.20
2	KCVG	7	121	0.63	0.92	0.58	KPHX	6	121	0.34	0.86	0.29
1	KDCA	7	1	0.50	0.86	0.43	KPIT	7	1	0.74	0.81	0.60
2	KDCA	7	121	0.48	0.93	0.45	KPIT	7	121	0.69	0.92	0.64
1	KDEN	8	1	0.54	0.78	0.42	KSAC	6	1	0.78	0.78	0.61
2	KDEN	8	91	0.51	0.89	0.45	KSAC	6	121	0.74	0.93	0.69
1	KDFW	9	1	0.46	0.86	0.40	KSAN	5	1	0.72	0.87	0.63
2	KDFW	9	1	0.46	0.86	0.40	KSAN	5	121	0.70	0.93	0.65
1	KDTW	8	1	0.59	0.83	0.50	KSAT	7	1	0.53	0.83	0.44
2	KDTW	8	121	0.57	0.94	0.53	KSAT	7	121	0.51	0.94	0.48
1	KEUG	6	1	0.73	0.83	0.61	KSEA	6	1	0.66	0.81	0.53
2	KEUG	6	121	0.70	0.92	0.64	KSEA	6	121	0.63	0.92	0.58
1	KIAH	7	1	0.72	0.83	0.60	KSFO	10	1	0.68	0.90	0.61
2	KIAH	7	121	0.68	0.93	0.63	KSFO	10	91	0.63	0.95	0.60
1	KLAS	8	1	0.44	0.77	0.34	KSMX	8	1	0.80	0.86	0.69
2	KLAS	8	1	0.44	0.77	0.34	KSMX	8	91	0.76	0.92	0.70
1	KLAX	7	1	0.88	0.90	0.79	KSTL	7	1	0.53	0.84	0.44
2	KLAX	7	61	0.85	0.93	0.79	KSTL	7	61	0.51	0.91	0.47
1	KLGA	9	1	0.42	0.80	0.34	KTPA	6	1	0.66	0.77	0.51
2	KLGA	9	121	0.40	0.92	0.37	KTPA	6	121	0.63	0.90	0.57

Table 34: Average performance of the processing of OLAs 1 and 2 over all 30 weather stations by Static (variant 0x2, i.e., weather prediction, transitions unlimited) Rev. 1.1

OLA	Average	Metric
OLA 1	0.609	NetNorm
OLA 2	0.534	MQNetNorm

A.11.2.2 OLAs 3 & 4 Please see Table 35 below and Table 36 below.

Table 35: Results of the processing of OLAs 3 and 4 by Static (variant 0x2, i.e., weather prediction, transitions unlimited) Rev. 1.1

OLA	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm
3	KATL	7	121	0.35	0.91	0.32	KMCI	8	1	0.30	0.83	0.25
4	KATL	7	121	0.35	0.91	0.32	KMCI	8	121	0.27	0.93	0.25
3	KBOS	9	1	0.33	0.83	0.27	KMCO	7	121	0.43	0.92	0.39
4	KBOS	9	121	0.31	0.93	0.29	KMCO	7	121	0.43	0.92	0.39
3	KBWI	7	1	0.45	0.69	0.31	KMSP	8	1	0.33	0.80	0.27
4	KBWI	7	121	0.45	0.91	0.41	KMSP	8	121	0.31	0.93	0.29
3	KCLE	8	1	0.32	0.80	0.25	KORD	8	1	0.30	0.80	0.24
4	KCLE	8	121	0.30	0.92	0.28	KORD	8	121	0.29	0.92	0.26
3	KCLT	5	1	0.36	0.69	0.25	KPHL	8	1	0.38	0.80	0.31
4	KCLT	5	121	0.35	0.90	0.31	KPHL	8	121	0.37	0.92	0.34
3	KCVG	7	1	0.32	0.75	0.24	KPHX	6	91	0.32	0.84	0.26
4	KCVG	7	121	0.30	0.90	0.27	KPHX	6	121	0.31	0.85	0.27
3	KDCA	7	1	0.35	0.81	0.29	KPIT	7	1	0.37	0.76	0.28
4	KDCA	7	121	0.34	0.93	0.32	KPIT	7	121	0.35	0.91	0.32
3	KDEN	8	1	0.32	0.74	0.24	KSAC	6	1	0.49	0.75	0.36
4	KDEN	8	121	0.31	0.90	0.27	KSAC	6	121	0.47	0.92	0.44
3	KDFW	9	1	0.29	0.82	0.24	KSAN	5	121	0.43	0.92	0.40
4	KDFW	9	121	0.29	0.93	0.27	KSAN	5	121	0.43	0.92	0.40
3	KDTW	8	1	0.38	0.77	0.29	KSAT	7	121	0.32	0.93	0.30
4	KDTW	8	121	0.38	0.93	0.35	KSAT	7	121	0.32	0.93	0.30
3	KEUG	6	1	0.45	0.79	0.36	KSEA	6	1	0.36	0.77	0.28
4	KEUG	6	121	0.44	0.91	0.40	KSEA	6	121	0.35	0.91	0.32
3	KIAH	7	121	0.38	0.91	0.35	KSFO	10	1	0.38	0.86	0.32
4	KIAH	7	121	0.38	0.91	0.35	KSFO	10	121	0.38	0.94	0.35
3	KLAS	8	1	0.37	0.75	0.28	KSMX	8	1	0.57	0.82	0.47
4	KLAS	8	121	0.35	0.91	0.32	KSMX	8	121	0.59	0.91	0.54
3	KLAX	7	91	0.44	0.92	0.41	KSTL	7	1	0.33	0.79	0.26
4	KLAX	7	91	0.44	0.92	0.41	KSTL	7	121	0.33	0.91	0.30
3	KLGA	9	1	0.32	0.76	0.24	KTPA	6	121	0.43	0.89	0.39
4	KLGA	9	121	0.32	0.92	0.30	KTPA	6	121	0.43	0.89	0.39

Table 36: Average performance of the processing of OLAs 3 and 4 over all 30 weather stations by Static (variant 0x2, i.e., weather prediction, transitions unlimited) Rev. 1.1

OLA	Average	Metric
OLA 3	0.373	NetNorm
OLA 4	0.334	MQNetNorm

A.11.2.3 OLAs 5 & 6

Please see Table 37 below and Table 38 below.

Table 37: Results of the processing of OLAs 5 and 6 by Static (variant 0x3, i.e., weather prediction, transitions limited) Rev. 1.1

OLA	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm
5	KATL	17	31	0.27	0.95	0.25	KMCI	18	61	0.24	0.93	0.22
6	KATL	17	31	0.27	0.95	0.25	KMCI	18	61	0.24	0.93	0.22
5	KBOS	19	121	0.28	0.95	0.27	KMCO	17	31	0.26	0.98	0.26
6	KBOS	19	121	0.28	0.95	0.27	KMCO	17	31	0.26	0.98	0.26
5	KBWI	17	31	0.23	0.97	0.23	KMSP	18	31	0.25	0.90	0.23
6	KBWI	17	31	0.22	0.98	0.21	KMSP	18	31	0.25	0.90	0.23
5	KCLE	18	31	0.26	0.86	0.22	KORD	18	31	0.29	0.90	0.26
6	KCLE	18	31	0.26	0.86	0.22	KORD	18	31	0.27	0.90	0.24
5	KCLT	15	31	0.20	0.97	0.20	KPHL	18	31	0.28	0.95	0.27
6	KCLT	15	31	0.20	0.97	0.20	KPHL	18	31	0.28	0.95	0.27
5	KCVG	17	31	0.24	0.93	0.22	KPHX	16	31	0.27	0.96	0.26
6	KCVG	17	31	0.24	0.93	0.22	KPHX	16	31	0.27	0.96	0.26
5	KDCA	17	31	0.28	0.95	0.27	KPIT	17	31	0.22	0.95	0.21
6	KDCA	17	31	0.28	0.95	0.27	KPIT	17	31	0.20	0.96	0.19
5	KDEN	18	61	0.24	0.94	0.23	KSAC	16	31	0.30	0.96	0.29
6	KDEN	18	61	0.24	0.94	0.23	KSAC	16	31	0.30	0.96	0.28
5	KDFW	19	61	0.27	0.93	0.25	KSAN	15	1	0.27	0.97	0.26
6	KDFW	19	61	0.27	0.93	0.25	KSAN	15	1	0.27	0.97	0.26
5	KDTW	18	31	0.30	0.89	0.27	KSAT	17	91	0.26	0.95	0.24
6	KDTW	18	31	0.28	0.89	0.25	KSAT	17	91	0.26	0.95	0.24
5	KEUG	16	31	0.26	0.95	0.25	KSEA	16	31	0.36	0.95	0.34
6	KEUG	16	31	0.26	0.95	0.25	KSEA	16	31	0.36	0.95	0.34
5	KIAH	17	31	0.26	0.98	0.25	KSFO	20	91	0.24	0.93	0.22
6	KIAH	17	31	0.25	0.98	0.24	KSFO	20	91	0.23	0.93	0.21
5	KLAS	18	31	0.31	0.92	0.28	KSMX	18	61	0.30	0.98	0.30
6	KLAS	18	31	0.31	0.92	0.28	KSMX	18	61	0.30	0.98	0.29
5	KLAX	17	61	0.26	0.97	0.25	KSTL	17	31	0.27	0.93	0.25
6	KLAX	17	61	0.24	0.97	0.24	KSTL	17	31	0.27	0.93	0.25
5	KLGA	19	61	0.28	0.95	0.27	KTPA	16	31	0.24	0.98	0.24
6	KLGA	19	61	0.28	0.95	0.27	KTPA	16	31	0.24	0.98	0.24

Table 38: Average performance of the processing of OLAs 5 and 6 over all 30 weather stations by Static (variant 0x3, i.e., weather prediction, transitions limited) Rev. 1.1

OLA	Average	Metric
OLA 5	0.267	NetNorm
OLA 6	0.247	MQNetNorm

A.12 RESULTS OF THE ALGORITHMS AGING

A.12.1 Using current weather only

A.12.1.1 OLAs 1 & 2

Please see Table 39 below and Table 40 on page 233.

Table 39: Results of the processing of OLAs 1 and 2 by Aging (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.1

OLA	station	$y(1)$	$r(1)$	$y(2)$	$r(2)$	$y(3)$	$r(3)$	$y(4)$	$r(4)$	$y(5)$	$r(5)$	$y(6)$	$r(6)$	$y(7)$	$r(7)$	$y(8)$	$r(8)$	$y(9)$	$r(9)$	$y(10)$	$r(10)$	$y(11)$	$r(11)$	$y(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $y(n)$ is the y-intercept and $r(n)$ is the running average size minutes for month n																										
1	KATL	7	31	7	1	7	31	7	31	7	31	7	31	7	31	7	31	7	91	7	31	7	1	7	1	0.57	0.90	0.51
2	KATL	7	121	7	61	7	121	7	61	7	121	7	61	7	61	7	91	7	91	7	91	7	121	7	121	0.56	0.93	0.52
1	KBOS	9	31	9	31	9	31	9	31	9	1	9	61	9	1	9	61	9	31	9	31	9	31	9	31	0.38	0.92	0.35
2	KBOS	9	61	9	31	9	31	9	61	9	121	9	61	9	31	9	61	9	61	9	31	9	31	9	121	0.38	0.93	0.35
1	KBWI	7	1	7	1	7	31	7	31	7	31	7	31	7	1	7	31	7	1	7	31	7	1	7	1	0.67	0.82	0.55
2	KBWI	7	31	7	121	7	61	7	91	7	91	7	31	7	31	7	91	7	91	7	91	7	61	7	61	0.68	0.91	0.63
1	KCLE	8	31	8	1	8	1	8	1	8	1	8	31	8	1	8	1	8	31	8	1	8	1	8	31	0.59	0.89	0.52
2	KCLE	8	91	8	121	8	61	8	121	8	61	8	61	8	61	8	61	8	61	8	91	8	61	8	61	0.58	0.93	0.54
1	KCLT	5	1	5	1	5	31	5	31	5	1	5	1	5	1	5	31	5	31	5	31	5	1	5	1	0.70	0.84	0.59
2	KCLT	5	121	5	61	5	91	5	61	5	91	5	91	5	121	5	91	5	121	5	91	5	121	5	121	0.69	0.91	0.63
1	KCVG	7	31	7	1	7	1	7	1	7	1	7	31	7	1	7	1	7	1	7	31	7	1	7	1	0.65	0.85	0.55
2	KCVG	7	121	7	61	7	91	7	121	7	61	7	91	7	91	7	61	7	91	7	121	7	121	7	91	0.65	0.93	0.60
1	KDCA	7	1	7	1	7	31	7	31	7	1	7	31	7	1	7	1	7	61	7	1	7	1	7	1	0.50	0.88	0.44
2	KDCA	7	61	7	121	7	61	7	61	7	91	7	121	7	91	7	61	7	61	7	61	7	31	7	61	0.50	0.93	0.46
1	KDEN	8	1	8	1	8	1	8	31	8	31	8	31	8	1	8	1	8	1	8	1	8	1	8	1	0.56	0.83	0.46
2	KDEN	8	31	8	91	8	31	8	91	8	61	8	61	8	91	8	61	8	61	8	91	8	61	8	91	0.52	0.90	0.47
1	KDFW	9	1	9	61	9	1	9	1	9	1	9	1	9	31	9	1	9	31	9	1	9	31	9	31	0.47	0.89	0.41
2	KDFW	9	61	9	61	9	91	9	31	9	91	9	31	9	91	9	61	9	31	9	61	9	91	9	61	0.46	0.94	0.44
1	KDTW	8	31	8	1	8	1	8	31	8	31	8	61	8	31	8	1	8	31	8	31	8	1	8	1	0.57	0.89	0.50
2	KDTW	8	61	8	121	8	61	8	91	8	121	8	61	8	61	8	61	8	61	8	31	8	61	8	61	0.59	0.93	0.55
1	KEUG	6	1	6	31	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	0.74	0.83	0.61
2	KEUG	6	91	6	121	6	61	6	91	6	31	6	31	6	31	6	61	6	31	6	121	6	121	6	91	0.72	0.92	0.66
1	KIAH	7	1	7	1	7	31	7	1	7	1	7	1	7	1	7	1	7	31	7	1	7	31	7	1	0.70	0.86	0.60
2	KIAH	7	31	7	61	7	31	7	61	7	121	7	31	7	31	7	61	7	31	7	61	7	31	7	121	0.71	0.93	0.66
1	KLAS	8	1	8	1	8	1	8	1	8	1	8	1	8	31	8	1	8	1	8	31	8	1	8	1	0.45	0.76	0.35
2	KLAS	8	61	8	61	8	61	8	91	8	121	8	31	8	61	8	61	8	31	8	61	8	61	8	61	0.43	0.91	0.39
1	KLAX	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	0.88	0.90	0.79
2	KLAX	7	91	7	31	7	31	7	31	7	31	7	31	7	31	7	31	7	31	7	31	7	61	7	31	0.86	0.93	0.80

Table 39: (continued)

1	KLGA	9	91	9	91	9	31	9	31	9	31	9	31	9	31	9	1	9	31	9	1	9	61	9	31	0.42	0.90	0.38
2	KLGA	9	91	9	91	9	61	9	61	9	61	9	31	9	61	9	31	9	31	9	91	9	61	9	121	0.42	0.91	0.38
1	KMCI	8	1	8	31	8	31	8	1	8	31	8	31	8	31	8	1	8	1	8	1	8	31	8	31	0.66	0.91	0.60
2	KMCI	8	91	8	121	8	61	8	61	8	31	8	121	8	61	8	61	8	91	8	61	8	61	8	91	0.66	0.94	0.62
1	KMCO	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	31	7	1	7	1	0.66	0.83	0.55
2	KMCO	7	61	7	31	7	61	7	61	7	31	7	61	7	31	7	61	7	61	7	31	7	61	7	61	0.65	0.92	0.60
1	KMSP	8	1	8	31	8	1	8	31	8	31	8	1	8	31	8	31	8	1	8	31	8	1	8	31	0.36	0.89	0.32
2	KMSP	8	91	8	61	8	61	8	61	8	91	8	91	8	31	8	121	8	91	8	91	8	91	8	31	0.37	0.93	0.35
1	KORD	8	31	8	31	8	31	8	1	8	1	8	31	8	31	8	31	8	31	8	121	8	1	8	1	0.59	0.91	0.54
2	KORD	8	61	8	31	8	91	8	61	8	91	8	61	8	61	8	31	8	61	8	121	8	61	8	61	0.59	0.93	0.55
1	KPHL	8	1	8	31	8	1	8	31	8	31	8	31	8	31	8	31	8	31	8	31	8	1	8	31	0.64	0.90	0.58
2	KPHL	8	61	8	91	8	61	8	31	8	61	8	31	8	61	8	61	8	31	8	31	8	61	8	91	0.65	0.93	0.60
1	KPHX	6	1	6	1	6	31	6	31	6	31	6	31	6	31	6	31	6	31	6	1	6	1	6	1	0.37	0.74	0.27
2	KPHX	6	61	6	91	6	91	6	91	6	61	6	61	6	61	6	91	6	61	6	61	6	61	6	121	0.36	0.86	0.31
1	KPIT	7	1	7	31	7	1	7	31	7	31	7	31	7	31	7	31	7	31	7	31	7	1	7	1	0.72	0.87	0.63
2	KPIT	7	61	7	91	7	121	7	91	7	61	7	61	7	31	7	61	7	61	7	121	7	61	7	61	0.73	0.92	0.67
1	KSAC	6	1	6	1	6	1	6	1	6	31	6	31	6	31	6	1	6	1	6	1	6	1	6	1	0.77	0.83	0.64
2	KSAC	6	121	6	91	6	61	6	91	6	61	6	61	6	31	6	31	6	91	6	61	6	121	6	121	0.77	0.93	0.71
1	KSAN	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1	0.72	0.87	0.63
2	KSAN	5	61	5	61	5	61	5	61	5	31	5	31	5	31	5	61	5	61	5	61	5	31	5	91	0.71	0.92	0.66
1	KSAT	7	1	7	1	7	1	7	1	7	31	7	31	7	31	7	31	7	1	7	1	7	1	7	1	0.52	0.87	0.45
2	KSAT	7	121	7	121	7	91	7	31	7	121	7	61	7	31	7	91	7	61	7	61	7	31	7	61	0.53	0.93	0.49
1	KSEA	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	0.66	0.81	0.53
2	KSEA	6	121	6	61	6	91	6	121	6	121	6	121	6	61	6	91	6	121	6	121	6	121	6	121	0.65	0.93	0.60
1	KSFO	10	1	10	1	10	1	10	1	10	1	10	1	10	1	10	1	10	1	10	1	10	1	10	1	0.68	0.90	0.61
2	KSFO	10	61	10	61	10	31	10	31	10	31	10	31	10	31	10	31	10	31	10	31	10	91	10	91	0.67	0.94	0.63
1	KSMX	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	0.80	0.86	0.69
2	KSMX	8	31	8	31	8	31	8	31	8	31	8	31	8	31	8	31	8	31	8	31	8	31	8	121	0.79	0.91	0.72
1	KSTL	7	1	7	1	7	1	7	121	7	31	7	1	7	1	7	1	7	31	7	1	7	1	7	1	0.53	0.89	0.47
2	KSTL	7	61	7	61	7	61	7	121	7	31	7	61	7	31	7	91	7	31	7	91	7	61	7	121	0.52	0.93	0.48
1	KTPA	6	1	6	1	6	1	6	1	6	31	6	1	6	1	6	1	6	1	6	61	6	31	6	31	0.66	0.81	0.54
2	KTPA	6	61	6	91	6	31	6	61	6	31	6	61	6	61	6	61	6	61	6	61	6	91	6	31	0.65	0.90	0.59

Table 40: Average performance of the processing of OLAs 1 and 2 over all 30 weather stations by Aging (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.1

OLA	Average	Metric
OLA 1	0.606	NetNorm
OLA 2	0.555	MQNetNorm

A.12.1.2 OLAs 3 & 4 Please see Table 41 below and Table 42 on the next page.

Table 41: Results of the processing of OLAs 3 and 4 by Aging (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.2

OLA	station	$y(1)$	$r(1)$	$y(2)$	$r(2)$	$y(3)$	$r(3)$	$y(4)$	$r(4)$	$y(5)$	$r(5)$	$y(6)$	$r(6)$	$y(7)$	$r(7)$	$y(8)$	$r(8)$	$y(9)$	$r(9)$	$y(10)$	$r(10)$	$y(11)$	$r(11)$	$y(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $y(n)$ is the y-intercept and $r(n)$ is the running average size minutes for month n																										
3	KATL	17	61	17	31	17	31	17	31	17	31	7	121	7	31	17	91	17	121	17	31	7	121	17	91	0.36	0.91	0.33
4	KATL	17	61	17	61	17	61	7	121	17	61	7	121	7	91	17	91	17	121	17	31	7	121	17	91	0.35	0.93	0.33
3	KBOS	19	61	9	31	9	121	19	31	9	121	9	91	9	91	9	121	9	121	9	91	9	121	19	91	0.32	0.93	0.30
4	KBOS	19	61	9	31	9	121	19	61	9	121	9	91	9	91	9	121	9	121	9	91	9	121	19	91	0.32	0.93	0.30
3	KBWI	17	31	17	31	7	31	17	31	7	31	7	121	7	91	17	121	7	31	17	61	7	91	17	121	0.45	0.90	0.41
4	KBWI	17	61	17	91	7	31	17	61	7	61	7	121	7	121	17	121	7	91	17	61	7	91	17	121	0.45	0.91	0.41
3	KCLE	18	31	18	91	18	121	18	121	18	31	18	31	8	61	8	91	18	61	18	61	18	61	18	121	0.38	0.91	0.35
4	KCLE	18	31	18	91	18	121	18	121	18	31	18	61	8	91	8	91	18	61	18	91	18	61	18	121	0.38	0.92	0.35
3	KCLT	15	31	15	31	15	31	15	121	15	91	15	31	5	61	15	31	15	91	15	31	15	1	15	31	0.43	0.88	0.38
4	KCLT	15	61	15	61	15	31	15	121	15	91	5	61	5	61	15	61	15	91	15	61	15	91	15	31	0.43	0.90	0.39
3	KCVG	17	61	17	31	17	31	17	121	17	31	17	31	7	121	7	121	17	121	17	31	17	121	17	31	0.40	0.90	0.36
4	KCVG	17	61	17	91	17	31	27	61	17	61	17	31	7	121	7	121	17	121	17	61	17	121	17	61	0.38	0.90	0.34
3	KDCA	17	61	17	31	17	31	17	31	17	31	7	121	7	121	17	31	7	91	7	121	17	31	17	121	0.37	0.93	0.34
4	KDCA	17	61	17	31	17	61	17	61	7	121	7	121	7	121	17	61	7	121	7	121	17	121	17	91	0.36	0.93	0.34
3	KDEN	18	1	18	1	18	31	18	121	18	91	8	121	18	1	8	91	18	1	8	31	18	1	18	1	0.34	0.89	0.30
4	KDEN	18	31	18	31	18	31	18	121	18	91	8	121	8	121	8	91	18	31	8	61	18	31	18	91	0.38	0.92	0.35
3	KDFW	19	61	19	31	19	121	19	61	19	121	19	31	9	121	9	1	9	31	9	121	19	31	19	61	0.34	0.92	0.31
4	KDFW	19	61	19	61	19	121	19	61	19	121	19	31	9	121	9	61	9	121	9	121	19	31	19	61	0.34	0.94	0.32
3	KDTW	18	31	18	31	18	91	18	61	18	31	8	61	8	61	8	121	18	61	18	31	18	61	18	91	0.43	0.92	0.39
4	KDTW	18	31	18	31	18	91	18	61	18	91	8	61	8	91	8	121	18	121	18	61	18	91	18	91	0.42	0.93	0.39
3	KEUG	16	31	16	61	16	31	16	31	16	31	6	121	16	61	16	31	16	1	16	31	16	61	6	91	0.46	0.90	0.41
4	KEUG	16	61	16	61	16	61	6	121	16	31	6	121	16	61	16	31	16	31	16	31	16	61	6	91	0.47	0.91	0.43
3	KIAH	17	121	17	61	17	91	17	91	17	31	17	61	7	121	7	61	7	121	7	61	17	61	17	91	0.43	0.92	0.40
4	KIAH	17	121	17	91	17	91	17	121	17	91	7	121	7	121	7	91	7	121	7	121	17	91	17	91	0.44	0.93	0.41
3	KLAS	8	1	8	1	18	31	8	91	18	31	18	61	8	121	18	31	8	1	8	91	8	1	8	1	0.37	0.83	0.31
4	KLAS	8	61	8	61	18	91	8	121	18	31	18	61	8	121	18	31	8	31	8	121	8	61	8	61	0.38	0.92	0.34
3	KLAX	7	1	17	31	17	31	17	121	17	121	17	31	17	61	17	121	17	1	17	1	7	61	7	91	0.50	0.90	0.45
4	KLAX	17	121	17	31	17	31	17	121	17	121	17	31	17	61	17	121	17	31	17	31	7	61	7	91	0.50	0.92	0.46
3	KLGA	19	61	19	31	9	61	9	121	9	121	9	61	9	121	9	31	9	121	9	121	19	31	19	121	0.33	0.91	0.30
4	KLGA	19	61	19	31	9	121	9	121	9	121	9	61	9	121	9	61	9	121	9	121	19	31	19	121	0.33	0.91	0.30
3	KMCI	18	31	18	121	28	31	18	121	18	121	8	31	18	61	8	91	18	61	18	31	18	121	18	91	0.36	0.93	0.34
4	KMCI	18	61	18	121	28	31	18	121	18	121	8	121	18	61	8	91	18	61	18	31	18	121	18	121	0.36	0.94	0.34
3	KMCO	7	91	7	121	7	121	17	31	17	91	7	61	7	31	17	1	17	121	17	31	17	31	17	31	0.40	0.89	0.36
4	KMCO	7	91	7	121	7	121	17	31	17	91	7	61	7	31	17	91	17	121	17	61	17	31	17	31	0.40	0.91	0.37
3	KMSP	8	31	8	61	8	121	8	121	8	31	8	31	8	61	8	61	8	91	8	61	8	1	8	1	0.34	0.92	0.32
4	KMSP	8	121	8	61	8	121	8	121	8	31	8	121	8	61	8	121	8	91	8	91	8	91	8	61	0.35	0.93	0.33
3	KORD	18	91	18	31	18	31	18	61	18	91	8	31	8	121	8	121	8	31	18	61	18	91	18	91	0.37	0.92	0.34

Table 41: (continued)

4	KORD	18	91	18	31	18	91	18	61	18	91	8	61	8	121	8	121	8	61	18	91	18	91	18	91	0.37	0.92	0.34
3	KPHL	18	91	18	121	18	31	18	61	8	121	8	61	8	121	18	61	8	91	18	61	18	31	28	31	0.42	0.92	0.39
4	KPHL	18	121	18	121	18	91	18	121	8	121	8	61	8	121	18	61	8	91	18	61	18	61	28	91	0.42	0.93	0.39
3	KPHX	6	1	6	1	6	31	6	31	6	121	6	31	6	91	6	91	6	31	6	31	6	1	6	1	0.34	0.77	0.27
4	KPHX	6	61	6	91	6	91	6	61	6	121	6	61	6	121	6	121	6	31	6	61	6	61	6	121	0.34	0.85	0.29
3	KPIT	17	121	17	31	17	31	27	61	17	31	7	61	7	31	7	31	7	91	17	61	17	31	17	61	0.44	0.89	0.39
4	KPIT	17	121	17	61	17	91	27	61	17	61	7	61	7	31	7	91	7	91	17	61	17	91	17	61	0.44	0.91	0.40
3	KSAC	6	121	6	1	6	61	16	31	16	31	16	31	16	31	16	31	16	31	6	91	6	1	6	91	0.53	0.89	0.48
4	KSAC	6	121	6	61	6	121	16	61	16	91	6	121	16	61	16	91	16	31	6	91	6	121	6	91	0.54	0.93	0.50
3	KSAN	5	1	15	1	15	31	15	31	15	31	15	31	5	121	15	31	5	61	5	121	5	1	5	31	0.45	0.88	0.39
4	KSAN	5	121	15	31	15	31	15	61	15	31	15	31	5	121	15	91	5	61	5	121	5	61	5	61	0.45	0.91	0.41
3	KSAT	7	121	7	31	17	31	17	121	17	91	17	121	17	31	7	121	7	61	7	121	17	61	17	31	0.33	0.92	0.30
4	KSAT	7	121	7	121	17	61	17	121	17	91	17	121	17	31	7	121	7	61	7	121	17	61	17	31	0.33	0.93	0.30
3	KSEA	16	91	6	91	16	31	16	31	16	1	6	91	6	61	16	1	6	31	16	1	16	121	16	61	0.40	0.89	0.36
4	KSEA	16	91	6	91	16	61	16	61	16	121	6	121	6	61	6	91	6	121	16	31	16	121	16	61	0.40	0.92	0.37
3	KSFO	10	61	10	61	20	91	20	121	30	91	30	91	20	121	20	121	20	121	20	1	10	31	10	1	0.39	0.94	0.37
4	KSFO	10	61	10	61	20	121	20	121	30	91	30	91	20	121	20	121	20	121	20	31	10	91	10	91	0.39	0.95	0.37
3	KSMX	8	121	8	91	18	61	18	61	18	91	8	121	8	61	8	31	8	121	8	31	8	31	8	1	0.64	0.91	0.58
4	KSMX	8	121	8	121	18	61	18	61	18	91	8	121	8	61	8	31	8	121	8	31	8	31	8	121	0.64	0.91	0.58
3	KSTL	7	121	17	31	17	1	17	121	17	31	17	31	7	121	7	121	7	61	17	31	17	31	17	31	0.34	0.91	0.31
4	KSTL	7	121	17	61	17	61	17	121	17	91	17	61	7	121	7	121	7	61	17	31	17	31	17	91	0.35	0.92	0.32
3	KTPA	6	121	6	121	16	31	6	31	6	121	16	31	6	1	6	121	16	1	16	31	16	31	16	31	0.40	0.87	0.35
4	KTPA	6	31	6	121	16	31	6	31	6	121	16	31	6	91	6	121	16	121	16	91	16	31	16	91	0.41	0.89	0.36

Table 42: Average performance of the processing of OLAs 3 and 4 over all 30 weather stations by Aging (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.1

OLA	Average	Metric
OLA 3	0.402	NetNorm
OLA 4	0.371	MQNetNorm

A.12.1.3 OLAs 5 & 6 Please see Table 43 below and Table 44 on page 238.

Table 43: Results of the processing of OLAs 5 and 6 by Aging (variant 0x1, i.e., current weather only, transitions limited) Rev. 1.1

OLA	station	$y(1)$	$r(1)$	$y(2)$	$r(2)$	$y(3)$	$r(3)$	$y(4)$	$r(4)$	$y(5)$	$r(5)$	$y(6)$	$r(6)$	$y(7)$	$r(7)$	$y(8)$	$r(8)$	$y(9)$	$r(9)$	$y(10)$	$r(10)$	$y(11)$	$r(11)$	$y(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $y(n)$ is the y-intercept and $r(n)$ is the running average size minutes for month n																										
5	KATL	7	31	7	1	7	61	7	1	7	121	7	1	7	61	27	91	37	1	7	1	7	1	7	1	0.24	0.92	0.22
6	KATL	7	31	7	1	7	61	7	1	7	121	7	1	7	61	27	91	37	1	7	1	7	1	7	1	0.24	0.92	0.22
5	KBOS	9	1	39	121	9	1	9	1	9	1	9	121	39	121	9	31	9	31	9	1	9	1	9	1	0.20	0.91	0.18
6	KBOS	9	1	39	121	9	1	9	1	9	1	9	121	39	121	9	31	9	61	9	1	9	1	9	1	0.20	0.91	0.18
5	KBWI	7	1	37	91	37	61	37	121	37	121	7	1	27	31	7	1	7	1	7	91	7	1	7	61	0.21	0.94	0.20
6	KBWI	7	1	37	91	37	61	37	121	37	121	7	1	27	31	7	1	7	1	7	91	7	1	7	61	0.21	0.94	0.20
5	KCLE	8	1	38	1	18	1	8	31	8	121	38	1	8	31	38	121	8	91	8	1	38	121	8	1	0.20	0.90	0.18
6	KCLE	8	1	38	1	18	1	8	31	8	121	38	1	8	31	38	121	8	91	8	1	38	121	8	1	0.20	0.90	0.18
5	KCLT	5	1	25	1	5	31	35	1	5	1	5	1	5	61	5	1	5	31	35	121	35	1	5	1	0.26	0.93	0.24
6	KCLT	5	1	25	1	5	31	35	1	5	1	5	1	5	61	5	1	5	91	35	121	35	1	5	1	0.26	0.93	0.24
5	KCVG	7	1	37	121	7	1	7	121	7	1	7	121	7	31	7	31	37	121	7	1	37	121	7	1	0.19	0.90	0.17
6	KCVG	7	1	37	121	7	1	7	1	7	1	7	121	7	31	7	31	37	121	7	1	37	121	7	1	0.19	0.90	0.17
5	KDCA	7	31	37	31	37	1	7	1	7	1	37	1	7	1	7	1	7	1	7	121	7	1	7	1	0.22	0.92	0.20
6	KDCA	7	31	37	31	37	1	7	1	7	1	37	1	7	1	7	1	7	1	7	121	7	1	7	1	0.22	0.92	0.20
5	KDEN	8	91	38	31	8	1	8	1	8	61	8	31	8	1	8	1	38	61	8	1	8	1	8	1	0.16	0.91	0.15
6	KDEN	8	91	38	31	8	1	8	1	8	61	8	31	8	1	8	1	38	61	8	1	8	1	8	1	0.16	0.91	0.15
5	KDFW	9	91	39	1	9	1	9	1	9	1	9	1	9	1	9	1	39	91	9	121	39	91	9	1	0.22	0.91	0.20
6	KDFW	9	91	9	1	9	1	9	1	9	1	9	1	9	1	9	1	39	91	9	121	39	91	9	1	0.22	0.91	0.20
5	KDTW	8	1	8	1	18	1	8	1	8	1	8	31	8	31	8	61	38	1	8	121	8	1	8	31	0.18	0.91	0.17
6	KDTW	8	1	8	1	18	1	8	1	8	1	8	31	8	31	8	61	38	1	8	121	8	1	8	31	0.18	0.91	0.17
5	KEUG	36	121	26	31	6	1	36	1	6	121	6	91	6	121	6	1	26	121	6	1	26	121	6	31	0.21	0.95	0.20
6	KEUG	36	121	26	31	6	1	36	1	6	121	6	91	6	121	6	1	26	121	6	1	26	121	6	31	0.21	0.95	0.20
5	KIAH	7	121	7	1	7	1	7	31	7	31	37	121	7	1	7	31	37	1	7	121	37	121	7	1	0.21	0.96	0.20
6	KIAH	7	121	7	1	7	1	7	31	7	31	37	121	7	1	7	31	37	1	7	121	37	121	7	1	0.21	0.96	0.20
5	KLAS	8	1	28	31	8	1	8	1	8	1	8	1	8	1	8	121	38	31	8	1	28	91	38	31	0.20	0.90	0.18
6	KLAS	8	1	28	31	8	1	8	1	8	1	8	1	8	1	8	121	38	31	8	1	28	91	38	31	0.20	0.90	0.18
5	KLAX	7	121	37	61	7	121	7	1	7	121	7	31	7	31	7	121	7	1	7	121	37	31	7	1	0.20	0.97	0.20
6	KLAX	7	121	37	61	7	121	7	1	7	121	7	31	7	31	7	121	7	1	7	121	37	61	7	1	0.20	0.97	0.20
5	KLGA	9	1	39	121	9	1	9	1	9	31	9	31	39	61	9	1	39	91	9	1	39	91	9	1	0.20	0.92	0.18
6	KLGA	9	1	39	121	9	1	9	1	9	31	9	31	39	61	9	1	39	91	9	1	39	91	9	1	0.20	0.92	0.18
5	KMCI	8	1	38	121	8	1	8	1	28	121	38	1	8	1	38	31	8	1	38	121	38	1	8	1	0.19	0.92	0.17
6	KMCI	8	1	38	121	8	1	8	1	28	121	38	1	8	1	38	31	8	1	38	121	38	1	8	1	0.19	0.92	0.17
5	KMCO	7	31	37	1	7	121	37	1	7	1	7	121	27	91	7	61	37	1	7	1	7	1	7	1	0.22	0.94	0.21
6	KMCO	7	31	37	1	7	121	37	1	7	1	7	121	27	91	7	61	37	1	7	1	7	1	7	1	0.22	0.94	0.21
5	KMSP	8	1	38	31	8	1	8	1	8	1	38	121	8	1	8	1	38	61	8	1	8	1	8	1	0.17	0.92	0.15
6	KMSP	8	1	38	31	8	1	8	1	8	1	38	121	8	1	8	1	38	61	8	1	8	1	8	1	0.17	0.92	0.15
5	KORD	8	1	8	1	28	121	8	1	8	1	38	121	8	1	8	31	38	61	8	31	8	1	8	1	0.19	0.92	0.17

Table 43: (continued)

6	KORD	8	1	8	1	28	121	8	1	8	1	38	121	8	1	8	31	38	61	8	31	8	1	8	1	0.19	0.92	0.17
5	KPHL	8	1	38	61	38	1	38	121	8	121	8	1	8	1	8	1	28	61	38	31	38	121	8	1	0.21	0.92	0.19
6	KPHL	8	1	38	61	38	1	38	121	8	121	8	1	8	1	8	1	28	61	38	31	38	121	8	1	0.21	0.92	0.19
5	KPHX	6	1	6	31	26	31	6	1	6	1	6	1	6	121	6	1	6	31	36	121	26	121	36	61	0.18	0.92	0.17
6	KPHX	6	1	6	1	26	31	6	1	6	1	6	1	6	121	6	1	6	31	36	121	26	121	36	61	0.18	0.92	0.17
5	KPIT	7	1	37	121	37	1	37	121	7	1	7	31	7	31	7	31	17	61	27	121	7	1	7	1	0.21	0.91	0.19
6	KPIT	7	1	37	121	37	1	37	121	7	1	7	31	7	31	7	121	17	121	27	121	7	1	7	1	0.21	0.91	0.19
5	KSAC	6	1	6	31	6	1	6	31	6	1	6	31	6	1	6	1	6	1	6	1	36	31	6	1	0.20	0.95	0.19
6	KSAC	6	1	6	31	6	1	6	31	6	1	6	31	6	1	6	1	6	1	6	1	36	31	6	1	0.20	0.95	0.19
5	KSAN	35	1	35	31	25	1	35	31	5	1	35	121	5	121	5	1	35	121	5	91	5	1	5	61	0.23	0.97	0.22
6	KSAN	35	1	35	31	25	1	35	31	5	1	35	121	5	91	5	1	35	121	5	91	5	1	5	61	0.23	0.97	0.22
5	KSAT	7	1	7	1	7	1	7	1	7	1	7	1	7	31	7	1	7	1	27	61	27	61	7	61	0.19	0.92	0.18
6	KSAT	7	1	7	1	7	1	7	1	7	1	7	1	7	31	7	1	7	1	27	61	27	61	7	61	0.19	0.92	0.18
5	KSEA	6	121	6	1	6	1	6	1	6	1	6	1	36	121	26	61	36	31	6	1	6	1	6	1	0.21	0.96	0.21
6	KSEA	6	121	6	1	6	1	6	1	6	1	6	1	36	121	26	61	36	31	6	1	6	1	6	1	0.21	0.96	0.21
5	KSFO	10	1	40	31	10	1	10	1	10	1	10	1	10	31	10	1	30	91	10	1	30	91	10	121	0.17	0.94	0.16
6	KSFO	10	1	40	31	10	1	10	1	10	1	10	1	10	31	10	1	30	91	10	1	30	91	10	121	0.17	0.94	0.16
5	KSMX	8	31	38	31	8	121	8	1	38	121	38	121	8	91	8	121	38	121	8	1	8	1	8	121	0.17	0.97	0.16
6	KSMX	8	31	38	31	8	121	8	1	38	121	38	121	8	91	8	121	38	121	8	1	8	1	8	121	0.17	0.97	0.16
5	KSTL	7	1	37	121	17	1	37	31	7	1	27	121	7	121	7	1	7	1	7	1	37	121	7	31	0.18	0.91	0.17
6	KSTL	7	1	37	121	17	1	37	61	7	1	27	121	7	121	7	1	7	1	7	1	37	121	7	31	0.18	0.91	0.17
5	KTPA	6	31	36	1	6	91	36	121	6	1	6	1	6	1	6	31	36	1	6	1	6	31	6	1	0.21	0.94	0.19
6	KTPA	6	31	36	1	6	91	36	121	6	1	6	1	6	1	6	31	36	1	6	1	6	31	6	1	0.21	0.94	0.19

Table 44: Average performance of the processing of OLAs 5 and 6 over all 30 weather stations by Aging (variant 0x1, i.e., current weather only, transitions limited) Rev. 1.1

OLA	Average	Metric
OLA 5	0.201	NetNorm
OLA 6	0.187	MQNetNorm

A.12.2 Using weather prediction

A.12.2.1 OLAs 1 & 2

Please see Table 45 below and Table 46 on page 241.

Table 45: Results of the processing of OLAs 1 and 2 by Aging (variant 0x2, i.e., weather prediction, transitions unlimited) Rev. 1.1

OLA	station	$y(1)$	$r(1)$	$y(2)$	$r(2)$	$y(3)$	$r(3)$	$y(4)$	$r(4)$	$y(5)$	$r(5)$	$y(6)$	$r(6)$	$y(7)$	$r(7)$	$y(8)$	$r(8)$	$y(9)$	$r(9)$	$y(10)$	$r(10)$	$y(11)$	$r(11)$	$y(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $y(n)$ is the y-intercept and $r(n)$ is the running average size minutes for month n																										
1	KATL	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	61	7	1	7	1	7	1	0.53	0.77	0.41
2	KATL	7	91	7	121	7	121	7	91	7	121	7	91	7	61	7	121	7	61	7	121	7	121	7	121	0.54	0.92	0.49
1	KBOS	9	1	9	1	9	1	9	1	9	1	9	1	9	1	9	1	9	31	9	1	9	1	9	121	0.29	0.87	0.25
2	KBOS	9	91	9	1	9	91	9	121	9	121	9	91	9	61	9	61	9	91	9	61	9	1	9	121	0.37	0.93	0.34
1	KBWI	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	0.69	0.74	0.52
2	KBWI	7	121	7	121	7	91	7	121	7	121	7	121	7	1	7	121	7	121	7	61	7	91	7	121	0.65	0.89	0.58
1	KCLE	8	61	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	0.61	0.94	0.57
2	KCLE	8	61	8	121	8	121	8	121	8	121	8	121	8	121	8	61	8	61	8	121	8	61	8	91	0.57	0.93	0.53
1	KCLT	5	1	5	1	5	31	5	31	5	61	5	31	5	1	5	1	5	1	5	1	5	1	5	1	0.70	0.89	0.62
2	KCLT	5	91	5	121	5	121	5	121	5	91	5	91	5	91	5	121	5	121	5	121	5	121	5	121	0.68	0.91	0.62
1	KCVG	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	0.66	0.81	0.54
2	KCVG	7	121	7	121	7	121	7	91	7	121	7	121	7	121	7	91	7	121	7	91	7	91	7	121	0.63	0.92	0.58
1	KDCA	7	1	7	1	7	91	7	1	7	1	7	31	7	1	7	1	7	1	7	1	7	1	7	1	0.48	0.89	0.43
2	KDCA	7	121	7	91	7	91	7	91	7	121	7	91	7	121	7	91	7	91	7	121	7	121	7	91	0.48	0.93	0.45
1	KDEN	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	0.54	0.78	0.42
2	KDEN	8	121	8	121	8	121	8	91	8	91	8	121	8	61	8	91	8	61	8	91	8	121	8	121	0.51	0.90	0.45
1	KDFW	9	1	9	1	9	61	9	1	9	1	9	1	9	1	9	1	9	1	9	61	9	1	9	1	0.45	0.87	0.39
2	KDFW	9	91	9	121	9	91	9	121	9	121	9	121	9	61	9	121	9	121	9	91	9	121	9	91	0.45	0.94	0.42
1	KDTW	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	0.59	0.83	0.50
2	KDTW	8	91	8	91	8	121	8	121	8	121	8	121	8	121	8	91	8	121	8	61	8	121	8	121	0.57	0.93	0.53
1	KEUG	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	0.73	0.83	0.61
2	KEUG	6	121	6	121	6	121	6	91	6	61	6	61	6	61	6	61	6	61	6	121	6	91	6	1	0.71	0.91	0.64
1	KIAH	7	1	7	1	7	1	7	1	7	31	7	1	7	1	7	1	7	1	7	1	7	1	7	1	0.71	0.84	0.59
2	KIAH	7	1	7	121	7	61	7	91	7	61	7	61	7	61	7	61	7	61	7	91	7	91	7	91	0.69	0.92	0.63
1	KLAS	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	31	8	1	8	1	8	1	0.45	0.73	0.32
2	KLAS	8	1	8	91	8	121	8	91	8	121	8	121	8	121	8	91	8	61	8	1	8	121	8	1	0.42	0.88	0.37
1	KLAX	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	0.88	0.90	0.79
2	KLAX	7	121	7	1	7	91	7	61	7	31	7	1	7	31	7	31	7	31	7	31	7	1	7	1	0.86	0.91	0.78
1	KLGA	9	91	9	1	9	1	9	1	9	1	9	1	9	1	9	1	9	1	9	1	9	1	9	1	0.27	0.81	0.22
2	KLGA	9	121	9	61	9	61	9	91	9	121	9	61	9	61	9	61	9	121	9	121	9	121	9	1	0.39	0.91	0.36
1	KMCI	8	1	8	1	8	1	8	1	8	1	8	61	8	1	8	1	8	1	8	1	8	1	8	1	0.64	0.89	0.57
2	KMCI	8	1	8	121	8	121	8	121	8	61	8	91	8	91	8	121	8	61	8	121	8	91	8	121	0.64	0.93	0.60
1	KMCO	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	0.66	0.82	0.55

Table 45: (continued)

2	KMCO	7	1	7	121	7	91	7	121	7	121	7	91	7	121	7	121	7	91	7	91	7	61	7	121	0.62	0.91	0.56
1	KMSP	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	0.38	0.84	0.32
2	KMSP	8	61	8	121	8	1	8	121	8	91	8	1	8	91	8	91	8	91	8	121	8	121	8	61	0.34	0.92	0.32
1	KORD	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	91	8	1	8	1	0.60	0.90	0.54
2	KORD	8	121	8	121	8	91	8	121	8	91	8	91	8	121	8	91	8	91	8	121	8	121	8	121	0.57	0.93	0.54
1	KPHL	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	0.65	0.85	0.55
2	KPHL	8	91	8	91	8	61	8	121	8	121	8	91	8	91	8	121	8	121	8	121	8	121	8	91	0.62	0.93	0.57
1	KPHX	6	1	6	1	6	1	6	61	6	1	6	61	6	61	6	1	6	61	6	1	6	1	6	1	0.36	0.77	0.28
2	KPHX	6	121	6	121	6	121	6	121	6	91	6	121	6	91	6	121	6	61	6	121	6	121	6	121	0.34	0.85	0.29
1	KPIT	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	0.74	0.81	0.60
2	KPIT	7	91	7	121	7	121	7	121	7	121	7	91	7	121	7	91	7	61	7	121	7	121	7	91	0.69	0.92	0.64
1	KSAC	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	0.78	0.78	0.61
2	KSAC	6	121	6	61	6	121	6	91	6	91	6	61	6	61	6	121	6	121	6	91	6	121	6	91	0.74	0.92	0.69
1	KSAN	5	1	5	31	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1	0.72	0.89	0.64
2	KSAN	5	91	5	121	5	31	5	31	5	91	5	31	5	61	5	31	5	61	5	61	5	91	5	121	0.71	0.92	0.65
1	KSAT	7	1	7	1	7	1	7	31	7	1	7	1	7	1	7	61	7	1	7	1	7	1	7	1	0.40	0.84	0.34
2	KSAT	7	121	7	91	7	91	7	61	7	91	7	121	7	121	7	61	7	61	7	91	7	91	7	91	0.51	0.93	0.48
1	KSEA	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	6	1	0.66	0.81	0.53
2	KSEA	6	121	6	121	6	91	6	121	6	121	6	121	6	91	6	121	6	91	6	121	6	121	6	121	0.63	0.92	0.58
1	KSFO	10	1	10	1	10	1	10	1	10	1	10	1	10	1	10	1	10	1	10	1	10	1	10	1	0.68	0.90	0.61
2	KSFO	10	31	10	1	10	1	10	1	10	31	10	121	10	1	10	1	10	1	10	1	10	1	10	121	0.59	0.92	0.54
1	KSMX	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	0.80	0.86	0.69
2	KSMX	8	1	8	1	8	1	8	61	8	31	8	61	8	31	8	31	8	31	8	31	8	1	8	1	0.80	0.89	0.71
1	KSTL	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	1	7	91	7	1	0.53	0.86	0.46
2	KSTL	7	121	7	121	7	91	7	121	7	91	7	121	7	121	7	91	7	1	7	91	7	91	7	91	0.50	0.92	0.46
1	KTPA	6	1	6	1	6	1	6	1	6	61	6	1	6	1	6	1	6	1	6	1	6	1	6	1	0.67	0.84	0.56
2	KTPA	6	91	6	61	6	61	6	31	6	91	6	61	6	91	6	91	6	61	6	61	6	91	6	1	0.64	0.89	0.57

Table 46: Average performance of the processing of OLAs 1 and 2 over all 30 weather stations by Aging (variant 0x2, i.e., weather prediction, transitions unlimited) Rev. 1.1

OLA	Average	Metric
OLA 1	0.595	NetNorm
OLA 2	0.533	MQNetNorm

A.12.2.2 OLAs 3 & 4

Please see Table 47 below and Table 48 on the next page.

Table 47: Results of the processing of OLAs 3 and 4 by Aging (variant 0x2, i.e., weather prediction, transitions unlimited) Rev. 1.1

OLA	station	$y(1)$	$r(1)$	$y(2)$	$r(2)$	$y(3)$	$r(3)$	$y(4)$	$r(4)$	$y(5)$	$r(5)$	$y(6)$	$r(6)$	$y(7)$	$r(7)$	$y(8)$	$r(8)$	$y(9)$	$r(9)$	$y(10)$	$r(10)$	$y(11)$	$r(11)$	$y(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $y(n)$ is the y-intercept and $r(n)$ is the running average size minutes for month n																										
3	KATL	17	1	17	1	17	1	7	121	17	1	7	121	7	1	7	121	17	121	17	1	7	121	17	1	0.33	0.82	0.27
4	KATL	17	1	7	121	17	121	7	121	17	91	7	121	7	91	7	121	17	121	7	121	7	121	17	1	0.34	0.88	0.30
3	KBOS	19	1	9	1	9	1	9	121	9	121	9	1	9	121	9	91	9	1	9	61	9	1	19	1	0.30	0.88	0.26
4	KBOS	19	1	9	1	9	61	9	121	9	121	9	121	9	121	9	91	9	91	9	91	9	91	19	1	0.31	0.91	0.28
3	KBWI	17	1	17	1	7	1	17	1	7	1	7	91	7	121	7	121	7	121	17	1	7	121	17	121	0.40	0.80	0.32
4	KBWI	7	121	17	1	7	1	17	1	7	91	7	91	7	121	7	121	7	121	17	1	7	121	17	121	0.40	0.85	0.34
3	KCLE	18	1	18	1	18	1	18	121	18	1	18	1	18	1	8	121	18	1	18	1	18	1	18	1	0.28	0.85	0.24
4	KCLE	18	1	18	121	18	1	18	121	8	121	18	1	8	121	8	121	8	91	18	1	18	1	18	1	0.30	0.88	0.26
3	KCLT	15	1	15	1	15	1	15	1	15	1	5	121	5	1	15	1	15	91	15	1	15	1	15	1	0.36	0.74	0.26
4	KCLT	15	121	15	91	15	121	15	1	15	91	5	121	5	91	5	91	15	121	5	61	5	91	15	1	0.34	0.84	0.28
3	KCVG	17	1	17	1	17	1	17	121	17	1	17	1	7	61	7	121	7	1	17	1	17	1	17	1	0.27	0.82	0.22
4	KCVG	17	121	17	121	17	1	17	121	17	121	17	1	7	121	7	121	7	91	17	1	17	121	17	91	0.30	0.87	0.26
3	KDCA	17	1	17	1	7	1	17	1	17	1	7	1	7	121	17	1	7	121	7	1	17	1	17	31	0.28	0.84	0.24
4	KDCA	17	1	17	1	7	1	7	121	7	91	17	91	7	121	17	121	7	121	7	1	17	1	17	61	0.29	0.90	0.26
3	KDEN	18	1	18	1	18	1	18	1	18	1	18	1	8	121	18	1	18	1	8	1	18	1	18	1	0.30	0.81	0.25
4	KDEN	18	1	18	91	18	1	18	1	18	91	8	121	8	121	8	91	8	121	18	1	18	1	18	1	0.30	0.86	0.26
3	KDFW	19	1	19	1	19	1	19	31	19	91	9	31	9	121	9	91	9	1	9	1	19	1	19	1	0.32	0.88	0.28
4	KDFW	19	1	19	1	19	91	19	91	19	121	9	61	9	121	9	121	9	121	9	121	19	1	19	1	0.31	0.91	0.28
3	KDTW	18	1	18	1	18	1	18	1	18	1	8	91	8	61	8	1	18	1	18	1	18	1	18	1	0.32	0.84	0.26
4	KDTW	18	1	18	1	18	1	18	1	8	61	8	91	8	121	8	121	18	1	18	1	18	1	18	121	0.38	0.87	0.33
3	KEUG	16	1	16	1	16	1	6	61	16	1	6	1	6	61	16	1	16	1	16	1	16	1	16	1	0.44	0.83	0.37
4	KEUG	16	1	16	1	16	121	6	91	16	61	6	121	6	61	6	61	16	1	16	1	16	1	16	1	0.42	0.88	0.37
3	KIAH	17	1	17	1	17	121	17	91	17	121	17	61	7	121	7	61	7	61	7	61	17	1	17	1	0.37	0.86	0.32
4	KIAH	17	61	7	121	17	121	17	121	17	121	17	61	7	121	7	91	7	61	7	61	7	61	17	1	0.38	0.89	0.34
3	KLAS	8	1	8	31	18	1	18	1	18	1	18	1	8	121	8	121	8	91	8	1	8	1	8	1	0.34	0.85	0.29
4	KLAS	8	1	8	91	18	1	18	1	18	1	18	1	8	121	8	121	8	91	8	91	8	121	8	1	0.33	0.85	0.28
3	KLAX	7	1	17	1	17	31	17	31	17	91	7	1	17	91	17	121	17	31	17	1	7	91	7	61	0.47	0.88	0.42
4	KLAX	7	121	7	121	17	31	17	121	17	121	7	31	17	91	17	121	17	31	7	61	7	91	7	121	0.49	0.91	0.45
3	KLGA	19	1	19	1	9	1	9	1	9	1	9	91	9	91	9	61	9	121	9	121	19	1	19	1	0.30	0.85	0.26
4	KLGA	19	1	19	91	9	1	9	1	9	121	9	91	9	91	9	61	9	121	9	121	19	1	19	1	0.29	0.88	0.25
3	KMCI	18	121	18	1	28	1	18	1	18	121	18	1	18	1	8	1	18	1	18	91	18	1	18	1	0.22	0.86	0.19
4	KMCI	18	121	18	1	18	91	28	1	18	121	18	1	8	121	8	91	18	91	18	91	18	1	8	1	0.27	0.89	0.24
3	KMCO	7	1	7	1	17	1	17	1	17	31	7	1	7	31	17	31	17	91	17	1	17	1	17	1	0.36	0.79	0.28
4	KMCO	7	61	7	121	7	121	17	61	17	61	7	121	7	121	17	121	17	121	17	61	7	61	17	1	0.36	0.90	0.32
3	KMSP	8	1	8	1	8	31	8	121	8	1	8	61	8	1	8	91	8	1	8	1	8	91	8	1	0.28	0.86	0.24
4	KMSP	8	61	8	121	8	91	8	121	8	91	8	121	8	121	8	121	8	121	8	121	8	121	8	1	0.30	0.92	0.27
3	KORD	18	1	18	1	18	1	18	91	18	1	8	1	8	91	8	91	8	1	18	1	18	1	18	1	0.29	0.85	0.25

Table 47: (continued)

4	KORD	18	91	18	1	18	121	18	91	18	91	8	61	8	121	8	121	8	1	18	1	18	1	18	121	0.29	0.90	0.26
3	KPHL	18	1	18	1	18	1	18	1	8	1	8	1	8	121	18	1	8	1	18	91	18	1	18	121	0.37	0.84	0.31
4	KPHL	18	1	18	1	18	1	18	61	8	121	8	121	8	121	8	121	8	91	18	121	18	1	18	121	0.38	0.89	0.34
3	KPHX	6	1	6	121	6	121	6	61	6	91	6	31	6	121	6	61	6	1	6	61	6	1	6	1	0.32	0.77	0.25
4	KPHX	6	121	6	121	6	121	6	91	6	121	6	121	6	121	6	121	6	1	6	121	6	121	6	121	0.31	0.84	0.26
3	KPIT	17	121	17	1	7	31	27	1	17	1	7	91	7	1	7	31	7	61	7	61	17	1	17	1	0.30	0.81	0.24
4	KPIT	17	121	17	1	7	31	27	61	7	1	7	91	7	121	7	121	7	91	7	121	17	1	17	91	0.31	0.86	0.27
3	KSAC	6	1	6	1	16	1	16	1	16	91	6	121	6	121	16	1	16	1	16	1	6	31	6	1	0.45	0.81	0.36
4	KSAC	6	121	6	121	6	61	16	61	16	91	6	121	6	61	6	121	6	91	6	121	6	121	6	91	0.48	0.91	0.44
3	KSAN	5	1	15	1	15	1	15	31	5	1	15	1	5	121	15	91	5	121	5	31	5	31	5	1	0.42	0.83	0.35
4	KSAN	5	121	5	91	15	31	5	121	5	1	15	1	5	121	15	91	5	91	5	31	5	91	5	61	0.39	0.89	0.35
3	KSAT	17	1	7	61	17	1	17	1	17	121	17	121	17	1	7	121	7	61	7	91	17	1	17	1	0.30	0.85	0.25
4	KSAT	17	1	7	121	17	61	17	121	17	121	17	121	17	91	7	121	7	121	7	91	17	1	7	91	0.29	0.90	0.27
3	KSEA	16	1	16	1	16	1	16	1	16	1	6	1	6	31	16	1	6	91	16	1	16	1	16	1	0.28	0.80	0.22
4	KSEA	16	1	6	91	16	121	6	121	6	121	6	121	6	91	6	121	6	91	16	1	16	1	16	1	0.33	0.91	0.30
3	KSFO	10	91	10	1	20	91	20	121	30	91	30	31	20	121	20	91	20	91	20	1	10	1	10	1	0.37	0.91	0.33
4	KSFO	10	121	10	91	20	91	20	121	30	91	30	31	20	121	20	91	20	121	20	1	10	1	10	121	0.36	0.93	0.34
3	KSMX	8	1	8	1	18	1	18	61	18	121	8	31	8	31	8	1	8	91	8	121	8	1	8	1	0.59	0.87	0.52
4	KSMX	8	121	8	121	18	61	18	61	18	121	8	61	8	31	8	61	8	91	8	121	8	1	8	91	0.58	0.90	0.52
3	KSTL	7	1	17	1	17	1	17	121	17	1	7	91	7	61	7	121	7	121	17	1	17	1	17	1	0.27	0.86	0.24
4	KSTL	7	1	17	1	17	91	17	121	7	1	7	121	7	61	7	91	7	121	17	1	17	1	17	91	0.29	0.87	0.25
3	KTPA	6	1	6	31	16	1	6	31	6	91	16	1	6	31	6	121	16	61	16	1	6	91	16	1	0.39	0.82	0.32
4	KTPA	6	91	6	61	16	1	6	121	6	91	16	1	6	121	6	121	16	61	16	121	6	121	16	61	0.37	0.88	0.33

Table 48: Average performance of the processing of OLAs 3 and 4 over all 30 weather stations by Aging (variant 0x2, i.e., weather prediction, transitions unlimited) Rev. 1.1

OLA	Average	Metric
OLA 3	0.342	NetNorm
OLA 4	0.310	MQNetNorm

A.12.2.3 OLAs 5 & 6

Please see Table 49 below and Table 50 on the next page.

Table 49: Results of the processing of OLAs 5 and 6 by Aging (variant 0x3, i.e., weather prediction, transitions limited) Rev. 1.1

OLA	station	$y(1)$	$r(1)$	$y(2)$	$r(2)$	$y(3)$	$r(3)$	$y(4)$	$r(4)$	$y(5)$	$r(5)$	$y(6)$	$r(6)$	$y(7)$	$r(7)$	$y(8)$	$r(8)$	$y(9)$	$r(9)$	$y(10)$	$r(10)$	$y(11)$	$r(11)$	$y(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $y(n)$ is the y-intercept and $r(n)$ is the running average size minutes for month n																										
5	KATL	7	31	7	1	7	61	7	1	7	121	7	1	7	61	27	91	37	1	7	1	7	1	7	1	0.24	0.92	0.22
6	KATL	7	31	7	1	7	61	7	1	7	121	7	1	7	61	27	91	37	1	7	1	7	1	7	1	0.24	0.92	0.22
5	KBOS	9	1	39	121	9	1	9	1	9	1	9	121	39	121	9	31	9	31	9	1	9	1	9	31	0.21	0.91	0.19
6	KBOS	9	1	39	121	9	1	9	1	9	1	9	121	39	121	9	31	9	61	9	1	9	1	9	31	0.21	0.91	0.19
5	KBWI	7	1	37	91	37	61	37	121	37	121	7	1	27	31	7	1	7	31	7	91	7	1	7	61	0.21	0.93	0.20
6	KBWI	7	1	37	91	37	61	37	121	37	121	7	1	27	31	7	1	7	1	7	91	7	1	7	61	0.21	0.93	0.20
5	KCLE	8	31	38	1	18	1	8	31	8	121	8	1	8	31	38	121	8	91	8	1	38	121	8	1	0.19	0.90	0.17
6	KCLE	8	31	38	1	18	1	8	31	8	121	8	1	8	31	38	121	8	91	8	1	38	121	8	1	0.19	0.90	0.17
5	KCLT	5	1	25	1	5	31	35	1	5	1	5	1	5	61	5	1	5	31	35	121	5	1	5	1	0.26	0.93	0.24
6	KCLT	5	1	25	1	5	31	35	1	5	1	5	1	5	61	5	1	5	91	35	121	5	1	5	1	0.26	0.93	0.24
5	KCVG	7	31	37	121	7	1	7	121	7	1	7	121	7	31	7	31	37	121	7	1	37	121	7	1	0.19	0.90	0.17
6	KCVG	7	31	37	121	7	1	7	121	7	1	7	121	7	31	7	31	37	121	7	1	37	121	7	1	0.19	0.90	0.17
5	KDCA	7	31	37	31	37	1	7	1	7	31	37	1	7	1	7	1	7	1	7	121	7	1	7	1	0.22	0.91	0.20
6	KDCA	7	31	37	31	37	1	7	1	7	31	37	1	7	1	7	1	7	1	7	121	7	1	7	1	0.22	0.91	0.20
5	KDEN	8	91	38	31	8	31	8	1	8	61	8	31	8	1	8	1	38	61	8	1	8	1	8	1	0.16	0.90	0.15
6	KDEN	8	91	38	31	8	31	8	1	8	61	8	31	8	1	8	1	38	61	8	1	8	1	8	1	0.16	0.90	0.15
5	KDFW	9	91	39	1	9	1	9	1	9	1	9	1	9	1	9	1	39	91	9	121	39	91	9	1	0.22	0.91	0.20
6	KDFW	9	91	9	1	9	1	9	1	9	1	9	1	9	1	9	1	39	91	9	121	39	91	9	1	0.22	0.91	0.20
5	KDTW	8	1	8	1	18	1	38	1	8	1	8	31	8	31	8	61	38	1	8	121	8	1	8	31	0.18	0.91	0.17
6	KDTW	8	1	8	1	18	1	38	1	8	1	8	31	8	31	8	61	38	1	8	121	8	1	8	31	0.18	0.91	0.17
5	KEUG	36	121	26	31	6	1	6	1	6	121	6	91	6	121	6	1	26	121	6	1	26	121	6	31	0.21	0.95	0.20
6	KEUG	36	121	26	31	6	1	36	1	6	121	6	91	6	121	6	1	26	121	6	1	26	121	6	31	0.21	0.95	0.20
5	KIAH	7	121	7	1	7	1	7	31	7	31	37	121	7	1	7	31	37	1	7	121	37	121	7	1	0.21	0.96	0.20
6	KIAH	7	121	7	1	7	1	7	31	7	31	37	121	7	1	7	31	37	1	7	121	37	121	7	1	0.21	0.96	0.20
5	KLAS	8	1	28	31	8	1	8	1	8	1	8	31	8	1	8	121	38	31	8	1	28	91	38	31	0.20	0.90	0.18
6	KLAS	8	1	28	31	8	1	8	1	8	1	8	31	8	1	8	121	38	31	8	1	28	91	38	31	0.20	0.90	0.18
5	KLAX	7	121	37	61	7	121	7	1	7	121	7	31	7	31	7	121	7	1	7	121	37	31	7	1	0.20	0.97	0.20
6	KLAX	7	121	37	61	7	121	7	1	7	121	7	31	7	31	7	121	7	1	7	121	37	61	7	1	0.20	0.97	0.20
5	KLGA	9	1	39	121	9	1	9	1	9	31	9	31	39	61	9	1	39	91	9	1	39	91	9	1	0.20	0.92	0.18
6	KLGA	9	1	39	121	9	1	9	1	9	31	9	31	39	61	9	1	39	91	9	1	39	91	9	1	0.20	0.92	0.18
5	KMCI	8	1	38	121	8	1	8	1	28	121	38	1	8	1	38	31	8	1	38	121	38	1	8	1	0.19	0.92	0.17
6	KMCI	8	1	38	121	8	1	8	1	28	121	38	1	8	1	38	31	8	1	38	121	38	1	8	1	0.19	0.92	0.17
5	KMCO	7	31	37	1	7	121	37	1	7	1	7	121	27	91	7	61	7	1	7	1	7	1	7	1	0.22	0.94	0.21
6	KMCO	7	31	37	1	7	121	37	1	7	1	7	121	27	91	7	61	7	1	7	1	7	1	7	1	0.22	0.94	0.21
5	KMSP	8	1	38	31	8	1	8	1	8	1	38	121	8	1	8	1	38	61	8	1	8	1	8	1	0.17	0.92	0.15
6	KMSP	8	1	38	31	8	1	8	1	8	1	38	121	8	1	8	1	38	61	8	1	8	1	8	1	0.17	0.92	0.15
5	KORD	8	1	8	31	28	121	8	1	8	1	38	121	8	1	8	31	38	61	8	31	8	1	8	1	0.19	0.92	0.17

6	KORD	8	1	8	31	28	121	8	1	8	1	38	121	8	1	8	31	38	61	8	31	8	1	8	1	0.19	0.92	0.17
5	KPHL	8	1	38	61	38	1	38	121	8	121	8	61	8	1	8	1	28	61	38	31	38	121	8	1	0.21	0.92	0.19
6	KPHL	8	1	38	61	38	1	38	121	8	121	8	61	8	1	8	1	28	61	38	31	38	121	8	1	0.21	0.92	0.19
5	KPHX	6	1	6	31	26	31	6	1	6	1	6	1	6	121	6	1	6	31	36	121	26	121	36	61	0.18	0.92	0.17
6	KPHX	6	1	6	31	26	31	6	1	6	1	6	1	6	121	6	1	6	31	36	121	26	121	36	61	0.18	0.92	0.17
5	KPIT	7	1	37	121	37	1	37	121	7	1	7	31	7	31	7	31	17	61	27	121	7	1	7	1	0.21	0.91	0.19
6	KPIT	7	1	37	121	37	1	37	121	7	1	7	31	7	31	7	121	17	121	27	121	7	1	7	1	0.21	0.91	0.19
5	KSAC	6	1	6	31	6	1	6	31	6	1	6	31	6	1	6	1	6	1	6	1	36	31	6	1	0.20	0.95	0.19
6	KSAC	6	1	6	31	6	1	6	31	6	1	6	31	6	1	6	1	6	1	6	1	36	31	6	1	0.20	0.95	0.19
5	KSAN	35	1	35	31	25	1	35	31	5	1	35	121	5	121	5	1	35	121	5	91	5	1	5	61	0.23	0.97	0.22
6	KSAN	35	1	35	31	25	1	35	31	5	1	35	121	5	91	5	1	35	121	5	91	5	1	5	61	0.23	0.97	0.22
5	KSAT	7	1	7	1	7	1	7	1	7	1	7	1	7	31	7	1	7	1	27	61	27	61	7	61	0.19	0.92	0.18
6	KSAT	7	1	7	1	7	1	7	1	7	1	7	1	7	31	7	1	7	1	27	61	27	61	7	61	0.19	0.92	0.18
5	KSEA	6	121	6	1	6	1	6	1	6	1	6	1	36	121	26	61	36	31	6	1	6	31	6	1	0.21	0.96	0.20
6	KSEA	6	121	6	1	6	1	6	1	6	1	6	1	36	121	26	61	36	31	6	1	6	31	6	1	0.21	0.96	0.20
5	KSFO	10	1	40	31	10	1	10	1	10	31	10	1	10	31	10	1	30	91	10	1	30	91	10	121	0.17	0.94	0.16
6	KSFO	10	1	40	31	10	1	10	1	10	31	10	1	10	31	10	1	30	91	10	1	30	91	10	121	0.17	0.94	0.16
5	KSMX	8	31	38	31	8	121	8	1	38	121	38	121	8	91	8	121	8	1	8	1	8	1	8	121	0.17	0.97	0.16
6	KSMX	8	31	38	31	8	121	8	1	38	121	38	121	8	91	8	121	8	1	8	1	8	1	8	121	0.17	0.97	0.16
5	KSTL	7	31	37	121	7	1	37	31	7	1	27	121	7	121	7	1	7	31	7	1	37	121	7	31	0.18	0.91	0.16
6	KSTL	7	31	37	121	7	1	37	61	7	1	27	121	7	121	7	1	7	31	7	1	37	121	7	31	0.18	0.91	0.17
5	KTPA	6	31	36	1	6	91	36	121	6	1	6	31	6	1	6	31	36	1	6	1	6	31	6	1	0.21	0.94	0.19
6	KTPA	6	31	36	1	6	91	36	121	6	1	6	31	6	1	6	31	36	1	6	1	6	31	6	1	0.21	0.94	0.19

Table 50: Average performance of the processing of OLAs 5 and 6 over all 30 weather stations by Aging (variant 0x3, i.e., weather prediction, transitions limited) Rev. 1.1

OLA	Average	Metric
OLA 5	0.201	NetNorm
OLA 6	0.186	MQNetNorm

A.13 RESULTS OF THE ALGORITHMS FUZZY-CRISP HYBRID

The retraction-threshold difference is 1 for the following results:

A.13.1 Using current weather only

A.13.1.1 OLAs 1 & 2

Please see Table 51 below and Table 52 on page 248.

Table 51: Results of the processing of OLAs 1 and 2 by Fuzzy-Crisp Hybrid (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.4

OLA	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
1	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.53	0.87	0.46
2	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.53	0.87	0.46
1	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.34	0.92	0.31
2	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.34	0.92	0.31
1	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.62	0.88	0.55
2	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.62	0.88	0.55
1	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.54	0.92	0.50
2	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.54	0.92	0.50
1	KCLT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.87	0.58
2	KCLT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.87	0.58
1	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.90	0.53
2	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.90	0.53
1	KDCA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.45	0.91	0.41
2	KDCA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.45	0.91	0.41
1	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.47	0.91	0.43
2	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.47	0.91	0.43
1	KDFW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.43	0.93	0.40
2	KDFW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.43	0.93	0.40
1	KDTW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.55	0.91	0.50
2	KDTW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.55	0.91	0.50
1	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.68	0.91	0.62
2	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.68	0.91	0.62

Table 51: (continued)

1	KIAH	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.66	0.91	0.60
2	KIAH	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.66	0.91	0.60
1	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.42	0.91	0.38
2	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.42	0.91	0.38
1	KLAX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.81	0.93	0.76
2	KLAX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.81	0.93	0.76
1	KLGA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.38	0.89	0.34
2	KLGA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.38	0.89	0.34
1	KMCI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.62	0.94	0.58
2	KMCI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.62	0.94	0.58
1	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.58	0.90	0.53
2	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.58	0.90	0.53
1	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.36	0.90	0.32
2	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.36	0.90	0.32
1	KORD	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.54	0.91	0.49
2	KORD	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.54	0.91	0.49
1	KPHL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.91	0.54
2	KPHL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.91	0.54
1	KPHX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.34	0.71	0.24
2	KPHX	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.29	0.79	0.23
1	KPIT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.90	0.60
2	KPIT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.90	0.60
1	KSAC	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.74	0.89	0.66
2	KSAC	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.74	0.89	0.66
1	KSAN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.68	0.91	0.62
2	KSAN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.68	0.91	0.62
1	KSAT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.49	0.90	0.44
2	KSAT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.49	0.90	0.44
1	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.92	0.55
2	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.92	0.55
1	KSFO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.61	0.97	0.59
2	KSFO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.61	0.97	0.59
1	KSMX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.70	0.92	0.65
2	KSMX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.70	0.92	0.65
1	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.49	0.91	0.44
2	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.49	0.91	0.44
1	KTPA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.84	0.54
2	KTPA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.84	0.54

Table 52: Average performance of the processing of OLAs 1 and 2 over all 30 weather stations by Fuzzy-Crisp Hybrid (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.4

OLA	Average	Metric
OLA 1	0.560	NetNorm
OLA 2	0.505	MQNetNorm

A.13.1.2 OLAs 3 & 4

Please see Table 53 below and Table 54 on the next page.

Table 53: Results of the processing of OLAs 3 and 4 by Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.4

OLA	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
3	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.5	91	0.5	91	0.37	0.90	0.33
4	KATL	0.5	91	0.5	31	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	121	0.5	31	0.5	91	0.5	91	0.35	0.93	0.33
3	KBOS	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.33	0.92	0.30
4	KBOS	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.33	0.92	0.30
3	KBWI	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.46	0.87	0.40
4	KBWI	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.48	0.89	0.42
3	KCLE	0.5	91	0.5	31	0.5	1	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.32	0.94	0.30
4	KCLE	0.5	91	0.5	31	0.5	1	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.32	0.94	0.30
3	KCLT	0.5	31	0.5	121	0.5	61	0.5	121	0.5	1	0.5	31	0.5	1	0.5	31	0.5	121	0.5	1	0.5	31	0.5	121	0.41	0.91	0.38
4	KCLT	0.5	31	0.5	121	0.5	61	0.5	121	0.5	1	0.5	31	0.5	1	0.5	31	0.5	121	0.5	31	0.5	31	0.5	121	0.41	0.91	0.38
3	KCVG	0.5	91	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.38	0.91	0.35
4	KCVG	0.5	91	0.5	61	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.37	0.91	0.33
3	KDCA	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.38	0.91	0.35
4	KDCA	0.5	1	0.5	31	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.32	0.93	0.30
3	KDEN	0.5	1	0.5	1	0.5	61	0.5	91	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.38	0.91	0.34
4	KDEN	0.5	1	0.5	1	0.5	61	0.5	91	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.38	0.91	0.34
3	KDFW	0.5	31	0.5	121	0.5	31	0.5	31	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.28	0.95	0.26
4	KDFW	0.5	31	0.5	121	0.5	31	0.5	31	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.28	0.95	0.27
3	KDTW	0.5	121	0.5	31	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.36	0.93	0.34
4	KDTW	0.5	121	0.5	31	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.37	0.94	0.35
3	KEUG	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.47	0.90	0.42
4	KEUG	0.5	31	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.49	0.90	0.44
3	KIAH	0.5	61	0.5	61	0.5	121	0.5	91	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.43	0.91	0.39
4	KIAH	0.5	61	0.5	61	0.5	121	0.5	121	0.5	121	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.42	0.92	0.38
3	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.91	0.37
4	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.91	0.37
3	KLAX	0.5	1	0.5	31	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.48	0.92	0.44
4	KLAX	0.5	1	0.5	31	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.48	0.92	0.44
3	KLGA	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.31	0.90	0.28
4	KLGA	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.31	0.90	0.28
3	KMCI	0.5	121	0.5	1	0.5	1	0.5	91	0.5	31	0.5	31	0.5	1	0.5	1	0.5	31	0.5	121	0.5	121	0.5	1	0.32	0.96	0.30
4	KMCI	0.5	121	0.5	1	0.5	121	0.5	91	0.5	31	0.5	31	0.5	1	0.5	1	0.5	31	0.5	121	0.5	121	0.5	1	0.32	0.96	0.31
3	KMCO	0.5	1	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.48	0.91	0.43
4	KMCO	0.5	1	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.48	0.91	0.43
3	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.35	0.90	0.31
4	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.35	0.90	0.31
3	KORD	0.5	1	0.5	1	0.5	91	0.5	61	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.30	0.93	0.28

Table 53: (continued)

4	KORD	0.5	1	0.5	1	0.5	91	0.5	61	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	121	0.5	31	0.5	1	0.31	0.95	0.29
3	KPHL	0.5	1	0.5	91	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	31	0.5	61	0.38	0.94	0.36
4	KPHL	0.5	1	0.5	91	0.5	1	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	31	0.5	61	0.39	0.93	0.37
3	KPHX	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.23	0.86	0.20
4	KPHX	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.28	0.78	0.22
3	KPIT	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	31	0.5	61	0.44	0.91	0.40
4	KPIT	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	31	0.5	61	0.44	0.91	0.40
3	KSAC	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.89	0.53
4	KSAC	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	31	0.5	1	0.5	31	0.5	1	0.5	31	0.5	1	0.5	1	0.51	0.92	0.47
3	KSAN	0.5	1	0.5	31	0.5	121	0.5	1	0.5	1	0.5	121	0.5	61	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.45	0.91	0.40
4	KSAN	0.5	1	0.5	31	0.5	121	0.5	1	0.5	1	0.5	121	0.5	61	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.44	0.91	0.40
3	KSAT	0.5	1	0.5	1	0.5	31	0.5	31	0.5	91	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.38	0.92	0.35
4	KSAT	0.5	1	0.5	1	0.5	31	0.5	31	0.5	91	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.38	0.92	0.35
3	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	121	0.5	121	0.46	0.92	0.42
4	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	121	0.5	121	0.46	0.92	0.42
3	KSFO	0.5	1	0.5	1	0.5	31	0.5	121	0.5	121	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.37	0.95	0.35
4	KSFO	0.5	1	0.5	1	0.5	31	0.5	121	0.5	121	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.37	0.95	0.35
3	KSMX	0.5	1	0.5	1	0.5	31	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.61	0.93	0.57
4	KSMX	0.5	1	0.5	1	0.5	31	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.61	0.93	0.57
3	KSTL	0.5	1	0.5	1	0.5	1	0.5	61	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.28	0.91	0.26
4	KSTL	0.5	1	0.5	1	0.5	1	0.5	61	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.32	0.92	0.29
3	KTPA	0.5	61	0.5	61	0.5	31	0.5	1	0.5	61	0.5	31	0.5	1	0.5	1	0.5	31	0.5	31	0.5	31	0.5	31	0.47	0.89	0.42
4	KTPA	0.5	61	0.5	31	0.5	31	0.5	31	0.5	31	0.5	31	0.5	1	0.5	31	0.5	31	0.5	31	0.5	31	0.5	31	0.47	0.90	0.42

Table 54: Average performance of the processing of OLAs 3 and 4 over all 30 weather stations by Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.4

OLA	Average	Metric
OLA 3	0.396	NetNorm
OLA 4	0.362	MQNetNorm

A.13.1.3 OLAs 5 & 6

Please see Table 55 below and Table 56 on the next page.

Table 55: Results of the processing of OLAs 5 and 6 by Fuzzy-Crisp Hybrid (variant 0x1, i.e., current weather only, transitions limited) Rev. 1.4

OLA	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
5	KATL	0.5	31	0.5	1	0.5	121	0.5	31	0.5	31	0.5	1	0.5	121	0.5	121	0.5	31	0.5	1	0.5	61	0.5	1	0.26	0.92	0.24
6	KATL	0.5	31	0.5	1	0.5	121	0.5	31	0.5	31	0.5	1	0.5	121	0.5	121	0.5	31	0.5	1	0.5	61	0.5	1	0.26	0.92	0.24
5	KBOS	0.5	31	0.5	91	0.5	1	0.5	1	0.5	31	0.5	61	0.5	91	0.5	1	0.5	121	0.5	1	0.5	31	0.5	91	0.23	0.92	0.21
6	KBOS	0.5	31	0.5	91	0.5	1	0.5	1	0.5	31	0.5	61	0.5	91	0.5	1	0.5	121	0.5	1	0.5	31	0.5	121	0.23	0.92	0.21
5	KBWI	0.5	121	0.5	91	0.5	1	0.5	121	0.5	121	0.5	1	0.5	1	0.5	31	0.5	1	0.5	31	0.5	31	0.5	1	0.23	0.94	0.22
6	KBWI	0.5	121	0.5	91	0.5	1	0.5	121	0.5	121	0.5	1	0.5	1	0.5	31	0.5	1	0.5	31	0.5	31	0.5	1	0.23	0.94	0.22
5	KCLE	0.5	31	0.5	1	0.5	31	0.5	31	0.5	1	0.5	121	0.5	1	0.5	91	0.5	121	0.5	61	0.5	31	0.5	1	0.20	0.89	0.18
6	KCLE	0.5	31	0.5	1	0.5	31	0.5	31	0.5	1	0.5	121	0.5	1	0.5	91	0.5	121	0.5	61	0.5	31	0.5	1	0.20	0.89	0.18
5	KCLT	0.5	121	0.5	1	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	121	0.5	31	0.5	31	0.5	1	0.30	0.93	0.27
6	KCLT	0.5	121	0.5	1	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	121	0.5	31	0.5	31	0.5	1	0.30	0.93	0.27
5	KCVG	0.5	31	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	61	0.5	31	0.5	91	0.5	121	0.5	121	0.21	0.91	0.19
6	KCVG	0.5	31	0.5	121	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	61	0.5	31	0.5	91	0.5	121	0.5	121	0.21	0.91	0.19
5	KDCA	0.5	31	0.5	121	0.5	121	0.5	31	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.5	1	0.5	31	0.5	1	0.26	0.91	0.24
6	KDCA	0.5	31	0.5	121	0.5	121	0.5	31	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.5	1	0.5	31	0.5	1	0.26	0.91	0.24
5	KDEN	0.5	91	0.5	1	0.5	31	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	121	0.5	31	0.5	91	0.5	31	0.17	0.89	0.15
6	KDEN	0.5	91	0.5	1	0.5	31	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	121	0.5	31	0.5	91	0.5	31	0.17	0.89	0.15
5	KDFW	0.5	31	0.5	121	0.5	121	0.5	1	0.5	1	0.5	121	0.5	91	0.5	31	0.5	31	0.5	1	0.5	31	0.5	1	0.23	0.91	0.21
6	KDFW	0.5	31	0.5	121	0.5	121	0.5	1	0.5	1	0.5	121	0.5	91	0.5	31	0.5	31	0.5	1	0.5	31	0.5	1	0.23	0.91	0.21
5	KDTW	0.5	61	0.5	1	0.5	121	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.5	121	0.5	121	0.5	31	0.5	31	0.22	0.92	0.21
6	KDTW	0.5	61	0.5	1	0.5	121	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.5	121	0.5	121	0.5	31	0.5	31	0.22	0.92	0.21
5	KEUG	0.5	1	0.5	91	0.5	1	0.5	61	0.5	121	0.5	121	0.5	121	0.5	1	0.5	31	0.5	31	0.5	31	0.5	121	0.24	0.95	0.23
6	KEUG	0.5	1	0.5	91	0.5	1	0.5	61	0.5	121	0.5	121	0.5	121	0.5	1	0.5	31	0.5	31	0.5	31	0.5	121	0.24	0.95	0.23
5	KIAH	0.5	31	0.5	31	0.5	1	0.5	1	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	31	0.5	121	0.5	61	0.25	0.96	0.24
6	KIAH	0.5	31	0.5	31	0.5	1	0.5	1	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	31	0.5	121	0.5	61	0.25	0.96	0.24
5	KLAS	0.5	1	0.5	121	0.5	1	0.5	1	0.5	121	0.5	1	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	31	0.20	0.90	0.18
6	KLAS	0.5	1	0.5	121	0.5	1	0.5	1	0.5	121	0.5	1	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	31	0.20	0.90	0.18
5	KLAX	0.5	31	0.5	1	0.5	1	0.5	121	0.5	121	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	61	0.5	61	0.22	0.97	0.21
6	KLAX	0.5	31	0.5	1	0.5	1	0.5	121	0.5	121	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	61	0.5	61	0.22	0.97	0.21
5	KLGA	0.5	121	0.5	121	0.5	31	0.5	1	0.5	121	0.5	121	0.5	61	0.5	1	0.5	31	0.5	121	0.5	1	0.5	1	0.24	0.93	0.22
6	KLGA	0.5	121	0.5	121	0.5	31	0.5	1	0.5	121	0.5	121	0.5	61	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.24	0.93	0.22
5	KMCI	0.5	61	0.5	61	0.5	121	0.5	1	0.5	31	0.5	1	0.5	61	0.5	91	0.5	121	0.5	31	0.5	121	0.5	121	0.21	0.91	0.19
6	KMCI	0.5	61	0.5	91	0.5	121	0.5	1	0.5	31	0.5	1	0.5	61	0.5	91	0.5	121	0.5	31	0.5	121	0.5	121	0.21	0.91	0.19
5	KMCO	0.5	1	0.5	31	0.5	121	0.5	121	0.5	91	0.5	31	0.5	31	0.5	121	0.5	121	0.5	31	0.5	121	0.5	1	0.26	0.96	0.25
6	KMCO	0.5	121	0.5	31	0.5	121	0.5	121	0.5	91	0.5	31	0.5	31	0.5	121	0.5	121	0.5	31	0.5	121	0.5	1	0.27	0.96	0.26
5	KMSP	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	121	0.5	91	0.5	61	0.5	61	0.5	121	0.5	31	0.5	31	0.17	0.91	0.16
6	KMSP	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	121	0.5	91	0.5	61	0.5	61	0.5	121	0.5	31	0.5	31	0.17	0.91	0.16
5	KORD	0.5	31	0.5	31	0.5	121	0.5	91	0.5	31	0.5	61	0.5	31	0.5	91	0.5	31	0.5	121	0.5	31	0.5	121	0.21	0.91	0.19

Table 55: (continued)

6	KORD	0.5	31	0.5	31	0.5	121	0.5	91	0.5	31	0.5	61	0.5	31	0.5	91	0.5	31	0.5	121	0.5	31	0.5	121	0.21	0.91	0.19
5	KPHL	0.5	1	0.5	121	0.5	1	0.5	121	0.5	61	0.5	121	0.5	91	0.5	31	0.5	1	0.5	121	0.5	91	0.5	61	0.27	0.92	0.25
6	KPHL	0.5	1	0.5	31	0.5	1	0.5	61	0.5	61	0.5	121	0.5	91	0.5	31	0.5	1	0.5	121	0.5	91	0.5	61	0.26	0.92	0.24
5	KPHX	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	121	0.5	91	0.5	31	0.5	61	0.5	121	0.5	31	0.22	0.93	0.20
6	KPHX	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	121	0.5	91	0.5	31	0.5	61	0.5	121	0.5	31	0.22	0.93	0.20
5	KPIT	0.5	1	0.5	61	0.5	1	0.5	121	0.5	121	0.5	61	0.5	1	0.5	121	0.5	1	0.5	31	0.5	121	0.5	1	0.21	0.91	0.20
6	KPIT	0.5	1	0.5	91	0.5	1	0.5	121	0.5	121	0.5	61	0.5	1	0.5	121	0.5	1	0.5	31	0.5	121	0.5	1	0.21	0.91	0.19
5	KSAC	0.5	121	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	91	0.5	61	0.5	1	0.5	31	0.5	1	0.5	1	0.26	0.96	0.25
6	KSAC	0.5	121	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	91	0.5	61	0.5	1	0.5	31	0.5	1	0.5	1	0.26	0.96	0.25
5	KSAN	0.5	61	0.5	31	0.5	121	0.5	1	0.5	1	0.5	91	0.5	1	0.5	121	0.5	91	0.5	1	0.5	61	0.5	121	0.24	0.97	0.24
6	KSAN	0.5	61	0.5	31	0.5	121	0.5	1	0.5	1	0.5	91	0.5	1	0.5	121	0.5	91	0.5	1	0.5	61	0.5	121	0.24	0.97	0.24
5	KSAT	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	121	0.5	31	0.5	61	0.5	91	0.21	0.91	0.19
6	KSAT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	121	0.5	31	0.5	61	0.5	91	0.20	0.91	0.18
5	KSEA	0.5	121	0.5	1	0.5	1	0.5	121	0.5	121	0.5	1	0.5	61	0.5	121	0.5	91	0.5	1	0.5	91	0.5	61	0.30	0.95	0.28
6	KSEA	0.5	121	0.5	1	0.5	1	0.5	121	0.5	121	0.5	1	0.5	61	0.5	121	0.5	91	0.5	1	0.5	91	0.5	61	0.30	0.95	0.28
5	KSFO	0.5	1	0.5	121	0.5	31	0.5	121	0.5	121	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.5	121	0.5	61	0.19	0.94	0.18
6	KSFO	0.5	1	0.5	121	0.5	31	0.5	61	0.5	121	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.5	121	0.5	61	0.20	0.94	0.18
5	KSMX	0.5	31	0.5	1	0.5	31	0.5	1	0.5	61	0.5	91	0.5	121	0.5	31	0.5	31	0.5	121	0.5	91	0.5	31	0.21	0.96	0.20
6	KSMX	0.5	31	0.5	1	0.5	31	0.5	1	0.5	61	0.5	91	0.5	121	0.5	31	0.5	31	0.5	121	0.5	91	0.5	31	0.21	0.96	0.20
5	KSTL	0.5	1	0.5	31	0.5	121	0.5	1	0.5	31	0.5	121	0.5	1	0.5	61	0.5	1	0.5	121	0.5	31	0.5	1	0.19	0.92	0.18
6	KSTL	0.5	1	0.5	31	0.5	121	0.5	31	0.5	31	0.5	121	0.5	1	0.5	61	0.5	1	0.5	121	0.5	31	0.5	1	0.20	0.92	0.19
5	KTPA	0.5	1	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	121	0.5	1	0.5	91	0.5	121	0.5	121	0.5	91	0.25	0.95	0.24
6	KTPA	0.5	1	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	121	0.5	1	0.5	91	0.5	121	0.5	121	0.5	91	0.25	0.95	0.24

Table 56: Average performance of the processing of OLAs 5 and 6 over all 30 weather stations by Fuzzy-Crisp Hybrid (variant 0x1, i.e., current weather only, transitions limited) Rev. 1.4

OLA	Average	Metric
OLA 5	0.230	NetNorm
OLA 6	0.213	MQNetNorm

A.13.2 Using weather prediction

A.13.2.1 OLAs 1 & 2

Please see Table 57 below and Table 58 on page 255.

Table 57: Results of the processing of OLAs 1 and 2 by Fuzzy-Crisp Hybrid (variant 0x02, i.e., weather prediction, transitions unlimited) Rev. 1.4

OLA	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
1	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.51	0.88	0.45
2	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.51	0.88	0.45
1	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.32	0.92	0.29
2	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.32	0.92	0.29
1	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.89	0.53
2	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.89	0.53
1	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.92	0.48
2	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.92	0.48
1	KCLT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.87	0.56
2	KCLT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.87	0.56
1	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.57	0.90	0.51
2	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.57	0.90	0.51
1	KDCA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.44	0.91	0.40
2	KDCA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.44	0.91	0.40
1	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.44	0.92	0.40
2	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.44	0.92	0.40
1	KDFW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.93	0.39
2	KDFW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.93	0.39
1	KDTW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.92	0.48
2	KDTW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.92	0.48
1	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.91	0.58
2	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.91	0.58
1	KIAH	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.91	0.57
2	KIAH	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.91	0.57
1	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.40	0.91	0.37
2	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.40	0.91	0.37
1	KLAX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.77	0.93	0.71
2	KLAX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.77	0.93	0.71
1	KLGA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.37	0.90	0.33
2	KLGA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.37	0.90	0.33
1	KMCI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.94	0.56
2	KMCI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.94	0.56
1	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.56	0.91	0.51

Table 57: (continued)

2	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.56	0.91	0.51
1	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.35	0.91	0.31
2	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.35	0.91	0.31
1	KORD	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.51	0.92	0.47
2	KORD	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.51	0.92	0.47
1	KPHL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.57	0.92	0.52
2	KPHL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.57	0.92	0.52
1	KPHX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.31	0.71	0.22
2	KPHX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.31	0.71	0.22
1	KPIT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.90	0.58
2	KPIT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.90	0.58
1	KSAC	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.68	0.89	0.61
2	KSAC	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.68	0.89	0.61
1	KSAN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.65	0.91	0.59
2	KSAN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.65	0.91	0.59
1	KSAT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.48	0.91	0.43
2	KSAT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.48	0.91	0.43
1	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.58	0.92	0.53
2	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.58	0.92	0.53
1	KSFO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.97	0.58
2	KSFO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.97	0.58
1	KSMX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.92	0.58
2	KSMX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.92	0.58
1	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.47	0.91	0.43
2	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.47	0.91	0.43
1	KTPA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.84	0.51
2	KTPA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.84	0.51

Table 58: Average performance of the processing of OLAs 1 and 2 over all 30 weather stations by Fuzzy-Crisp Hybrid (variant 0x2, i.e., weather prediction, transitions unlimited) Rev. 1.4

OLA	Average	Metric
OLA 1	0.534	NetNorm
OLA 2	0.483	MQNetNorm

A.13.2.2 OLAs 3 & 4

Please see Table 59 below and Table 60 on the next page.

Table 59: Results of the processing of OLAs 3 and 4 by Fuzzy-Crisp Hybrid (variant 0x02, i.e., weather prediction, transitions unlimited) Rev. 1.4

OLA	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
3	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.44	0.87	0.39
4	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.44	0.87	0.39
3	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.32	0.92	0.29
4	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.32	0.92	0.29
3	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.55	0.88	0.49
4	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.55	0.88	0.49
3	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.91	0.38
4	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.91	0.38
3	KCLT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.50	0.87	0.44
4	KCLT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.50	0.87	0.44
3	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.42	0.89	0.38
4	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.42	0.89	0.38
3	KDCA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.42	0.91	0.38
4	KDCA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.42	0.91	0.38
3	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.92	0.38
4	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.92	0.38
3	KDFW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.36	0.93	0.34
4	KDFW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.36	0.93	0.34
3	KDTW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.47	0.91	0.43
4	KDTW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.47	0.91	0.43
3	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.56	0.91	0.51
4	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.56	0.91	0.51
3	KIAH	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.48	0.91	0.43
4	KIAH	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.48	0.91	0.43
3	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.40	0.91	0.36
4	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.40	0.91	0.36
3	KLAX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.92	0.48
4	KLAX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.92	0.48
3	KLGA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.37	0.90	0.33
4	KLGA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.37	0.90	0.33
3	KMCI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.93	0.38
4	KMCI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.93	0.38
3	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.51	0.90	0.46
4	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.51	0.90	0.46
3	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.34	0.91	0.31
4	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.34	0.91	0.31
3	KORD	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.91	0.37

Table 59: (continued)

4	KORD	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.91	0.37
3	KPHL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.47	0.91	0.43
4	KPHL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.47	0.91	0.43
3	KPHX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.31	0.71	0.22
4	KPHX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.31	0.71	0.22
3	KPIT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.48	0.89	0.42
4	KPIT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.48	0.89	0.42
3	KSAC	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.90	0.53
4	KSAC	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.90	0.53
3	KSAN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.90	0.47
4	KSAN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.90	0.47
3	KSAT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.40	0.90	0.36
4	KSAT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.40	0.90	0.36
3	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.50	0.92	0.46
4	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.50	0.92	0.46
3	KSFO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.43	0.96	0.42
4	KSFO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.43	0.96	0.42
3	KSMX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.92	0.58
4	KSMX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.92	0.58
3	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.91	0.37
4	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.41	0.91	0.37
3	KTPA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.84	0.43
4	KTPA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.84	0.43

Table 60: Average performance of the processing of OLAs 3 and 4 over all 30 weather stations by Fuzzy-Crisp Hybrid (variant 0x2, i.e., weather prediction, transitions unlimited) Rev. 1.4

OLA	Average	Metric
OLA 3	0.451	NetNorm
OLA 4	0.406	MQNetNorm

A.13.2.3 OLAs 5 & 6

Please see Table 61 below and Table 62 on the next page.

Table 61: Results of the processing of OLAs 5 and 6 by Fuzzy-Crisp Hybrid (variant 0x03, i.e., weather prediction, transitions limited) Rev. 1.4

OLA	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
5	KATL	0.5	31	0.5	1	0.5	121	0.5	31	0.5	31	0.5	1	0.5	121	0.5	1	0.5	31	0.5	1	0.5	61	0.5	31	0.28	0.92	0.26
6	KATL	0.5	31	0.5	1	0.5	121	0.5	31	0.5	31	0.5	1	0.5	121	0.5	1	0.5	31	0.5	1	0.5	61	0.5	31	0.28	0.92	0.26
5	KBOS	0.5	31	0.5	91	0.5	1	0.5	31	0.5	31	0.5	61	0.5	121	0.5	1	0.5	121	0.5	91	0.5	31	0.5	91	0.24	0.91	0.22
6	KBOS	0.5	31	0.5	91	0.5	1	0.5	31	0.5	31	0.5	61	0.5	121	0.5	1	0.5	121	0.5	91	0.5	31	0.5	121	0.24	0.91	0.22
5	KBWI	0.5	121	0.5	91	0.5	1	0.5	121	0.5	121	0.5	31	0.5	1	0.5	31	0.5	1	0.5	31	0.5	31	0.5	1	0.28	0.94	0.26
6	KBWI	0.5	121	0.5	91	0.5	1	0.5	1	0.5	121	0.5	31	0.5	1	0.5	31	0.5	1	0.5	31	0.5	31	0.5	121	0.29	0.94	0.28
5	KCLE	0.5	31	0.5	91	0.5	31	0.5	31	0.5	1	0.5	121	0.5	1	0.5	91	0.5	121	0.5	61	0.5	31	0.5	1	0.20	0.89	0.18
6	KCLE	0.5	31	0.5	91	0.5	31	0.5	31	0.5	1	0.5	121	0.5	1	0.5	91	0.5	121	0.5	61	0.5	31	0.5	1	0.20	0.89	0.18
5	KCLT	0.5	121	0.5	121	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	121	0.5	31	0.5	31	0.5	1	0.30	0.93	0.28
6	KCLT	0.5	121	0.5	121	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	121	0.5	31	0.5	31	0.5	1	0.30	0.93	0.28
5	KCVG	0.5	31	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.5	61	0.5	31	0.5	31	0.5	91	0.5	121	0.5	121	0.20	0.91	0.18
6	KCVG	0.5	31	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.5	61	0.5	31	0.5	31	0.5	91	0.5	121	0.5	121	0.20	0.91	0.18
5	KDCA	0.5	31	0.5	121	0.5	121	0.5	121	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.5	1	0.5	31	0.5	1	0.26	0.91	0.24
6	KDCA	0.5	31	0.5	121	0.5	121	0.5	121	0.5	1	0.5	1	0.5	1	0.5	121	0.5	1	0.5	1	0.5	31	0.5	1	0.26	0.91	0.24
5	KDEN	0.5	1	0.5	1	0.5	31	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	121	0.5	31	0.5	1	0.5	31	0.16	0.89	0.15
6	KDEN	0.5	1	0.5	1	0.5	31	0.5	121	0.5	31	0.5	121	0.5	1	0.5	91	0.5	121	0.5	31	0.5	1	0.5	31	0.16	0.89	0.15
5	KDFW	0.5	31	0.5	121	0.5	121	0.5	1	0.5	1	0.5	121	0.5	91	0.5	31	0.5	31	0.5	1	0.5	31	0.5	1	0.23	0.91	0.21
6	KDFW	0.5	31	0.5	121	0.5	121	0.5	1	0.5	1	0.5	121	0.5	91	0.5	31	0.5	31	0.5	1	0.5	31	0.5	1	0.23	0.91	0.21
5	KDTW	0.5	61	0.5	1	0.5	121	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.5	121	0.5	121	0.5	31	0.5	31	0.24	0.92	0.22
6	KDTW	0.5	61	0.5	1	0.5	121	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.5	121	0.5	121	0.5	31	0.5	31	0.24	0.92	0.22
5	KEUG	0.5	1	0.5	91	0.5	1	0.5	61	0.5	121	0.5	121	0.5	121	0.5	1	0.5	31	0.5	31	0.5	1	0.5	1	0.25	0.95	0.24
6	KEUG	0.5	1	0.5	91	0.5	1	0.5	61	0.5	121	0.5	121	0.5	121	0.5	1	0.5	31	0.5	31	0.5	1	0.5	1	0.25	0.95	0.24
5	KIAH	0.5	1	0.5	31	0.5	1	0.5	1	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	31	0.5	121	0.5	61	0.25	0.96	0.24
6	KIAH	0.5	1	0.5	31	0.5	1	0.5	1	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	31	0.5	121	0.5	61	0.25	0.96	0.24
5	KLAS	0.5	31	0.5	61	0.5	1	0.5	1	0.5	121	0.5	1	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.19	0.93	0.18
6	KLAS	0.5	31	0.5	61	0.5	1	0.5	1	0.5	121	0.5	1	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.19	0.93	0.18
5	KLAX	0.5	31	0.5	1	0.5	1	0.5	121	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	61	0.22	0.97	0.22
6	KLAX	0.5	31	0.5	1	0.5	1	0.5	121	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	61	0.22	0.97	0.22
5	KLGA	0.5	121	0.5	91	0.5	31	0.5	1	0.5	121	0.5	121	0.5	61	0.5	1	0.5	1	0.5	121	0.5	91	0.5	31	0.24	0.94	0.23
6	KLGA	0.5	121	0.5	91	0.5	31	0.5	1	0.5	121	0.5	121	0.5	61	0.5	121	0.5	1	0.5	121	0.5	91	0.5	1	0.25	0.94	0.23
5	KMCI	0.5	31	0.5	31	0.5	121	0.5	1	0.5	31	0.5	1	0.5	61	0.5	91	0.5	121	0.5	1	0.5	121	0.5	91	0.22	0.92	0.20
6	KMCI	0.5	31	0.5	91	0.5	121	0.5	1	0.5	31	0.5	1	0.5	61	0.5	91	0.5	121	0.5	1	0.5	121	0.5	91	0.22	0.92	0.20
5	KMCO	0.5	1	0.5	31	0.5	121	0.5	121	0.5	91	0.5	31	0.5	1	0.5	121	0.5	121	0.5	1	0.5	121	0.5	1	0.26	0.95	0.25
6	KMCO	0.5	121	0.5	31	0.5	121	0.5	121	0.5	91	0.5	31	0.5	1	0.5	121	0.5	121	0.5	1	0.5	121	0.5	1	0.27	0.95	0.25
5	KMSP	0.5	1	0.5	121	0.5	61	0.5	31	0.5	1	0.5	121	0.5	91	0.5	61	0.5	61	0.5	121	0.5	31	0.5	31	0.19	0.92	0.17
6	KMSP	0.5	1	0.5	121	0.5	61	0.5	1	0.5	1	0.5	121	0.5	91	0.5	61	0.5	61	0.5	121	0.5	31	0.5	31	0.18	0.91	0.17
5	KORD	0.5	31	0.5	31	0.5	121	0.5	91	0.5	1	0.5	1	0.5	31	0.5	91	0.5	31	0.5	121	0.5	31	0.5	121	0.21	0.91	0.19

Table 61: (continued)

6	KORD	0.5	31	0.5	31	0.5	121	0.5	91	0.5	1	0.5	1	0.5	31	0.5	91	0.5	31	0.5	121	0.5	31	0.5	121	0.21	0.91	0.19
5	KPHL	0.5	1	0.5	1	0.5	1	0.5	121	0.5	61	0.5	121	0.5	91	0.5	1	0.5	121	0.5	1	0.5	91	0.5	1	0.24	0.92	0.22
6	KPHL	0.5	1	0.5	1	0.5	1	0.5	61	0.5	61	0.5	121	0.5	91	0.5	1	0.5	121	0.5	1	0.5	91	0.5	1	0.24	0.92	0.22
5	KPHX	0.5	91	0.5	61	0.5	1	0.5	61	0.5	31	0.5	1	0.5	121	0.5	91	0.5	31	0.5	61	0.5	121	0.5	1	0.21	0.92	0.19
6	KPHX	0.5	91	0.5	31	0.5	1	0.5	61	0.5	31	0.5	1	0.5	121	0.5	91	0.5	31	0.5	61	0.5	121	0.5	1	0.21	0.92	0.19
5	KPIT	0.5	1	0.5	1	0.5	1	0.5	121	0.5	121	0.5	61	0.5	1	0.5	121	0.5	1	0.5	31	0.5	121	0.5	31	0.20	0.91	0.19
6	KPIT	0.5	1	0.5	1	0.5	1	0.5	121	0.5	121	0.5	61	0.5	1	0.5	121	0.5	1	0.5	31	0.5	121	0.5	31	0.20	0.91	0.19
5	KSAC	0.5	121	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	91	0.5	61	0.5	1	0.5	31	0.5	1	0.5	1	0.28	0.95	0.27
6	KSAC	0.5	121	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	91	0.5	61	0.5	1	0.5	31	0.5	1	0.5	1	0.28	0.95	0.27
5	KSAN	0.5	61	0.5	31	0.5	121	0.5	1	0.5	1	0.5	91	0.5	1	0.5	121	0.5	91	0.5	1	0.5	61	0.5	121	0.25	0.97	0.24
6	KSAN	0.5	61	0.5	31	0.5	121	0.5	1	0.5	1	0.5	91	0.5	1	0.5	121	0.5	91	0.5	1	0.5	61	0.5	121	0.25	0.97	0.24
5	KSAT	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.5	61	0.5	1	0.21	0.91	0.19
6	KSAT	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.5	61	0.5	1	0.22	0.91	0.20
5	KSEA	0.5	121	0.5	31	0.5	1	0.5	121	0.5	121	0.5	31	0.5	61	0.5	121	0.5	91	0.5	1	0.5	91	0.5	61	0.31	0.94	0.29
6	KSEA	0.5	121	0.5	31	0.5	1	0.5	121	0.5	121	0.5	31	0.5	61	0.5	121	0.5	91	0.5	1	0.5	61	0.5	61	0.31	0.94	0.29
5	KSFO	0.5	1	0.5	1	0.5	31	0.5	121	0.5	121	0.5	1	0.5	1	0.5	121	0.5	121	0.5	1	0.5	121	0.5	61	0.20	0.94	0.19
6	KSFO	0.5	1	0.5	1	0.5	31	0.5	1	0.5	121	0.5	1	0.5	1	0.5	91	0.5	121	0.5	1	0.5	121	0.5	61	0.20	0.94	0.18
5	KSMX	0.5	1	0.5	1	0.5	31	0.5	1	0.5	61	0.5	61	0.5	121	0.5	31	0.5	31	0.5	121	0.5	91	0.5	31	0.20	0.96	0.19
6	KSMX	0.5	1	0.5	1	0.5	31	0.5	1	0.5	61	0.5	61	0.5	121	0.5	31	0.5	31	0.5	121	0.5	91	0.5	31	0.20	0.96	0.19
5	KSTL	0.5	1	0.5	31	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.5	61	0.5	1	0.5	121	0.5	31	0.5	1	0.20	0.91	0.18
6	KSTL	0.5	1	0.5	31	0.5	1	0.5	1	0.5	31	0.5	121	0.5	1	0.5	61	0.5	1	0.5	121	0.5	31	0.5	1	0.20	0.91	0.18
5	KTPA	0.5	1	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	121	0.5	31	0.5	1	0.5	121	0.5	121	0.5	91	0.25	0.95	0.24
6	KTPA	0.5	1	0.5	121	0.5	31	0.5	121	0.5	1	0.5	1	0.5	121	0.5	31	0.5	1	0.5	121	0.5	121	0.5	91	0.25	0.95	0.24

Table 62: Average performance of the processing of OLAs 5 and 6 over all 30 weather stations by Fuzzy-Crisp Hybrid (variant 0x3, i.e., weather prediction, transitions limited) Rev. 1.4

OLA	Average	Metric
OLA 5	0.233	NetNorm
OLA 6	0.217	MQNetNorm

The membership function of the fuzzy set “Plenty of Allocated Visibility Minutes Remaining” is described by the following function:

$$1 - \frac{(\text{Minutes Visible Monthly}) + (\text{Time to Retract Minutes})}{(\text{Maximum Minutes Allowed to Be Visible})} \tag{15}$$

where

- *Minutes Visible Monthly* is how many minutes that harvester has been visible this month,
- *Time to Retract Minutes* is the time it takes for the harvester to retract, and
- *Maximum Minutes Allowed to Be Visible* is the OLA-specifified allocation of minutes that the harvester is permitted to be visible each month.

An example plot of the membership function of the fuzzy set “Plenty of Allocated Visibility Minutes Remaining” is shown in Figure 42 for the following case:

- the harvester is allocated 8760 minutes per month (e.g., OLAs 3-6) and
- the harvester starts the month deployed and remains deployed.

A.14 PROBABILITY FOR FALSE PREDICTION AT KBOS

Let us analyze Fuzzy variant 0x3 which requires that today and tomorrow be windy, which is defined as when the rolling average windspeed has a membership value of 0.9 or more. (We chose the lambda-cut value to be 0.9 in Section 3.1.2.1.) Let us use the rolling-average window size of 1, which is sometimes the rolling-average window size chosen by our training algorithm for OLA5 for Fuzzy (Table 61 on page 258).

Using our assumption that the day-ahead windspeed prediction error for a site is approximately 0.3 times that site's average windspeed, the standard deviation (σ) for the day-ahead prediction error for the site having the highest average windspeed listed in Appendix A.8 (KBOS at 9.42 knots) is 0.3×9.42 knots $= 2.8$ knots. Thus, for all sites, the least probabilities that the actual day-ahead windspeed will be within $\sigma = 2.8$ knots of the predicted windspeed is 68% and that the actual windspeed will be within $2\sigma = 5.6$ knots is 95%.

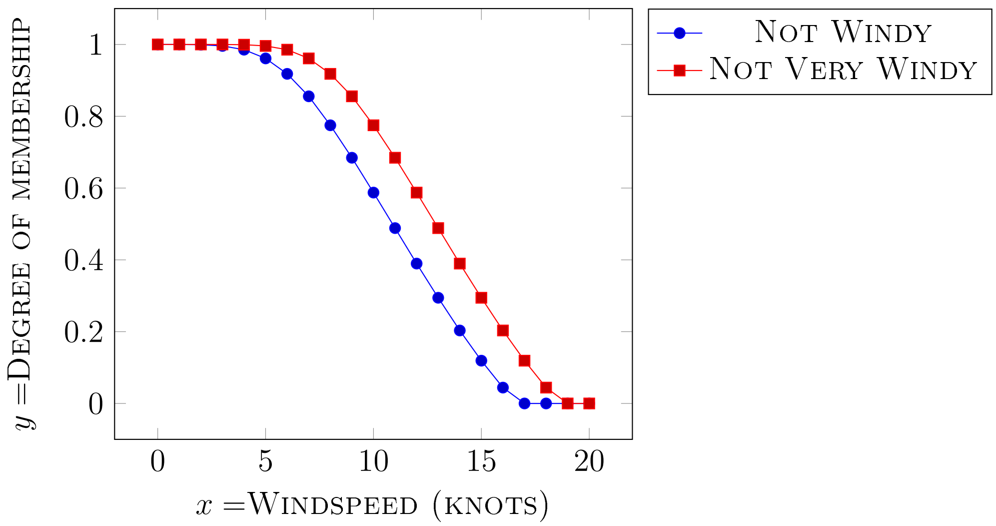

Figure 41: Example membership functions for the fuzzy set "Not Windy" and the set "Not Very Windy"

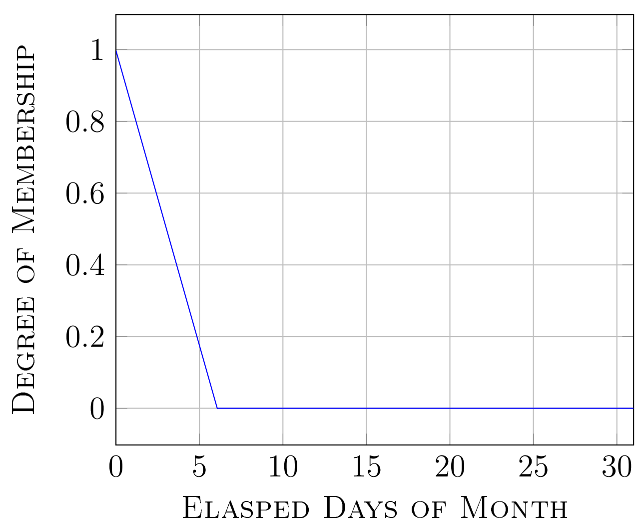

Figure 42: Example membership function for the fuzzy set "Plenty of Allocated Visibility Minutes Remaining" (See also Equation 15)

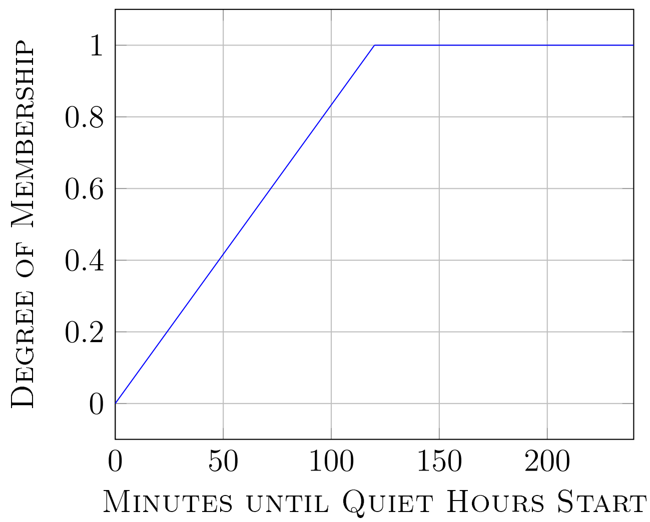

Figure 43: Membership function for the fuzzy set “Approaching Near the Start of Quiet Hours”

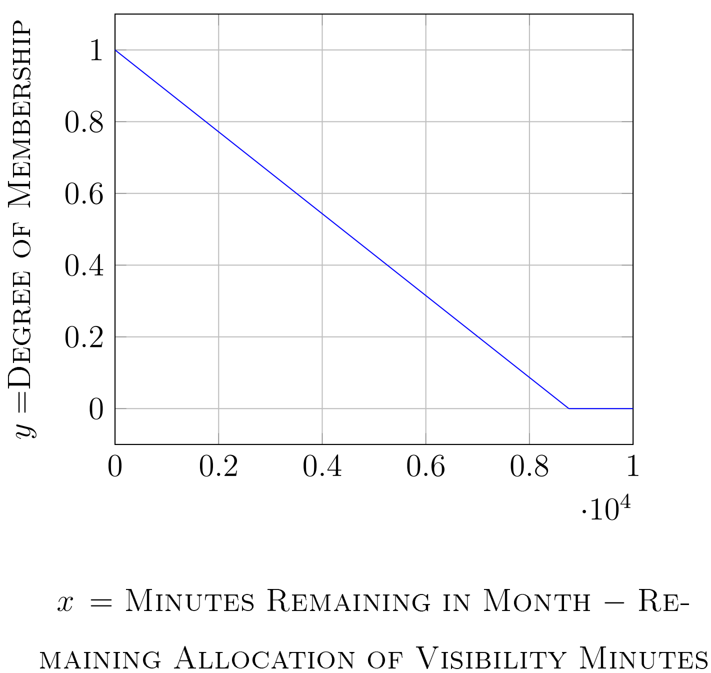

Figure 44: Example membership function for the fuzzy set “Approaching the Use Visibility Allocation or Lose It Point”

What is the probability that Fuzzy variant 0x3, will be fooled into deeming an actually windy tomorrow as being not windy (Type II error) or vice-a-versa (Type I error)?

Table 63 shows the probability for predicting tomorrow's windy day as not windy, given actual day-ahead windspeeds. (The table was substantially calculated using Excel's NORM.DIST() formula.) If the actual day-ahead windspeed is the lowest windspeed deemed to be windy at KBOS, which is 9 knots (Appendix A.2.4) since we chose the lambda-cut value to be 0.9, then because our simulated windspeed predictor uses a Gaussian distribution [47], the probability that the algorithm will receive a weather prediction that tomorrow will not be windy is 50%. If tomorrow's actual windspeed will be 10 knots, then the probability for a false negative is 36%.

Table 63: Probabilities at station KBOS for predicting tomorrow's windy day as not windy

Actual day-ahead windspeed (knots)	Probability of predicting windspeed less than 9 knots
9	50%
10	36%
11	24%
12	14%
13	8%
14	4%
15	2%
16	1%
17	0%

A.15 POPULATION OF MSA CORRESPONDING TO EACH OF THE 30 WEATHER STATIONS

Each of the 30 weather stations corresponds to a metropolitan statistical area (MSA). That correspondence is shown in Table 64. Also shown in Table 64 are the April 1, 2010 populations of each MSA [98]. The bottom lines of Table 64 indicate that the combined populations of the 30 MSAs comprise nearly 42% of the total U.S. population as counted on April 1, 2010 [98].

Table 64: Population of MSA corresponding to each of the 30 weather stations

Station	MSA	April 1, 2010 (Census)
KATL	Atlanta-Sandy Springs-Roswell, GA Metro Area	5,286,728
KBOS	Boston-Cambridge-Newton, MA-NH Metro Area	4,552,402
KBWI	Baltimore-Columbia-Towson, MD Metro Area	2,710,489
KCLE	Cleveland-Elyria, OH Metro Area	2,077,240
KCLT	Charlotte-Concord-Gastonia, NC-SC Metro Area	2,217,012
KCVG	Cincinnati, OH-KY-IN Metro Area	2,114,580
KDCA	Washington-Arlington-Alexandria, DC-VA-MD-WV Metro Area	5,636,232
KDEN	Denver-Aurora-Lakewood, CO Metro Area	2,543,482
KDFW	Dallas-Fort Worth-Arlington, TX Metro Area	6,426,214
KDTW	Detroit-Warren-Dearborn, MI Metro Area	4,296,250
KEUG	Eugene, OR Metro Area	351,715
KIAH	Houston-The Woodlands-Sugar Land, TX Metro Area	5,920,416
KLAS	Las Vegas-Henderson-Paradise, NV Metro Area	1,951,269
KLAX	Los Angeles-Long Beach-Anaheim, CA Metro Area	12,828,837
KLGA	New York-Newark-Jersey City, NY-NJ-PA Metro Area	19,567,410
KMCI	Kansas City, MO-KS Metro Area	2,009,342
KMCO	Orlando-Kissimmee-Sanford, FL Metro Area	2,134,411
KMSP	Minneapolis-St. Paul-Bloomington, MN-WI Metro Area	3,348,859
KORD	Chicago-Naperville-Elgin, IL-IN-WI Metro Area	9,461,105
KPHL	Philadelphia-Camden-Wilmington, PA-NJ-DE-MD Metro Area	5,965,343
KPHX	Phoenix-Mesa-Scottsdale, AZ Metro Area	4,192,887
KPIT	Pittsburgh, PA Metro Area	2,356,285
KSAC	Sacramento–Roseville–Arden-Arcade, CA Metro Area	2,149,127
KSAN	San Diego-Carlsbad, CA Metro Area	3,095,313
KSAT	San Antonio-New Braunfels, TX Metro Area	2,142,508
KSEA	Seattle-Tacoma-Bellevue, WA Metro Area	3,439,809

Table 64: (continued)

KSFO	San Francisco-Oakland-Hayward, CA Metro Area	4,335,391
KSMX	Santa Maria-Santa Barbara, CA Metro Area	423,895
KSTL	St. Louis, MO-IL Metro Area	2,787,701
KTPA	Tampa-St. Petersburg-Clearwater, FL Metro Area	2,783,243
	Total population of the 30 MSA's listed above:	129,105,495
	Total U.S. population, census, April 1, 2010:	308,745,538
	30 MSA's population as a percentage of total:	41.82%

A.16 ENERGY CAPTURED AT EACH WEATHER STATION DURING TRAINING AND TESTING BY A PERMANENTLY DEPLOYED HARVESTER

The sources of the ASOS data are ASOS weather stations, which usually have anemometer heights of approximately 10 or 8 meters; " Typical ASOS wind sensor heights are 33 feet or 27 feet, depending on local site-specific restrictions or requirements" [59]. Please note that because windspeeds typically increase with altitude[17, p. 668] and wind turbines may be much higher than 10 meters, be careful to not underestimate wind resources at the site of a particular ASOS weather station. For information on assessing wind resources, please visit the National Renewable Energy Laboratory's web page entitled "Wind Resource Assessment" [65].)

Table 65: Energy captured at each weather station ws during training and testing by permanently deployed harvester hm defined in Appendix A.3

	$E_{\text{Harvested}}^{(\text{Permanent},wrk)}$ (GWh)	
Station	Training	Testing
KATL	11.08164373	2.313831367
KBOS	24.62576615	5.124163067

Table 65: (continued)

KBWI	8.529008367	1.6589851
KCLE	17.7525112	4.426799383
KCLT	4.898122183	1.082148483
KCVG	11.54389702	2.520545983
KDCA	12.9294752	2.79856035
KDEN	20.52796085	4.692598817
KDFW	25.16715067	6.362182917
KDTW	14.93492915	3.617316633
KEUG	7.234854417	1.490479333
KIAH	10.82712013	2.402683117
KLAS	14.10018065	3.675125867
KLAX	9.040475867	1.947226767
KLGA	24.14711153	5.16765185
KMCI	21.16693805	4.87104135
KMCO	10.06265663	1.886246417
KMSP	15.34392997	3.693996417
KORD	17.19378537	3.938879683
KPHL	16.6037386	3.530311733
KPHX	6.110979783	1.2980193
KPIT	9.992927967	2.311737383
KSAC	6.023616267	1.217522833
KSAN	4.329993183	0.783599983
KSAT	13.56972367	3.12457265
KSEA	9.219178417	2.016763933
KSFO	29.17697397	6.20793595
KSMX	11.70718298	2.629237233
KSTL	14.20177952	3.131923483

Table 65: (continued)

KTPA	5.465520333	1.06674115

A.17 ESTIMATED SHAPE AND SCALE PARAMETERS OF WEIBULL DISTRIBUTION FOR EACH STATION

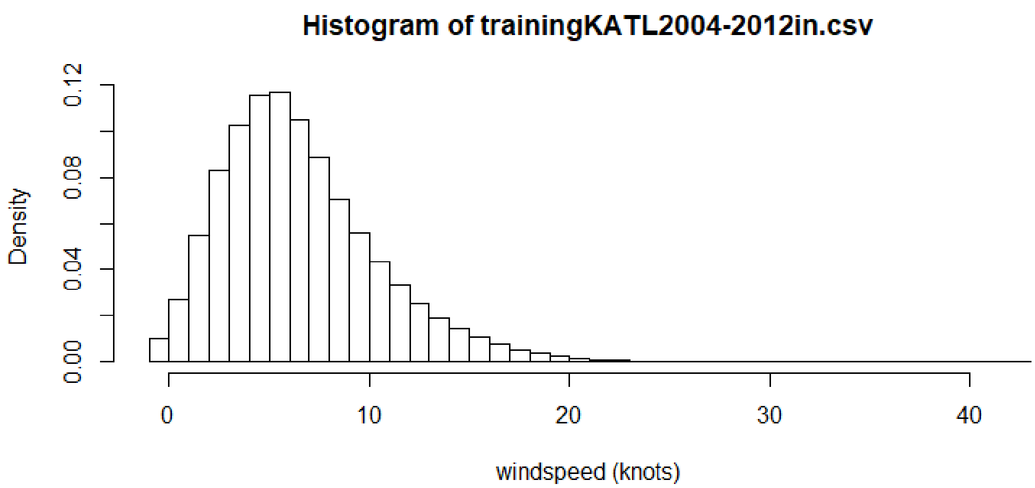

Figure 45: Histogram of densities of windspeeds in KATL's training file: trainingKTPA2004-2012in.csv

We used the R script below to generate the estimates (Table 66) of the shape (B) and scale (A) parameters of a Weibull distribution [6, Equation 5.45, p.185] (e.g., Figure 45) fitted to the densities of windspeeds for each station. Note that in some cases, the script may truncate data to avoid the following error:

```
Error in checkparamlist(arg_startfix$start.arg, arg_startfix$fix.arg
    , :
  'start' should not have NA or NaN values.
```

Also note that the script estimates the Weibull distribution for a location parameter (ν) of -1. (The location parameter is -1 because zero-knot windspeeds are placed in the first bucket of the histogram generated by the script's line starting with

```
x = hist(DF$wind_knots, freq=FALSE, plot=FALSE, breaks=
seq(-1,max(DF$wind_knots))
```

.)

```
stations <- c("KATL",
"KBOS",
"KBWI",
"KCLE",
"KCLT",
"KCVG",
"KDCA",
"KDEN",
"KDFW",
"KDTW",
"KEUG",
"KIAH",
"KLAS",
"KLAX",
"KLGA",
"KMCI",
"KMCO",
"KMSP",
"KORD",
"KPHL",
"KPHX",
"KPIT",
"KSAC",
"KSAN",
"KSAT",
"KSEA",
"KSFO",
"KSMX",
"KSTL",
"KTPA")

#stations <- c("KATL")

library("fitdistrplus")

iter <- length(stations)+1
m <- matrix(NA, nrow=iter, ncol=6)
```

```r
rowIndex = 1
m[rowIndex,] <- c("station", "shape", "scale", "Kolmogorov-Smirnov statistic", "Cramer-von Mises statistic", "Anderson-Darling statistic")
rowIndex=rowIndex+1
for(station in stations) {

 print(station)

 inputFilenameBase = paste("training", station, "2004-2012in", sep="")
        inputFilename = paste(inputFilenameBase, ".csv", sep="")

 DF <- read.csv(inputFilename, header=TRUE)

        x = hist(DF$wind_knots, freq=FALSE, plot=FALSE, breaks=seq(-1,max(DF$wind_knots)),main=paste("Histogram of", inputFilename),xlab="windspeed (knots)")
 #If a zero exists, truncate data to avoid the following error:
 ## Error in checkparamlist(arg_startfix$start.arg, arg_startfix$fix.arg, :
 ##  'start' should not have NA or NaN values.
 finalIndex = if(x$density[which.min(x$density)] == 0) which.min(x$density)-1 else length(x$density)
 #d = fitdistr(x$density[1:finalIndex], "weibull")
 d = fitdist(x$density[1:finalIndex], "weibull",lower=0)
 k = gofstat(d)
 m[rowIndex,] <- c(station, d$estimate, k$ks, k$cvm, k$ad)
 rowIndex=rowIndex+1

}

options(width = 200)
sink("WeibullAllStations.txt")
print(m)
sink()
```

Table 66: Estimated shape and scale parameters of Weibull distribution for each station where the location parameter (ν) is -1

	estimated		Goodness-of-fit statistics		
station	shape	scale	Kolmogorov-Smirnov	Cramer-von Mises	Anderson-Darling

Table 66: (continued)

KATL	0.28	0.004	0.12	0.21	1.42
KBOS	0.37	0.008	0.11	0.14	0.96
KBWI	0.31	0.006	0.10	0.12	0.93
KCLE	0.30	0.005	0.12	0.19	1.34
KCLT	0.29	0.005	0.11	0.13	0.94
KCVG	0.27	0.003	0.13	0.22	1.48
KDCA	0.36	0.009	0.11	0.14	0.97
KDEN	0.35	0.006	0.09	0.09	0.72
KDFW	0.31	0.006	0.14	0.26	1.74
KDTW	0.31	0.005	0.11	0.15	1.10
KEUG	0.28	0.004	0.12	0.17	1.19
KIAH	0.31	0.004	0.21	0.44	2.55
KLAS	0.35	0.008	0.11	0.17	1.13
KLAX	0.36	0.009	0.14	0.12	0.79
KLGA	0.39	0.011	0.11	0.15	1.12
KMCI	0.31	0.006	0.12	0.23	1.62
KMCO	0.26	0.001	0.15	0.36	2.21
KMSP	0.31	0.006	0.13	0.22	1.54
KORD	0.38	0.011	0.13	0.18	1.23
KPHL	0.35	0.008	0.11	0.15	1.04
KPHX	0.26	0.002	0.11	0.15	1.12
KPIT	0.35	0.008	0.12	0.16	1.06
KSAC	0.25	0.002	0.16	0.26	1.82
KSAN	0.31	0.006	0.13	0.14	0.95
KSAT	0.29	0.005	0.13	0.24	1.63
KSEA	0.31	0.005	0.11	0.17	1.16

Table 66: (continued)

KSFO	0.36	0.007	0.14	0.24	1.65
KSMX	0.31	0.006	0.12	0.18	1.23
KSTL	0.32	0.006	0.12	0.17	1.16
KTPA	0.32	0.005	0.13	0.16	1.01

A.18 LOOKING FOR TRENDS IN WINDSPEED DATA

In Section 4.2.0.2, we suggested that control algorithms look for trends in the training data. We looked for yearly trends using the following method for each station:

1. For each year of windspeeds in the nine years of training data (2004 to 2012), create a density distribution table.
2. From the density distribution table, estimate Weibull scale and shape parameters to create two eleven-year time series.
3. For the training partition (the first nine years), run two modified Mann-Kendall (MK) tests; one test for the scale yearly time series and another test for the shape yearly time series.
4. For the entire eleven years, run two modified MK tests; one test for the scale yearly time series and another test for the shape yearly time series.

The first three steps of the method directly above are implemented by the R script at the end of this section.

Table 67 includes an eleven-year time series of Weibull shape estimates for each station. The question can be addressed of whether or not each shape-estimate time series by examining the data in Table 68 which has statistics generated from original and modified MK tests. If the MK statistics suggest that a trend exists within the reader's toleration, then the reader may find the slope statistics in Table 67 helpful.

The scale estimates and MK statistics are in Tables 69 and 70.

Table 67: Weibull shape estimates and Sen's Slopes

												average		Sen's Slope		normalized to avg.	
station	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	9-year	11-year	9-year	11-year	9-year	11-year
KATL	0.40	0.36	0.39	0.39	0.38	0.44	0.48	0.30	0.47	0.34	0.41	0.40	0.39	0.010	0.003	2%	1%
KBOS	0.47	0.50	0.38	0.49	0.49	0.50	0.42	0.42	0.48	0.45	0.43	0.46	0.46	-0.003	-0.004	-1%	-1%
KBWI	0.33	0.41	0.43	0.45	0.42	0.36	0.47	0.36	0.35	0.42	0.32	0.40	0.39	-0.003	-0.004	-1%	-1%
KCLE	0.37	0.40	0.40	0.50	0.37	0.39	0.35	0.38	0.40	0.38	0.41	0.40	0.40	-0.001	0.001	0%	0%
KCLT	0.37	0.43	0.40	0.34	0.41	0.39	0.36	0.35	0.37	0.39	0.43	0.38	0.38	-0.007	0.000	-2%	0%
KCVG	0.41	0.47	0.45	0.42	0.36	0.36	0.40	0.35	0.35	0.41	0.44	0.40	0.40	-0.013	-0.006	-3%	-1%
KDCA	0.44	0.40	0.40	0.44	0.48	0.39	0.42	0.41	0.39	0.45	0.40	0.42	0.42	-0.004	-0.001	-1%	0%
KDEN	0.38	0.38	0.42	0.44	0.45	0.37	0.44	0.40	0.47	0.47	0.43	0.42	0.42	0.008	0.006	2%	1%
KDFW	0.45	0.47	0.49	0.40	0.45	0.44	0.43	0.43	0.48	0.45	0.46	0.45	0.45	-0.003	-0.001	-1%	0%
KDTW	0.39	0.37	0.40	0.31	0.40	0.42	0.45	0.46	0.42	0.38	0.40	0.40	0.40	0.009	0.004	2%	1%
KEUG	0.40	0.38	0.34	0.41	0.41	0.45	0.34	0.31	0.41	0.41	0.36	0.38	0.38	0.001	0.000	0%	0%
KIAH	0.41	0.38	0.36	0.41	0.39	0.37	0.41	0.37	0.36	0.35	0.40	0.39	0.38	-0.003	-0.002	-1%	-1%
KLAS	0.43	0.36	0.46	0.41	0.39	0.39	0.42	0.39	0.40	0.52	0.49	0.41	0.42	-0.002	0.005	0%	1%
KLAX	0.48	0.41	0.37	0.41	0.41	0.39	0.50	0.53	0.42	0.44	0.46	0.44	0.44	0.006	0.006	1%	1%
KLGA	0.43	0.45	0.52	0.52	0.40	0.43	0.47	0.41	0.40	0.44	0.52	0.45	0.45	-0.007	0.000	-2%	0%
KMCI	0.42	0.42	0.46	0.47	0.39	0.49	0.35	0.42	0.39	0.50	0.37	0.42	0.43	-0.003	-0.003	-1%	-1%
KMCO	0.34	0.42	0.37	0.40	0.40	0.34	0.43	0.36	0.39	0.38	0.34	0.38	0.38	0.002	-0.003	1%	-1%
KMSP	0.41	0.32	0.43	0.37	0.39	0.39	0.43	0.47	0.34	0.37	0.29	0.40	0.38	0.007	-0.003	2%	-1%
KORD	0.53	0.42	0.49	0.46	0.50	0.45	0.39	0.42	0.42	0.40	0.45	0.45	0.45	-0.012	-0.009	-3%	-2%
KPHL	0.42	0.44	0.44	0.52	0.37	0.45	0.43	0.40	0.47	0.45	0.49	0.44	0.44	0.002	0.004	0%	1%
KPHX	0.37	0.36	0.29	0.38	0.31	0.37	0.35	0.39	0.34	0.39	0.38	0.35	0.36	0.002	0.003	0%	1%
KPIT	0.43	0.39	0.42	0.42	0.39	0.39	0.41	0.37	0.39	0.34	0.38	0.40	0.39	-0.004	-0.005	-1%	-1%
KSAC	0.29	0.43	0.39	0.43	0.42	0.36	0.30	0.29	0.42	0.45	0.40	0.37	0.38	-0.001	0.000	0%	0%
KSAN	0.33	0.35	0.36	0.38	0.41	0.39	0.40	0.40	0.38	0.44	0.36	0.38	0.38	0.007	0.007	2%	2%
KSAT	0.40	0.44	0.37	0.41	0.46	0.44	0.32	0.46	0.37	0.41	0.45	0.41	0.41	0.000	0.001	0%	0%
KSEA	0.39	0.36	0.36	0.41	0.40	0.47	0.37	0.36	0.40	0.34	0.45	0.39	0.39	0.001	0.001	0%	0%
KSFO	0.54	0.49	0.36	0.56	0.42	0.50	0.52	0.45	0.57	0.51	0.53	0.49	0.50	0.005	0.003	1%	1%
KSMX	0.51	0.49	0.40	0.54	0.39	0.41	0.40	0.35	0.46	0.47	0.43	0.44	0.44	-0.013	-0.007	-3%	-2%
KSTL	0.39	0.35	0.33	0.43	0.42	0.39	0.32	0.34	0.39	0.32	0.35	0.37	0.37	-0.002	-0.004	-1%	-1%
KTPA	0.39	0.41	0.41	0.35	0.42	0.39	0.40	0.30	0.39	0.39	0.40	0.39	0.39	-0.003	-0.002	-1%	0%

Table 68: Original and modified Mann-Kendall statistics for shape estimates

	Z-Value		S		Kendall's Tau		Kendall's Tau Empirical Bootstrap CI		Z-value Empirical Bootstrap CI	
station	9-year	11-year	9-year	11-year	9-year	11-year	9-year	11-year	9-year	11-year
KATL	0.73	0.31	8	5	0.22	0.09	(-0.4444444,0.4166667)	(-0.3818182,0.3818182)	(-1.5067416,1.5811388)	(-1.492781,1.5794944)
KBOS	-0.52	-0.93	-6	-13	-0.17	-0.24	(-0.3888889,0.4444444)	(-0.3818182,0.3818182)	(-1.497786,1.6050968)	(-1.6499158,1.5762208)
KBWI	-0.10	-0.78	-2	-11	-0.06	-0.20	(-0.4444444,0.4166667)	(-0.3636364,0.3636364)	(-1.5811388,1.5990054)	(-1.5665209,1.5894388)

Table 68: (continued)

KCLE	-0.10	0.16	-2	3	-0.06	0.05	(-0.4166667,0.4444444)	(-0.3818182,0.3636364)	(-1.4895865,1.5811388)	(-1.5794944,1.6029951)
KCLT	-1.15	-0.16	-12	-3	-0.33	-0.05	(-0.4444444,0.4444444)	(-0.3818182,0.3636364)	(-1.483997,1.6050968)	(-1.5894388,1.5794944)
KCVG	-2.40	-1.25	-24	-17	-0.67	-0.31	(-0.4166667,0.4166667)	(-0.3818182,0.3636364)	(-1.483997,1.5811388)	(-1.5927956,1.723173)
KDCA	-0.73	-0.47	-8	-7	-0.22	-0.13	(-0.4444444,0.4444444)	(-0.4,0.3818182)	(-1.483997,1.6050968)	(-1.5762208,1.505221)
KDEN	1.15	1.40	12	19	0.33	0.35	(-0.4444444,0.4444444)	(-0.3818182,0.3818182)	(-1.630178,1.5429086)	(-1.5665209,1.505221)
KDFW	-0.73	0.00	-8	-1	-0.22	-0.02	(-0.4166667,0.4166667)	(-0.3818182,0.3818182)	(-1.5067416,1.483997)	(-1.6099045,1.5762208)
KDTW	2.19	1.09	22	15	0.61	0.27	(-0.4166667,0.4166667)	(-0.3818182,0.3818182)	(-1.5067416,1.483997)	(-1.5665209,1.6636654)
KEUG	0.10	0.00	2	1	0.06	0.02	(-0.4166667,0.4444444)	(-0.3818182,0.3636364)	(-1.6050968,1.5990054)	(-1.6499158,1.505221)
KIAH	-0.94	-1.09	-10	-15	-0.28	-0.27	(-0.4166667,0.4444444)	(-0.3636364,0.4)	(-1.5067416,1.5429086)	(-1.5794944,1.4958621)
KLAS	-0.10	1.09	-2	15	-0.06	0.27	(-0.4444444,0.4166667)	(-0.3636364,0.3818182)	(-1.5811388,1.5811388)	(-1.505221,1.537735)
KLAX	0.94	1.25	10	17	0.28	0.31	(-0.4166667,0.4444444)	(-0.3818182,0.3636364)	(-1.5811388,1.5067416)	(-1.5794944,1.5762208)
KLGA	-0.94	0.00	-10	1	-0.28	0.02	(-0.4444444,0.4444444)	(-0.3818182,0.3636364)	(-1.5811388,1.5811388)	(-1.6636654,1.5927956)
KMCI	-0.52	-0.31	-6	-5	-0.17	-0.09	(-0.4166667,0.4444444)	(-0.3636364,0.3818182)	(-1.5067416,1.4895865)	(-1.5794944,1.4489141)
KMCO	0.31	-0.31	4	-5	0.11	-0.09	(-0.3333333,0.3333333)	(-0.3636364,0.3818182)	(-0.9742786,1.1468293)	(-1.5665209,1.6499158)
KMSP	0.52	-0.47	6	-7	0.17	-0.13	(-0.4166667,0.4444444)	(-0.3818182,0.4)	(-1.5067416,1.483997)	(-1.505221,1.6099045)
KORD	-1.77	-1.71	-18	-23	-0.50	-0.42	(-0.4166667,0.4166667)	(-0.3636364,0.3818182)	(-1.5811388,1.5811388)	(-1.5794944,1.5794944)
KPHL	0.10	1.09	2	15	0.06	0.27	(-0.4166667,0.4444444)	(-0.3818182,0.3818182)	(-1.5067416,1.5067416)	(-1.492781,1.5762208)
KPHX	0.10	1.09	2	15	0.06	0.27	(-0.4166667,0.4444444)	(-0.3818182,0.3818182)	(-1.5811388,1.5067416)	(-1.6029951,1.5665209)
KPIT	-2.19	-2.80	-22	-37	-0.61	-0.67	(-0.4444444,0.4166667)	(-0.3636364,0.3818182)	(-1.5429086,1.5067416)	(-1.537735,1.5665209)
KSAC	-0.73	0.00	-8	1	-0.22	0.02	(-0.4166667,0.4444444)	(-0.3818182,0.3818182)	(-1.5811388,1.5685187)	(-1.6499158,1.6099045)
KSAN	2.19	1.87	22	25	0.61	0.45	(-0.4166667,0.4444444)	(-0.4,0.3818182)	(-1.6431677,1.5067416)	(-1.5762208,1.6533212)
KSAT	0.10	0.31	0	5	0.00	0.09	(-0.4166667,0.4166667)	(-0.3818182,0.3818182)	(-1.5685187,1.5811388)	(-1.6499158,1.5794944)
KSEA	0.10	0.00	2	1	0.06	0.02	(-0.4444444,0.4444444)	(-0.3818182,0.4)	(-1.483997,1.5811388)	(-1.6499158,1.505221)
KSFO	0.52	0.78	6	11	0.17	0.20	(-0.4444444,0.4166667)	(-0.4,0.3818182)	(-1.6050968,1.5685187)	(-1.505221,1.6499158)
KSMX	-1.36	-0.78	-14	-11	-0.39	-0.20	(-0.4444444,0.4722222)	(-0.3818182,0.3818182)	(-1.5811388,1.5811388)	(-1.6499158,1.5762208)
KSTL	-0.52	-0.93	-6	-13	-0.17	-0.24	(-0.4166667,0.4166667)	(-0.3818182,0.3818182)	(-1.6050968,1.5429086)	(-1.6601958,1.5794944)
KTPA	-0.94	-0.62	-10	-9	-0.28	-0.16	(-0.4444444,0.4166667)	(-0.3636364,0.3636364)	(-1.483997,1.5990054)	(-1.6499158,1.5244784)

Table 69: Weibull scale estimates and Sen's Slopes

station	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	9-year	11-year	9-year	11-year	9-year	11-year
KATL	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.00	0.02	0.01	0.01	0.01	0.01	0.0009	0.0003	7%	3%
KBOS	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.0003	-0.0004	-2%	-3%
KBWI	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	-0.0002	-0.0002	-1%	-2%
KCLE	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0000	0.0000	0%	0%
KCLT	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	-0.0005	0.0000	-4%	0%
KCVG	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.0008	-0.0003	-7%	-2%
KDCA	0.02	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	-0.0003	-0.0001	-2%	-1%
KDEN	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0003	0.0004	3%	4%
KDFW	0.01	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.01	0.01	-0.0001	0.0001	-1%	0%

Table 69: (continued)

KDTW	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0007	0.0002	7%	2%
KEUG	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.00	0.01	0.02	0.01	0.01	0.01	-0.0003	-0.0001	-3%	-1%
KIAH	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	-0.0001	0.0001	0%	1%
KLAS	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.01	0.01	-0.0002	0.0003	-2%	3%
KLAX	0.02	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.01	0.02	0.02	0.01	0.01	0.0000	0.0003	0%	2%
KLGA	0.01	0.01	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	-0.0004	-0.0001	-3%	0%
KMCI	0.01	0.01	0.02	0.02	0.01	0.02	0.01	0.01	0.01	0.02	0.01	0.01	0.01	-0.0004	-0.0003	-3%	-2%
KMCO	0.00	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.0005	0.0000	5%	0%
KMSP	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.00	0.01	0.01	0.0004	-0.0003	3%	-3%
KORD	0.02	0.01	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.02	0.01	-0.0009	-0.0007	-6%	-5%
KPHL	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	-0.0001	0.0003	-1%	2%
KPHX	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0000	0.0003	0%	4%
KPIT	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.0002	-0.0003	-2%	-3%
KSAC	0.00	0.02	0.01	0.02	0.01	0.01	0.01	0.00	0.01	0.02	0.01	0.01	0.01	-0.0005	0.0001	-5%	1%
KSAN	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.0006	0.0006	5%	4%
KSAT	0.01	0.02	0.01	0.01	0.02	0.01	0.01	0.02	0.01	0.01	0.02	0.01	0.01	-0.0003	0.0000	-2%	0%
KSEA	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.0000	0.0000	0%	0%
KSFO	0.02	0.02	0.01	0.02	0.01	0.01	0.02	0.01	0.02	0.01	0.02	0.01	0.01	0.0002	0.0002	2%	1%
KSMX	0.02	0.02	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	-0.0010	-0.0005	-7%	-4%
KSTL	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.0004	-0.0004	-4%	-5%
KTPA	0.01	0.01	0.01	0.01	0.02	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.0002	0.0002	2%	1%

Table 70: Original and modified Mann-Kendall statistics for scales estimates

	Z-Value		S		Kendall's Tau		Kendall's Tau Empirical Bootstrap CI		Z-value Empirical Bootstrap CI	
station	9-year	11-year	9-year	11-year	9-year	11-year	9-year	11-year	9-year	11-year
KATL	1.36	0.78	14	11	0.39	0.20	(-0.4444444,0.4166667)	(-0.4,0.3818182)	(-1.5067416,1.483997)	(-1.6099045,1.5794944)
KBOS	-0.31	-0.78	-4	-11	-0.11	-0.20	(-0.4444444,0.4444444)	(-0.3818182,0.3818182)	(-1.483997,1.5811388)	(-1.6499158,1.5794944)
KBWI	-0.10	-0.47	-2	-7	-0.06	-0.13	(-0.4444444,0.4444444)	(-0.3818182,0.3818182)	(-1.483997,1.5067416)	(-1.6398245,1.5894388)
KCLE	0.10	0.31	0	5	0.00	0.09	(-0.4166667,0.4166667)	(-0.3818182,0.3636364)	(-1.5305651,1.5067416)	(-1.5762208,1.505221)
KCLT	-1.15	0.16	-12	3	-0.33	0.05	(-0.4166667,0.4166667)	(-0.3818182,0.3818182)	(-1.5305651,1.5811388)	(-1.5995735,1.5794944)
KCVG	-1.98	-0.62	-20	-9	-0.56	-0.16	(-0.4166667,0.4444444)	(-0.3636364,0.3818182)	(-1.5429086,1.5811388)	(-1.505221,1.5665209)
KDCA	-1.36	-0.31	-14	-5	-0.39	-0.09	(-0.4166667,0.4166667)	(-0.3818182,0.3818182)	(-1.6050968,1.483997)	(-1.5762208,1.505221)
KDEN	0.94	1.25	10	17	0.28	0.31	(-0.4166667,0.4166667)	(-0.3636364,0.3636364)	(-1.5811388,1.483997)	(-1.5894388,1.5179774)
KDFW	-0.31	0.31	-4	5	-0.11	0.09	(-0.4444444,0.4444444)	(-0.3818182,0.3636364)	(-1.5429086,1.5685187)	(-1.505221,1.505221)
KDTW	1.77	0.62	18	9	0.50	0.16	(-0.4444444,0.4444444)	(-0.3636364,0.3818182)	(-1.5429086,1.5067416)	(-1.505221,1.5665209)
KEUG	-0.52	-0.31	-6	-5	-0.17	-0.09	(-0.4444444,0.4166667)	(-0.3636364,0.3636364)	(-1.483997,1.5811388)	(-1.5762208,1.5762208)
KIAH	0.10	0.31	0	5	0.00	0.09	(-0.4444444,0.4166667)	(-0.3818182,0.3818182)	(-1.483997,1.5811388)	(-1.505221,1.5762208)
KLAS	-0.52	0.78	-6	11	-0.17	0.20	(-0.4166667,0.4166667)	(-0.4,0.3818182)	(-1.5305651,1.5811388)	(-1.5665209,1.5894388)
KLAX	0.10	0.47	0	7	0.00	0.13	(-0.4166667,0.4166667)	(-0.3636364,0.3818182)	(-1.6050968,1.4895865)	(-1.5653199,1.505221)

Table 70: (continued)

KLGA	-1.36	-0.16	-14	-3	-0.39	-0.05	(-0.4444444,0.4166667)	(-0.3818182,0.3636364)	(-1.5067416,1.6237976)	(-1.5762208,1.5762208)
KMCI	-0.94	-0.62	-10	-9	-0.28	-0.16	(-0.4444444,0.4166667)	(-0.3818182,0.3818182)	(-1.5990054,1.5067416)	(-1.5762208,1.537735)
KMCO	0.31	0.00	4	1	0.11	0.02	(-0.4444444,0.4166667)	(-0.4,0.3636364)	(-1.5811388,1.5067416)	(-1.5179774,1.5762208)
KMSP	0.73	-0.31	8	-5	0.22	-0.09	(-0.4444444,0.4166667)	(-0.3818182,0.4)	(-1.5811388,1.5811388)	(-1.5894388,1.6099045)
KORD	-1.98	-2.18	-20	-29	-0.56	-0.53	(-0.4166667,0.4444444)	(-0.3818182,0.3818182)	(-1.5429086,1.483997)	(-1.5762208,1.6499158)
KPHL	-0.10	0.93	-2	13	-0.06	0.24	(-0.4444444,0.4444444)	(-0.4,0.3636364)	(-1.5067416,1.5685187)	(-1.505221,1.492781)
KPHX	0.10	0.93	0	13	0.00	0.24	(-0.4444444,0.4166667)	(-0.3818182,0.4)	(-1.5067416,1.5811388)	(-1.5665209,1.5927956)
KPIT	-0.94	-1.71	-10	-23	-0.28	-0.42	(-0.4166667,0.4166667)	(-0.3818182,0.3818182)	(-1.5067416,1.5305651)	(-1.5794944,1.6499158)
KSAC	-0.73	0.00	-8	1	-0.22	0.02	(-0.4444444,0.4166667)	(-0.3818182,0.3636364)	(-1.5811388,1.6050968)	(-1.505221,1.5653199)
KSAN	1.56	1.40	16	19	0.44	0.35	(-0.4444444,0.4166667)	(-0.3636364,0.3818182)	(-1.5067416,1.5429086)	(-1.5762208,1.5310636)
KSAT	-0.52	0.00	-6	-1	-0.17	-0.02	(-0.4166667,0.4444444)	(-0.3636364,0.3818182)	(-1.5067416,1.5811388)	(-1.6029951,1.5244784)
KSEA	0.10	0.00	0	1	0.00	0.02	(-0.4444444,0.4166667)	(-0.3636364,0.3818182)	(-1.5429086,1.4895865)	(-1.5653199,1.505221)
KSFO	0.31	0.62	4	9	0.11	0.16	(-0.4444444,0.4444444)	(-0.3818182,0.3636364)	(-1.5429086,1.5811388)	(-1.5665209,1.505221)
KSMX	-1.77	-1.09	-18	-15	-0.50	-0.27	(-0.4166667,0.4444444)	(-0.3818182,0.3818182)	(-1.5811388,1.483997)	(-1.505221,1.6499158)
KSTL	-0.94	-1.71	-10	-23	-0.28	-0.42	(-0.4444444,0.4166667)	(-0.4,0.3818182)	(-1.5067416,1.5429086)	(-1.5894388,1.5794944)
KTPA	0.31	0.62	4	9	0.11	0.16	(-0.4444444,0.4166667)	(-0.3818182,0.3636364)	(-1.5305651,1.5811388)	(-1.505221,1.6099045)

An R script generating trend statistics for the first nine years of data follows.

Listing 3: R script generating trend statistics for the first nine years of data

```
stations <- c("KATL",
"KBOS",
"KBWI",
"KCLE",
"KCLT",
"KCVG",
"KDCA",
"KDEN",
"KDFW",
"KDTW",
"KEUG",
"KIAH",
"KLAS",
"KLAX",
"KLGA",
"KMCI",
"KMCO",
"KMSP",
"KORD",
"KPHL",
"KPHX",
"KPIT",
"KSAC",
"KSAN",
"KSAT",
"KSEA",
"KSFO",
"KSMX",
"KSTL",
"KTPA")

#stations <- c("KATL")

library("fitdistrplus")
library(modifiedmk)


for(station in stations) {

 inputFilenameBase = paste("training", station, "2004-2012in", sep="
    ")
         inputFilename = paste(inputFilenameBase, ".csv", sep="")
         outputFilename = paste(inputFilenameBase, "
```

```
TrendStatisticsModifiedMK.txt", sep="")
DF <- read.csv(inputFilename, header=TRUE)
 iter <- 9+1 #for header
m <- matrix(NA, nrow=iter, ncol=7)
 rowIndex = 1
m[rowIndex,] <- c("station", "year", "shape", "scale", "Kolmogorov-Smirnov statistic","Cramer-von Mises statistic", "Anderson-Darling statistic")
 rowIndex=rowIndex+1
DF <- read.csv(inputFilename, header=TRUE)
 for (year in 2004:2012){
  print(year)
  prex = DF[as.numeric(substr(DF$timestamp, 0, 4)) == year,]$wind_knots
        x = hist(prex, freq=FALSE, plot=FALSE, breaks=seq(-1,max(DF$wind_knots)),main=paste("Histogram of", inputFilename), xlab="windspeed (knots)")
  #If a zero exists, truncate data to avoid the following error:
  ## Error in checkparamlist(arg_startfix$start.arg, arg_startfix$fix.arg, :
  ##  'start' should not have NA or NaN values.
  finalIndex = if(x$density[which.min(x$density)] == 0) which.min(x$density)-1 else length(x$density)
  d = fitdist(x$density[1:finalIndex], "weibull",lower=0)
  k = gofstat(d)
 m[rowIndex,] <- c(station, year, d$estimate, k$ks, k$cvm, k$ad)
  rowIndex=rowIndex+1
 }
 print(station)
# print to file
 options(width = 200)
 sink(outputFilename)
 print(station)
 print(m)
 shape = m[,3]
 print("ModifiedMKshape")
# The 2 index in the shape array omits heading row
```

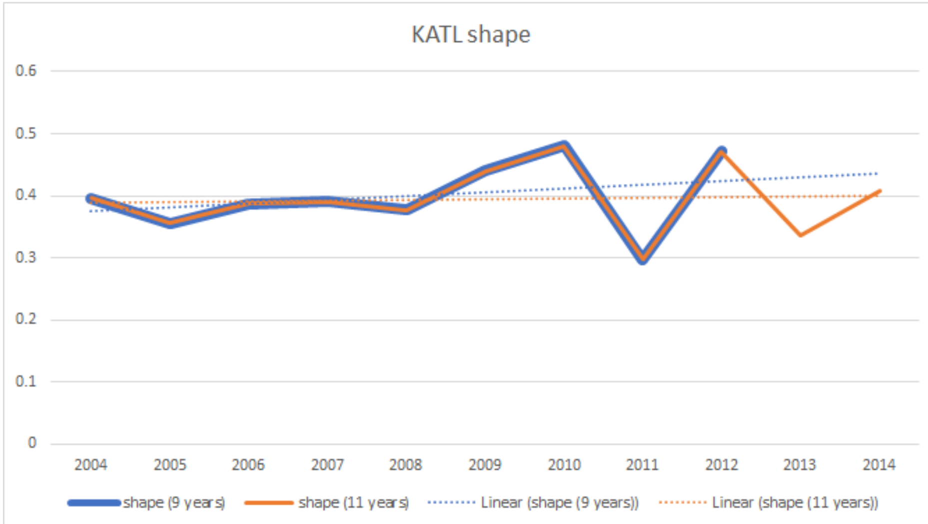

Figure 46: Shape estimates of Weibull distributions for each year of windspeeds for station KATL with 9-year and 11-year trend lines

```
bbsmk(as.numeric(shape[2:length(shape)]), ci=0.95, nsim=2000, eta
    =1, bl.len=NULL)

print("ModifiedMKscale")
scale = m[,4]
# The 2 index in the scale array omits heading row
bbsmk(as.numeric(scale[2:length(shape)]), ci=0.95, nsim=2000, eta
    =1, bl.len=NULL)

sink()

}
```

Plots for estimates of the Weibull shape and scale parameters for station KATL are shown in Figs. 46 and 47, which show trend lines for the first nine years of data and the entire eleven years. In both cases, the 9-year trend lines have a greater slope than the 11-year trend lines.

However, does a modified MK test suggest that trends exist for station KATL? Let us work through the following example to determine whether or not to behave as if there is a trend there. We shall walk through the following steps:

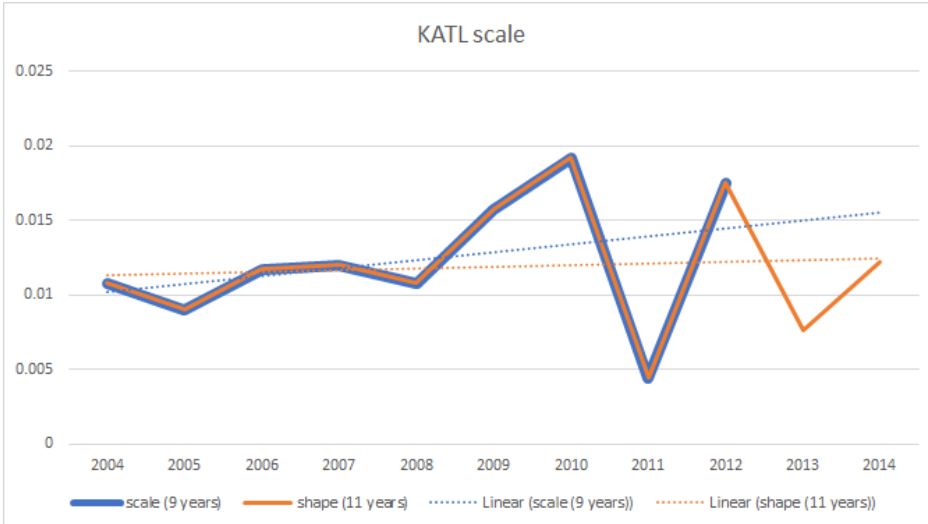

Figure 47: Scale estimates of Weibull distributions for each year of windspeeds for station KATL with 9-year and 11-year trend lines

- "[Make] an initial assumption.
- "[Collect] evidence (data).
- "Based on the available evidence (data), [decide] whether to reject or not reject the initial assumption" [15].

In the context of the p-value approach, the steps immediately above are realized by the following four steps:

1. "Specify the null and alternative hypotheses" [15].
    - The null hypothesis H_0 is KATL's Weibull scale parameter exhibits no yearly monotonic trend.
    - The alternative hypothesis H_A is KATL's Weibull scale parameter exhibits a yearly monotonic trend.
2. "Using the sample data and assuming the null hypothesis is true, calculate the value of the test statistic" [15]. To calculate the value of the test statistic, we passed nine estimates of the Weibull scales (one estimate for each of the nine years of training data) for station KATL to the R function `bbsmk()` in the context of the R script shown in Listing 3 on

page 276. That script shows that the parameter "ci=0.95" is passed to `bbsmk()`. Thus, the returned confidence interval will contain a value that applies to the entire population 95% of the time. (For example, if we had access to the entire population of windspeeds for station KATL for all years and if we could measure the windspeed with 100% accuracy and if we could somehow not affect the wind by our measurements, then we would be able to make with 100% accuracy statements about the entire population of windspeeds because we would have access to the entire population of KATL's windpseeds, which is data from every year in history.) The function `bbsmk()`returned (-1.5067416,1.483997) as the Z-value empirical bootstrap confidence interval for KATL's scale (and returned other statistical information for KATL and the other 29 stations as shown in Table 70). Thus, we have 95% confidence that the Z-value of the modified MK test of the scale of the Weibull estimate for entire population of windspeeds around KATL is in the range (-1.5067416,1.483997).

3. "Using the known distribution of the test statistic, calculate the **P-value**: 'If the null hypothesis is true, what is the probability that we'd observe a more extreme test statistic in the direction of the alternative hypothesis than we did?' [15]. We are assuming that the distribution of the test statistic is normal. If the Z-score is at the lowest part of that range, namely at -1.51 rounding to two decimal places, then the Z-score's corresponding P-value is 0.06552 according to a Z-score table that gives the area to the left of the Z-score. If the Z-score is the highest part of that range, namely at 1.48 rounding to two decimal places, then the Z-score's corresponding P-value is $1 - .93056 = 0.06944$. Thus, the Z-score confidence interval corresponds to the two P-values: 0.06552 and 0.06944. Of those two P-values, we are interested in the lower, 0.07 rounded to two decimal places, in the next step.

4. "Set the significance level, A, the probability of making a Type I error to be small — 0.01, 0.05, or 0.10" [15]. We would commit a Type I error if we reject the null hypothesis when we the null hypothesis is actually true [15, Table S.3.2]. We examine two cases for the significance level. "Compare the P-value to A. If the P-value is less than (or equal to) A, reject the null hypothesis in favor of the alternative hypothesis. If the P-value is greater than A, do not reject the null hypothesis"[15].

- If we set the significance level A to 0.05, then the P-value of 0.07 is larger than A and we would not reject the null hypothesis thereby behaving as if no trend exists in KATL's scale estimates.
- If we loosen the significance level A to 0.10 (increasing the probability of making a Type I error), then the P-value of 0.07 is smaller than A and we would reject the null hypothesis thereby behaving as if a trend exists in KATL scale estimates, examining the Sen's slope of the nine-year series of scale estimates (Table 69).

Now that we have examined the 9-year scale data for KATL, let us repeat the process for KATL's 9-year shape data.

1. - The null hypothesis H_0 is KATL's Weibull shape parameter exhibits no yearly monotonic trend.
   - The alternative hypothesis H_A is KATL's Weibull shape parameter exhibits a yearly monotonic trend.
2. As we did for the scale parameter above, we calculate the test statistic for the shape data. The R function `bbsmk()` returned $(-1.5067416, 1.5811388)$ as the 95% confidence interval for the Z-value empirical bootstrap (Table 68 on page 272).
3. Assuming that the Z-value distribution is normal, we calculate the P-values: The lowest part of the Z-score range, -1.51 rounding to two decimal places, corresponds to the P-value 0.06552 according to a Z-score table that gives the area to the left of the Z-score. The highest part of the Z-score range, 1.58, maps to the P-value $1 - 0.94295 = 0.05705$. The lower of the two P-values is $0.05705 \approx 0.06$.
4. We choose two significance levels, A. We compare the lower of the two P-values to the two choices for A:
   - If we set the significance level A to 0.05, then the P-value of 0.06 is larger than A and we would not reject the null hypothesis thereby behaving as if no trend exists in KATL's shape estimates.
   - If we loosen the significance level A to 0.10 (increasing the probability of making a Type I error), then the P-value of 0.06 is smaller than A and we would reject

the null hypothesis thereby behaving as if a trend exists in KATL shape estimates, examining the Sen's slope of the nine-year series of shape estimates (Table 67).

A.19 LOOKING FOR TRENDS IN HOURLY ELECTRICITY PRICE DATA

In Section 4.1.6.3, we raised the possibility that a trend could be developing where the second peak's dominance of the first peak is growing. In this section, we explore that possibility using the same procedure we used in Appendix A.18:

1. - Let the null hypothesis H_0 be that the proportions plotted in Figure 12 on page 66 do *not* exhibit a yearly monotonic trend.
   - The alternative hypothesis H_A be that they do.
2. As we did in Appendix A.18, we calculate the test statistic for the proportions plotted in Figure 12 on page 66. As shown in Listing 4, the R function `bbsmk()` returned $(-1.6431677, 1.596677)$ as the 95% confidence interval for the Z-value empirical bootstrap (Line 15).
3. Assuming that the Z-value distribution is normal, we calculate the P-values: The lowest part of the Z-score range, -1.64 rounding to two decimal places, corresponds to the P-value 0.05050258 according to R's `pnorm()` function (which gives the area to the left of the Z-score). The highest part of the Z-score range, 1.61, maps to the P-value $1-0.9452007 = 0.05479929$. The lower of the two P-values is $0.05050258 \approx 0.051$.
4. We choose two significance levels, A. We compare the lower of the two P-values to the two choices for A:
   - If we set the significance level A to 0.05, then the P-value of 0.051 is larger than A and we would not reject the null hypothesis thereby behaving as if no trend exists in yearly proportions.
   - If we loosen the significance level A to 0.10 (increasing the probability of making a Type I error), then the P-value of 0.051 is smaller than A and we would reject the null hypothesis thereby behaving as if a trend exists in the yearly proportions,

examining the Sen's slope of fifteen-year series of proportions, which is 0.025, as shown in Line 7 of Listing 4.

Note the re-running `bbsmk()` results in different returned intervals in Lines 34, 57, 60, respectively, than Line 15: (-1.606034,1.6394097), (-1.6391088,1.6431677), and (-1.6006671,1.5670696). Those intervals mapped to no P-values less than 0.05 (Lines 62 to 72.)

Listing 4: R commands to calculate trend statistics for proportions of electricity price data plotted in Figure 12

```
1  > library(modifiedmk)
2  > proportions = c(1.01, 1.06, 1.02, 0.98, 0.97, 0.98, 0.95, 1.08,
       1.05, 1.20, 1.20, 1.29, 0.99, 1.65, 1.36)
3  > bbsmk(proportions, ci=0.95, nsim=2000, eta=1, bl.len=NULL)
4  Z-Value =
5  2.2820051
6  Sen's Slope =
7  0.025
8  S =
9  47
10 Kendall's Tau =
11 0.447619047619048
12 Kendall's Tau Empirical Bootstrapped CI =
13 (-0.3238095,0.3142857)
14 Z-value Empirical Bootstrapped CI =
15 (-1.6431677,1.596677)>
16 > pnorm(-1.64)
17 [1] 0.05050258
18 > pnorm(1.60)
19 [1] 0.9452007
20 > 1-pnorm(1.60)
21 [1] 0.05479929
22 > results = bbsmk(proportions, ci=0.95, nsim=2000, eta=1, bl.len=
      NULL)
23 Z-Value =
24 2.2820051
25 Sen's Slope =
26 0.025
27 S =
28 47
29 Kendall's Tau =
30 0.447619047619048
31 Kendall's Tau Empirical Bootstrapped CI =
```

```
(-0.3333333,0.3047619)
Z-value Empirical Bootstrapped CI =
(-1.606034,1.6394097)>
> results = bbsmk(proportions, ci=0.95, nsim=2000, eta=1, bl.len=
    NULL)
Z-Value =
2.2820051
Sen's Slope =
0.025
S =
47
Kendall's Tau =
0.447619047619048
Kendall's Tau Empirical Bootstrapped CI =
(-0.3047619,0.3047619)
Z-value Empirical Bootstrapped CI =
(-1.6391088,1.6431677)>
> results = bbsmk(proportions, ci=0.95, nsim=2000, eta=1, bl.len=
    NULL)
Z-Value =
2.2820051
Sen's Slope =
0.025
S =
47
Kendall's Tau =
0.447619047619048
Kendall's Tau Empirical Bootstrapped CI =
(-0.3142857,0.2952381)
Z-value Empirical Bootstrapped CI =
(-1.6006671,1.5670696)>
> pnorm(-1.606034)
[1] 0.05413322
> 1-pnorm(1.6394097)
[1] 0.05056398
> pnorm(-1.6391088)
[1] 0.0505953
> 1-pnorm(1.6431677)
[1] 0.05017412
> pnorm(-1.6006671)
[1] 0.05472534
> 1-pnorm(1.5670696)
[1] 0.05854921
```

Listing 5: R commands to calculate trend statistics for monotonic increasing data

```
> test
```

```
 [1]  1.01  2.01  3.01  4.01  5.01  6.01  7.01  8.01  9.01 10.01
     11.01 12.01 13.01 14.01 15.01
> bbsmk(test, ci=0.95, nsim=2000, eta=1, bl.len=NULL)
Z-Value =
5.1466653
Sen's Slope =
1
S =
105
Kendall's Tau =
1
Kendall's Tau Empirical Bootstrapped CI =
(-0.3333333,0.4761905)
Z-value Empirical Bootstrapped CI =
(-1.7903285,2.3028804)
```

A.20 ALGORITHM PERFORMANCE PER STATION PER OLA

Recall that algorithms are trained to try to maximize the metric by which the algorithm's performance will be measured. Odd-numbered OLAs measure performance using NetNorm. Even-numbered OLAs use MQNetNorm. Thus, an algorithm may achieve different scores for odd-numbered OLAs than it does for even-numbered OLAs because training an algorithm in the context of odd-numbered OLAs may chose different settings (e.g., running-average-window size) than the training chooses for the algorithm in the context of even-numbered OLAs.

Tables 71, 72, and 73 show algorithm performance per station for OLAs 1 and 2, OLAs 3 and 4, and OLAs 5 and 6, respectively. Each row is a unique OLA-and-station pair. For each row, the column of the highest scoring algorithm is marked with a capital X. Rows having multiple X's indicate that ties occurred. (Trained fuzzy variants 0x0 and 0x2 tie for OLAs 1 - 4.) Each row's highest score is in either the NetNorm column (odd OLAs) or the MQNetNorm column (even OLAs). The performance of each row's applicable non-highest algorithms is given by a negative value x, which is how many points the non-highest algorithm's score y is less the row's highest score z, i.e., $x = y - z$. Each non-applicable cell is indicated by a hyphen.

Table 71: Highest scoring algorithms for each station for OLAs 1 and 2

Legend: 'X' indicates algorithm(s) achieving best NetNorm or MQNetNorm where applicable.
'-' indicates non-applicability.
Negative value indicates algorithm's score less best.

OLA	Station	Best NetNorm	Best MQNetNorm	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)
1	KATL	0.57	-	X	-	X	-	X	-	-0.16	-	-0.04	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KATL	-	0.52	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.02	-	X	-	-0.03	-	-0.06	-	-0.07	-
1	KBOS	0.38	-	X	-	X	-	X	-	-0.13	-	-0.04	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KBOS	-	0.35	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.01	-	X	-	-0.01	-	-0.04	-	-0.06	-
1	KBWI	0.69	-	X	-	X	-	-0.02	-	-0.17	-	-0.07	-	-0.09	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KBWI	-	0.63	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.04	-	X	-	-0.05	-	-0.08	-	-0.1	-
1	KCLE	0.61	-	-0.02	-	-0.02	-	-0.02	-	-0.04	-	-0.07	-	-0.09	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KCLE	-	0.54	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.01	-	X	-	-0.01	-	-0.04	-	-0.06	-
1	KCLT	0.7	-	X	-	X	-	X	-	-0.08	-	-0.03	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KCLT	-	0.63	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.01	-	X	-	-0.01	-	-0.05	-	-0.07	-
1	KCVG	0.66	-	X	-	X	-	-0.01	-	-0.12	-	-0.07	-	-0.09	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KCVG	-	0.6	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.02	-	X	-	-0.02	-	-0.07	-	-0.09	-
1	KDCA	0.5	-	X	-	X	-	X	-	-0.07	-	-0.05	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KDCA	-	0.46	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.01	-	X	-	-0.01	-	-0.05	-	-0.06	-
1	KDEN	0.56	-	-0.02	-	-0.02	-	X	-	-0.14	-	-0.09	-	-0.12	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KDEN	-	0.47	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.02	-	X	-	-0.02	-	-0.04	-	-0.07	-
1	KDFW	0.47	-	X	-	-0.01	-	X	-	-0.08	-	-0.04	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KDFW	-	0.44	-	-	-	-	-	-	-	-	-	-	-	-	-0.01	-	-0.04	-	X	-	-0.02	-	-0.04	-	-0.05	-
1	KDTW	0.6	-	X	-	-0.01	-	-0.03	-	-0.1	-	-0.05	-	-0.08	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KDTW	-	0.55	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.02	-	X	-	-0.02	-	-0.05	-	-0.07	-
1	KEUG	0.74	-	-0.01	-	-0.01	-	X	-	-0.13	-	-0.06	-	-0.1	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KEUG	-	0.66	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.02	-	X	-	-0.02	-	-0.04	-	-0.08	-
1	KIAH	0.72	-	X	-	X	-	-0.02	-	-0.13	-	-0.06	-	-0.09	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KIAH	-	0.66	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.03	-	X	-	-0.03	-	-0.06	-	-0.09	-
1	KLAS	0.45	-	-0.01	-	-0.01	-	X	-	-0.13	-	-0.03	-	-0.05	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KLAS	-	0.39	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.05	-	X	-	-0.02	-	-0.01	-	-0.02	-
1	KLAX	0.88	-	X	-	X	-	X	-	-0.09	-	-0.07	-	-0.11	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KLAX	-	0.8	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.01	-	X	-	-0.02	-	-0.04	-	-0.09	-
1	KLGA	0.42	-	X	-	X	-	X	-	-0.2	-	-0.04	-	-0.05	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KLGA	-	0.38	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.01	-	X	-	-0.02	-	-0.04	-	-0.05	-
1	KMCI	0.66	-	X	-	X	-	X	-	-0.09	-	-0.04	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KMCI	-	0.62	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.03	-	X	-	-0.02	-	-0.04	-	-0.06	-
1	KMCO	0.66	-	X	-	X	-	X	-	-0.11	-	-0.08	-	-0.1	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KMCO	-	0.6	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.03	-	X	-	-0.04	-	-0.07	-	-0.09	-
1	KMSP	0.38	-	X	-	X	-	-0.02	-	-0.06	-	-0.02	-	-0.03	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KMSP	-	0.35	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.02	-	X	-	-0.03	-	-0.03	-	-0.04	-
1	KORD	0.6	-	-0.01	-	-0.01	-	-0.01	-	-0.06	-	-0.06	-	-0.09	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KORD	-	0.55	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.01	-	X	-	-0.01	-	-0.06	-	-0.08	-

Table 71: (continued)

				Legend: 'X' indicates algorithm(s) achieving best NetNorm or MQNetNorm where applicable. '-' indicates non-applicability. Negative value indicates algorithm's score less best.																											
OLA	Station	Best NetNorm	Best MQNetNorm	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)
1	KPHL	0.65	-	X	-	X	-	-0.01	-	-0.1	-	-0.06	-	-0.08	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KPHL	-	0.6	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.03	-	X	-	-0.03	-	-0.06	-	-0.08	-
1	KPHX	0.37	-	-0.01	-	-0.01	-	X	-	-0.09	-	-0.03	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KPHX	-	0.31	-	-	-	-	-	-	-	-	-	-	-	-	-0.01	-	-0.02	-	X	-	-0.02	-	-0.08	-	-0.09	-
1	KPIT	0.74	-	X	-	X	-	-0.02	-	-0.14	-	-0.07	-	-0.1	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KPIT	-	0.67	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.03	-	X	-	-0.03	-	-0.07	-	-0.09	-
1	KSAC	0.78	-	X	-	X	-	-0.01	-	-0.17	-	-0.04	-	-0.1	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KSAC	-	0.72	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.03	-	-0.01	-	-0.03	-	-0.06	-	-0.11	-
1	KSAN	0.72	-	X	-	X	-	X	-	-0.08	-	-0.04	-	-0.07	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KSAN	-	0.66	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.01	-	X	-	-0.01	-	-0.04	-	-0.07	-
1	KSAT	0.53	-	X	-	X	-	-0.01	-	-0.19	-	-0.04	-	-0.05	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KSAT	-	0.49	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.01	-	X	-	-0.01	-	-0.05	-	-0.06	-
1	KSEA	0.66	-	X	-	X	-	X	-	-0.13	-	-0.06	-	-0.08	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KSEA	-	0.6	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.02	-	X	-	-0.02	-	-0.05	-	-0.07	-
1	KSFO	0.68	-	X	-	X	-	X	-	-0.07	-	-0.07	-	-0.09	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KSFO	-	0.63	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.03	-	X	-	-0.09	-	-0.04	-	-0.05	-
1	KSMX	0.8	-	X	-	X	-	X	-	-0.11	-	-0.1	-	-0.16	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KSMX	-	0.72	-	-	-	-	-	-	-	-	-	-	-	-	-0.01	-	-0.02	-	X	-	-0.01	-	-0.07	-	-0.14	-
1	KSTL	0.53	-	X	-	X	-	X	-	-0.07	-	-0.04	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KSTL	-	0.48	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.01	-	X	-	-0.02	-	-0.04	-	-0.05	-
1	KTPA	0.67	-	-0.01	-	-0.01	-	-0.01	-	-0.11	-	-0.03	-	-0.07	-	-	-	-	-	-	-	-	-	-	-	-	-
2	KTPA	-	0.59	-	-	-	-	-	-	-	-	-	-	-	-	-0.01	-	-0.02	-	X	-	-0.02	-	-0.05	-	-0.08	-
				23	-	21	-	18	-	0	-	0	-	0	-	26	-	0	-	29	-	0	-	0	-	0	-

Table 72: Highest scoring algorithms for each station for OLAs 3 and 4

Legend: ‘X’ indicates algorithm(s) achieving best NetNorm or MQNetNorm where applicable.
‘-’ indicates non-applicability.
Negative value indicates algorithm’s score less best.

OLA	Station	Best NetNorm	Best MQNetNorm	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)
3	KATL	0.44	-	-0.05	-	-0.09	-	-0.08	-	-0.11	-	-0.07	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KATL	-	0.39	-	-	-	-	-	-	-	-	-	-	-	-	-0.03	-	-0.07	-	-0.06	-	-0.09	-	-0.06	-	X	-
3	KBOS	0.34	-	X	-	-0.01	-	-0.02	-	-0.04	-	-0.01	-	-0.02	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KBOS	-	0.32	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.03	-	-0.02	-	-0.04	-	-0.02	-	-0.03	-
3	KBWI	0.55	-	-0.05	-	-0.1	-	-0.1	-	-0.15	-	-0.09	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KBWI	-	0.49	-	-	-	-	-	-	-	-	-	-	-	-	-0.04	-	-0.08	-	-0.08	-	-0.15	-	-0.07	-	X	-
3	KCLE	0.41	-	-0.07	-	-0.09	-	-0.03	-	-0.13	-	-0.09	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KCLE	-	0.38	-	-	-	-	-	-	-	-	-	-	-	-	-0.06	-	-0.1	-	-0.03	-	-0.12	-	-0.08	-	X	-
3	KCLT	0.5	-	-0.12	-	-0.14	-	-0.07	-	-0.14	-	-0.09	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KCLT	-	0.44	-	-	-	-	-	-	-	-	-	-	-	-	-0.09	-	-0.13	-	-0.05	-	-0.16	-	-0.06	-	X	-
3	KCVG	0.42	-	-0.08	-	-0.1	-	-0.02	-	-0.15	-	-0.04	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KCVG	-	0.38	-	-	-	-	-	-	-	-	-	-	-	-	-0.07	-	-0.11	-	-0.04	-	-0.12	-	-0.05	-	X	-
3	KDCA	0.42	-	-0.05	-	-0.07	-	-0.05	-	-0.14	-	-0.04	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KDCA	-	0.38	-	-	-	-	-	-	-	-	-	-	-	-	-0.03	-	-0.06	-	-0.04	-	-0.12	-	-0.08	-	X	-
3	KDEN	0.41	-	-0.07	-	-0.09	-	-0.07	-	-0.11	-	-0.03	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KDEN	-	0.38	-	-	-	-	-	-	-	-	-	-	-	-	-0.07	-	-0.11	-	-0.03	-	-0.12	-	-0.04	-	X	-
3	KDFW	0.36	-	-0.04	-	-0.07	-	-0.02	-	-0.04	-	-0.08	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KDFW	-	0.34	-	-	-	-	-	-	-	-	-	-	-	-	-0.04	-	-0.07	-	-0.02	-	-0.06	-	-0.07	-	X	-
3	KDTW	0.47	-	-0.06	-	-0.09	-	-0.04	-	-0.15	-	-0.11	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KDTW	-	0.43	-	-	-	-	-	-	-	-	-	-	-	-	-0.05	-	-0.08	-	-0.04	-	-0.1	-	-0.08	-	X	-
3	KEUG	0.56	-	-0.07	-	-0.11	-	-0.1	-	-0.12	-	-0.09	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KEUG	-	0.51	-	-	-	-	-	-	-	-	-	-	-	-	-0.06	-	-0.11	-	-0.08	-	-0.14	-	-0.07	-	X	-
3	KIAH	0.48	-	-0.07	-	-0.1	-	-0.05	-	-0.11	-	-0.05	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KIAH	-	0.43	-	-	-	-	-	-	-	-	-	-	-	-	-0.05	-	-0.08	-	-0.02	-	-0.09	-	-0.05	-	X	-
3	KLAS	0.41	-	-0.01	-	-0.04	-	-0.04	-	-0.07	-	X	-	-0.01	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KLAS	-	0.37	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.05	-	-0.03	-	-0.09	-	X	-	-0.01	-
3	KLAX	0.52	-	-0.07	-	-0.08	-	-0.02	-	-0.05	-	-0.04	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KLAX	-	0.48	-	-	-	-	-	-	-	-	-	-	-	-	-0.06	-	-0.07	-	-0.02	-	-0.03	-	-0.04	-	X	-
3	KLGA	0.37	-	-0.02	-	-0.05	-	-0.04	-	-0.07	-	-0.06	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KLGA	-	0.33	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.03	-	-0.03	-	-0.08	-	-0.05	-	X	-
3	KMCI	0.41	-	-0.09	-	-0.11	-	-0.05	-	-0.19	-	-0.09	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KMCI	-	0.38	-	-	-	-	-	-	-	-	-	-	-	-	-0.08	-	-0.13	-	-0.04	-	-0.14	-	-0.07	-	X	-
3	KMCO	0.51	-	-0.06	-	-0.08	-	-0.11	-	-0.15	-	-0.03	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KMCO	-	0.46	-	-	-	-	-	-	-	-	-	-	-	-	-0.04	-	-0.07	-	-0.09	-	-0.14	-	-0.03	-	X	-
3	KMSP	0.35	-	X	-	-0.02	-	-0.01	-	-0.07	-	X	-	-0.01	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KMSP	-	0.33	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.04	-	X	-	-0.06	-	-0.02	-	-0.02	-
3	KORD	0.41	-	-0.09	-	-0.11	-	-0.04	-	-0.12	-	-0.11	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KORD	-	0.37	-	-	-	-	-	-	-	-	-	-	-	-	-0.07	-	-0.11	-	-0.03	-	-0.11	-	-0.08	-	X	-

Table 72: (continued)

				Legend: 'X' indicates algorithm(s) achieving best NetNorm or MQNetNorm where applicable. '-' indicates non-applicability. Negative value indicates algorithm's score less best.																							
OLA	Station	Best NetNorm	Best MQNetNorm	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)
3	KPHL	0.47	-	-0.07	-	-0.09	-	-0.05	-	-0.1	-	-0.09	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KPHL	-	0.43	-	-	-	-	-	-	-	-	-	-	-	-	-0.06	-	-0.09	-	-0.04	-	-0.09	-	-0.06	-	X	-
3	KPHX	0.34	-	X	-	-0.02	-	X	-	-0.02	-	-0.11	-	-0.03	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KPHX	-	0.29	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.02	-	X	-	-0.03	-	-0.07	-	-0.07	-
3	KPIT	0.48	-	-0.09	-	-0.11	-	-0.04	-	-0.18	-	-0.04	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KPIT	-	0.42	-	-	-	-	-	-	-	-	-	-	-	-	-0.06	-	-0.1	-	-0.02	-	-0.15	-	-0.02	-	X	-
3	KSAC	0.59	-	-0.08	-	-0.1	-	-0.06	-	-0.14	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KSAC	-	0.53	-	-	-	-	-	-	-	-	-	-	-	-	-0.05	-	-0.09	-	-0.03	-	-0.09	-	-0.06	-	X	-
3	KSAN	0.52	-	-0.09	-	-0.09	-	-0.07	-	-0.1	-	-0.07	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KSAN	-	0.47	-	-	-	-	-	-	-	-	-	-	-	-	-0.07	-	-0.07	-	-0.06	-	-0.12	-	-0.07	-	X	-
3	KSAT	0.4	-	-0.08	-	-0.08	-	-0.07	-	-0.1	-	-0.02	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KSAT	-	0.36	-	-	-	-	-	-	-	-	-	-	-	-	-0.06	-	-0.06	-	-0.06	-	-0.09	-	-0.01	-	X	-
3	KSEA	0.5	-	-0.12	-	-0.14	-	-0.1	-	-0.22	-	-0.04	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KSEA	-	0.46	-	-	-	-	-	-	-	-	-	-	-	-	-0.11	-	-0.14	-	-0.09	-	-0.16	-	-0.04	-	X	-
3	KSFO	0.43	-	-0.03	-	-0.05	-	-0.04	-	-0.06	-	-0.06	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KSFO	-	0.42	-	-	-	-	-	-	-	-	-	-	-	-	-0.04	-	-0.07	-	-0.05	-	-0.08	-	-0.07	-	X	-
3	KSMX	0.64	-	-0.02	-	-0.07	-	X	-	-0.05	-	-0.03	-	-0.01	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KSMX	-	0.59	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-0.05	-	-0.01	-	-0.07	-	-0.02	-	-0.01	-
3	KSTL	0.41	-	-0.05	-	-0.08	-	-0.07	-	-0.14	-	-0.13	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KSTL	-	0.37	-	-	-	-	-	-	-	-	-	-	-	-	-0.04	-	-0.07	-	-0.05	-	-0.12	-	-0.08	-	X	-
3	KTPA	0.52	-	-0.05	-	-0.09	-	-0.12	-	-0.13	-	-0.05	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
4	KTPA	-	0.43	-	-	-	-	-	-	-	-	-	-	-	-	-0.01	-	-0.04	-	-0.07	-	-0.1	-	-0.01	-	X	-
				3	-	0	-	2	-	0	-	3	-	25	-	6	-	0	-	2	-	0	-	1	-	25	-

Table 73: Highest scoring algorithms for each station for OLAs 5 and 6

				Legend: 'X' indicates algorithm(s) achieving best NetNorm or MQNetNorm where applicable. '-' indicates non-applicability. Negative value indicates algorithm's score less best.																							
OLA	Station	Best NetNorm	Best MQNetNorm	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)
5	KATL	0.28	-	-	-0.01	-	-0.01	-	-0.04	-	-0.04	-	-0.02	-	X	-	-	-	-	-	-	-	-	-	-	-	-
6	KATL	-	0.26	-	-	-	-	-	-	-	-	-	-	-	-	-	-0.01	-	-0.01	-	-0.04	-	-0.04	-	-0.02	-	X
5	KBOS	0.28	-	-	X	-	X	-	-0.08	-	-0.07	-	-0.05	-	-0.04	-	-	-	-	-	-	-	-	-	-	-	-
6	KBOS	-	0.27	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.09	-	-0.08	-	-0.06	-	-0.05
5	KBWI	0.28	-	-	-0.05	-	-0.05	-	-0.07	-	-0.07	-	-0.05	-	X	-	-	-	-	-	-	-	-	-	-	-	-
6	KBWI	-	0.28	-	-	-	-	-	-	-	-	-	-	-	-	-	-0.07	-	-0.07	-	-0.08	-	-0.08	-	-0.06	-	X
5	KCLE	0.26	-	-	X	-	X	-	-0.06	-	-0.07	-	-0.06	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-
6	KCLE	-	0.22	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.04	-	-0.05	-	-0.04	-	-0.04
5	KCLT	0.3	-	-	-0.1	-	-0.1	-	-0.04	-	-0.04	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-
6	KCLT	-	0.28	-	-	-	-	-	-	-	-	-	-	-	-	-	-0.08	-	-0.08	-	-0.04	-	-0.04	-	-0.01	-	X
5	KCVG	0.24	-	-	X	-	X	-	-0.05	-	-0.05	-	-0.03	-	-0.04	-	-	-	-	-	-	-	-	-	-	-	-
6	KCVG	-	0.22	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.05	-	-0.05	-	-0.03	-	-0.04
5	KDCA	0.28	-	-	X	-	X	-	-0.06	-	-0.06	-	-0.02	-	-0.02	-	-	-	-	-	-	-	-	-	-	-	-
6	KDCA	-	0.27	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.07	-	-0.07	-	-0.03	-	-0.03
5	KDEN	0.24	-	-	X	-	X	-	-0.08	-	-0.08	-	-0.07	-	-0.08	-	-	-	-	-	-	-	-	-	-	-	-
6	KDEN	-	0.23	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.08	-	-0.08	-	-0.08	-	-0.08
5	KDFW	0.27	-	-	X	-	X	-	-0.05	-	-0.05	-	-0.04	-	-0.04	-	-	-	-	-	-	-	-	-	-	-	-
6	KDFW	-	0.25	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.05	-	-0.05	-	-0.04	-	-0.04
5	KDTW	0.3	-	-	X	-	X	-	-0.12	-	-0.12	-	-0.08	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-
6	KDTW	-	0.25	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.08	-	-0.08	-	-0.04	-	-0.03
5	KEUG	0.26	-	-	X	-	X	-	-0.05	-	-0.05	-	-0.02	-	-0.01	-	-	-	-	-	-	-	-	-	-	-	-
6	KEUG	-	0.25	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.05	-	-0.05	-	-0.02	-	-0.01
5	KIAH	0.26	-	-	X	-	X	-	-0.05	-	-0.05	-	-0.01	-	-0.01	-	-	-	-	-	-	-	-	-	-	-	-
6	KIAH	-	0.24	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.04	-	-0.04	-	X	-	X
5	KLAS	0.31	-	-	X	-	X	-	-0.11	-	-0.11	-	-0.11	-	-0.12	-	-	-	-	-	-	-	-	-	-	-	-
6	KLAS	-	0.28	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.1	-	-0.1	-	-0.1	-	-0.1
5	KLAX	0.26	-	-	X	-	X	-	-0.06	-	-0.06	-	-0.04	-	-0.04	-	-	-	-	-	-	-	-	-	-	-	-
6	KLAX	-	0.24	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.04	-	-0.04	-	-0.03	-	-0.02
5	KLGA	0.28	-	-	X	-	X	-	-0.08	-	-0.08	-	-0.04	-	-0.04	-	-	-	-	-	-	-	-	-	-	-	-
6	KLGA	-	0.27	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.09	-	-0.09	-	-0.05	-	-0.04
5	KMCI	0.24	-	-	X	-	X	-	-0.05	-	-0.05	-	-0.03	-	-0.02	-	-	-	-	-	-	-	-	-	-	-	-
6	KMCI	-	0.22	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.05	-	-0.05	-	-0.03	-	-0.02
5	KMCO	0.26	-	-	X	-	X	-	-0.04	-	-0.04	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-
6	KMCO	-	0.26	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.05	-	-0.05	-	X	-	-0.01
5	KMSP	0.25	-	-	X	-	X	-	-0.08	-	-0.08	-	-0.08	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-
6	KMSP	-	0.23	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.08	-	-0.08	-	-0.07	-	-0.06
5	KORD	0.29	-	-	X	-	X	-	-0.1	-	-0.1	-	-0.08	-	-0.08	-	-	-	-	-	-	-	-	-	-	-	-
6	KORD	-	0.24	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.07	-	-0.07	-	-0.05	-	-0.05

Table 73: (continued)

Legend: ‘X’ indicates algorithm(s) achieving best NetNorm or MQNetNorm where applicable.
‘-’ indicates non-applicability.
Negative value indicates algorithm’s score less best.

OLA	Station	Best NetNorm	Best MQNetNorm	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)	Static (variant 0x0)	Static (variant 0x1)	Static (variant 0x2)	Static (variant 0x3)	Aging (variant 0x0)	Aging (variant 0x1)	Aging (variant 0x2)	Aging (variant 0x3)	Fuzzy (variant 0x0)	Fuzzy (variant 0x1)	Fuzzy (variant 0x2)	Fuzzy (variant 0x3)
5	KPHL	0.28	-	-	X	-	X	-	-0.07	-	-0.07	-	-0.01	-	-0.04	-	-	-	-	-	-	-	-	-	-	-	-
6	KPHL	-	0.27	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.08	-	-0.08	-	-0.03	-	-0.05
5	KPHX	0.27	-	-	X	-	X	-	-0.09	-	-0.09	-	-0.05	-	-0.06	-	-	-	-	-	-	-	-	-	-	-	-
6	KPHX	-	0.26	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.09	-	-0.09	-	-0.06	-	-0.07
5	KPIT	0.22	-	-	X	-	X	-	-0.01	-	-0.01	-	-0.01	-	-0.02	-	-	-	-	-	-	-	-	-	-	-	-
6	KPIT	-	0.19	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	X	-	X	-	X	-	X
5	KSAC	0.3	-	-	X	-	X	-	-0.1	-	-0.1	-	-0.04	-	-0.02	-	-	-	-	-	-	-	-	-	-	-	-
6	KSAC	-	0.28	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.09	-	-0.09	-	-0.03	-	-0.01
5	KSAN	0.27	-	-	X	-	X	-	-0.04	-	-0.04	-	-0.03	-	-0.02	-	-	-	-	-	-	-	-	-	-	-	-
6	KSAN	-	0.26	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.04	-	-0.04	-	-0.02	-	-0.02
5	KSAT	0.26	-	-	X	-	X	-	-0.07	-	-0.07	-	-0.05	-	-0.05	-	-	-	-	-	-	-	-	-	-	-	-
6	KSAT	-	0.24	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.06	-	-0.06	-	-0.06	-	-0.04
5	KSEA	0.36	-	-	X	-	X	-	-0.15	-	-0.15	-	-0.06	-	-0.05	-	-	-	-	-	-	-	-	-	-	-	-
6	KSEA	-	0.34	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.13	-	-0.14	-	-0.06	-	-0.05
5	KSFO	0.24	-	-	X	-	X	-	-0.07	-	-0.07	-	-0.05	-	-0.04	-	-	-	-	-	-	-	-	-	-	-	-
6	KSFO	-	0.21	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.05	-	-0.05	-	-0.03	-	-0.03
5	KSMX	0.3	-	-	X	-	X	-	-0.13	-	-0.13	-	-0.09	-	-0.1	-	-	-	-	-	-	-	-	-	-	-	-
6	KSMX	-	0.29	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.13	-	-0.13	-	-0.09	-	-0.1
5	KSTL	0.27	-	-	X	-	X	-	-0.09	-	-0.09	-	-0.08	-	-0.07	-	-	-	-	-	-	-	-	-	-	-	-
6	KSTL	-	0.25	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.08	-	-0.08	-	-0.06	-	-0.07
5	KTPA	0.25	-	-	-0.01	-	-0.01	-	-0.04	-	-0.04	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-
6	KTPA	-	0.24	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-0.05	-	-0.05	-	X	-	X
			Totals:	-	26	-	26	-	0	-	0	-	3	-	5	-	27	-	27	-	1	-	1	-	4	-	6

A.21 ANALYSIS OF ALGORITHM TO FIND OPTIMUM FOR OLA 3

We implemented an algorithm by Smith et al. [86] that solves the weight constrained shortest path problem with replenishment (WCSPP-R), which is NP-Hard, to attempt to find the optimum solution for OLA 3 [86]. The weight is allocated visibility time per month. Thus, replenishment occurs as each month begins. The path length is inversely proportional to the net energy harvested. We tried to run the algorithm to completion, but because the run-time did not complete after days and system problems frequently aborted the job, we abandoned the effort to find optimum for OLA 3. In this section, we examine a reason why the run-time takes so long: labels are not being dominated quickly enough, as we shall see. To explain what we mean by *labels are not being dominated*, we first need to describe the graph we built and Smith's WCSPP-R algorithm:

We built an acyclic, directed weighted graph with replenishment edges. The graph has two nodes for each time step: one node represents the stowed state; the other node represents the deployed states. Each node has two exit edges, which represent two possibilities: 1. remain in the current state, and 2. change states (i.e., deploy if stowed, stow if deployed). Recall that we model each minute. Thus, each full year has over 500,000 minutes, which means each year has over 1 million nodes.

Smith's algorithm preprocesses each node i by finding the least cost from the source node s to i and the least cost from the destination node t to i. Weight is also tracked during pre-processing in case the shortest path (i.e., least cost path) through i happens to not violate the weight budget. If pre-processing finds a shortest path that does not violate the weight budget, then preprocessing found a solution to the given WCSPP-R problem and Smith's algorithm ends.

If preprocessing does not find a solution, then preprocessing prunes infeasible nodes from the graph. It also prunes nodes that can only be part of a non-optimum path. Preprocessing passes the pruned graph to a label-correcting algorithm.

The label-correcting algorithm that we implemented "maintain[s] a set of non-dominated labels, where a label $L = (i, c_L, w_L)$, said to be on node i, represents a partial path from s to node i with cost c_L and weight w_L, and a label L_1 is said to dominate label L_2 if they are on the same node, and if $c_{L_1} < c_{L_2}$ and $w_{L_1} \leq w_{L_2}$ or $c_{L_1} \leq c_{L_2}$ and $w_{L_1} < w_{L_2}$" [86]. Thus, each node i has a set of labels. If a label is dominated by another label, then the dominated label is removed.

The problem with our implementation of the label-correcting algorithm is that the list of labels grows too large because labels are not dominated. For example, consider, the six-node, dual-weighted directed graph in Figure 48. The harvester in this particular simulation takes 1 minute to deploy and 1 minute to retract. Each node represents the state of the harvester, where nodes named with the prefix 'r' indicate that the harvester is in the retracted state and nodes named with the prefix 'd' indicate that the harvester is in the deployed state. Each node's is named with a number indicating a timestep. The retractable harvester begins and ends the simulation in the retracted state. Hence, for timesteps 0 and 3, each timestep has only one node, which indicates a retracted state. For timesteps 1 and 2, each timestep has two nodes; Having two nodes at the same timestep indicates that the harvester can either be fully deployed or fully stowed at that timestep.

Each directed edge $i \rightarrow j$ has two weights expressed by an ordered pair: (x, y) where x is the minutes of the harvester's monthly visibility budget the harvester consumes and y is a cost the harvester incurs if the harvester is the state represented by node i at timestep n and in the state represented by node j and timestep $n + 1$. (The cost y, which we seek to minimize, is an inverse of the energy harvested during that timestep. A cost of 0 represents the highest amount of energy that the harvester converts during any timestep of the entire simulation. Our seeking to minimize y allows us to use shortest-path algorithms instead of longest-path ones.)

Let node r0 be the source node. And let us visit each node in ascending timestep order.

Node r0 is adjacent to two nodes: r1 and d1. Node r1 has only one label since there is only one path to it: (0, 10). Likewise, node d1 has only one label: (1, 15).

Node r1 leads to two nodes: r2 and d2. To obtain a label for r2, we add the weights of the r1→r2 edge (0, 10) to r1's label (0, 10) to obtain (0, 20). Similarly, we add the weights of the r1→d2 edge to obtain (1, 25).

Node d1 leads to two nodes: r2 and d2. To obtain a second label for r2, we add the weights of the d1→d2 edge (1, 0) to d1's label (1, 15) to obtain (2, 15). Note that r2 now

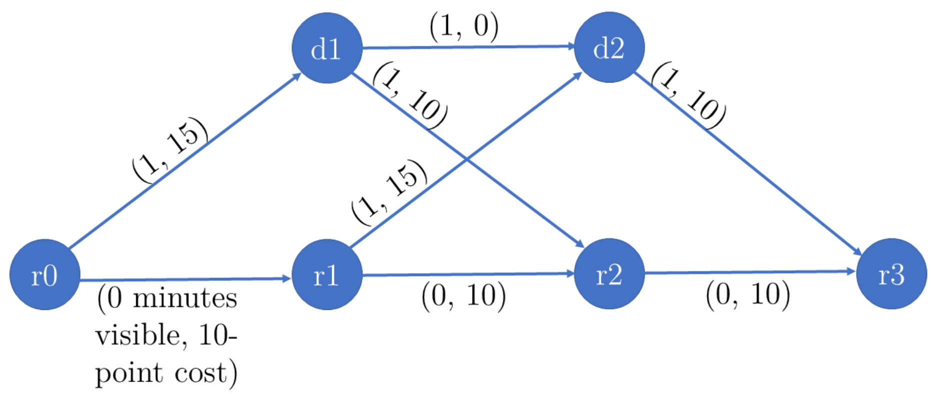

Figure 48: Weighted, directed graph through which to find shortest path within time budget

has two labels: (0, 20) and (2, 15). Neither label dominates the other because one label has a lower cost and the other label has a lower weight.

Now that we have explained what we mean by *labels are not being dominated*, we return to describing our implementation having over 1 million nodes. As we ran that implementation, the list of labels for each node was growing without seeming to decrease. Thus, we turned our attention to finding the "optimum" for a simpler case, OLA 5. We describe how we found "optimum" for OLA 5 in Section 4.3.

A.22 AN IN-DEPTH LOOK AT THE FUZZY-CRISP ALGORITHMS

In this section, we examine earlier versions (v1.2 and v1.3) of the Fuzzy-Crisp algorithms. Such analysis led to the creation of Fuzzy-Crisp v1.4.

A.22.1 Fuzzy v1.2's implementation of the retraction threshold has a greater margin than necessary

A difference between Fuzzy v1.2 and the other two algorithms (Static and Aging) is that Fuzzy is ensured to retract when the running windspeed average drops below the lowest windspeed deemed to be windy k_{ws} at station ws instead of at $k_{ws} - 1$ knot. Fuzzy is observing a retraction threshold that is 1 knot higher than required by the OLA. As shown in the following code listing, Fuzzy is being extra careful that it does not violate the terms of the OLAs requiring the harvester to retract when the weather is not windy (OLAs 1 through 4).

```
/* Ensure that algorithm meets the minimum requirements of OLA */
if ((windspeed_knots_average < windy.getLowestWindspeedThatIsWindyKnots())
   || !(noiseAllowed.bIsNoiseAllowed(sample.date))
   || (harvester.getFractionVisbilePlusTimeToRetractMonthly(
     ws.iUsedAllItsAllocatedVisibilityMinutesPerMonth)) >= 0.99) {
  harvester.resetMode();
}
```

Note that Fuzzy's code directly above is actually crisp code; The code directly above does not calculate a membership value of the variable `windspeed_knots_average`, but directly compares its value to the value returned by `windy.getLowestWindspeedThatIsWindyKnots()`. The code directly above uses the boolean `bIsNoiseAllowed(sample.date)` instead of a fuzzy set. Also the code ensures that the algorithm does not use more visibility time than it has been allocated by directly comparing the constant 0.99 to the sum of the allocated visibility time it has consumed plus the time the harvester takes to retract.

The Static and Aging algorithms use the following code:

```
if (windspeed_knots_average > deploymentThresholdKnots) {
  harvester.setMode();
} else if (windspeed_knots_average < retractionThresholdKnots) {
  harvester.resetMode();
}
```

where `retractionThresholdKnots` is calculated by the following lines

```
final int RETRACTION_THRESHOLD_DIFFERENCE = 1;

// Determine what control signal to output.
// Use visibility-time-remaining to control deployment threshold

int deploymentThresholdKnots = getDeploymentThresholdKnots();
int retractionThresholdKnots = deploymentThresholdKnots -
    RETRACTION_THRESHOLD_DIFFERENCE;
```

where `getDeploymentThresholdKnots()` gets the deployment threshold chosen by the training algorithm (or chosen by the user). (As stated in Section 3.7, the training algorithm searches a design space which starts at the lowest windspeed deemed to be windy k_{ws} at weather station ws.)

Is Fuzzy performing on OLAs 3 and 4 much worse than Static and Aging because Fuzzy is observing a stricter retraction threshold than Fuzzy and Aging? We answer that question by running tests on a modified version Fuzzy shown here:

```

/* Modified retraction threshold for Fuzzy */

final int RETRACTION_THRESHOLD_DIFFERENCE = 1;

/* Ensure that algorithm meets the minimum requirements of OLA */
if ((windspeed_knots_average < (windy.
    getLowestWindspeedThatIsWindyKnots() -
    RETRACTION_THRESHOLD_DIFFERENCE)
  || !(noiseAllowed.bIsNoiseAllowed(sample.date))
  || (harvester.getFractionVisbilePlusTimeToRetractMonthly(
    ws.iUsedAllItsAllocatedVisibilityMinutesPerMonth)) >= 0.99) {

 harvester.resetMode();

}
```

The test results (Appendix A.22.5 on page 306) indicate that the modified Fuzzy (which uses the RETRACTION-THRESHOLD-DIFFERENCE-equaling-1 code directly above) operation performed the same as the original Fuzzy (where RETRACTION THRESHOLD DIFFERENCE = 0) for OLAs 1 through 4. Thus, the test results imply that Fuzzy's crisp code listed directly above is being dominated by Fuzzy-Crisp's fuzzy code.

We explore whether Fuzzy-Crisp's crisp code listed directly above is being dominated by

Fuzzy-Crisp's fuzzy code in the following paragraphs. We added code to collect statistics about whether or not the crisp conditional in Fuzzy-Crisp evaluates the true when the fuzzy conditional evaluates to false as shown here:

```

 boolean bCrispRetractionThresholdReached = false;

 lCrispConditionalEvaluated++;

 /* Ensure that algorithm meets the minimum requirements of OLA */
 if (bCrispRetractionThresholdReached = ((windspeed_knots_average <
     (windy.getLowestWindspeedThatIsWindyKnots()-
    RETRACTION_THRESHOLD_DIFFERENCE)))
 || !(noiseAllowed.bIsNoiseAllowed(sample.date))
 || (harvester.getFractionVisbilePlusTimeToRetractMonthly(
  ws.iUsedAllItsAllocatedVisibilityMinutesPerMonth)) >= 0.99) {

 {if (bCrispRetractionThresholdReached)
     lCrispRetractionThresholdReached++;}
 harvester.resetMode();

} else // use fuzzy logic to decide deployment and retraction
```

The code immediately above increments a long typed variable `lCrispConditionalEvaluated` that holds the number of times the code is reached. The code's boolean variable `bCrispRetractionThresholdReached` is incremented when the crisp inequality

```
windspeed_knots_average < (windy.getLowestWindspeedThatIsWindyKnots
    ()-RETRACTION_THRESHOLD_DIFFERENCE)
```

evaluates to true. If that conditional or either of the two other terms of the conditional evaluate to true, then the following fuzzy code is not executed:

```
} else // use fuzzy logic to decide deployment and retraction
{

 lCrispFoundNoReasonToRetract++;

 final double retractionThresholdMembershipValue =
     deployment_threshold_mv - 0.2;

 if (resultingMembershipValue >= deployment_threshold_mv) {

```

```
    harvester.setMode();
} else if (resultingMembershipValue <
     retractionThresholdMembershipValue) {
    lFuzzyStricterThanCrisp++;
    harvester.resetMode();
}
```

The code immediately above can deploy (set) and retract (reset) a harvester. The value of the variable `resultingMembershipValue` is the harvester's membership value in a fuzzy set which we describe below. If `resultingMembershipValue` is greater than or equal to a deployment threshold membership value (which is expressed in the code above as `deployment_threshold_mv` and chosen by the training algorithm or by the user), then the code deploys the harvester (`.setMode()`).

If the harvester's membership value is not greater than or equal to the deployment threshold membership value, the code evaluates whether or not the resulting membership value is less than the retraction threshold membership value, which is 0.2 less than the deployment threshold membership value (`deployment_threshold_mv - 0.2`). If the membership value is less than the retraction threshold membership value, then the code reaches a statement involved with collecting statistics; The code increments the variable `lFuzzyStricterThanCrisp`, which means that the crisp `if` conditional (which is in the code listing above the the listing directly above) evaluated to false and the fuzzy inequality `(resultingMembershipValue < retractionThresholdMembershipValue)` evaluated to true. The `else if` clause ends by executing the code that retracts the harvester (`.resetMode()`).

How often does Fuzzy-Crisp's fuzzy code overrule Fuzzy-Crisp's crisp code? That is, how often does the fuzzy code deem that the harvester's membership value is less than the `retractionThresholdMembershipValue` when the crisp code finds no reason to retract? An answer to that question is given in Table 74 on the next page, which is a summary of Table 79 on page 315. Table 74 indicates that on average, when the crisp code finds no reason to retract, the fuzzy code retracts an average 46% of the time when the OLA is 3 and the RETRACTION THRESHOLD DIFFERENCE is 0 knots. That average increases to 54% of the time when the RETRACTION THRESHOLD DIFFERENCE is increased to 1 knot. (When the OLA is 4, that average number of times the fuzzy code overrides the crisp code likewise increases when the RETRACTION THRESHOLD DIFFERENCE is increased from 0 to 1 knots.) That overruling of Fuzzy-Crisp's crisp code by Fuzzy-Crisp's fuzzy code helps to explain why increasing the RETRACTION THRESHOLD DIFFERENCE from 0 knots to 1 knot does not improve Fuzzy-Crisp's NetNorm and MQNetNorm scores for OLA 3. Increasing the RETRACTION THRESHOLD DIFFERENCE from 0 knots to 1 knot increases the crisp code's leniency and does not affect the fuzzy code's strictness. *Leniency* is defined as not retracting the harvester.

Table 74: Summary of comparison of two values for RETRACTION THRESHOLD DIFFERENCE for OLAs 3 and 4 for all stations

		lFuzzyStricterThanCrisp/lCrispFoundNoReasonToRetract	
OLA	R.T.D.* (knots)	Average	Standard Deviation
3	0	46%	8%
3	1	54%	8%
4	0	51%	8%
4	1	59%	8%
**R.T.D.* is an acronym for *RETRACTION THRESHOLD DIFFERENCE*			

Thus, we ask, Does the fuzzy code retract the harvester in every case that the average windspeed is between the two crisp retraction thresholds, i.e., when the more lenient crisp does not retract and when the less lenient crisp code retracts (if the fuzzy code were to examine every case)? To find an answer, we added boolean variables to the code to track whether or not each case causes retraction by the following tests, where the added boolean variable is enclosed in parentheses:

- The crisp retraction threshold is reached when the RETRACTION THRESHOLD DIFFERENCE is 0 knots (`bCrispRetractionThresholdReachedWhenRTDIsZero`)

- The crisp retraction threshold is reached when the RETRACTION THRESHOLD DIFFERENCE is 1 knot (`bCrispRetractionThresholdReachedWhenRTDIsOne`)
- The fuzzy retraction threshold is reached (`bFuzzyRetractionThresholdReached`) (Technically, the variable `bFuzzyRetractionThresholdReached` is crisp, being the result of defuzzification, as shown in the code listing which evaluates the boolean expression determining `bFuzzyRetractionThresholdReached`.)

We defined the boolean variable `bMoreLenientCase` to tell use when the average windspeed is between the two crisp retraction thresholds as shown in the following code:

```
boolean bCrispRetractionThresholdReachedWhenRTDIsZero = (windspeed_knots_average <windy.getLowestWindspeedThatIsWindyKnots());
boolean bCrispRetractionThresholdReachedWhenRTDIsOne = (windspeed_knots_average <(c));

boolean bMoreLenientCase = !bCrispRetractionThresholdReachedWhenRTDIsOne &&
  bCrispRetractionThresholdReachedWhenRTDIsZero;
```

The boolean variable `bMoreLenientCase` tells us when the average windspeed is in the interval [`.getLowestWindspeedThatIsWindyKnots()-1`,`.getLowestWindspeedThatIsWindyKnots()`). When the average windspeed is in that interval, does the fuzzy code cause retraction (if the fuzzy code is allowed to process that case)? The following code tells us:

```
/* "MoreLenientCase" is defined as a windspeed that does not cause retraction when R.T.D. is 1, but
 *  does cause retraction when R.T.D. is 0.
 */
boolean bCrispRetractionThresholdReachedWhenRTDIsZero = (windspeed_knots_average <windy.getLowestWindspeedThatIsWindyKnots());
boolean bCrispRetractionThresholdReachedWhenRTDIsOne = (windspeed_knots_average <(windy.getLowestWindspeedThatIsWindyKnots()-1));

boolean bMoreLenientCase = !bCrispRetractionThresholdReachedWhenRTDIsOne &&
  bCrispRetractionThresholdReachedWhenRTDIsZero;

boolean bRetractionConditionReached = false;

```

```java
/* Ensure that algorithm meets the minimum requirements of OLA */
if (((windspeed_knots_average < (windy.getLowestWindspeedThatIsWindyKnots()-RETRACTION_THRESHOLD_DIFFERENCE)))
  || !(noiseAllowed.bIsNoiseAllowed(sample.date))
  || (harvester.getFractionVisbilePlusTimeToRetractMonthly(
    ws.iUsedAllItsAllocatedVisibilityMinutesPerMonth)) >= 0.99) {

 harvester.resetMode();
 bRetractionConditionReached = true;

} else // use fuzzy logic to decide deployment and retraction
{

 final double retractionThresholdMembershipValue = deployment_threshold_mv - 0.2;

 if (resultingMembershipValue >= deployment_threshold_mv) {

  harvester.setMode();

 } else if (resultingMembershipValue < retractionThresholdMembershipValue) {

  bRetractionConditionReached = true;

  harvester.resetMode();

 }

}

if (bMoreLenientCase && !bRetractionConditionReached) {

 bEveryMoreLenientCaseRestricted = false;

}

boolean bFuzzyRetractionThresholdReached = resultingMembershipValue < (deployment_threshold_mv - 0.2);
if (bMoreLenientCase && !bFuzzyRetractionThresholdReached) {

 bEveryMoreLenientCaseReachesFuzzyRetractionThreshold = false;

}
```

Now, we are ready to answer the question, Does the fuzzy code retract the harvester in every case that the average windspeed is between the two crisp retraction thresholds? Running the code immediately above for every station for OLAs 3 and 4 indicates that the answer is Yes.

And we have the answer to our original question, Is Fuzzy-Crisp performing on OLAs 3 and 4 much worse than Static and Aging because Fuzzy-Crisp is observing a stricter retraction threshold than Static and Aging? The answer is No.

Now that we examined Fuzzy-Crisp's crisp code, let us turn to Fuzzy-Crisp's fuzzy code. Would Fuzzy-Crisp perform comparably on OLAs 3 and 4 to Static and Aging if Fuzzy-Crisp used only its crisp code to retract? In the next section, we ask, How does Fuzzy-Crisp perform on OLAs 3 and4 using only its crisp code to retract?

A.22.2 How does Fuzzy-Crisp v1.2 perform on OLAs 3 and4 using only its crisp code to retract?

We commented Line 34 in the code listing above so that Fuzzy-Crisp's fuzzy code does not retract the harvester; Fuzzy-Crisp relies solely on Fuzzy-Crisp's crisp code to retract to achieve the following test results: The modified Fuzzy-Crisp that uses crisp code to retract and not its fuzzy code achieves

- an average NetNorm() score of 0.386 for OLA 3, and
- an average MQNetNorm() score of 0.353 for OLA 4,

which are comparable to the average scores for Static and Aging for OLAs 3 and 4 (Table 4 on page 84). (Per-station results for the "retract-using-crisp-code-only" version of Fuzzy-Crisp are shown in Table 80 on page 323.)

Thus, one or more aspects of Fuzzy-Crisp's fuzzy code is causing Fuzzy-Crisp to perform relatively poorly for OLAs 3 and 4. We explore what in Fuzzy-Crisp's fuzzy code in the next section.

A.22.3 How Fuzzy-Crisp combines fuzzy sets to determine a resulting membership value

In Section A.22.1 above, we refer to a fuzzy set in which the harvester has the degree of membership indicated by the variable `resultingMembershipValue`. In the following paragraph, we describe that fuzzy set. The fuzzy set is a combination of fuzzy sets such as WINDY AT <STATION>, APPROACHING QUIET HOURS, and PLENTY OF ALLOCATED VISIBILITY MINUTES REMAINING. The combination of fuzzy sets we use is shown in the following code excerpt:

```
double resultingMembershipValue = Math.min(membershipValueConditional,

noPredictionResult = Math.min(windy.getMembershipValueForWindy((int) windspeed_knots_average),
 Math.max(
  not(noiseAllowed.getMembershipValueForApproachingQuietHours(sample.date,
  MINUTES_BEFORE_QUIET_HOURS_X_INTERCEPT)),
  harvester.getMembershipValueForPlentyOfAllocatedVisibilityMinutesRemaining(
  ws.iUsedAllItsAllocatedVisibilityMinutesPerMonth)))
 );
```

where the variable `membershipValueConditional` depends on whether or not Fuzzy is using weather prediction:

```
if (bUseWeatherPrediction) {
 // If very windy tomorrow and running out of time, retract
 // -or-
 // Allow deployment if not very windy tomorrow
 ////  -or-
 // if not running out of time

 membershipValueConditional = Math.max(
  not(windy.getMembershipValueForVeryWindy((int) windspeed_knots_average_future)),
  harvester.getMembershipValueForPlentyOfAllocatedVisibilityMinutesRemaining(
  ws.iUsedAllItsAllocatedVisibilityMinutesPerMonth
  )
```

```
  );
} else {
  membershipValueConditional = 1;
}
```

We briefly discuss the listing directly above in Section 3.6.2.3 and do not repeat that discussion here, but refer to the result of the code directly above as `conditional prediction` because it depends on whether Fuzzy is using weather prediction.

The two code listings directly above translate into the statement
result = (conditional prediction AND (windy AND ((not approaching quiet hours) OR (plenty of allocated visibility minutes remaining))))
which, in turn, can be expressed by the following linguistic statement:

If weather prediction is not being used, then the resulting membership value is higher than otherwise when the weather is windy and if either of the following two cases are occurring:

1. the time is not approaching quiet hours or
2. plenty of allocated visibility minutes are remaining.

For example, if the time of day is approaching quiet hours but there are plenty of allocated visibility minutes remaining, then the harvester will tend to deploy if the weather is windy.

For another example, if the time of day is approaching quiet hours and the harvester has consumed most of its visibility minutes, the harvester will tend to not deploy, even if the weather is windy. Not only will the fuzzy code of version 1.2 of Fuzzy tend not to deploy, but the code will tend to retract under those conditions.

A.22.4 Effect the effect of Fuzzy-Crisp's "plenty-of-allocated-visibility-minutes-are-remaining" condition

The membership value of the harvester in the fuzzy set PLENTY OF ALLOCATED MINUTES REMAINING is calculated by the following code:

```
  public float
      getMembershipValueForPlentyOfAllocatedVisibilityMinutesRemaining
      (long maximumMinutesAllowedToBeVisible) {

```

```
  float membershipValue = 1 - getFractionVisbilePlusTimeToRetractMonthly(maximumMinutesAllowedToBeVisible);

  float ADJUSTMENT = (float) -0.1; // account for adjective "Plenty". The plenty of time left overcomes the negative adjustment
  membershipValue += ADJUSTMENT;

  if (membershipValue > 1) {

   membershipValue = 1;

  } else if (membershipValue < 0) {

   membershipValue = 0;

  }

  return(membershipValue);

}
```

An example plot of the membership value function is shown in Figure 42 on page 261. The membership value function is a linear function dependent on the visibility minutes the harvester has consumed during the month and dependent on two constants: the retraction time (Section A.3.5) and an adjustment factor of -0.1.

That adjustment factor had been intended to account for the adjective *plenty* in the linguistic variable's name PLENTY OF ALLOCATED MINUTES REMAINING. It was intended to shift the x-axis from the perspective that if the adjustment factor were zero, then the code would be finding the harvester's membership value in the set A MEDIUM AMOUNT OF MINUTES REMAINING. However, the way that the adjustment factor is implemented in version 1.2 of Fuzzy actually shifts the y-axis.

That unintended implementation may be part of the reason that Fuzzy-Crisp 1.2 is not performing comparably to Static and Aging for OLAs 3 and 4. Thus, let us test Fuzzy-Crisp 1.3 where the adjustment factor shifts the x-axis. The Fuzzy-Crisp 1.3, where the only difference in code from Fuzzy-Crisp 1.2 is that Fuzzy-Crisp 1.3 shifts the x-axis instead of the y-axis as shown in the code listing immediately below achieves

- an average NetNorm() score of 0.367 for OLA 3, and

- an average MQNetNorm() score of 0.289 for OLA 4,

which is much better than Fuzzy-Crisp 1.2 which scored 0.221 and 0.191, respectively, but still not as good as the top-scoring Aging for OLAs 3 and 4, which scored 0.402 and 0.371 respectively (Table 4 on page 84) and less than the version of Fuzzy-Crisp 1.2 that does not use its fuzzy code to retract (Section A.22.2), which scored 0.386 and 0.353 respectively. (Per station results of Fuzzy-Crisp 1.3 are shown in Table 81 on page 325).

```
public float
getMembershipValueForPlentyOfAllocatedVisibilityMinutesRemainingShiftX(
   long maximumMinutesAllowedToBeVisible) {

  float ADJUSTMENT = (float) 0.9; // account for adjective "Plenty".
     The plenty of time left overcomes the fractional adjustment

  float membershipValue = 1 -
     getFractionVisbilePlusTimeToRetractMonthly((int) ADJUSTMENT *
     maximumMinutesAllowedToBeVisible);

  if (membershipValue > 1) {

   membershipValue = 1;

  } else if (membershipValue < 0) {

   membershipValue = 0;

  }

  return(membershipValue);

 }
```

A.22.5 Data of exploration the effect of RETRACTION THRESHOLD DIFFERENCE in Fuzzy

Below is the data to which we refer from Section A.22.1.

A.22.5.1 OLA 1 Please see Table 75.

Table 75: Results of the processing of OLA 1 by Fuzzy-Crsip (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.2 when the subtraction Deployment Threshold less Retraction Threshold equals 0 and when it equals 1

Deployment Threshold less Retraction Threshold	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
0	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.58	0.62	0.36
1	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.58	0.62	0.36
0	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.53	0.75	0.39
1	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.53	0.75	0.39
0	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.62	0.66	0.40
1	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.62	0.66	0.40
0	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.69	0.44
1	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.69	0.44
0	KCLT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.70	0.48	0.34
1	KCLT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.70	0.48	0.34
0	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.60	0.36
1	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.60	0.36
0	KDCA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.62	0.67	0.41
1	KDCA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.62	0.67	0.41
0	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.57	0.79	0.45
1	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.57	0.79	0.45
0	KDFW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.65	0.80	0.52
1	KDFW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.65	0.80	0.52
0	KDTW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.72	0.48
1	KDTW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.72	0.48
0	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.66	0.69	0.45
1	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.66	0.69	0.45
0	KIAH	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.61	0.66	0.40

Table 75: (continued)

1	KIAH	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.61	0.66	0.40
0	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.77	0.45
1	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.77	0.45
0	KLAX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.77	0.83	0.64
1	KLAX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.77	0.83	0.64
0	KLGA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.56	0.74	0.41
1	KLGA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.56	0.74	0.41
0	KMCI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.74	0.50
1	KMCI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.74	0.50
0	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.53	0.60	0.32
1	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.53	0.60	0.32
0	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.62	0.40
1	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.62	0.40
0	KORD	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.67	0.39
1	KORD	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.67	0.39
0	KPHL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.68	0.41
1	KPHL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.68	0.41
0	KPHX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.49	0.31
1	KPHX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.64	0.49	0.31
0	KPIT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.62	0.60	0.37
1	KPIT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.62	0.60	0.37
0	KSAC	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.76	0.63	0.47
1	KSAC	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.76	0.63	0.47
0	KSAN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.68	0.57	0.39
1	KSAN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.68	0.57	0.39
0	KSAT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.40	0.79	0.31
1	KSAT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.40	0.79	0.31
0	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.69	0.69	0.47
1	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.69	0.69	0.47
0	KSFO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.95	0.60
1	KSFO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.95	0.60
0	KSMX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.85	0.57
1	KSMX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.85	0.57
0	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.64	0.41
1	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.64	0.41
0	KTPA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.70	0.55	0.38
1	KTPA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.70	0.55	0.38

A.22.5.2 OLA 2

Please see Table 76 on the next page.

Table 76: Results of the processing of OLA 2 by Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.2 when the subtraction Deployment Threshold less Retraction Threshold equals 0 and when it equals 1

Deployment Threshold less Retraction Threshold	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
0	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.60	0.69	0.42
1	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.60	0.69	0.42
0	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.53	0.75	0.39
1	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.53	0.75	0.39
0	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.66	0.39
1	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.66	0.39
0	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.59	0.73	0.43
1	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.59	0.73	0.43
0	KCLT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	61	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.65	0.64	0.42
1	KCLT	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	61	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.65	0.64	0.42
0	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.60	0.36
1	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.60	0.60	0.36
0	KDCA	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.44	0.80	0.35
1	KDCA	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.44	0.80	0.35
0	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.57	0.79	0.45
1	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.57	0.79	0.45
0	KDFW	0.5	1	0.5	1	0.5	61	0.5	1	0.5	91	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.58	0.84	0.49
1	KDFW	0.5	1	0.5	1	0.5	61	0.5	1	0.5	91	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.58	0.84	0.49
0	KDTW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.72	0.48
1	KDTW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.72	0.48
0	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.66	0.69	0.45
1	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.66	0.69	0.45
0	KIAH	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	61	0.5	1	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.53	0.84	0.45
1	KIAH	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	61	0.5	1	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.53	0.84	0.45
0	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.77	0.45
1	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.59	0.77	0.45
0	KLAX	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.34	0.79	0.27

where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n

Table 76: (continued)

1	KLAX	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.34	0.79	0.27
0	KLGA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.46	0.82	0.38
1	KLGA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.46	0.82	0.38
0	KMCI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.74	0.50
1	KMCI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.74	0.50
0	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	91	0.5	1	0.5	1	0.56	0.68	0.38
1	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	91	0.5	1	0.5	1	0.56	0.68	0.38
0	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.57	0.65	0.37
1	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.57	0.65	0.37
0	KORD	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.74	0.49
1	KORD	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.74	0.49
0	KPHL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.73	0.38
1	KPHL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.73	0.38
0	KPHX	0.5	1	0.5	1	0.5	91	0.5	61	0.5	61	0.5	1	0.5	61	0.5	91	0.5	1	0.5	91	0.5	1	0.5	1	0.43	0.67	0.29
1	KPHX	0.5	1	0.5	1	0.5	91	0.5	61	0.5	61	0.5	1	0.5	61	0.5	91	0.5	1	0.5	91	0.5	1	0.5	1	0.43	0.67	0.29
0	KPIT	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.29	0.77	0.23
1	KPIT	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.29	0.77	0.23
0	KSAC	0.5	1	0.5	1	0.5	1	0.5	61	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.73	0.73	0.53
1	KSAC	0.5	1	0.5	1	0.5	1	0.5	61	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.73	0.73	0.53
0	KSAN	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.74	0.65	0.48
1	KSAN	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.74	0.65	0.48
0	KSAT	0.5	91	0.5	1	0.5	1	0.5	61	0.5	91	0.5	91	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.42	0.82	0.34
1	KSAT	0.5	91	0.5	1	0.5	1	0.5	61	0.5	91	0.5	91	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.42	0.82	0.34
0	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.64	0.70	0.45
1	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.64	0.70	0.45
0	KSFO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.95	0.60
1	KSFO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.63	0.95	0.60
0	KSMX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.85	0.57
1	KSMX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.67	0.85	0.57
0	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.56	0.66	0.37
1	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.56	0.66	0.37
0	KTPA	0.5	1	0.5	1	0.5	1	0.5	61	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.5	91	0.5	91	0.5	1	0.58	0.74	0.42
1	KTPA	0.5	1	0.5	1	0.5	1	0.5	61	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.5	91	0.5	91	0.5	1	0.58	0.74	0.42

A.22.5.3 OLA 3

Please see Table 77 on the following page.

Table 77: Results of the processing of OLA 3 by Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.2 when the subtraction Deployment Threshold less Retraction Threshold equals 0 and when it equals 1

Deployment Threshold less Retraction Threshold	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
0	KATL	0.5	61	0.5	91	0.5	31	0.5	91	0.5	91	0.5	1	0.5	1	0.5	91	0.5	61	0.5	91	0.5	91	0.5	1	0.21	0.83	0.17
1	KATL	0.5	61	0.5	91	0.5	31	0.5	91	0.5	91	0.5	1	0.5	1	0.5	91	0.5	61	0.5	91	0.5	91	0.5	1	0.21	0.83	0.17
0	KBOS	0.5	91	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	61	0.5	61	0.5	91	0.17	0.86	0.15
1	KBOS	0.5	91	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	61	0.5	61	0.5	91	0.17	0.86	0.15
0	KBWI	0.5	91	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	91	0.5	61	0.5	31	0.23	0.87	0.20
1	KBWI	0.5	91	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	91	0.5	61	0.5	31	0.23	0.87	0.20
0	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	61	0.5	61	0.5	1	0.5	61	0.5	61	0.5	91	0.5	61	0.18	0.82	0.15
1	KCLE	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	61	0.5	61	0.5	1	0.5	61	0.5	61	0.5	91	0.5	61	0.18	0.82	0.15
0	KCLT	0.5	61	0.5	61	0.5	91	0.5	1	0.5	31	0.5	31	0.5	1	0.5	31	0.5	1	0.5	91	0.5	91	0.5	61	0.26	0.70	0.18
1	KCLT	0.5	61	0.5	61	0.5	91	0.5	1	0.5	31	0.5	31	0.5	1	0.5	31	0.5	1	0.5	91	0.5	91	0.5	61	0.26	0.70	0.18
0	KCVG	0.5	91	0.5	91	0.5	1	0.5	91	0.5	91	0.5	61	0.5	1	0.5	1	0.5	91	0.5	91	0.5	1	0.5	1	0.15	0.76	0.12
1	KCVG	0.5	91	0.5	91	0.5	1	0.5	91	0.5	91	0.5	61	0.5	1	0.5	1	0.5	91	0.5	91	0.5	1	0.5	1	0.15	0.76	0.12
0	KDCA	0.5	91	0.5	61	0.5	91	0.5	1	0.5	61	0.5	61	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.19	0.86	0.17
1	KDCA	0.5	91	0.5	61	0.5	91	0.5	1	0.5	61	0.5	61	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.19	0.86	0.17
0	KDEN	0.5	1	0.5	31	0.5	61	0.5	91	0.5	31	0.5	1	0.5	91	0.5	61	0.5	1	0.5	61	0.5	61	0.5	31	0.19	0.85	0.16
1	KDEN	0.5	1	0.5	31	0.5	61	0.5	91	0.5	31	0.5	1	0.5	91	0.5	61	0.5	1	0.5	61	0.5	61	0.5	31	0.19	0.85	0.16
0	KDFW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	91	0.5	31	0.5	91	0.5	61	0.5	91	0.5	1	0.17	0.90	0.15
1	KDFW	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	91	0.5	31	0.5	91	0.5	61	0.5	91	0.5	1	0.17	0.90	0.15
0	KDTW	0.5	31	0.5	91	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	31	0.5	1	0.5	91	0.5	1	0.5	61	0.20	0.83	0.17
1	KDTW	0.5	31	0.5	91	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	31	0.5	1	0.5	91	0.5	1	0.5	61	0.20	0.83	0.17
0	KEUG	0.5	1	0.5	91	0.5	61	0.5	1	0.5	31	0.5	31	0.5	91	0.5	31	0.5	1	0.5	31	0.5	91	0.5	1	0.26	0.86	0.23
1	KEUG	0.5	1	0.5	91	0.5	61	0.5	1	0.5	31	0.5	31	0.5	91	0.5	31	0.5	1	0.5	31	0.5	91	0.5	1	0.26	0.86	0.23
0	KIAH	0.5	61	0.5	61	0.5	1	0.5	1	0.5	91	0.5	61	0.5	31	0.5	31	0.5	91	0.5	31	0.5	1	0.5	91	0.19	0.88	0.17
1	KIAH	0.5	61	0.5	61	0.5	1	0.5	1	0.5	91	0.5	61	0.5	31	0.5	31	0.5	91	0.5	31	0.5	1	0.5	91	0.19	0.88	0.17
0	KLAS	0.5	1	0.5	1	0.5	91	0.5	61	0.5	61	0.5	61	0.5	91	0.5	61	0.5	1	0.5	61	0.5	1	0.5	61	0.21	0.83	0.18
1	KLAS	0.5	1	0.5	1	0.5	91	0.5	61	0.5	61	0.5	61	0.5	91	0.5	61	0.5	1	0.5	61	0.5	1	0.5	61	0.21	0.83	0.18

0	KLAX	0.5	31	0.5	31	0.5	91	0.5	31	0.5	1	0.5	91	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	1	0.26	0.86	0.22
1	KLAX	0.5	31	0.5	31	0.5	91	0.5	31	0.5	1	0.5	91	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	1	0.26	0.86	0.22
0	KLGA	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.5	1	0.5	31	0.5	91	0.5	91	0.5	91	0.5	1	0.16	0.86	0.14
1	KLGA	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.5	1	0.5	31	0.5	91	0.5	91	0.5	91	0.5	1	0.16	0.86	0.14
0	KMCI	0.5	1	0.5	91	0.5	1	0.5	1	0.5	61	0.5	61	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.19	0.88	0.16
1	KMCI	0.5	1	0.5	91	0.5	1	0.5	1	0.5	61	0.5	61	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.19	0.88	0.16
0	KMCO	0.5	91	0.5	91	0.5	31	0.5	1	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	91	0.5	61	0.5	31	0.19	0.81	0.16
1	KMCO	0.5	91	0.5	91	0.5	31	0.5	1	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	91	0.5	61	0.5	31	0.19	0.81	0.16
0	KMSP	0.5	91	0.5	61	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	61	0.5	91	0.5	31	0.5	1	0.5	91	0.17	0.82	0.14
1	KMSP	0.5	91	0.5	61	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	61	0.5	91	0.5	31	0.5	1	0.5	91	0.17	0.82	0.14
0	KORD	0.5	1	0.5	1	0.5	31	0.5	61	0.5	31	0.5	1	0.5	31	0.5	1	0.5	91	0.5	91	0.5	1	0.5	31	0.17	0.82	0.14
1	KORD	0.5	1	0.5	1	0.5	31	0.5	61	0.5	31	0.5	1	0.5	31	0.5	1	0.5	91	0.5	91	0.5	1	0.5	31	0.17	0.82	0.14
0	KPHL	0.5	31	0.5	91	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	91	0.5	31	0.5	91	0.22	0.84	0.18
1	KPHL	0.5	31	0.5	91	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	91	0.5	31	0.5	91	0.22	0.84	0.18
0	KPHX	0.5	31	0.5	1	0.5	91	0.5	61	0.5	1	0.5	91	0.5	91	0.5	31	0.5	61	0.5	61	0.5	1	0.5	1	0.25	0.71	0.18
1	KPHX	0.5	31	0.5	1	0.5	91	0.5	61	0.5	1	0.5	91	0.5	91	0.5	31	0.5	61	0.5	61	0.5	1	0.5	1	0.25	0.71	0.18
0	KPIT	0.5	91	0.5	61	0.5	91	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.21	0.83	0.17
1	KPIT	0.5	91	0.5	61	0.5	91	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.21	0.83	0.17
0	KSAC	0.5	91	0.5	91	0.5	1	0.5	91	0.5	1	0.5	31	0.5	31	0.5	31	0.5	61	0.5	91	0.5	31	0.5	1	0.32	0.83	0.27
1	KSAC	0.5	91	0.5	91	0.5	1	0.5	91	0.5	1	0.5	31	0.5	31	0.5	31	0.5	61	0.5	91	0.5	31	0.5	1	0.32	0.83	0.27
0	KSAN	0.5	31	0.5	1	0.5	31	0.5	1	0.5	61	0.5	1	0.5	1	0.5	61	0.5	61	0.5	61	0.5	1	0.5	1	0.29	0.74	0.21
1	KSAN	0.5	31	0.5	1	0.5	31	0.5	1	0.5	61	0.5	1	0.5	1	0.5	61	0.5	61	0.5	61	0.5	1	0.5	1	0.29	0.74	0.21
0	KSAT	0.5	91	0.5	91	0.5	61	0.5	1	0.5	61	0.5	61	0.5	91	0.5	61	0.5	31	0.5	61	0.5	1	0.5	91	0.20	0.86	0.17
1	KSAT	0.5	91	0.5	91	0.5	61	0.5	1	0.5	61	0.5	61	0.5	91	0.5	61	0.5	31	0.5	61	0.5	1	0.5	91	0.20	0.86	0.17
0	KSEA	0.5	31	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.23	0.85	0.19
1	KSEA	0.5	31	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.23	0.85	0.19
0	KSFO	0.5	1	0.5	61	0.5	1	0.5	91	0.5	61	0.5	1	0.5	61	0.5	91	0.5	61	0.5	91	0.5	1	0.5	1	0.20	0.93	0.18
1	KSFO	0.5	1	0.5	61	0.5	1	0.5	91	0.5	61	0.5	1	0.5	61	0.5	91	0.5	61	0.5	91	0.5	1	0.5	1	0.20	0.93	0.18
0	KSMX	0.5	1	0.5	31	0.5	31	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.36	0.89	0.32
1	KSMX	0.5	1	0.5	31	0.5	31	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.36	0.89	0.32
0	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.43	0.64	0.28
1	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.43	0.64	0.28
0	KTPA	0.5	91	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	31	0.5	31	0.5	1	0.5	61	0.5	31	0.5	61	0.26	0.71	0.18
1	KTPA	0.5	91	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	31	0.5	31	0.5	1	0.5	61	0.5	31	0.5	61	0.26	0.71	0.18

A.22.5.4 OLA 4 Please see Table 78 on the next page.

Table 78: Results of the processing of OLA 4 by Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.2 when the subtraction Deployment Threshold less Retraction Threshold equals 0 and when it equals 1

Deployment Threshold less Retraction Threshold	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
0	KATL	0.5	91	0.5	91	0.5	31	0.5	91	0.5	91	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.5	91	0.5	1	0.20	0.86	0.17
1	KATL	0.5	91	0.5	91	0.5	31	0.5	91	0.5	91	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.5	91	0.5	1	0.20	0.86	0.17
0	KBOS	0.5	91	0.5	61	0.5	91	0.5	91	0.5	91	0.5	1	0.5	1	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.17	0.90	0.15
1	KBOS	0.5	91	0.5	61	0.5	91	0.5	91	0.5	91	0.5	1	0.5	1	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.17	0.90	0.15
0	KBWI	0.5	91	0.5	91	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.24	0.85	0.20
1	KBWI	0.5	91	0.5	91	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.24	0.85	0.20
0	KCLE	0.5	1	0.5	1	0.5	91	0.5	61	0.5	61	0.5	61	0.5	61	0.5	31	0.5	61	0.5	61	0.5	91	0.5	91	0.18	0.87	0.15
1	KCLE	0.5	1	0.5	1	0.5	91	0.5	61	0.5	61	0.5	61	0.5	61	0.5	31	0.5	61	0.5	61	0.5	91	0.5	91	0.18	0.87	0.15
0	KCLT	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	61	0.5	91	0.5	1	0.5	91	0.5	91	0.5	91	0.26	0.80	0.21
1	KCLT	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	61	0.5	91	0.5	1	0.5	91	0.5	91	0.5	91	0.26	0.80	0.21
0	KCVG	0.5	91	0.5	91	0.5	1	0.5	91	0.5	91	0.5	91	0.5	1	0.5	61	0.5	91	0.5	91	0.5	61	0.5	91	0.16	0.85	0.14
1	KCVG	0.5	91	0.5	91	0.5	1	0.5	91	0.5	91	0.5	91	0.5	1	0.5	61	0.5	91	0.5	91	0.5	61	0.5	91	0.16	0.85	0.14
0	KDCA	0.5	91	0.5	61	0.5	91	0.5	61	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.21	0.90	0.19
1	KDCA	0.5	91	0.5	61	0.5	91	0.5	61	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.21	0.90	0.19
0	KDEN	0.5	1	0.5	91	0.5	61	0.5	91	0.5	31	0.5	1	0.5	91	0.5	61	0.5	31	0.5	61	0.5	61	0.5	61	0.19	0.86	0.17
1	KDEN	0.5	1	0.5	91	0.5	61	0.5	91	0.5	31	0.5	1	0.5	91	0.5	61	0.5	31	0.5	61	0.5	61	0.5	61	0.19	0.86	0.17
0	KDFW	0.5	1	0.5	61	0.5	1	0.5	91	0.5	1	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.16	0.91	0.15
1	KDFW	0.5	1	0.5	61	0.5	1	0.5	91	0.5	1	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.16	0.91	0.15
0	KDTW	0.5	91	0.5	91	0.5	1	0.5	1	0.5	61	0.5	31	0.5	31	0.5	31	0.5	1	0.5	91	0.5	1	0.5	91	0.19	0.86	0.17
1	KDTW	0.5	91	0.5	91	0.5	1	0.5	1	0.5	61	0.5	31	0.5	31	0.5	31	0.5	1	0.5	91	0.5	1	0.5	91	0.19	0.86	0.17
0	KEUG	0.5	91	0.5	91	0.5	61	0.5	91	0.5	31	0.5	31	0.5	91	0.5	31	0.5	61	0.5	91	0.5	91	0.5	91	0.31	0.92	0.28
1	KEUG	0.5	91	0.5	91	0.5	61	0.5	91	0.5	31	0.5	31	0.5	91	0.5	31	0.5	61	0.5	91	0.5	91	0.5	91	0.31	0.92	0.28
0	KIAH	0.5	61	0.5	61	0.5	1	0.5	91	0.5	91	0.5	61	0.5	61	0.5	31	0.5	91	0.5	91	0.5	1	0.5	91	0.20	0.90	0.18
1	KIAH	0.5	61	0.5	61	0.5	1	0.5	91	0.5	91	0.5	61	0.5	61	0.5	31	0.5	91	0.5	91	0.5	1	0.5	91	0.20	0.90	0.18
0	KLAS	0.5	1	0.5	1	0.5	31	0.5	61	0.5	61	0.5	61	0.5	91	0.5	91	0.5	1	0.5	91	0.5	91	0.5	91	0.20	0.84	0.17
1	KLAS	0.5	1	0.5	1	0.5	31	0.5	61	0.5	61	0.5	61	0.5	91	0.5	91	0.5	1	0.5	91	0.5	91	0.5	91	0.20	0.84	0.17
0	KLAX	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.5	91	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	61	0.25	0.88	0.22

Table 78: (continued)

1	KLAX	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.5	91	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	61	0.25	0.88	0.22
0	KLGA	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.5	1	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.16	0.86	0.14
1	KLGA	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.5	1	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.16	0.86	0.14
0	KMCI	0.5	61	0.5	91	0.5	1	0.5	1	0.5	61	0.5	61	0.5	91	0.5	31	0.5	91	0.5	91	0.5	1	0.5	61	0.19	0.92	0.18
1	KMCI	0.5	61	0.5	91	0.5	1	0.5	1	0.5	61	0.5	61	0.5	91	0.5	31	0.5	91	0.5	91	0.5	1	0.5	61	0.19	0.92	0.18
0	KMCO	0.5	91	0.5	91	0.5	31	0.5	1	0.5	91	0.5	91	0.5	61	0.5	61	0.5	1	0.5	91	0.5	61	0.5	1	0.20	0.83	0.16
1	KMCO	0.5	91	0.5	91	0.5	31	0.5	1	0.5	91	0.5	91	0.5	61	0.5	61	0.5	1	0.5	91	0.5	61	0.5	1	0.20	0.83	0.16
0	KMSP	0.5	91	0.5	91	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	61	0.5	91	0.5	31	0.5	91	0.5	91	0.17	0.85	0.15
1	KMSP	0.5	91	0.5	91	0.5	1	0.5	91	0.5	1	0.5	91	0.5	61	0.5	61	0.5	91	0.5	31	0.5	91	0.5	91	0.17	0.85	0.15
0	KORD	0.5	91	0.5	61	0.5	31	0.5	61	0.5	61	0.5	1	0.5	91	0.5	1	0.5	91	0.5	91	0.5	1	0.5	31	0.17	0.85	0.15
1	KORD	0.5	91	0.5	61	0.5	31	0.5	61	0.5	61	0.5	1	0.5	91	0.5	1	0.5	91	0.5	91	0.5	1	0.5	31	0.17	0.85	0.15
0	KPHL	0.5	61	0.5	91	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	61	0.5	61	0.5	91	0.5	31	0.5	91	0.21	0.87	0.19
1	KPHL	0.5	61	0.5	91	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	61	0.5	61	0.5	91	0.5	31	0.5	91	0.21	0.87	0.19
0	KPHX	0.5	61	0.5	1	0.5	91	0.5	61	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	31	0.26	0.81	0.21
1	KPHX	0.5	61	0.5	1	0.5	91	0.5	61	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	31	0.26	0.81	0.21
0	KPIT	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.5	31	0.5	1	0.5	91	0.5	1	0.5	91	0.21	0.91	0.19
1	KPIT	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	1	0.5	31	0.5	1	0.5	91	0.5	1	0.5	91	0.21	0.91	0.19
0	KSAC	0.5	91	0.5	91	0.5	1	0.5	91	0.5	31	0.5	31	0.5	61	0.5	61	0.5	61	0.5	91	0.5	61	0.5	1	0.32	0.86	0.28
1	KSAC	0.5	91	0.5	91	0.5	1	0.5	91	0.5	31	0.5	31	0.5	61	0.5	61	0.5	61	0.5	91	0.5	61	0.5	1	0.32	0.86	0.28
0	KSAN	0.5	61	0.5	61	0.5	61	0.5	1	0.5	61	0.5	61	0.5	1	0.5	61	0.5	91	0.5	91	0.5	1	0.5	61	0.31	0.80	0.25
1	KSAN	0.5	61	0.5	61	0.5	61	0.5	1	0.5	61	0.5	61	0.5	1	0.5	61	0.5	91	0.5	91	0.5	1	0.5	61	0.31	0.80	0.25
0	KSAT	0.5	91	0.5	91	0.5	91	0.5	1	0.5	61	0.5	61	0.5	91	0.5	91	0.5	31	0.5	61	0.5	31	0.5	91	0.20	0.87	0.17
1	KSAT	0.5	91	0.5	91	0.5	91	0.5	1	0.5	61	0.5	61	0.5	91	0.5	91	0.5	31	0.5	61	0.5	31	0.5	91	0.20	0.87	0.17
0	KSEA	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.5	61	0.5	91	0.5	91	0.26	0.86	0.22
1	KSEA	0.5	91	0.5	91	0.5	91	0.5	91	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.5	61	0.5	91	0.5	91	0.26	0.86	0.22
0	KSFO	0.5	1	0.5	61	0.5	1	0.5	91	0.5	61	0.5	1	0.5	61	0.5	91	0.5	61	0.5	91	0.5	1	0.5	1	0.20	0.93	0.18
1	KSFO	0.5	1	0.5	61	0.5	1	0.5	91	0.5	61	0.5	1	0.5	61	0.5	91	0.5	61	0.5	91	0.5	1	0.5	1	0.20	0.93	0.18
0	KSMX	0.5	91	0.5	31	0.5	31	0.5	31	0.5	61	0.5	1	0.5	31	0.5	1	0.5	1	0.5	31	0.5	61	0.5	1	0.33	0.91	0.31
1	KSMX	0.5	91	0.5	31	0.5	31	0.5	31	0.5	61	0.5	1	0.5	31	0.5	1	0.5	1	0.5	31	0.5	61	0.5	1	0.33	0.91	0.31
0	KSTL	0.5	1	0.5	1	0.5	91	0.5	91	0.5	91	0.5	1	0.5	1	0.5	31	0.5	1	0.5	61	0.5	1	0.5	91	0.34	0.84	0.29
1	KSTL	0.5	1	0.5	1	0.5	91	0.5	91	0.5	91	0.5	1	0.5	1	0.5	31	0.5	1	0.5	61	0.5	1	0.5	91	0.34	0.84	0.29
0	KTPA	0.5	91	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.5	91	0.5	91	0.5	1	0.5	61	0.5	91	0.5	91	0.28	0.83	0.23
1	KTPA	0.5	91	0.5	91	0.5	1	0.5	91	0.5	61	0.5	91	0.5	91	0.5	91	0.5	1	0.5	61	0.5	91	0.5	91	0.28	0.83	0.23

A.22.5.5 Comparison of two values for RETRACTION THRESHOLD DIFFERENCE for OLAs 3 through 4 for all stations

Please see Table 79.

Table 79: Comparison of two values for RETRACTION THRESHOLD DIFFERENCE for OLAs 3 through 4 for all stations

OLA	station	RETRACTION THRESHOLD DIFFERENCE	lCrispConditionalEvaluated	lCrispRetractionThresholdReached	lCrispFoundNoReasonToRetract	lFuzzyStricterThanCrisp	lFuzzyStricterThanCrisp/lCrispFoundNoReasonToRetract	Increase in percentage of lFuzzyStricterThanCrisp/lCrispFoundNoReasonToRetract	RETRACTION THRESHOLD DIFFERENCE=0, OLA=3	RETRACTION THRESHOLD DIFFERENCE=1, OLA=3	RETRACTION THRESHOLD DIFFERENCE=0, OLA=4	RETRACTION THRESHOLD DIFFERENCE=1, OLA=4
3	KATL	0	1051150	754117	297033	112906	38%		38%			
3	KATL	1	1051150	695702	355448	171321	48%	10%		48%		
4	KATL	0	1051150	787199	263951	116086	44%				44%	

Table 79: (continued)

4	KATL	1	1051150	722770	328380	180515	55%	11%				55%
3	KBOS	0	1051117	837708	213409	87130	41%		41%			
3	KBOS	1	1051117	808810	242307	116028	48%	7%		48%		
4	KBOS	0	1051117	831026	220091	99309	45%				45%	
4	KBOS	1	1051117	797782	253335	132553	52%	7%				52%
3	KBWI	0	1051080	761673	289407	103547	36%		36%			
3	KBWI	1	1051080	703920	347160	161300	46%	11%		46%		
4	KBWI	0	1051080	794185	256895	110283	43%				43%	
4	KBWI	1	1051080	725042	326038	179426	55%	12%				55%
3	KCLE	0	1051112	722009	329103	155492	47%		47%			
3	KCLE	1	1051112	670422	380690	207079	54%	7%		54%		
4	KCLE	0	1051112	727893	323219	179610	56%				56%	
4	KCLE	1	1051112	668454	382658	239049	62%	7%				62%
3	KCLT	0	1051188	661719	389469	233003	60%		60%			
3	KCLT	1	1051188	572034	479154	322688	67%	8%		67%		
4	KCLT	0	1051188	692493	358695	258937	72%				72%	
4	KCLT	1	1051188	587116	464072	364314	79%	6%				79%
3	KCVG	0	1051140	717780	333360	196232	59%		59%			
3	KCVG	1	1051140	648034	403106	265978	66%	7%		66%		
4	KCVG	0	1051140	730888	320252	194928	61%				61%	
4	KCVG	1	1051140	654816	396324	271000	68%	8%				68%
3	KDCA	0	1050534	793999	256535	126236	49%		49%			
3	KDCA	1	1050534	751441	299093	168794	56%	7%		56%		
4	KDCA	0	1050534	800075	250459	147483	59%				59%	
4	KDCA	1	1050534	747238	303296	200320	66%	7%				66%
3	KDEN	0	1051057	694819	356238	135351	38%		38%			

Table 79: (continued)

3	KDEN	1	1051057	642345	408712	187825	46%	8%		46%		
4	KDEN	0	1051057	756437	294620	146218	50%				50%	
4	KDEN	1	1051057	693007	358050	209648	59%	9%				59%
3	KDFW	0	1051013	782964	268049	110034	41%		41%			
3	KDFW	1	1051013	744976	306037	148022	48%	7%		48%		
4	KDFW	0	1051013	790443	260570	110559	42%				42%	
4	KDFW	1	1051013	752790	298223	148212	50%	7%				50%
3	KDTW	0	1051112	790573	260539	88498	34%		34%			
3	KDTW	1	1051112	741503	309609	137568	44%	10%		44%		
4	KDTW	0	1051112	792642	258470	94016	36%				36%	
4	KDTW	1	1051112	741013	310099	145645	47%	11%				47%
3	KEUG	0	1050961	777877	273084	135119	49%		49%			
3	KEUG	1	1050961	725335	325626	187661	58%	8%		58%		
4	KEUG	0	1050961	766882	284079	157295	55%				55%	
4	KEUG	1	1050961	705901	345060	218276	63%	8%				63%
3	KIAH	0	1051050	741267	309783	163510	53%		53%			
3	KIAH	1	1051050	674190	376860	230587	61%	8%		61%		
4	KIAH	0	1051050	732256	318794	170859	54%				54%	
4	KIAH	1	1051050	664574	386476	238541	62%	8%				62%
3	KLAS	0	1050899	866482	184417	85292	46%		46%			
3	KLAS	1	1050899	825713	225186	126061	56%	10%		56%		
4	KLAS	0	1050899	867701	183198	86095	47%				47%	
4	KLAS	1	1050899	827420	223479	126376	57%	10%				57%
3	KLAX	0	1050894	689170	361724	114184	32%		32%			
3	KLAX	1	1050894	633204	417690	170150	41%	9%		41%		
4	KLAX	0	1050894	708321	342573	113244	33%				33%	

Table 79: (continued)

4	KLAX	1	1050894	652456	398438	169109	42%	9%				42%
3	KLGA	0	1051144	848065	203079	105134	52%		52%			
3	KLGA	1	1051144	809147	241997	144052	60%	8%		60%		
4	KLGA	0	1051144	826118	225026	113496	50%				50%	
4	KLGA	1	1051144	784214	266930	155400	58%	8%				58%
3	KMCI	0	1051087	645927	405160	181086	45%		45%			
3	KMCI	1	1051087	581491	469596	245522	52%	8%		52%		
4	KMCI	0	1051087	652458	398629	185316	46%				46%	
4	KMCI	1	1051087	585640	465447	252134	54%	8%				54%
3	KMCO	0	1051136	820396	230740	133647	58%		58%			
3	KMCO	1	1051136	767817	283319	186226	66%	8%		66%		
4	KMCO	0	1051136	806564	244572	152546	62%				62%	
4	KMCO	1	1051136	746158	304978	212952	70%	7%				70%
3	KMSP	0	1051121	860814	190307	74626	39%		39%			
3	KMSP	1	1051121	826127	224994	109313	49%	9%		49%		
4	KMSP	0	1051121	857447	193674	78742	41%				41%	
4	KMSP	1	1051121	821154	229967	115035	50%	9%				50%
3	KORD	0	1051064	724595	326469	164170	50%		50%			
3	KORD	1	1051064	669328	381736	219437	57%	7%		57%		
4	KORD	0	1051064	717450	333614	174162	52%				52%	
4	KORD	1	1051064	657672	393392	233940	59%	7%				59%
3	KPHL	0	1051124	696194	354930	140689	40%		40%			
3	KPHL	1	1051124	646562	404562	190321	47%	7%		47%		
4	KPHL	0	1051124	741888	309236	166639	54%				54%	
4	KPHL	1	1051124	676945	374179	231582	62%	8%				62%
3	KPHX	0	1050999	793937	257062	68279	27%		27%			

Table 79: (continued)

3	KPHX	1	1050999	762349	288650	99867	35%	8%		35%		
4	KPHX	0	1050999	916283	134716	72323	54%				54%	
4	KPHX	1	1050999	864753	186246	123853	66%	13%				66%
3	KPIT	0	1051105	737087	314018	175265	56%		56%			
3	KPIT	1	1051105	667649	383456	244703	64%	8%		64%		
4	KPIT	0	1051105	742269	308836	180177	58%				58%	
4	KPIT	1	1051105	669443	381662	253003	66%	8%				66%
3	KSAC	0	1050950	755064	295886	132824	45%		45%			
3	KSAC	1	1050950	681701	369249	206187	56%	11%		56%		
4	KSAC	0	1050950	774246	276704	139123	50%				50%	
4	KSAC	1	1050950	694806	356144	218563	61%	11%				61%
3	KSAN	0	1050942	724452	326490	187399	57%		57%			
3	KSAN	1	1050942	682387	368555	229464	62%	5%		62%		
4	KSAN	0	1050942	739801	311141	183089	59%				59%	
4	KSAN	1	1050942	695258	355684	227632	64%	5%				64%
3	KSAT	0	1051059	759105	291954	155122	53%		53%			
3	KSAT	1	1051059	702972	348087	211255	61%	8%		61%		
4	KSAT	0	1051059	776116	274943	152739	56%				56%	
4	KSAT	1	1051059	717576	333483	211279	63%	8%				63%
3	KSEA	0	1050356	675363	374993	195477	52%		52%			
3	KSEA	1	1050356	605911	444445	264929	60%	7%		60%		
4	KSEA	0	1050356	667525	382831	198766	52%				52%	
4	KSEA	1	1050356	594022	456334	272269	60%	8%				60%
3	KSFO	0	1050904	782433	268471	115018	43%		43%			
3	KSFO	1	1050904	754832	296072	142619	48%	5%		48%		
4	KSFO	0	1050904	782433	268471	115018	43%				43%	

Table 79: (continued)

4	KSFO	1	1050904	754832	296072	142619	48%	5%				48%
3	KSMX	0	1050960	822403	228557	96097	42%		42%			
3	KSMX	1	1050960	788031	262929	130469	50%	8%		50%		
4	KSMX	0	1050960	824210	226750	99886	44%				44%	
4	KSMX	1	1050960	790506	260454	133590	51%	7%				51%
3	KSTL	0	1051128	769605	281523	123509	44%		44%			
3	KSTL	1	1051128	728849	322279	164265	51%	7%		51%		
4	KSTL	0	1051128	833532	217596	112619	52%				52%	
4	KSTL	1	1051128	787110	264018	159041	60%	8%				60%
3	KTPA	0	1051144	794480	256664	136915	53%		53%			
3	KTPA	1	1051144	716268	334876	215127	64%	11%		64%		
4	KTPA	0	1051144	796919	254225	137017	54%				54%	
4	KTPA	1	1051144	717706	333438	216230	65%	11%				65%
							Average:	8%	46%	54%	51%	59%
							Standard Deviation:	2%	8%	8%	8%	8%

In Section A.22.1, we determined the answer to the question, Does the fuzzy code retract the harvester in every case that the average windspeed is between the two crisp retraction thresholds, i.e., when the more lenient crisp does not retract and when the less lenient crisp code retracts (if the fuzzy code were to examine every case)? The answer is Yes for every station for OLAs 3 and 4. Details are below, which are printed by the following print statement:

```
    System.out.println("FuzzyVsCrisp: " + station + "," +
                RETRACTION_THRESHOLD_DIFFERENCE + "," +
                bEveryMoreLenientCaseRestricted + "," +
                bEveryMoreLenientCaseReachesFuzzyRetractionThreshold);
```

```
FuzzyVsCrisp: KATL,1,true,true
```

```
FuzzyVsCrisp: KATL,1,true,true
FuzzyVsCrisp: KBOS,1,true,true
FuzzyVsCrisp: KBOS,1,true,true
FuzzyVsCrisp: KBWI,1,true,true
FuzzyVsCrisp: KBWI,1,true,true
FuzzyVsCrisp: KCLE,1,true,true
FuzzyVsCrisp: KCLE,1,true,true
FuzzyVsCrisp: KCLT,1,true,true
FuzzyVsCrisp: KCLT,1,true,true
FuzzyVsCrisp: KCVG,1,true,true
FuzzyVsCrisp: KCVG,1,true,true
FuzzyVsCrisp: KDCA,1,true,true
FuzzyVsCrisp: KDCA,1,true,true
FuzzyVsCrisp: KDEN,1,true,true
FuzzyVsCrisp: KDEN,1,true,true
FuzzyVsCrisp: KDFW,1,true,true
FuzzyVsCrisp: KDFW,1,true,true
FuzzyVsCrisp: KDTW,1,true,true
FuzzyVsCrisp: KDTW,1,true,true
FuzzyVsCrisp: KEUG,1,true,true
FuzzyVsCrisp: KEUG,1,true,true
FuzzyVsCrisp: KIAH,1,true,true
FuzzyVsCrisp: KIAH,1,true,true
FuzzyVsCrisp: KLAS,1,true,true
FuzzyVsCrisp: KLAS,1,true,true
FuzzyVsCrisp: KLAX,1,true,true
FuzzyVsCrisp: KLAX,1,true,true
FuzzyVsCrisp: KLGA,1,true,true
FuzzyVsCrisp: KLGA,1,true,true
FuzzyVsCrisp: KMCI,1,true,true
FuzzyVsCrisp: KMCI,1,true,true
FuzzyVsCrisp: KMCO,1,true,true
FuzzyVsCrisp: KMCO,1,true,true
FuzzyVsCrisp: KMSP,1,true,true
FuzzyVsCrisp: KMSP,1,true,true
FuzzyVsCrisp: KORD,1,true,true
FuzzyVsCrisp: KORD,1,true,true
FuzzyVsCrisp: KPHL,1,true,true
FuzzyVsCrisp: KPHL,1,true,true
FuzzyVsCrisp: KPHX,1,true,true
FuzzyVsCrisp: KPHX,1,true,true
FuzzyVsCrisp: KPIT,1,true,true
FuzzyVsCrisp: KPIT,1,true,true
FuzzyVsCrisp: KSAC,1,true,true
FuzzyVsCrisp: KSAC,1,true,true
FuzzyVsCrisp: KSAN,1,true,true
FuzzyVsCrisp: KSAN,1,true,true
```

```
FuzzyVsCrisp: KSAT,1,true,true
FuzzyVsCrisp: KSAT,1,true,true
FuzzyVsCrisp: KSEA,1,true,true
FuzzyVsCrisp: KSEA,1,true,true
FuzzyVsCrisp: KSFO,1,true,true
FuzzyVsCrisp: KSFO,1,true,true
FuzzyVsCrisp: KSMX,1,true,true
FuzzyVsCrisp: KSMX,1,true,true
FuzzyVsCrisp: KSTL,1,true,true
FuzzyVsCrisp: KSTL,1,true,true
FuzzyVsCrisp: KTPA,1,true,true
FuzzyVsCrisp: KTPA,1,true,true
```

A.22.5.6 OLAs 3 and 4: Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) 1.2 using only crisp code to retract

Please see Table 80.

Table 80: Results of the processing of OLA 3 by Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.2 when the subtraction Deployment Threshold less Retraction Threshold equals 0 and when it equals 1

OLA	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
3	KATL	0.5	1	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	61	0.5	31	0.368	0.878	0.323
4	KATL	0.5	61	0.5	1	0.5	31	0.5	31	0.5	31	0.5	1	0.5	1	0.5	61	0.5	91	0.5	31	0.5	61	0.5	31	0.365	0.921	0.336
3	KBOS	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.325	0.900	0.292
4	KBOS	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.325	0.900	0.292
3	KBWI	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.452	0.842	0.381
4	KBWI	0.5	1	0.5	31	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.451	0.871	0.392
3	KCLE	0.5	91	0.5	31	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.329	0.920	0.302
4	KCLE	0.5	91	0.5	31	0.5	91	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	91	0.5	1	0.332	0.934	0.310
3	KCLT	0.5	1	0.5	91	0.5	61	0.5	91	0.5	91	0.5	31	0.5	1	0.5	31	0.5	61	0.5	1	0.5	1	0.5	31	0.446	0.884	0.394
4	KCLT	0.5	31	0.5	91	0.5	61	0.5	91	0.5	91	0.5	31	0.5	1	0.5	31	0.5	61	0.5	31	0.5	1	0.5	31	0.419	0.900	0.377
3	KCVG	0.5	61	0.5	61	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	91	0.5	1	0.291	0.920	0.268
4	KCVG	0.5	91	0.5	61	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	91	0.5	1	0.292	0.920	0.269
3	KDCA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.283	0.902	0.255
4	KDCA	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.320	0.922	0.295
3	KDEN	0.5	1	0.5	1	0.5	31	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.342	0.892	0.305
4	KDEN	0.5	1	0.5	1	0.5	31	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.367	0.889	0.326
3	KDFW	0.5	1	0.5	1	0.5	91	0.5	61	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.330	0.927	0.306
4	KDFW	0.5	1	0.5	1	0.5	91	0.5	61	0.5	91	0.5	91	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.344	0.931	0.320
3	KDTW	0.5	61	0.5	31	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.363	0.924	0.335
4	KDTW	0.5	61	0.5	31	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.381	0.925	0.353
3	KEUG	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.505	0.890	0.450
4	KEUG	0.5	91	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.481	0.901	0.433
3	KIAH	0.5	61	0.5	31	0.5	91	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.426	0.896	0.382
4	KIAH	0.5	61	0.5	31	0.5	91	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.426	0.896	0.382
3	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.401	0.891	0.357
4	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.401	0.891	0.357
3	KLAX	0.5	1	0.5	31	0.5	1	0.5	1	0.5	91	0.5	1	0.5	31	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.507	0.911	0.462
4	KLAX	0.5	1	0.5	31	0.5	1	0.5	31	0.5	91	0.5	1	0.5	31	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.527	0.912	0.481
3	KLGA	0.5	61	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.317	0.880	0.279
4	KLGA	0.5	61	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.317	0.880	0.279
3	KMCI	0.5	91	0.5	1	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	61	0.5	1	0.342	0.936	0.320
4	KMCI	0.5	91	0.5	1	0.5	91	0.5	91	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	91	0.5	61	0.5	1	0.349	0.947	0.331
3	KMCO	0.5	1	0.5	1	0.5	1	0.5	31	0.5	91	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.414	0.890	0.368
4	KMCO	0.5	1	0.5	1	0.5	1	0.5	31	0.5	91	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.414	0.890	0.368

Table 80: (continued)

3	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.336	0.864	0.290
4	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.326	0.871	0.284
3	KORD	0.5	1	0.5	1	0.5	91	0.5	61	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.274	0.940	0.257
4	KORD	0.5	1	0.5	1	0.5	91	0.5	61	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.275	0.931	0.256
3	KPHL	0.5	1	0.5	91	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.350	0.911	0.319
4	KPHL	0.5	1	0.5	91	0.5	1	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.401	0.913	0.366
3	KPHX	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.217	0.849	0.184
4	KPHX	0.5	1	0.5	1	0.5	1	0.5	31	0.5	31	0.5	31	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.271	0.761	0.206
3	KPIT	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	31	0.5	31	0.423	0.885	0.374
4	KPIT	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	31	0.5	31	0.436	0.878	0.383
3	KSAC	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.575	0.833	0.479
4	KSAC	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	31	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.513	0.887	0.455
3	KSAN	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	91	0.5	61	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.461	0.894	0.412
4	KSAN	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	91	0.5	61	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.461	0.894	0.412
3	KSAT	0.5	1	0.5	1	0.5	31	0.5	31	0.5	61	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.359	0.897	0.322
4	KSAT	0.5	1	0.5	31	0.5	31	0.5	31	0.5	91	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.371	0.898	0.333
3	KSEA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.476	0.894	0.425
4	KSEA	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	91	0.464	0.903	0.419
3	KSFO	0.5	1	0.5	1	0.5	31	0.5	91	0.5	91	0.5	1	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.353	0.973	0.343
4	KSFO	0.5	1	0.5	1	0.5	31	0.5	91	0.5	91	0.5	61	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.359	0.977	0.351
3	KSMX	0.5	1	0.5	1	0.5	31	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.602	0.929	0.559
4	KSMX	0.5	1	0.5	1	0.5	31	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.602	0.929	0.559
3	KSTL	0.5	1	0.5	1	0.5	1	0.5	61	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.273	0.899	0.245
4	KSTL	0.5	1	0.5	1	0.5	1	0.5	61	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.273	0.899	0.245
3	KTPA	0.5	1	0.5	31	0.5	31	0.5	1	0.5	31	0.5	31	0.5	1	0.5	1	0.5	31	0.5	31	0.5	31	0.5	31	0.451	0.861	0.388
4	KTPA	0.5	61	0.5	31	0.5	31	0.5	31	0.5	31	0.5	31	0.5	1	0.5	61	0.5	31	0.5	61	0.5	31	0.5	31	0.456	0.891	0.406

A.22.5.7 OLAs 3 and 4: Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.3

Please see Table 81.

Table 81: Results of the processing of OLA 3 by Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) Rev. 1.3 when the subtraction Deployment Threshold less Retraction Threshold equals 0 and when it equals 1

where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n

OLA	station	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
3	KATL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	31	0.31	0.72	0.22
4	KATL	0.5	61	0.5	61	0.5	91	0.5	31	0.5	61	0.5	61	0.5	1	0.5	91	0.5	91	0.5	61	0.5	91	0.5	31	0.31	0.92	0.28
3	KBOS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.26	0.76	0.20
4	KBOS	0.5	91	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.22	0.91	0.21
3	KBWI	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.37	0.73	0.27
4	KBWI	0.5	1	0.5	61	0.5	1	0.5	31	0.5	1	0.5	31	0.5	1	0.5	91	0.5	61	0.5	1	0.5	1	0.5	61	0.39	0.77	0.30
3	KCLE	0.5	91	0.5	1	0.5	91	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.29	0.88	0.26
4	KCLE	0.5	91	0.5	31	0.5	91	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.5	91	0.5	91	0.29	0.90	0.26
3	KCLT	0.5	1	0.5	91	0.5	91	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.42	0.65	0.27
4	KCLT	0.5	91	0.5	91	0.5	91	0.5	91	0.5	61	0.5	91	0.5	61	0.5	1	0.5	91	0.5	61	0.5	1	0.5	1	0.33	0.80	0.27
3	KCVG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	1	0.31	0.72	0.22
4	KCVG	0.5	91	0.5	61	0.5	61	0.5	61	0.5	31	0.5	1	0.5	1	0.5	61	0.5	61	0.5	31	0.5	61	0.5	1	0.30	0.84	0.25
3	KDCA	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.36	0.72	0.26
4	KDCA	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	31	0.5	31	0.5	1	0.5	61	0.28	0.83	0.23
3	KDEN	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.39	0.83	0.33
4	KDEN	0.5	1	0.5	1	0.5	1	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.38	0.82	0.31
3	KDFW	0.5	1	0.5	1	0.5	91	0.5	1	0.5	61	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.31	0.87	0.27
4	KDFW	0.5	1	0.5	61	0.5	91	0.5	31	0.5	61	0.5	91	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.29	0.90	0.26
3	KDTW	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.36	0.86	0.31
4	KDTW	0.5	31	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.35	0.85	0.30
3	KEUG	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.52	0.71	0.37
4	KEUG	0.5	31	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.48	0.75	0.37
3	KIAH	0.5	1	0.5	31	0.5	91	0.5	91	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.38	0.79	0.30
4	KIAH	0.5	31	0.5	31	0.5	91	0.5	91	0.5	91	0.5	61	0.5	1	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.32	0.90	0.29
3	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.36	0.82	0.30
4	KLAS	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.36	0.82	0.30
3	KLAX	0.5	1	0.5	1	0.5	31	0.5	1	0.5	31	0.5	31	0.5	31	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.57	0.86	0.49
4	KLAX	0.5	91	0.5	31	0.5	61	0.5	31	0.5	31	0.5	31	0.5	31	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.57	0.89	0.50
3	KLGA	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.29	0.76	0.22
4	KLGA	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	61	0.5	1	0.5	1	0.5	61	0.27	0.79	0.21
3	KMCI	0.5	61	0.5	1	0.5	61	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.32	0.88	0.28
4	KMCI	0.5	61	0.5	1	0.5	61	0.5	91	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	91	0.5	1	0.33	0.90	0.29
3	KMCO	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.35	0.73	0.26
4	KMCO	0.5	1	0.5	1	0.5	31	0.5	61	0.5	31	0.5	1	0.5	1	0.5	91	0.5	91	0.5	91	0.5	1	0.5	1	0.34	0.81	0.27
3	KMSP	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.30	0.73	0.22

Table 81: (continued)

4	KMSP	0.5	31	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.26	0.84	0.22
3	KORD	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.5	1	0.30	0.81	0.24
4	KORD	0.5	1	0.5	31	0.5	31	0.5	61	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.5	61	0.5	91	0.5	1	0.30	0.87	0.26
3	KPHL	0.5	91	0.5	31	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.30	0.87	0.26
4	KPHL	0.5	91	0.5	91	0.5	1	0.5	31	0.5	61	0.5	1	0.5	1	0.5	61	0.5	1	0.5	61	0.5	1	0.5	91	0.31	0.87	0.27
3	KPHX	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.29	0.48	0.14
4	KPHX	0.5	1	0.5	1	0.5	31	0.5	61	0.5	31	0.5	61	0.5	31	0.5	61	0.5	1	0.5	61	0.5	1	0.5	1	0.23	0.71	0.16
3	KPIT	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.31	0.79	0.24
4	KPIT	0.5	1	0.5	1	0.5	31	0.5	91	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	91	0.5	1	0.5	91	0.27	0.84	0.23
3	KSAC	0.5	1	0.5	1	0.5	1	0.5	61	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.47	0.68	0.32
4	KSAC	0.5	1	0.5	1	0.5	1	0.5	61	0.5	91	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.50	0.76	0.38
3	KSAN	0.5	1	0.5	1	0.5	31	0.5	91	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.42	0.68	0.29
4	KSAN	0.5	91	0.5	1	0.5	31	0.5	91	0.5	91	0.5	61	0.5	61	0.5	91	0.5	1	0.5	1	0.5	1	0.5	1	0.34	0.82	0.28
3	KSAT	0.5	1	0.5	1	0.5	1	0.5	91	0.5	31	0.5	91	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.35	0.75	0.26
4	KSAT	0.5	91	0.5	1	0.5	31	0.5	91	0.5	61	0.5	91	0.5	31	0.5	31	0.5	1	0.5	61	0.5	1	0.5	1	0.28	0.87	0.24
3	KSEA	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	61	0.5	61	0.38	0.84	0.32
4	KSEA	0.5	1	0.5	1	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	91	0.5	61	0.39	0.84	0.33
3	KSFO	0.5	1	0.5	1	0.5	31	0.5	91	0.5	61	0.5	91	0.5	31	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.34	0.98	0.34
4	KSFO	0.5	1	0.5	1	0.5	31	0.5	91	0.5	61	0.5	91	0.5	31	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.34	0.98	0.34
3	KSMX	0.5	1	0.5	1	0.5	1	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.57	0.88	0.50
4	KSMX	0.5	1	0.5	1	0.5	31	0.5	61	0.5	61	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.55	0.89	0.49
3	KSTL	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.37	0.69	0.26
4	KSTL	0.5	1	0.5	1	0.5	61	0.5	61	0.5	31	0.5	91	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.26	0.87	0.23
3	KTPA	0.5	1	0.5	1	0.5	91	0.5	1	0.5	31	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	31	0.5	1	0.45	0.70	0.31
4	KTPA	0.5	91	0.5	1	0.5	91	0.5	31	0.5	91	0.5	91	0.5	1	0.5	61	0.5	61	0.5	61	0.5	91	0.5	1	0.39	0.82	0.32

A.23 SENSITIVITY ANALYSIS OF THE RETRACTION THRESHOLD DIFFERENCE ON AGING (VARIANT 0X0) FOR OLAS 3 & 4

A.23.1 RTD is 0

Please see Table 82 on the next page.

Table 82: Results of the processing of OLAs 3 and 4 by Aging (variant 0x0) Rev. 1.1 when the RTD is 0

where $y(n)$ is the y-intercept and $r(n)$ is the running average size minutes for month n

OLA	station	$y(1)$	$r(1)$	$y(2)$	$r(2)$	$y(3)$	$r(3)$	$y(4)$	$r(4)$	$y(5)$	$r(5)$	$y(6)$	$r(6)$	$y(7)$	$r(7)$	$y(8)$	$r(8)$	$y(9)$	$r(9)$	$y(10)$	$r(10)$	$y(11)$	$r(11)$	$y(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
3	KATL	17	31	17	31	17	121	17	31	17	31	7	121	7	31	17	31	17	121	17	31	17	1	17	61	0.36	0.92	0.34
4	KATL	17	31	17	61	17	121	17	61	17	61	7	121	7	91	17	31	17	121	17	31	17	61	17	91	0.37	0.94	0.35
3	KBOS	19	61	9	31	19	31	19	31	9	121	9	91	9	121	9	121	9	121	9	121	9	31	19	91	0.32	0.94	0.30
4	KBOS	19	61	9	31	19	31	19	31	9	121	9	91	9	121	9	121	9	121	9	121	9	121	19	91	0.32	0.94	0.30
3	KBWI	17	31	17	31	7	31	17	91	7	31	7	121	7	91	17	31	7	121	17	31	7	61	17	31	0.46	0.91	0.42
4	KBWI	17	31	17	61	7	31	17	91	7	31	7	121	7	91	17	31	7	121	17	31	7	61	17	31	0.46	0.91	0.42
3	KCLE	18	91	18	31	18	31	18	121	18	31	18	31	18	31	8	121	18	31	18	31	18	61	18	61	0.38	0.92	0.35
4	KCLE	18	91	18	61	18	31	18	121	18	31	18	31	18	31	8	121	18	61	18	31	18	61	18	61	0.38	0.92	0.35
3	KCLT	15	31	15	61	15	91	15	121	25	31	15	31	5	1	15	31	15	31	15	61	15	31	15	1	0.43	0.89	0.38
4	KCLT	15	31	15	61	15	91	15	121	25	31	15	31	5	121	15	61	15	61	15	61	15	91	15	31	0.42	0.91	0.38
3	KCVG	17	121	17	31	17	31	27	31	17	31	17	31	7	91	7	121	17	61	17	31	17	121	17	91	0.39	0.91	0.35
4	KCVG	17	121	17	61	17	31	27	31	17	31	17	31	7	91	7	121	17	61	17	31	17	121	17	91	0.39	0.91	0.35
3	KDCA	17	31	17	31	17	31	17	31	17	31	7	121	7	121	17	31	7	121	17	31	17	31	17	91	0.37	0.93	0.34
4	KDCA	17	31	17	31	17	61	17	31	17	61	7	121	7	121	17	31	7	121	17	31	17	91	17	91	0.37	0.93	0.34
3	KDEN	18	31	18	1	18	31	18	121	18	61	18	61	18	1	18	1	18	1	18	1	18	1	18	1	0.37	0.87	0.32
4	KDEN	18	31	18	31	18	31	18	121	18	61	18	61	18	31	18	31	18	31	18	31	18	31	18	91	0.37	0.93	0.34
3	KDFW	19	91	19	31	19	121	19	91	19	121	19	31	9	61	9	31	9	31	9	91	19	1	19	31	0.34	0.93	0.32
4	KDFW	19	91	19	31	19	121	19	91	19	121	19	31	9	61	9	91	9	31	9	121	19	31	19	31	0.35	0.94	0.33
3	KDTW	18	61	18	31	18	91	18	31	18	31	8	31	8	121	8	31	18	31	18	121	18	91	18	61	0.43	0.93	0.41
4	KDTW	18	61	18	31	18	91	18	31	18	31	8	61	8	121	8	31	18	31	18	121	18	91	18	61	0.43	0.93	0.41
3	KEUG	16	61	16	61	16	31	16	1	16	31	16	31	16	91	16	61	16	1	16	1	16	91	16	1	0.46	0.90	0.42
4	KEUG	16	61	16	61	16	31	16	31	16	31	16	31	16	91	16	61	16	31	16	31	16	91	16	31	0.48	0.92	0.44
3	KIAH	17	121	17	61	17	61	17	121	17	31	17	91	17	1	7	91	7	121	17	31	17	91	17	121	0.42	0.92	0.38
4	KIAH	17	121	17	61	17	61	17	121	17	31	17	91	7	121	7	91	7	121	17	31	17	91	17	121	0.42	0.93	0.39
3	KLAS	8	1	8	1	18	31	8	121	18	31	18	121	8	91	18	31	8	1	8	91	8	1	8	1	0.37	0.86	0.32
4	KLAS	8	31	8	61	18	61	8	121	18	31	18	121	8	91	18	31	8	31	8	91	8	61	8	61	0.38	0.92	0.35
3	KLAX	17	31	17	1	17	31	17	121	17	91	17	61	17	61	17	61	17	1	17	31	7	31	7	31	0.49	0.92	0.45
4	KLAX	17	91	17	31	17	31	17	121	17	91	17	61	17	61	17	31	17	31	17	31	7	61	7	31	0.51	0.93	0.47
3	KLGA	19	31	19	31	9	121	9	121	9	121	9	61	9	121	9	91	9	121	9	121	19	31	19	121	0.33	0.92	0.31
4	KLGA	19	61	19	31	9	121	9	121	9	121	9	61	9	121	9	91	9	121	9	121	19	31	19	121	0.33	0.92	0.31
3	KMCI	18	31	18	31	28	31	18	121	18	91	18	31	18	31	18	1	18	31	18	91	28	31	18	31	0.37	0.93	0.34
4	KMCI	18	31	18	31	28	31	18	121	18	91	18	31	18	31	18	31	18	31	18	121	28	31	18	31	0.37	0.94	0.34
3	KMCO	17	31	7	91	17	1	17	31	17	121	7	121	7	91	17	1	17	121	17	1	17	1	17	31	0.40	0.89	0.36
4	KMCO	17	31	7	91	17	31	17	31	17	121	7	121	7	91	17	31	17	121	17	31	17	31	17	31	0.42	0.92	0.39
3	KMSP	8	121	8	91	8	1	8	121	8	31	8	61	8	121	8	61	8	91	8	121	8	1	8	1	0.34	0.93	0.31
4	KMSP	8	121	8	91	8	121	8	121	8	31	8	61	8	121	8	121	8	91	8	121	8	31	8	31	0.34	0.93	0.32
3	KORD	18	31	18	31	18	91	18	121	18	31	8	31	18	31	8	121	8	121	18	121	18	121	18	31	0.37	0.93	0.34
4	KORD	18	31	18	31	18	91	18	121	18	31	8	31	8	61	8	121	8	121	18	121	18	121	18	121	0.36	0.93	0.34
3	KPHL	18	121	18	31	18	31	18	91	18	31	8	121	8	121	18	31	18	31	18	61	18	31	28	61	0.43	0.93	0.40
4	KPHL	18	121	18	31	18	31	18	91	8	121	8	121	8	121	18	31	18	31	18	61	18	31	28	61	0.42	0.93	0.39
3	KPHX	6	1	6	1	6	31	6	31	6	121	6	61	6	121	6	31	6	31	6	61	6	1	6	1	0.33	0.82	0.27
4	KPHX	6	61	6	121	6	91	6	61	6	121	6	61	6	121	6	61	6	121	6	91	6	31	6	121	0.33	0.88	0.29
3	KPIT	17	121	17	31	17	31	27	61	17	31	7	31	7	61	7	61	7	31	17	31	17	61	17	91	0.43	0.91	0.40
4	KPIT	17	121	17	91	17	121	27	61	17	31	7	61	7	61	7	91	7	91	17	61	17	61	17	91	0.43	0.92	0.39

Table 82: (continued)

3	KSAC	6	121	6	1	16	31	16	91	16	91	16	31	16	91	16	31	16	1	16	1	6	1	6	1	0.45	0.90	0.41
4	KSAC	6	121	6	61	16	31	16	91	16	91	16	61	16	91	16	31	16	31	16	31	6	91	6	121	0.52	0.94	0.48
3	KSAN	5	1	15	1	15	31	15	1	15	1	15	1	5	91	15	91	5	61	5	121	5	31	5	61	0.42	0.88	0.37
4	KSAN	5	61	15	31	15	31	15	31	15	31	15	31	5	91	15	91	5	61	5	121	5	31	5	61	0.45	0.91	0.42
3	KSAT	17	31	7	91	17	31	17	121	17	31	17	121	17	31	17	1	7	121	7	91	17	61	17	31	0.32	0.93	0.30
4	KSAT	17	61	7	121	17	31	17	121	17	31	17	121	17	31	7	121	7	121	7	91	17	61	17	31	0.33	0.93	0.31
3	KSEA	16	91	16	1	16	31	16	1	16	1	16	1	6	31	16	1	6	31	16	61	16	61	16	121	0.40	0.89	0.35
4	KSEA	16	91	16	31	16	31	16	91	16	61	6	121	6	31	16	61	6	31	16	61	16	61	16	121	0.42	0.93	0.39
3	KSFO	10	121	10	61	20	121	20	121	30	31	30	91	20	121	20	91	20	121	20	1	10	1	10	1	0.40	0.94	0.37
4	KSFO	10	121	10	91	20	121	20	121	30	31	30	91	20	121	20	121	20	121	20	1	10	61	10	91	0.40	0.95	0.38
3	KSMX	8	121	8	121	18	91	18	61	18	91	18	1	8	61	8	61	8	61	8	61	8	31	8	31	0.62	0.91	0.57
4	KSMX	8	121	8	121	18	91	18	61	18	91	18	31	8	61	8	61	8	61	8	61	8	31	8	91	0.62	0.91	0.56
3	KSTL	17	31	17	31	17	1	17	91	17	91	17	31	7	121	7	121	7	61	17	31	17	31	17	31	0.35	0.92	0.32
4	KSTL	17	61	17	31	17	61	17	61	17	91	17	31	7	121	7	121	7	61	17	31	17	31	17	31	0.38	0.92	0.35
3	KTPA	6	61	6	121	16	31	16	1	6	121	16	1	6	1	6	121	16	31	16	1	16	1	16	1	0.39	0.85	0.34
4	KTPA	6	61	6	121	16	31	6	61	6	121	16	31	6	61	6	121	16	31	16	31	16	31	16	91	0.41	0.90	0.37

A.23.2 RTD is 1

Please see Table 41 on page 234.

A.23.3 RTD is 2

Please see Table 83 on the next page.

Table 83: Results of the processing of OLAs 3 and 4 by Aging (variant 0x0) Rev. 1.1 when the RTD is 2

OLA	station	$y(1)$	$r(1)$	$y(2)$	$r(2)$	$y(3)$	$r(3)$	$y(4)$	$r(4)$	$y(5)$	$r(5)$	$y(6)$	$r(6)$	$y(7)$	$r(7)$	$y(8)$	$r(8)$	$y(9)$	$r(9)$	$y(10)$	$r(10)$	$y(11)$	$r(11)$	$y(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $y(n)$ is the y-intercept and $r(n)$ is the running average size minutes for month n																										
3	KATL	7	91	7	121	17	61	7	121	17	61	7	61	7	61	7	91	17	91	7	91	7	31	17	91	0.38	0.90	0.35
4	KATL	7	91	7	121	17	121	7	121	17	61	7	91	7	91	7	121	17	91	7	121	7	121	17	91	0.38	0.91	0.35
3	KBOS	19	31	9	61	9	121	9	61	9	121	9	121	9	61	9	121	9	121	9	121	9	91	9	61	0.34	0.92	0.31
4	KBOS	19	91	9	121	9	121	9	91	9	121	9	121	9	61	9	121	9	121	9	121	9	121	9	61	0.34	0.93	0.31
3	KBWI	17	31	17	61	7	31	17	61	7	61	7	61	7	121	7	121	7	31	17	61	7	61	17	121	0.46	0.88	0.40
4	KBWI	17	61	17	91	7	91	17	121	7	121	7	61	7	121	7	121	7	121	17	61	7	61	17	121	0.45	0.90	0.40
3	KCLE	18	61	18	31	18	61	18	121	8	61	18	61	8	91	8	31	8	91	18	61	18	61	18	31	0.39	0.89	0.35
4	KCLE	18	61	18	61	18	91	18	121	8	61	18	61	8	121	8	61	8	91	18	91	18	61	18	91	0.39	0.90	0.35
3	KCLT	15	31	15	31	15	31	15	61	15	121	5	61	5	91	5	61	15	61	15	61	15	31	15	61	0.43	0.86	0.37
4	KCLT	15	91	15	121	15	91	15	121	15	121	5	91	5	121	5	91	15	121	15	91	5	121	15	91	0.43	0.89	0.38
3	KCVG	17	31	17	31	17	61	17	121	17	61	17	31	7	121	7	121	7	61	17	61	17	91	17	31	0.40	0.87	0.35
4	KCVG	17	91	17	61	17	61	17	121	17	61	17	31	7	121	7	121	7	91	17	61	17	91	17	91	0.40	0.88	0.36
3	KDCA	17	91	7	121	17	61	7	31	7	121	7	121	7	61	17	61	7	61	7	61	17	61	17	61	0.36	0.92	0.33
4	KDCA	17	91	7	121	17	61	7	121	7	121	7	121	7	61	17	121	7	61	7	121	17	61	17	91	0.36	0.92	0.34
3	KDEN	18	1	18	31	18	1	18	121	18	61	8	121	8	121	8	121	8	61	8	31	18	1	8	91	0.37	0.88	0.32
4	KDEN	18	31	18	31	18	31	18	121	18	61	8	121	8	121	8	121	8	121	8	61	18	61	18	91	0.38	0.91	0.35
3	KDFW	19	31	19	31	19	91	19	31	19	121	9	31	9	121	9	31	9	121	9	121	19	61	19	61	0.34	0.92	0.31
4	KDFW	19	61	19	61	19	91	19	31	19	121	9	61	9	121	9	61	9	121	9	121	19	61	19	121	0.34	0.93	0.31
3	KDTW	18	91	18	31	18	61	18	61	18	31	8	91	8	61	8	61	8	121	18	31	18	61	18	121	0.41	0.90	0.37
4	KDTW	18	91	18	31	18	61	18	61	18	121	8	91	8	91	8	61	8	121	18	31	18	91	18	121	0.41	0.91	0.37
3	KEUG	16	91	6	121	16	61	6	1	16	31	6	121	16	31	16	31	16	1	16	31	16	1	6	61	0.46	0.87	0.40
4	KEUG	16	91	6	121	16	91	6	91	16	91	6	121	6	121	16	61	16	31	16	61	16	61	6	61	0.47	0.91	0.43
3	KIAH	17	61	17	31	17	121	17	61	17	121	17	61	7	91	7	121	7	91	7	91	17	61	17	91	0.43	0.91	0.39
4	KIAH	17	91	17	61	17	121	17	121	17	91	7	121	7	91	7	121	7	121	7	91	17	91	17	91	0.44	0.91	0.40
3	KLAS	8	31	8	1	8	61	8	91	18	31	18	31	8	61	18	31	8	1	8	31	8	1	8	31	0.37	0.83	0.31
4	KLAS	8	91	8	61	8	91	8	121	18	61	18	31	8	121	18	31	8	31	8	121	8	61	8	61	0.38	0.91	0.34
3	KLAX	7	31	7	31	17	61	17	61	17	121	7	61	17	61	17	91	17	31	7	31	7	61	7	91	0.53	0.90	0.48
4	KLAX	7	121	7	31	17	61	17	61	17	121	7	61	17	91	17	91	17	31	7	31	7	91	7	91	0.53	0.91	0.48
3	KLGA	19	61	19	61	9	121	9	121	9	91	9	121	9	91	9	31	9	91	9	91	9	121	19	121	0.34	0.90	0.31
4	KLGA	19	61	19	61	9	121	9	121	9	91	9	91	9	91	9	121	9	121	9	121	9	121	19	121	0.34	0.91	0.31
3	KMCI	18	121	18	31	18	61	18	61	18	121	8	91	18	91	8	91	18	31	18	61	18	121	18	121	0.37	0.92	0.34
4	KMCI	18	121	18	61	18	91	18	91	18	121	8	121	18	91	8	91	18	121	18	91	18	121	18	121	0.37	0.93	0.35
3	KMCO	7	121	7	91	7	61	17	31	17	91	7	31	7	31	7	121	17	61	7	121	7	61	17	31	0.43	0.89	0.38
4	KMCO	7	121	7	121	7	91	17	61	17	91	7	61	7	91	7	121	17	91	7	121	7	121	17	61	0.42	0.91	0.38
3	KMSP	8	31	8	31	8	121	8	121	8	31	8	121	8	61	8	61	8	61	8	31	8	61	8	31	0.35	0.91	0.32
4	KMSP	8	91	8	91	8	121	8	121	8	91	8	121	8	61	8	61	8	121	8	61	8	91	8	91	0.35	0.92	0.32
3	KORD	18	31	18	91	18	61	18	61	18	91	8	91	8	91	8	121	8	31	18	61	18	121	18	91	0.37	0.90	0.33
4	KORD	18	121	18	91	18	121	18	61	18	91	8	91	8	91	8	121	8	121	18	61	18	121	18	91	0.37	0.91	0.34
3	KPHL	18	121	18	61	18	31	18	121	8	121	8	61	8	91	8	121	8	31	18	61	18	31	28	61	0.43	0.91	0.39
4	KPHL	18	121	18	61	18	61	18	121	8	121	8	61	8	121	8	121	8	61	18	121	18	61	28	61	0.42	0.92	0.39
3	KPHX	6	1	6	1	6	31	6	31	6	91	6	61	6	61	6	61	6	31	6	31	6	1	6	1	0.34	0.73	0.25
4	KPHX	6	61	6	61	6	121	6	121	6	91	6	61	6	61	6	91	6	61	6	91	6	91	6	121	0.34	0.83	0.28
3	KPIT	17	121	17	31	17	61	27	91	7	31	7	91	7	91	7	91	7	91	7	61	17	31	17	91	0.43	0.88	0.38
4	KPIT	17	121	7	121	17	91	27	91	17	91	7	91	7	121	7	91	7	121	7	121	17	91	17	121	0.44	0.91	0.40

Table 83: (continued)

3	KSAC	6	121	6	91	6	121	16	61	16	31	6	121	6	121	6	61	6	121	6	31	6	1	6	61	0.56	0.90	0.51
4	KSAC	6	121	6	121	6	121	16	61	16	91	6	121	6	121	6	61	6	121	6	121	6	121	6	121	0.56	0.92	0.52
3	KSAN	5	1	15	31	15	61	5	121	5	121	15	61	5	121	15	31	5	61	5	121	5	31	5	1	0.43	0.87	0.37
4	KSAN	5	121	15	31	15	61	5	121	5	121	15	91	5	121	15	61	5	121	5	121	5	61	5	91	0.43	0.90	0.39
3	KSAT	7	121	7	31	17	31	17	121	17	121	17	121	17	31	7	91	7	91	7	91	17	61	17	31	0.33	0.90	0.29
4	KSAT	7	121	7	91	17	61	17	121	17	121	17	121	17	61	7	121	7	121	7	121	17	61	17	91	0.33	0.92	0.30
3	KSEA	16	31	6	91	16	61	16	31	16	31	6	121	6	61	6	61	6	61	16	31	16	121	16	91	0.41	0.89	0.36
4	KSEA	16	121	6	91	16	91	16	91	16	61	6	121	6	61	6	91	6	121	16	91	16	121	16	91	0.40	0.90	0.36
3	KSFO	10	31	10	31	20	61	20	121	30	91	30	61	20	121	20	121	20	91	10	121	10	1	10	1	0.40	0.92	0.37
4	KSFO	10	61	10	91	20	61	20	121	30	91	30	61	20	121	20	121	20	91	10	121	10	61	10	91	0.40	0.95	0.37
3	KSMX	8	121	8	91	18	31	18	61	18	61	8	121	8	61	8	31	8	121	8	31	8	1	8	1	0.63	0.89	0.56
4	KSMX	8	121	8	91	18	31	18	61	18	61	8	121	8	61	8	61	8	121	8	31	8	31	8	61	0.65	0.90	0.59
3	KSTL	7	91	7	121	17	31	17	91	7	121	7	61	7	61	7	121	7	61	17	31	7	121	17	61	0.37	0.91	0.33
4	KSTL	7	91	7	121	17	61	17	91	7	121	7	91	7	121	7	121	7	91	17	31	7	121	17	61	0.37	0.91	0.33
3	KTPA	6	91	6	121	16	31	6	121	6	121	16	31	6	1	6	121	16	1	16	31	6	61	16	31	0.41	0.85	0.35
4	KTPA	6	91	6	121	16	31	6	121	6	121	16	91	6	91	6	121	16	121	16	91	6	121	16	91	0.42	0.89	0.37

A.23.4 RTD is 3

Please see Table 84.

Table 84: Results of the processing of OLAs 3 and 4 by Aging (variant 0x0) Rev. 1.1 when the RTD is 3

OLA	station	$y(1)$	$r(1)$	$y(2)$	$r(2)$	$y(3)$	$r(3)$	$y(4)$	$r(4)$	$y(5)$	$r(5)$	$y(6)$	$r(6)$	$y(7)$	$r(7)$	$y(8)$	$r(8)$	$y(9)$	$r(9)$	$y(10)$	$r(10)$	$y(11)$	$r(11)$	$y(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
		where $y(n)$ is the y-intercept and $r(n)$ is the running average size minutes for month n																										
3	KATL	17	31	17	1	17	121	17	31	17	31	7	121	7	31	17	1	17	121	17	31	17	1	17	91	0.37	0.92	0.34
4	KATL	17	31	17	31	17	121	17	31	17	31	7	121	7	31	17	31	17	121	17	31	17	61	17	91	0.38	0.94	0.36
3	KBOS	19	61	9	31	19	31	19	31	9	121	9	91	9	121	9	121	9	121	9	91	9	61	19	91	0.32	0.94	0.30
4	KBOS	19	61	9	31	19	31	19	31	9	121	9	91	9	121	9	121	9	121	9	91	9	91	19	91	0.32	0.94	0.30
3	KBWI	17	31	17	31	17	121	17	61	7	31	7	91	7	31	17	31	7	121	17	31	17	31	17	121	0.48	0.92	0.44
4	KBWI	17	31	17	31	17	121	17	61	7	31	7	91	7	31	17	61	7	121	17	31	17	31	17	121	0.48	0.92	0.44
3	KCLE	18	61	18	91	18	31	18	121	18	31	18	1	18	1	8	121	18	31	18	31	18	31	18	1	0.38	0.91	0.34
4	KCLE	18	61	18	91	18	31	18	121	18	61	18	31	18	31	8	121	18	31	18	31	18	31	18	61	0.38	0.93	0.35

Table 84: (continued)

3	KCLT	15	31	15	91	15	121	15	121	25	31	15	31	5	31	15	61	15	61	15	31	15	1	15	31	0.40	0.91	0.36
4	KCLT	15	31	15	91	15	121	15	121	25	31	15	31	5	61	15	61	15	61	15	31	15	61	15	31	0.41	0.91	0.37
3	KCVG	17	121	17	61	17	91	27	31	17	61	17	1	7	121	7	61	17	31	17	31	17	121	17	91	0.39	0.91	0.35
4	KCVG	17	121	17	61	17	91	27	31	17	61	17	1	7	121	7	61	17	31	17	61	17	121	17	91	0.39	0.91	0.35
3	KDCA	17	1	17	1	17	31	17	31	17	1	17	31	7	91	17	31	7	121	17	1	17	1	17	31	0.37	0.91	0.34
4	KDCA	17	31	17	1	17	31	17	31	17	31	17	31	7	91	17	31	7	121	17	31	17	31	17	31	0.37	0.93	0.34
3	KDEN	18	1	18	31	18	31	18	31	18	121	18	31	18	1	18	1	18	1	18	1	18	31	18	1	0.37	0.89	0.33
4	KDEN	18	31	18	31	18	31	18	31	18	121	18	61	18	31	18	31	18	31	18	31	18	31	18	31	0.38	0.94	0.35
3	KDFW	19	121	19	1	19	121	19	121	19	91	19	31	9	121	9	31	9	1	9	91	19	1	19	1	0.33	0.93	0.31
4	KDFW	19	121	19	1	19	121	19	121	19	121	19	61	9	91	9	61	9	31	9	121	19	31	19	1	0.33	0.94	0.31
3	KDTW	18	61	18	91	18	91	18	61	18	1	8	31	8	61	8	91	18	1	18	91	18	91	18	121	0.42	0.93	0.39
4	KDTW	18	61	18	91	18	91	18	61	18	31	8	91	8	91	8	91	18	31	18	91	18	91	18	121	0.43	0.94	0.40
3	KEUG	16	61	16	31	16	91	16	1	16	31	16	1	16	91	16	31	16	1	16	1	16	91	16	1	0.47	0.91	0.43
4	KEUG	16	61	16	61	16	91	16	31	16	31	16	1	16	91	16	91	16	31	16	31	16	91	16	1	0.48	0.93	0.44
3	KIAH	17	31	17	31	27	31	17	121	17	61	17	121	17	1	7	121	7	121	17	31	17	91	17	121	0.40	0.92	0.36
4	KIAH	17	31	17	31	27	31	17	121	17	61	17	91	17	31	7	121	7	121	17	31	17	91	17	121	0.40	0.93	0.37
3	KLAS	8	1	8	1	18	31	8	121	18	31	18	121	8	121	18	31	8	1	8	121	8	1	8	1	0.37	0.89	0.33
4	KLAS	8	31	8	91	18	31	8	121	18	61	18	121	8	121	18	31	8	31	8	121	8	61	8	91	0.38	0.93	0.36
3	KLAX	17	31	17	1	17	31	17	121	27	1	17	1	17	61	17	121	17	31	17	1	7	91	17	1	0.49	0.92	0.45
4	KLAX	17	91	17	1	17	31	17	121	27	1	17	1	17	61	17	31	17	31	17	1	7	91	17	1	0.49	0.92	0.45
3	KLGA	19	61	19	31	19	1	9	121	9	121	9	91	9	121	9	91	9	121	9	121	19	1	19	121	0.33	0.92	0.30
4	KLGA	19	61	19	31	9	121	9	121	9	121	9	91	9	121	9	91	9	121	9	121	19	31	19	121	0.33	0.93	0.31
3	KMCI	18	121	18	91	28	31	28	1	18	91	18	1	18	1	18	1	18	61	18	121	28	1	18	1	0.32	0.93	0.30
4	KMCI	18	121	18	91	28	31	28	31	18	91	18	31	18	31	18	1	18	61	18	121	28	31	18	31	0.36	0.94	0.34
3	KMCO	17	31	7	61	17	1	17	31	17	91	7	121	7	91	17	1	17	91	17	1	17	1	17	31	0.41	0.91	0.37
4	KMCO	17	31	7	61	17	31	17	31	17	121	7	121	7	91	17	31	17	121	17	31	17	31	17	31	0.43	0.93	0.40
3	KMSP	8	121	8	91	8	31	8	121	8	31	8	61	8	61	8	61	8	91	8	121	8	1	18	31	0.33	0.94	0.31
4	KMSP	8	121	8	91	8	121	8	121	8	31	8	61	8	61	8	91	8	91	8	121	8	31	18	31	0.34	0.94	0.31
3	KORD	18	31	18	31	18	121	18	121	18	31	18	31	18	1	8	121	18	61	18	121	18	121	18	1	0.34	0.93	0.31
4	KORD	18	31	18	31	18	121	18	121	18	31	8	31	18	31	8	121	18	61	18	121	18	121	18	121	0.36	0.93	0.33
3	KPHL	18	121	18	61	18	31	18	91	18	1	8	1	8	121	18	1	18	1	18	31	18	31	28	61	0.42	0.90	0.38
4	KPHL	18	121	18	61	18	31	18	91	18	31	8	121	8	121	18	31	18	31	18	31	18	31	28	61	0.42	0.93	0.40
3	KPHX	6	1	6	61	6	121	6	1	16	1	6	31	6	121	6	31	6	1	6	91	6	1	6	1	0.30	0.82	0.25
4	KPHX	6	61	6	121	6	121	6	61	6	121	6	91	6	121	6	31	6	121	6	91	6	61	6	121	0.31	0.89	0.28
3	KPIT	17	121	17	31	17	31	27	31	17	31	17	1	7	91	7	91	7	91	17	121	17	91	17	61	0.42	0.92	0.39
4	KPIT	17	121	17	91	17	121	27	31	17	31	7	91	7	91	7	91	7	91	17	121	17	91	17	61	0.42	0.93	0.39
3	KSAC	6	121	6	1	16	1	16	121	16	121	16	31	16	91	16	31	16	1	16	1	6	91	6	1	0.49	0.92	0.45
4	KSAC	6	121	6	31	16	31	16	121	16	121	16	31	16	91	16	31	16	31	16	31	6	91	6	121	0.52	0.94	0.49
3	KSAN	5	1	15	1	15	31	15	31	15	1	15	1	5	91	15	91	15	1	15	1	5	31	5	91	0.46	0.91	0.42
4	KSAN	5	31	15	31	15	31	15	31	15	1	15	1	5	91	15	91	15	1	15	1	5	31	5	91	0.46	0.91	0.42
3	KSAT	17	31	7	91	17	31	17	31	17	31	17	121	17	31	17	31	7	31	17	1	17	31	17	1	0.33	0.93	0.30
4	KSAT	17	31	7	91	17	31	17	31	17	31	17	121	17	31	17	31	7	91	17	31	17	31	17	31	0.33	0.93	0.31
3	KSEA	16	121	16	1	16	1	16	61	16	31	16	1	16	1	16	1	16	1	16	91	26	1	16	91	0.36	0.92	0.33
4	KSEA	16	121	16	61	16	61	16	91	16	61	16	31	6	31	16	31	16	31	16	91	26	61	16	91	0.40	0.94	0.38
3	KSFO	10	91	10	61	20	121	20	121	30	31	30	121	20	121	20	91	20	121	20	1	10	31	10	1	0.40	0.94	0.38
4	KSFO	10	121	10	61	20	121	20	121	30	31	30	121	20	121	20	121	20	121	20	1	10	31	10	61	0.40	0.95	0.38
3	KSMX	8	121	18	1	18	31	18	61	18	91	18	1	8	61	8	61	8	61	18	1	8	31	8	31	0.60	0.91	0.55
4	KSMX	8	121	8	121	18	31	18	61	18	91	18	1	8	61	8	61	8	61	8	61	8	31	8	61	0.62	0.91	0.57
3	KSTL	17	1	17	1	17	1	17	61	17	31	17	1	17	1	7	91	7	121	17	1	17	31	17	31	0.37	0.90	0.33

Table 84: (continued)

4	KSTL	17	61	17	31	17	31	17	61	17	31	17	31	17	31	7	91	7	121	17	31	17	31	17	31	0.38	0.93	0.35
3	KTPA	6	91	16	31	16	31	16	1	16	1	16	1	6	1	6	121	16	1	16	1	16	1	16	1	0.38	0.87	0.33
4	KTPA	6	91	16	31	16	31	6	31	6	121	16	31	6	91	6	121	16	31	16	31	16	1	16	31	0.41	0.90	0.37

A.24 EXPLORATION 2 FULL RESULTS

Please see Table 85.

Table 85: Results of the processing of OLAs 3 and 4 by Static (variant 0x0, i.e., current weather only, transitions unlimited) Future revision

OLA	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm	station	deployment threshold (knots)	running average size (minutes)	NetNorm	MQMP	MQNetNorm
3	KATL	9	40	0.44	0.90	0.40	KMCI	12	41	0.43	0.93	0.40
4	KATL	9	101	0.44	0.93	0.41	KMCI	12	62	0.43	0.94	0.41
3	KBOS	10	30	0.35	0.92	0.32	KMCO	9	23	0.51	0.88	0.45
4	KBOS	9	113	0.34	0.94	0.32	KMCO	9	76	0.50	0.92	0.46
3	KBWI	9	32	0.55	0.89	0.49	KMSP	8	67	0.35	0.93	0.33
4	KBWI	9	81	0.54	0.92	0.50	KMSP	8	76	0.35	0.93	0.33
3	KCLE	10	43	0.40	0.90	0.37	KORD	11	35	0.42	0.90	0.38
4	KCLE	10	71	0.40	0.91	0.37	KORD	11	56	0.42	0.92	0.39
3	KCLT	8	37	0.53	0.88	0.47	KPHL	10	48	0.47	0.91	0.43
4	KCLT	8	50	0.52	0.90	0.47	KPHL	10	104	0.47	0.93	0.44
3	KCVG	10	51	0.44	0.89	0.39	KPHX	6	46	0.34	0.84	0.29
4	KCVG	10	81	0.44	0.91	0.40	KPHX	7	66	0.33	0.86	0.28
3	KDCA	9	45	0.43	0.92	0.39	KPIT	10	57	0.52	0.91	0.47
4	KDCA	9	89	0.42	0.93	0.39	KPIT	10	72	0.52	0.92	0.47
3	KDEN	10	36	0.41	0.89	0.36	KSAC	8	28	0.63	0.90	0.56

Table 85: (continued)

4	KDEN	11	109	0.38	0.94	0.36	KSAC	8	80	0.62	0.93	0.57
3	KDFW	11	28	0.36	0.92	0.34	KSAN	8	16	0.55	0.86	0.48
4	KDFW	12	68	0.36	0.95	0.34	KSAN	8	43	0.55	0.90	0.49
3	KDTW	10	44	0.48	0.92	0.44	KSAT	10	37	0.39	0.91	0.35
4	KDTW	10	44	0.48	0.92	0.44	KSAT	10	46	0.39	0.92	0.36
3	KEUG	8	38	0.57	0.90	0.51	KSEA	9	27	0.50	0.88	0.44
4	KEUG	8	71	0.57	0.92	0.52	KSEA	9	67	0.49	0.92	0.45
3	KIAH	10	31	0.49	0.90	0.44	KSFO	14	34	0.43	0.96	0.41
4	KIAH	10	52	0.49	0.92	0.45	KSFO	15	26	0.43	0.97	0.41
3	KLAS	9	34	0.40	0.91	0.36	KSMX	9	38	0.66	0.90	0.59
4	KLAS	9	63	0.40	0.93	0.37	KSMX	9	50	0.66	0.91	0.60
3	KLAX	10	22	0.56	0.90	0.50	KSTL	9	42	0.40	0.91	0.37
4	KLAX	10	72	0.54	0.92	0.50	KSTL	9	105	0.40	0.93	0.37
3	KLGA	10	53	0.37	0.92	0.34	KTPA	8	26	0.53	0.86	0.45
4	KLGA	10	86	0.36	0.92	0.34	KTPA	8	39	0.52	0.88	0.46

A.25 EXPLORATIONS 5 AND 6

A.25.1 Exploration 5: KATL

Please see Table 86.

Table 86: For KATL, results of the λ sensitivity analysis and extended dense design space search for processing of OLAs 3 and 4 by Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) Future version

OLA	station	λ	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
			where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
5	KATL	0.1	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	329	0.9	3	0.8	121	0.1	1	0.26	0.96	0.25
6	KATL	0.1	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	331	0.9	3	0.8	117	0.1	1	0.27	0.96	0.26
5	KATL	0.2	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	329	0.9	3	0.8	121	0.1	1	0.26	0.96	0.25
6	KATL	0.2	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	331	0.9	3	0.8	117	0.1	1	0.27	0.96	0.26
5	KATL	0.3	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	329	0.9	3	0.8	121	0.1	1	0.26	0.96	0.25
6	KATL	0.3	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	331	0.9	3	0.8	117	0.1	1	0.27	0.96	0.26
5	KATL	0.4	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	329	0.9	3	0.8	121	0.1	1	0.26	0.96	0.25
6	KATL	0.4	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	331	0.9	3	0.8	117	0.1	1	0.27	0.96	0.26
5	KATL	0.5	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	329	0.9	3	0.8	121	0.1	1	0.26	0.96	0.25
6	KATL	0.5	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	331	0.9	3	0.8	117	0.1	1	0.27	0.96	0.26
5	KATL	0.6	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	329	0.9	3	0.8	121	0.1	1	0.26	0.96	0.25
6	KATL	0.6	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	331	0.9	3	0.8	117	0.1	1	0.27	0.96	0.26
5	KATL	0.7	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	329	0.9	3	0.8	121	0.1	1	0.26	0.96	0.25
6	KATL	0.7	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	331	0.9	3	0.8	117	0.1	1	0.27	0.96	0.26
5	KATL	0.8	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	329	0.9	3	0.8	121	0.1	1	0.26	0.96	0.25
6	KATL	0.8	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	331	0.9	3	0.8	117	0.1	1	0.27	0.96	0.26
5	KATL	0.9	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	329	0.9	3	0.8	121	0.1	1	0.26	0.96	0.25
6	KATL	0.9	0.9	174	0.9	20	0.5	153	0.9	186	0.9	36	0.8	92	0.8	43	0.7	131	0.2	331	0.9	3	0.8	117	0.1	1	0.27	0.96	0.26

A.25.2 Exploration 6: KBOS

Please see Table 87 on the next page.

Table 87: For KBOS, results of the λ sensitivity analysis and extended dense design space search for processing of OLAs 3 and 4 by Fuzzy-Crisp (variant 0x0, i.e., current weather only, transitions unlimited) Future version

OLA	station	λ	$d(1)$	$r(1)$	$d(2)$	$r(2)$	$d(3)$	$r(3)$	$d(4)$	$r(4)$	$d(5)$	$r(5)$	$d(6)$	$r(6)$	$d(7)$	$r(7)$	$d(8)$	$r(8)$	$d(9)$	$r(9)$	$d(10)$	$r(10)$	$d(11)$	$r(11)$	$d(12)$	$r(12)$	NetNorm	MQMP	MQNetNorm
			where $d(n)$ is the deployment threshold of the combined degree of membership and $r(n)$ is the running average size minutes for month n																										
5	KBOS	0.1	0.9	93	0.3	280	0.8	157	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.4	86	0.8	202	0.27	0.91	0.25
6	KBOS	0.1	0.9	29	0.4	197	0.8	243	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.5	15	0.8	202	0.27	0.91	0.24
5	KBOS	0.2	0.9	93	0.3	280	0.8	157	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.4	86	0.8	202	0.27	0.91	0.25
6	KBOS	0.2	0.9	29	0.4	197	0.8	243	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.5	15	0.8	202	0.27	0.91	0.24
5	KBOS	0.3	0.9	93	0.3	280	0.8	157	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.4	86	0.8	202	0.27	0.91	0.25
6	KBOS	0.3	0.9	29	0.4	197	0.8	243	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.5	15	0.8	202	0.27	0.91	0.24
5	KBOS	0.4	0.9	93	0.3	280	0.8	157	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.4	86	0.8	202	0.27	0.91	0.25
6	KBOS	0.4	0.9	29	0.4	197	0.8	243	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.5	15	0.8	202	0.27	0.91	0.24
5	KBOS	0.5	0.9	93	0.3	280	0.8	157	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.4	86	0.8	202	0.27	0.91	0.25
6	KBOS	0.5	0.9	29	0.4	197	0.8	243	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.5	15	0.8	202	0.27	0.91	0.24
5	KBOS	0.6	0.9	93	0.3	280	0.8	157	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.4	86	0.8	202	0.27	0.91	0.25
6	KBOS	0.6	0.9	29	0.4	197	0.8	243	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.5	15	0.8	202	0.27	0.91	0.24
5	KBOS	0.7	0.9	93	0.3	280	0.8	157	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.4	86	0.8	202	0.27	0.91	0.25
6	KBOS	0.7	0.9	29	0.4	197	0.8	243	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.5	15	0.8	202	0.27	0.91	0.24
5	KBOS	0.8	0.9	93	0.3	280	0.8	157	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.4	86	0.8	202	0.27	0.91	0.25
6	KBOS	0.8	0.9	29	0.4	197	0.8	243	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.5	15	0.8	202	0.27	0.91	0.24
5	KBOS	0.9	0.9	93	0.3	280	0.8	157	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.4	86	0.8	202	0.27	0.91	0.25
6	KBOS	0.9	0.9	29	0.4	197	0.8	243	0.9	283	0.9	359	0.4	167	0.5	147	0.6	58	0.7	5	0.7	352	0.5	15	0.8	202	0.27	0.91	0.24

A.26 FULL RESULTS FOR EXPLORATION 8

Please see Table 88 on the following page and Table 89 on the next page.

Table 88: Effect of time horizon on Static (variant 0x3) processing KATL's data within OLA 5

OLA	station	time horizon	deployment threshold (knots)	window size (minutes)	NetNorm	MQMP	MQNetNorm
5	KATL	30	14	14	0.273	0.932	0.254
5	KATL	60	13	105	0.281	0.928	0.261
5	KATL	120	14	84	0.274	0.940	0.258
5	KATL	240	14	116	0.271	0.943	0.255
5	KATL	480	14	117	0.273	0.947	0.258
5	KATL	720	13	119	0.267	0.941	0.251
5	KATL	1440	14	121	0.277	0.962	0.267

Table 89: Effect of time horizon on Static (variant 0x3) processing KATL's data within OLA 6

OLA	station	time horizon	deployment threshold (knots)	window size (minutes)	NetNorm	MQMP	MQNetNorm
6	KATL	30	14	99	0.276	0.935	0.258
6	KATL	60	14	106	0.272	0.935	0.254
6	KATL	120	13	105	0.286	0.940	0.269
6	KATL	240	14	116	0.271	0.943	0.255
6	KATL	480	14	117	0.273	0.947	0.258
6	KATL	720	13	119	0.267	0.941	0.251
6	KATL	1440	14	121	0.277	0.962	0.267