

Noname manuscript No.
(will be inserted by the editor)

A Decomposition Approach for Commodity Pickup and Delivery with Time-Windows Under Uncertainty

Lavanya Marla · Cynthia Barnhart · Varun Biyani

Received: date / Accepted: date

Abstract We consider a special class of large-scale, network-based, resource allocation problems under uncertainty, namely that of multi-commodity flows with time-windows under uncertainty. In this class, we focus on problems involving commodity pickup and delivery with time windows. Our work examines methods of proactive planning, that is, robust plan generation to protect against future uncertainty. By *a priori* modeling uncertainties in data corresponding to service times, resource availability, supplies and demands, we generate solutions that are more *robust* operationally, that is, more likely to be executed or easier to repair when disrupted. We propose a novel modeling and solution framework involving a decomposition scheme that separates problems into a routing master problem and scheduling sub-problems; and iterates to find the optimal solution. Uncertainty is captured in part by the master problem and in part by the scheduling sub-problem. We present proof-of-concept for our approach using real data involving routing and scheduling for a large shipment carrier's ground network, and demonstrate the improved robustness of solutions from our approach.

Lavanya Marla
iLab, Heinz College, Carnegie Mellon University
Pittsburgh, PA.
E-mail: lavanyamarla@cmu.edu

Cynthia Barnhart
Department of Civil and Environmental Engineering, Massachusetts Institute of Technology
Cambridge, MA.
E-mail: cbarnhart@mit.edu

Varun Biyani
Heinz College, Carnegie Mellon University
Pittsburgh, PA.
E-mail: vbiyani@andrew.cmu.edu

Keywords robust routing and scheduling · multi-commodity routing and scheduling · uncertainty · decomposition

1 Introduction

In this paper, we consider the class of large-scale network-based problems including multi-commodity flow problems with time-windows, which are at the core of problems arising in transportation, communications and logistics. Such resource allocation problems with their large-scale nature and associated complexity, have been ideal candidates for the application of optimization techniques (Barnhart et al. 1994; Desrosiers et al. 1995; Barnhart et al. 1998a; Dumas et al. 1991). However, conventional optimization techniques usually involve assumptions of deterministic inputs, leading to solutions that are easily disrupted when realized parameter values are different; and thus exhibit a lack of *robustness* and high costs of recovery or repair. Such solutions are rarely (if ever) executed, and certainly, never truly optimal. In this work, our objective is to build robust network resource allocation solutions that: (i) are less fragile to disruption, (ii) easier to repair if needed, and (iii) minimize the realized, not just planned, problem costs.

Uncertainty in multi-commodity flows with time-windows can occur in the form of stochasticity in the supplies and demands of commodities; available capacities of the network links; and travel and service times on the network. The multi-commodity flows with time-windows is at the core of network design problems, and hence, in finding robust solutions to the multi-commodity flows with time-windows we expect to pro-

vide insights into the more complex problem of network design under uncertainty.

1.1 Problem Description

To illustrate and evaluate our approach, we consider a specific problem, namely the *Commodity Routing Problem with Time Windows Under Uncertainty (CRTW-UU)*. Under CRTW-UU, for each vehicle v (such as a plane or truck) in the set of vehicles V , we are given a set of vehicle routes defining a network of locations with time-independent travel times and capacities u_{ij} corresponding to vehicle capacities on the links, and service times at locations. Each commodity k (such as a trailer, package, crew member or passenger) in the set of commodities K with demand d_k needs to be routed over this network, from its origin $O(k)$ to its destination $D(k)$. Transshipment routing is allowed. All d_k units of commodity k are assumed to have the same route and schedule. (In cases where different units of a commodity can have different routes and schedules, each unit can be treated as a commodity by creating d_k separate commodities. Thus this assumption has no loss of generality.) Commodity k must be picked up after its earliest available time at its origin ($EAT_{O(k)}^k$) and delivered before its latest delivery time at its destination ($LDT_{D(k)}^k$). The objective is to find commodity routes, and vehicle and commodity schedules, which minimize costs due to vehicle operations, and non-service of commodities. We consider early drop-offs to have no bonuses, and we disallow late drop-offs (that is, if a commodity is late, it will not be delivered). We are therefore interested in determining commodity routes and commodity and vehicle schedules, given the *sequence of stops* each vehicle makes. In this work, we are particularly interested in addressing the stochastic nature of input data as seen in vehicle capacities, demands of commodities, and service times. In the remainder of the paper, we use the words ‘commodity’, ‘shipment’ and ‘trailer’ interchangeably.

CRTW-UU is at the core of the classic network design problem of vehicle routing with pickup and delivery of shipments under time-windows *and under uncertainty* (Cordeau et al. 2006). The problem of vehicle routing with pickup and delivery of shipments under time-windows under uncertainty reduces to the CRTW-UU if we assume the routes of vehicles to be known, with the schedule still unspecified. Instances (and variants) of the CRTW-UU arise in package delivery, container scheduling, airline scheduling, etc. These problems have been shown to be NP-hard Cordeau et al. (2006), and are more so in the case of uncertainty.

Approaches to capture uncertainty and build robust solutions have been in three categories: (i) tailored approaches, (ii) general, distribution-free approaches, and (iii) general-distribution-based approaches. Tailored approaches like Shebalov and Klabjan (2004), Lan et al. (2006), Paraskevopoulos et al. (1991) and Kang and Clarke (2002) identify specific features of the problem that can make the solution flexible, and maximize such attributes. General, distribution free approaches do not capture distribution information, but use information about uncertainty sets, as described in Soyster (1973), Ben-Tal and Nemirovski (1999), Ben-Tal and Nemirovski (2000), Bertsimas and Sim (2004) and Bertsimas and Sim (2003). General, distribution-based approaches such as Birge and Louveaux (1997), Charnes and Cooper (1959), Charnes and Cooper (1963), Rockafellar and Uryasev (2000) and Mulvey et al. (1981) model the underlying distributions analytically or through scenarios, to generate robust solutions for those distributions. In this work, our goal is to develop analytical and computational frameworks that help us take advantage of partial knowledge of data distributions, for example, in the form of quantiles.

Other studies have modeled uncertainty in the vehicle routing and multi-commodity flow contexts. Bertsimas and Simchi-Levi (1996) survey the vehicle routing problem (VRP) literature and studies the deterministic, dynamic and stochastic variants for the VRP. In the stochastic and dynamic cases, they study uncertainty in demands, location or arrival time of requests. For these, in particular, for the dynamic version, good solutions can be found by adapting the static methods appropriately. For stochastic VRPs, they consider the VRP under congestion, and based on structural insights from these problems, construct algorithms for the stochastic and dynamic cases. However, commodity pickup and delivery under uncertainty is not considered. Dror et al. (1989) study the vehicle routing with stochastic demands. They propose two new solution frameworks - one is a stochastic programming with recourse model that can be applied for structures with relatively general recourse actions, and the second is a Markov decision process based model. The authors do not consider uncertainty in other elements of the problem, and do not provide computational proof-of-concept. Mahr (2011) studies the problem of truckload pickup-delivery-and-return problem with time-windows, with release time-uncertainty, truck breakdown uncertainty and service time uncertainty. The author proposes a substitution algorithm that improves the performance of the agent-based approach in cases with and without uncertainty. He also shows that distributed heuristics are comparable to centralized optimization

methods in the case of dynamic pickup and delivery problems. Yang et al. (2004) consider a real-time (dynamic) multi-vehicle pickup and delivery problem, where requests arrive in real time, and their pickup and delivery windows are known at arrival. They propose formulations for the offline and online contexts of the problem, and describe that the best policy is one that takes some future demand distribution into consideration. This points to the necessity of robust models, although the paper does not explicitly plan for robustness.

Dessouky et al. (1999) study the impact of managing uncertainty by increasing the amount of information available, in the context of bus dispatching. They use technologies that enable greater control of systems by tracking information in real-time and using the information to control schedules, thus improving service levels. However, they consider in this paper an information-sparse scenario without implementation of intelligent transportation systems. Sungur et al. (2010) consider the courier delivery problem with probabilistic customer arrivals and uncertain travel times, and use an approach that combines stochastic programming with recourse to model customer arrival uncertainty and robust optimization to capture uncertainty in travel times. This scenario-based approach maximizes customer coverage and route similarity over scenarios, and minimizes earliness and lateness penalties and total travel times. This is a network design problem unlike the CRTW-UU. For large-scale problem instances, therefore, the authors use insertion-based heuristics to balance the multiple objectives presented. Wollmer (1980) considers multi-commodity flow networks where link capacities are uncertain and commodities are to be transported from origin to destination. The objective is to find an investment strategy that adds link capacities while minimizing associated expected investment costs for increasing link capacities. A two-stage stochastic program is formulated, wherein the objective of the second stage is to minimize transportation costs once link capacities are realized. This is again a network design problem, solved using stochastic optimization, which requires extensive scenario generation and knowledge of distributions of uncertain parameters to solve the problem. Ordonez and Zhao (2011) solve a similar problem by applying a robust optimization framework to the problem of expanding network capacity when demand and travel times are uncertain. This work is a network design problem, but is closest to our work in the goal of capturing both demand and travel time uncertainty in the presence of partial information about uncertainty.

1.2 Motivation for a new approach

While there has been extensive work on capturing specific types of uncertainty (such as demand uncertainty or travel time uncertainty) separately, there is relatively less work (for example, Sungur et al. (2010) and Ordonez and Zhao (2011)) on capturing both types of uncertainty and generating robust solutions. Moreover, most models require knowledge of uncertainty distributions, whereas in practice, data generated from the field has only partial knowledge of the underlying distribution. Our goal is to develop a framework for the CRTW-UU that helps capture multiple kinds of uncertainty simultaneously, while having the ability to make use of data distributions, if known, or partial information, if available (for example, in the form of quantiles).

1.3 Contributions

In addressing the CRTW-UU problem, our contributions are as follows. First, to capture demand and capacity uncertainty, we extend the Chance-Constrained model of Charnes and Cooper (1959) Charnes and Cooper (1963) and present our new Extended Chance-Constrained Programming (ECCP) model. Second, we develop a decomposition scheme that provides a new modeling and algorithmic approach, which captures travel time and demand uncertainty (building on the ECCP), and provides robust solutions that are less vulnerable to uncertainty. Our approach: (i) provides a new way of modeling the problem by separating the routing and scheduling elements of the problem, capturing different types of uncertainty in each, and maintaining accuracy by iterating among the modules; and (ii) is flexible with respect to data requirements in that it can be applied with partial or complete knowledge of data distributions.

1.4 Outline of the paper

In §2 we present our new decomposition modeling approach. We present the different elements of the model and discuss how uncertainty is captured using this model. In §3 we present an algorithm to solve the decomposition model. We then discuss the advantages and limitations of the approach. In §4 we apply our approach and discuss results for the problems of truckload routing and scheduling for a U.S. carrier. We summarize and conclude in §6.

2 Decomposition Modeling Approach

2.1 Decomposition Overview

Our decomposition approach for CRTW-UU involves the repeated solution of a master problem and sub-problems. At each iteration of the procedure, the master problem is solved to generate a proposed solution to the CRTW-UU problem. The master chooses one path for each commodity from multiple paths available, while satisfying capacity constraints. The set of paths for each commodity include an ‘artificial’ path (a high cost path with zero travel time), which means a shipment cannot be delivered within the specified time windows and has to use a higher cost alternative. Each solution to the master problem ensures satisfaction of all constraints in the problem except scheduling constraints; and minimum cost with respect to the satisfied constraints. To test if schedule infeasibilities exist in the solution, we solve sub-problems in which infeasibilities are detected using efficient network node-labeling algorithms. If no infeasibilities are found, a feasible schedule exists for the CRTW-UU solution and the CRTW-UU problem is solved. If, however, scheduling conflicts are identified, these scheduling conflicts are translated into inequalities that are added to the master problem to eliminate the current infeasible solution. After a finite (but possibly large) number of iterations, our approach is guaranteed to find a feasible, and hence optimal, solution to the CRTW-UU problem, as will be discussed in §3.6 and §3.7. A diagrammatic overview of our decomposition approach is presented in Figure 1.

2.2 Flow Master Problem (CRTW-UU-MP)

When scheduling constraints are relaxed, the CRTW-UU reduces to flowing shipments on the vehicles. In solving it deterministically without modeling uncertainty, the flow master problem involves solving the standard path-based multi-commodity flow formulation detailed in Ahuja et al. (1993), that is, choosing paths for each commodity k on its subnetwork G_k .

To model uncertainty in demands and capacities, we extend the Chance-Constrained Programming (CCP) model (Charnes and Cooper 1959, 1963) of Charnes and Cooper and present our *Extended Chance-Constrained Programming* model (ECCP). CCP is based on constraint satisfaction, that is, constraints containing uncertain capacity parameters are required to be satisfied for a pre-specified probability of protection γ . In our ECCP approach, however, the achieved level of protection is modeled as a variable. We use partial information about the distributions of uncertain capacity pa-

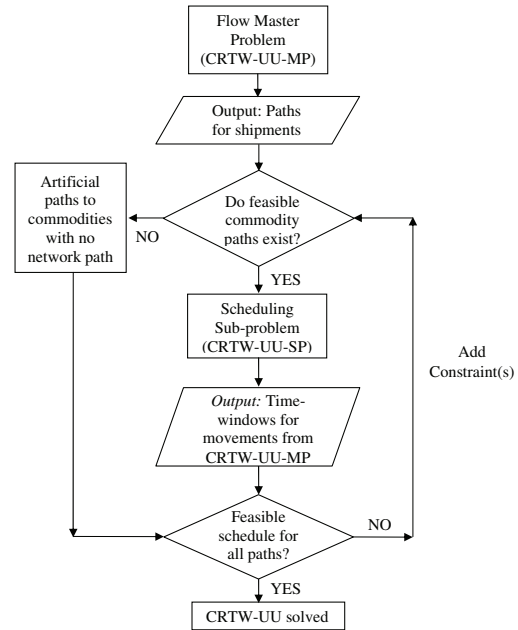


Fig. 1 Schematic diagram of the Decomposition Approach

rameters, in the form of quantiles; or full information in the form of distribution parameters. With each quantile is associated a level of protection, as defined by the Chance-Constrained Programming approach and defined more formally later in this section. Among these different levels of protection, our ECCP approach allows the model to maximize the level of protection under a robustness budget Δ . Compared to the Chance-Constrained Programming approach, the ECCP allows costs to be contained when achieving robustness and thus control the level of conservatism, and also avoids the necessity for the user to specify protection levels *a priori*, which can be difficult when multiple uncertain parameters in several constraints are involved Marla (2007). For further details of the approach, we refer the reader to Marla (2010). In the CRTW-UU, we capture uncertainty explicitly in capacities, that is, we protect against capacity drops. Through the increased slack in the capacity constraint, this acts as a proxy for protecting against demand uncertainty.

We now describe the network underlying the deterministic and ECCP models. For each shipment k we build a network $G_k = (N_k, A_k)$ that is a copy of network $G = (N, A)$ described here. In $G = (N, A)$, each node $j \in N$ has three attributes: a location $l(j)$, vehicle $v(j) : v(j) \in V$, and information if it represents arrival or departure of v at $l(j)$. Connecting these nodes are arcs $a \in A$ of three types: travel arcs, connection arcs and transfer arcs. Travel arcs (i, j) on the network represent movement of vehicle $v(i)(= v(j))$ departing from

$l(i)$ and arriving at $l(j)$. Flows on these arcs represent the movement of shipments on $v(i)$ from $l(i)$ to $l(j)$. Connection arcs connect the arrival node of $v(i)$ at location $l(i)$ and the departure node of $v(i) = v(j)$ at $l(j) = l(i)$. Flows on these arcs represent shipments remaining on $v(i)$ while it is positioned at $l(i)$. Travel and connection arcs belong to A_v , the set of arcs of vehicle v . Flows on *transfer arcs* (i, j) , which connect the arrival node of $v(i)$ at $l(i)$ to the departure node of $v(j) (\neq v(i))$ at $l(j) = l(i)$, represent the transfer of shipments between vehicles, and have an associated transfer time. $G = (N, A)$ is used to aggregate information (detailed in §3) from the networks $G_k = (N_k, A_k) \forall k \in K$. Each arc $(i, j) \in G$ has a capacity of u_{ij} , that is determined based on the type of arc (travel, transfer or connection arc). P^k is the set of origin to destination paths for commodity k in G_k . δ_{ij}^p is an arc-path indicator variable that is equal to 1 if arc (i, j) is on path p , 0 otherwise.

In addition, we have an *artificial path* for each shipment $k \in K$, which is not physically present in the shipment network G_k , but is used to model the case when the shipment does not have any feasible path. The use of an artificial path in the solution denotes the schedule infeasibility of that shipment within the specified time windows. c_{ij} is the cost on arc (i, j) . c_p is the cost due to 1 unit of flow on path p , equivalent to the sum of costs of the arcs on the path. $c_p = \sum_{(i,j) \in p} c_{ij}$.

Because we focus on finding feasible schedules, $c_{ij} = 0$, and hence $c_p = 0$ for all arcs and paths, except for the artificial path (denoting schedule infeasibility). For each shipment k , the cost associated with the artificial path is a penalty cost (for no service or late service, or for subcontracting out the service to another carrier).

We summarize the notation for the model as follows:

- K = set of shipments k
- G_k = network for shipment k , constructed as described above.
- G = network that aggregates information from networks $G_k, \forall k \in K$
- P^k = set of origin to destination paths for commodity k in G_k , including the ‘artificial’ path for k
- d_k = number of units of commodity k to be transported from origin to destination
- u_{ij} = capacity of $arc(i, j) \in G$
- δ_{ij}^p = arc-path indicator that is equal to 1 if arc (i, j) is on path p , 0 otherwise
- $f_p = 1$ if all d_k units of commodity k flow on any path $p \in P_k$; and 0 otherwise.
- c_p = cost due to 1 unit of flow on path p (if p is the artificial path this corresponds to penalties for no service, late service or subcontracting)

The deterministic formulation to find a path for each shipment on its subnetwork G_k , without capturing any uncertainty, is the same as the path-based multi-commodity flow formulation, and is as follows:

$$\min \sum_{k \in K} \sum_{p \in P^k} d_k c_p f_p \quad (1)$$

$$\text{s.t.} \sum_{p \in P^k} f_p = 1 \quad \forall k \in K \quad (2)$$

$$\sum_{k \in K} \sum_{p \in P^k} d_k f_p \delta_{ij}^p \leq u_{ij} \quad \forall (i, j) \in A \quad (3)$$

$$f_p \in \{0, 1\} \quad \forall p \in P^k, \forall k \in K \quad (4)$$

The objective (1) minimizes costs of commodity flows on the network. Constraints (2) correspond to finding exactly one feasible path on the network for each commodity, constraints (3) to ensure that flows satisfy arc capacity constraints, and constraints (4) correspond to integrality of commodity flows. We model the f_p variables as binary because it is more advantageous when adding constraints from the Scheduling Sub-Problem into the Flow Master problem; and furthermore, make our approach easier to explain. This is without any loss of generality, as the d_k units of each commodity k can also be split into multiple commodities of one unit each.

In order to capture uncertainty in demand or supplies (or, uncertainty in capacities as a proxy for demand uncertainty), we apply our ECCP model to (1) - (4). We first define the following additional notation.

- f_p^* = optimal solution to the deterministic problem ((1) - (4)) that minimizes costs when data assume nominal values,
- u_{ij} = capacity of arc (i, j) , indicating vehicle capacities or transshipment capacities, depending on the type of arc,
- Q_{ij} = set of quantiles $q = 1, \dots, |Q_{ij}|$ of uncertain capacity parameters u_{ij} , for each constraint corresponding to arc (i, j) ,
- u_{ij}^q = capacity associated with quantile $q \in Q_{ij}$,
- p_{ij}^q = protection level probability associated with quantile $q \in Q_{ij}$ for the capacity constraint corresponding to arc (i, j) , $0 \leq p_{ij}^q \leq 1$; such that $P(u_{ij} \leq u_{ij}^q) = p_{ij}^q$,
- y_{ij}^q is the binary variable that is equal to 1 if the protection level expressed as a probability p_{ij}^q , represented by the q th quantile, is attained in the capacity constraint for arc (i, j) ; and 0 otherwise,
- δ = pre-specified budget of cost from the nominal value $\sum_{k \in K} \sum_{p \in P^k} d_k c_p f_p^*$, and
- γ_{ij} = achieved protection level for the capacity of arc (i, j) .

The ECCP formulation corresponding to (1)-(4) is:

CRTW-UU-MP:

$$\max \sum_{(i,j) \in A} w_{ij} \gamma_{ij} \quad (5)$$

$$\text{s.t.} \sum_{k \in K} \sum_{p \in P^k} d_k c_p f_p \leq \sum_{k \in K} \sum_{p \in P^k} d_k c_p f_p^* + \Delta \quad (6)$$

$$\sum_{p \in P^k} f_p = 1 \quad \forall k \in K \quad (7)$$

$$\sum_{k \in K} \sum_{p \in P^k} d_k f_p \delta_{ij}^p \leq \sum_{q=1}^{|Q_{ij}|} u_{ij}^q (y_{ij}^q - y_{ij}^{q-1}) \quad \forall (i,j) \in A \quad (8)$$

$$y_{ij}^q \geq y_{ij}^{q-1} \quad \forall q \in Q_{ij}, \forall (i,j) \in A \quad (9)$$

$$y_{ij}^0 = 0 \quad \forall (i,j) \in A \quad (10)$$

$$y_{ij}^{|Q_{ij}|} = 1 \quad \forall (i,j) \in A \quad (11)$$

$$\gamma_{ij} \leq \sum_{q=1}^{|Q_{ij}|} p_{ij}^q (y_{ij}^q - y_{ij}^{q-1}) \quad \forall (i,j) \in A \quad (12)$$

$$f_p \in \{0, 1\} \quad \forall p \in P^k, \forall k \in K \quad (13)$$

$$y_{ij}^q \in \{0, 1\} \quad \forall q \in Q_{ij}, \forall (i,j) \in A \quad (14)$$

$$0 \leq \gamma_{ij} \leq 1 \quad \forall (i,j) \in A \quad (15)$$

The goal of our model is to choose the solution with the highest level of protection, within a pre-specified budget δ .

(5) is the ECCP objective function that maximizes a weighted sum of protection levels (with weights w_{ij} and achieved protection γ_{ij}) over constraints with uncertain parameters. Constraints (6) limit the expected cost of the robust solution to no more than a user-specified budget of Δ more than the expected optimal cost when using nominal parameter values. Constraints (7) assign one path to each shipment. Constraints (8) find the highest protection level attainable for the uncertain parameters. Inequalities (12) set γ_{ij} to be no greater than the highest protection level provided to the capacity constraint corresponding to (i,j) . (9) ensure that the protection level variables follow a step function, that is, if a higher level of protection is achieved, all lower levels of protection are also achieved. (10) and (11) set the boundary values of the step functions. Constraints (13), (14) and (15) describe the variable ranges.

2.3 Scheduling Sub-Problem (CRTW-UU-SP)

Given a shipment flow solution F from the CRTW-UU-MP, the objective of the Scheduling Sub-Problem (CRTW-UU-SP) is to determine if the shipment flows

obtained from solving the Flow Master Problem have a feasible *and robust* schedule that obeys shipment pickup and delivery time-windows, allows connections and transfers, and has sufficient slack in its path. If a shipment is assigned to any path other than the artificial path in the solution to (5) - (15), it is possible that infeasibility might exist in its schedule. We determine the existence of a feasible robust schedule for the Flow Master Problem solution by solving a series of shortest path and network-labeling algorithms, detailed in §3.1 and §3.3. If we find a feasible schedule, we are done, otherwise, the same algorithm identifies the sources of infeasibility.

To find a robust schedule, we assign as a proxy a ‘protection level’ for each shipment path in the Flow Master Problem solution. The intuition behind this is that each shipment path is protected up to a certain probability level, the entire schedule is better protected because there is more flexibility in schedule movements. The higher protection basically adds slack in travel times by assuming higher quantiles of travel times (according to the protection level) than the average, thus adding buffers and allowing for movements under uncertain (and higher) travel time realizations. Note also that slack in the travel time also corresponds to wider time-windows of movements on the shipment path. The quantile value for the *arc* travel time to be used in the Scheduling Sub Problem can be determined based on the desired protection level for the *path*, using our knowledge of uncertainty based on historical data. Note that using this model, we protect only a subset of all arcs in the network, namely, those that are present on paths in the Flow Master Problem solution. This also helps avoid over-conservatism or ‘guessing’ in choosing which arcs to selectively protect, which would be the case in a non-decomposition based approach where path flows were not known before assigning protection levels to paths or arcs.

The protection level assigned to each arc in a shipment path needs to be chosen a priori by the user, and can be an iterative process. Our decomposition approach also includes additional flexibility in changing the level of protection over iterations. The actual quantile of travel time used on an arc can be changed during the iterations of the decomposition approach. During subsequent applications of the sub-problem to the master problem solution, higher quantile values may be used if a higher level of protection for the service times is desired.

2.4 Iterative Feedback Mechanism

We iterate between solving the CRTW-UU-MP and the CRTW-UU-SP, identifying infeasibilities in the CRTW-

UU-SP solutions and adding them as new constraints into the CRTW-UU-MP in order to eliminate current infeasibilities. With each iteration, the constraints added to the CRTW-UU-MP increase its size minimally, but decrease the size of its feasible solution space. We will show that we converge to the optimal solution with each iteration of the algorithm. The CRTW-UU is solved when the shipment flows obtained from the CRTW-UU-MP have associated feasible schedules and no cuts are added, and the iterative procedure terminates.

3 Solution algorithm for the decomposition modeling approach

In this section, we more formally describe the algorithm, depicted in Figure 2. As described in §2.2 and §2.3, uncertainty is captured using the ECCP in the Flow Master Problem and using quantiles for travel time using the CCP in the Scheduling Sub-Problem.

We improve the solvability of the Flow Master Problem and Scheduling Sub-Problem by invoking a pre-processing step that identifies time-infeasible path assignments *a priori*. The Pre-processing step is executed before the commencement of iterations solving the Flow Master Problem and Scheduling Sub-Problem. We use the same notation introduced in §2, and detail each module of the Decomposition algorithm in the following sections.

3.1 Step 1: Network Pre-processing

On G_k , for all $k \in K$, we define EAT_j^k as the earliest time shipment k can reach node j after starting from $O(k)$ and LDT_j^k as the latest time shipment k can leave node j to get to its destination $D(k)$ on time. $sp_{i,j}^k$ is the shortest path distance for shipment k from i to node j . EAT_j^v and LDT_j^v are the earliest arrival time and latest departure time of vehicle v at each node $j \in N$.

The steps of the network Pre-processing phase are detailed as follows:

- (i) For each shipment network G_k for all $k \in K$, based on $EAT_{O(k)}^k$ and $LDT_{D(k)}^k$, find time-windows, expressed as the earliest arrival time and latest departure time $[EAT_i^k, LDT_i^k]$, of any shipment $k \in K$ at each node $i \in N_k$. Thus,

$$EAT_i^k = EAT_{O(k)}^k + sp_{O(k),i}^k; \text{ and} \quad (16)$$

$$LDT_i^k = LDT_{D(k)}^k - sp_{j,D(k)}^k \quad (17)$$

- (ii) Find time-windows (EAT_i^v, LDT_i^v) for each vehicle $v \in V$ at each node $i \in N$ along its route based on the earliest start time and latest allowed return time

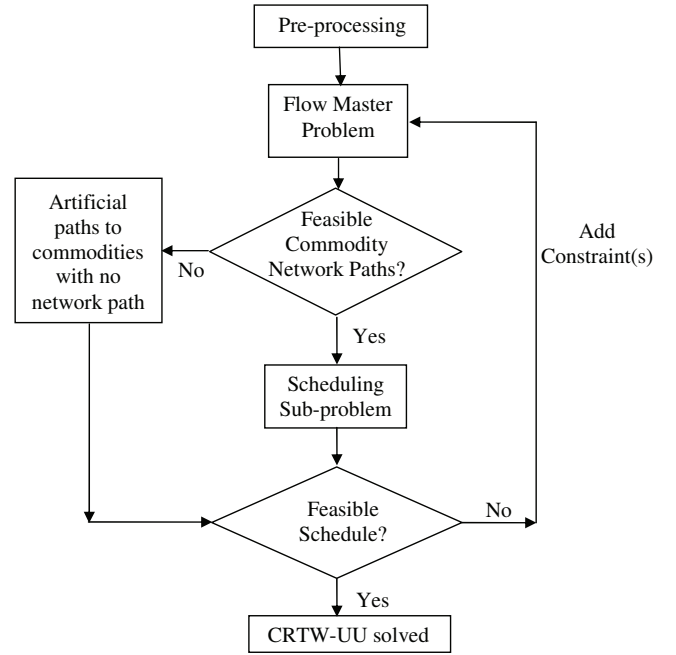


Fig. 2 Flow Chart of the Decomposition Approach

at the depot. These constraints are driven by driver work rules. If vehicle v does not pass through a node i , i is labeled unreachable for v . For all reachable arcs $(i, j) \in A_v$, for all $v \in V$, we have

$$EAT_j^v = EAT_i^v + tt_{ij}; \text{ and} \quad (18)$$

$$LDT_i^v = LDT_j^v - tt_{ij} \quad (19)$$

We label the time-windows and labels of vehicles and shipments thus obtained as *pre-processing time windows or labels*. They are the broadest set of time windows possible over any travel path for the vehicles and shipments (because these time-windows are formed using the *shortest* paths).

- (iii) If $LDT_i^k - EAT_i^k < 0$, the time-window duration of a shipment k is negative at a node $i \in N_k$. Because the pre-processing time-windows are the broadest time-windows, no schedule feasible path for shipment k passes through i . For each shipment network $G_k \forall k \in K$, remove all the schedule infeasible nodes, and all arcs incident to these nodes. This results in a reduced version of G_k containing only those arcs and nodes through which shipment k may pass.
- (iv) We say that the time-windows of shipment k and vehicle v overlap at node i if the combined *vehicle-shipment pre-processing time-window* has non-negative duration. We denote the time-window at node i for the vehicle-shipment pair v and k as: $(EAT_i^{k,v}, LDT_i^{k,v})$, where $EAT_i^{k,v} = \max\{EAT_i^k, EAT_i^v\}$ and $LDT_i^{k,v} =$

$\min\{LDT_i^k, LDT_i^v\}$. Find all possible overlaps of shipment-vehicle pairs at each node $i \in N, \forall k \in K, \forall v \in V$. If the overlap between time-windows of shipment k and vehicle v is negative at node i , that is, $LDT_i^{k,v} - EAT_i^{k,v} < 0$, shipment k cannot travel on the vehicle arcs in A_v incident to node i . We delete such arcs and nodes from G_k , further reducing its size. If the time-windows of a vehicle-shipment pair (v, k) are non-zero at both ends of an arc $(i, j) \in A_v$, shipment k can travel on (i, j) , that is, on that segment of vehicle v 's path, within the specified time-windows.

- (v) Consider the aggregate network G with the vehicle-shipment pre-processing time-windows for all $k \in K$ superimposed on each other at each node $i \in N$. Suppose the time windows of two shipment-vehicle pairs (v, k_1) and (v, k_2) at a node i are individually positive; but do not overlap with each other, thus making it infeasible for both k_1 and k_2 to travel together on any $(i, j) \in A_v$ in any schedule-feasible solution to the CRTW-UU. This infeasibility can be eliminated from the set of feasible solutions to the Flow Master Problem by adding the following constraint to CRTW-UU-SP:

$$\sum_{p_1 \in P_{k_1} | (i,j) \in p_1} f_{p_1} + \sum_{p_2 \in P_{k_2} | (i,j) \in p_2} f_{p_2} \leq 1, \quad (20)$$

where $f_p, p \in P_k$ is defined as in (5)-(15), that is, it is a binary variable that takes on value 1 if shipment k is assigned to path p ; and 0 otherwise. We add these constraints to the Flow Master Problem in the Pre-processing step to eliminate known infeasible solutions.

3.2 Step 2: Flow Master Problem (CRTW-UU-MP)

The goal of the Flow Master Problem is to assign a route to each shipment in the network. We formulate the basic route choice without uncertainty as a multi-commodity flow problem of choosing paths on the networks G_k (resulting after the Pre-processing step), for each shipment in K . To capture uncertainty in capacities and demands, we use the Extended Chance Constrained Programming (ECCP) Marla (2010) that results in the CRTW-UU-MP. The ECCP maintains the structure of the multi-commodity flow problem and allows the use of implicit or explicit column generation techniques for large instances Marla (2007). (Column generation is a technique used in very large-scale formulations where variables number in millions or billions, however only a subset of variables are included in the formulation to begin with. Implicit column generation uses a mathematical formulation to decide which

variables not already included should be brought into the Master Problem, without explicitly enumerating the variables. Explicit column generation on the other hand, enumerates the list of all (or most) variables and evaluates the value of adding them to the Master Problem. Explicit column generation in very large-scale instances is often intractable, while implicit column generation is efficient.) Other approaches that change the structure of the multi-commodity flow formulation, such as Bertsimas and Sim's robust framework Bertsimas and Sim (2004) Bertsimas and Sim (2003), can also be used, and explicit column generation may have to be used for large-scale instances.

3.3 Step 3: Scheduling Sub-problem (CRTW-UU-SP)

After solving the CRTW-UU-MP, vehicle routes and assigned shipment paths $p'_1, \dots, p'_{|K|}$ for shipments $k = 1, \dots, |K|$ are known. Though the paths introduced into the Flow Master Problem are individually schedule-feasible, it is still possible that interactions between these shipment paths $p'_1, \dots, p'_{|K|}$ produce infeasible schedules. Therefore, in the Scheduling Sub-Problem, we 'flow' these shipments on the network G to determine a combined feasible schedule. Thus all the computations in this step are on the aggregate network.

When we wish to protect against travel time uncertainty, we do the following. Use an a priori chosen quantile (higher quantile) of travel time for all the *shipment* paths in the network. For all arcs belonging to shipment paths output by the Flow Master Problem *in that iteration*, the travel time is selected to be a higher quantile as described in §2.3. All other arcs in the network are assumed to have travel time equal to the mean travel time.

The Scheduling Sub-Problem consists of the following steps on the aggregate network:

- i) Initialization: Let $k = 0$ represent the vehicle flows, and commodities $k = 1, \dots, |K|$ represent the shipments.
 - (a) Set $EAT_i^k = 0$, and $LDT_i^k = M$, a very large number, $\forall i \in N, \forall k \in K$. Set $EAT_{O(k)}^k$ to EAT_k at its origin, $LDT_{O(k)}^k = M$, $LDT_{D(k)}^k$ to LDT_k at its destination, and $EAT_{D(k)}^k = 0$.
 - (b) Set $EAT_i = 0$, and $LDT_i = M$.
 - (c) Let L_i be the label set at node i , consisting of the list of shipments that impact the time-windows at i . Set $L_i = \phi$ (empty).
 - (d) Set processing list to empty.
 - (e) For $k = 0, 1, \dots, |K|$, determine the time-windows for shipment k in sequence along its currently assigned path p'_k , (forwards from $O(k)$ for EAT

values and backwards from $D(k)$ for LDT values) in the aggregate network, as indicated in (21) and (22).

$$EAT_j^k = \max\{EAT_j^k, EAT_i^k + tt_{ij}\} \\ \forall (i, j) \in p'_k, \forall k = 0, 1, \dots, |K| \quad (21)$$

$$LDT_i^k = \min\{LDT_i^k, LDT_j^k - tt_{ij}\} \\ \forall (i, j) \in p'_k, \forall k = 0, 1, \dots, |K| \quad (22)$$

These time-windows will at least be as tight as the Pre-processing step time-windows, because each shipment $k \in K$ is restricted to path p'_k . The time-windows for the vehicles ($k = 0$) remain the same as those in the Pre-processing step, because the vehicle routes are given inputs that do not change in solving the CRTW-UU.

- ii) For node $i \in N$ and $k = 0, 1, \dots, |K|$,
- if $EAT_i^k > EAT_i$ and $EAT_i^k \neq 0$, set $EAT_i = EAT_i^k$, add k to L_i if not already present.
 - if $LDT_i^k < LDT_i$ and $LDT_i^k \neq M$, set $LDT_i = LDT_i^k$, add k to L_i if not already present.
 - and add k to the processing list, if it is not already present.

We refer to (EAT_i, LDT_i) as the *movement time windows* at node i .

- iii) For node $i \in N$ if the movement time windows (EAT_i, LDT_i) satisfy $LDT_i < EAT_i$ then there exist two shipments k_1 and k_2 in L_i with paths p_1 and p_2 respectively passing through i , such that $EAT_i^{k_1} > LDT_i^{k_2}$.

- (a) Without loss of generality, if $k_1 = 0$ (it is a vehicle path), add a constraint of the form

$$f_{p_2} \leq 0, \quad (23)$$

- (b) Else if $k_1 > 0$ and $k_2 > 0$ add a constraint of the form:

$$f_{p_1} + f_{p_2} \leq 1, \quad (24)$$

to the Flow Master Problem.

- iv) If the processing list is not empty, remove the first element from the list, add it at the end of the list, and go to step v.
- v) Update EAT_i, LDT_i for all $(i, j) \in p'_k \forall k \in 0, 1, \dots, |K|$, (that is, for each arc in each vehicle path and each shipment path chosen by the Master Problem), processing (i, j) in sequence along p'_k (propagating in the forward direction for the EAT values and in the backward direction for the LDT values) as:

If $EAT_j < EAT_i + tt_{ij}$, then

$$EAT_j = EAT_i + tt_{ij}, L_j = L_j \cup L_i, \quad (25)$$

If $LDT_i > LDT_j - tt_{ij}$, then

$$LDT_i = LDT_j - tt_{ij}, L_i = L_i \cup L_j \quad (26)$$

Remove any repeated shipments from L_i and L_j and return to Step (iv).

- vi) One execution of (iv) and (v) for all vehicles and shipments constitutes one iteration. If the change in EAT_i, LDT_i for successive iterations of (iv) and (v) is significant, repeat step (iv) and (v), else go to step (vii).
- vii) For any arc $(i, j) \in G$ such that $(i, j) \in p_k, k = 0, 1, \dots, |K|$, if $LDT_j - EAT_i < tt_{ij}$, this indicates an infeasibility in schedule caused by the interaction of paths selected by the Flow Master Problem for shipments belonging to the set $L_i \cup L_j$. If no such arcs (i, j) are found, stop; a set of feasible time-windows is found. Else, go to step (viii).
- viii) Add the following constraint to the Flow Master Problem:

$$\sum_{k \in (L_i \cup L_j)} f_{p'_k} \leq |\overline{L_i \cup L_j}| - 1; \quad (27)$$

where $\overline{L_i \cup L_j}$ is the set $L_i \cup L_j$ with no elements repeated; and $|\overline{L_i \cup L_j}|$ is the cardinality of $\overline{L_i \cup L_j}$. Constraint (27) states that the set of paths causing schedule infeasibility of the current solution should not be repeated in further iterations.

3.4 Stopping Criterion

If no schedule infeasibilities are identified in the scheduling algorithm, that is, $LDT_j - EAT_i < tt_{ij}$ for all $(i, j) \in G$, the CRTW-UU is solved; otherwise, the algorithm returns to Step 2 with added constraints of the type (27) in the Flow Master Problem. We iterate between solving the Flow Master Problem and the Scheduling Sub-Problem until no schedule infeasibilities are identified in the Flow Master Problem solution by the Scheduling Sub-Problem. Then the algorithm terminates with a feasible routing and schedule. (Note that a feasible solution is guaranteed because there exists at least one feasible solution of assigning to each shipment its artificial path - which has high cost but is, by definition, always schedule feasible.)

3.5 Output

The solution obtained from the above algorithm is a set of paths to which shipments are assigned and a set of *time-windows* indicating the earliest and latest time each vehicle and shipment movement can occur. The solution might route a shipment along one or more vehicles; or on its artificial arc, in which case it is not served and incurs a penalty. The schedule time-windows for the routing provide bounds within which the current set of vehicle and shipment flows may be scheduled.

3.6 Correctness of the Algorithm

The correctness of the decomposition approach is dependent on the fact that the cuts introduced in the Pre-processing and Scheduling Sub-Problems eliminate only regions of the Flow Master Problem solution space that are infeasible to the original CRTW(-UU) problem, and ensure that the current infeasible solution is not repeated.

Proposition: The cuts generated in the Pre-processing module and Scheduling Sub-Problem correspond to infeasible CRTW(-UU) solutions, and do not eliminate any feasible CRTW(-UU) solutions.

Proof. In the Pre-processing stage, the pre-processing time-windows are identified using shortest path computations, and therefore the time-windows identified are the broadest possible time-windows for any possible shipment and vehicle movements. Therefore, when we eliminate nodes and arcs from a shipment network $G_k \forall k \in K$, we eliminate those solutions that cannot satisfy schedule constraints under *any* conditions. For the same reason, adding constraints of the type (20) eliminates all vehicle-shipment pairs that are schedule-infeasible even under the broadest (shortest-path-based) time-windows. Hence, such pairs of vehicle-shipment pairs or shipment-shipment pairs cannot travel together in *any* solution. Thus, constraints (20) are valid, and do not eliminate more feasible space from the Flow Master Problem than necessary.

The correctness of the Scheduling Sub-Problem (CRTW-UU-SP) is due to two reasons - first, that the movement time-windows calculated via steps (i) - (vi) are the broadest possible time-windows for the shipments and vehicles *as assigned* in the Flow Master Problem solution; and second, that the labeling procedure employed identifies shipment paths that are infeasible together.

Step (i) of the scheduling sub-problem first identifies the possible time-windows EAT_i^k and LDT_i^k of each vehicle-shipment pair along the paths of each shipment. These are the broadest possible time-windows that allow each individual shipment to travel on the network, without ensuring combined consistency of shipment paths in the solution. If no overlaps exist between a pair of shipment paths at a node in step (ii), it indicates that the paths that do not have overlapping time-windows are not feasible together even under the broadest time windows for each path, ensuring correctness of constraints (23) and (24). The node labels at this step consist of all shipments that tighten the time windows. We then iterate through steps (iv) and (v) to generate the broadest time windows that allow all vehicle and shipment movements assigned by the Flow

Master Problem to take place together. As the iterations take place via propagation along paths, the labeling procedure identifies all preceding and consecutive nodes that tighten the time windows at a node; and add the list of shipments that determine the time-windows at the preceding or consecutive steps. Moreover, because the EAT s are propagated forward and the LDT s are propagated backwards along all paths, when the time-windows converge, we have the *broadest* time windows that allow *all* path movements to occur.

If after step (vi), all nodes and arcs have time-windows that are non-negative, then one or more feasible schedules can be constructed. One trivial case is to set the scheduled time at each node $i \in N$ to EAT_i . Alternatively, a feasible schedule can be constructed by setting the scheduled time at each node $i \in N$ to LDT_i . If we find instead, that $tt_{ij} \leq LDT_j - EAT_i$ for some $(i, j) \in A$, the minimum time necessary to traverse the arc (i, j) exceeds the maximum allowable time to get from i to j , and there is a schedule infeasibility. Then, the labels at i and j together identify the shipment paths that lead to the time-windows, and consequently, to infeasibility. The constraint (27) added to the Master Problem then prevents re-occurrence of the same infeasible solution. \square

3.7 Convergence of the Algorithm

There exists at least one feasible solution to the CRTW-UU, namely, the choice of artificial paths for all shipments, which is also the maximum cost solution. Therefore, there exists an optimal solution to the CRTW-UU.

In each iteration of our decomposition approach, we solve a relaxed version of the CRTW-UU formulation in the CRTW-UU-MP by relaxing scheduling constraints. Thus the cost incurred by the solution to the CRTW-UU-MP is a lower bound on the objective function cost of the CRTW-UU. As we add cuts to the Flow Master Problem, the objective function cost increases or stays the same. Each cut corresponds to the elimination of at least one infeasible solution to the CRTW-UU. Hence, each cut is unique, and after a finite (but possibly large) number of iterations, all infeasible solutions are eliminated and the Flow Master Problem solution will be feasible to the CRTW-UU. Because the Flow Master Problem is a relaxation of CRTW-UU and the added cuts eliminate only infeasible CRTW-UU solutions, finding a feasible schedule to the Flow Master Problem corresponds to solving, that is, finding an optimal solution to, the CRTW-UU.

3.8 Running Time of the Algorithm

Pre-processing involves employing network labeling and shortest path algorithms. We solve $O(2(|K|+|V|))$ shortest path problems, for which highly efficient algorithms such as Dijkstra's algorithm are available. Note that in the Pre-processing stage, the shipment networks are reduced in size, due to elimination of arcs and nodes. Computation of time-windows involves computation at each node and arc of each shipment network, requiring a maximum of $|K|(|N| + |A|)$ computations.

The Flow Master Problem is solved using standard integer programming techniques. Its size is smaller than conventional multi-commodity flow formulations with time windows (captured either as time variables through time-space networks) and therefore expected to be less complex, with fewer variables, as well as more tractable.

The Scheduling Sub-Problem involves network labeling algorithms to compute the time-windows, similar to the Pre-processing stage. Tracing the paths of commodities involves $O(|N|^2(|K| + |V|) + |N|(|K| + |V|))$ operations, because label initialization requires $O(|N|(|K| + |V|))$ and each time a shipment or vehicle path is traced, at least one label at some node i is tightened (except for the last set of propagations, where we stop). The number of labels at each node is restricted to $(|K| + |V|)$ and each relabeling takes $O(|N|)$ steps.

One can add multiple constraints of the form (23), (24) or (27) in each iteration by identifying all infeasibilities corresponding to a selected set of paths in a single iteration, or by adding constraints for a subset of infeasibilities and breaking out of the scheduling sub-problem. Because calling the optimization engine to solve the Flow Master Problem multiple times is more computationally expensive than the scheduling sub-problem, we recommend identifying and eliminating as many infeasibilities as possible in each iteration.

It is possible that the number of iterations between the Flow Master Problem and the Scheduling Sub-Problem will be large. Each cuts that is added to the space will always be effective as atleast one infeasible schedule solution is eliminated at each iteration. However, it may happen that several such cuts will need to be added, increasing the number of iterations. In theory, we can construct pathological instances where the number of iterations can be very large. However, computationally, we observe that the number of iterations of the decomposition approach is sensitive to the starting solution from the Flow Master Problem in the first iteration. To speed it up, a seed solution such as one from a traditional modeling approach, or one that is being implemented by the carrier, may be provided. In case of a high degree of infeasibility in the optimal solu-

tion (specifically, if several shipments cannot be served and take on artificial paths in the optimal solution) the decomposition approach can consume a lot of time iterating between the Flow Master Problem and the Scheduling Sub-Problem. This is because the Flow master Problem minimizes the penalties and maximizes the number of shipments served. It will therefore examine all possible combinations before assigning a shipment to an artificial path. To decrease the number of iterations in such cases, we use the techniques described in §3.9 and §3.10.

3.9 Identification of Dominant Cuts

We can strengthen the constraints in the Pre-processing and Scheduling Sub-Problem as described below.

Consider constraints in the Scheduling Sub-Problem of the form:

$$f_{p_1} + f_{p_2} \leq 1, \quad (28)$$

$$f_{p_1} + f_{p_3} \leq 1, \quad \text{and} \quad (29)$$

$$f_{p_2} + f_{p_3} \leq 1. \quad (30)$$

Notice that these three constraints can be modeled effectively with a single, *dominant* constraint, of the form:

$$f_{p_1} + f_{p_2} + f_{p_3} \leq 1. \quad (31)$$

Similarly, in the Pre-processing step, suppose shipments k_1 , k_2 and k_3 are identified, such that each pair cannot travel together on an arc (i, j) , pair-wise constraints of type (20) are generated for k_1 and k_2 , k_2 and k_3 , k_3 and k_1 . These constraints can be replaced by a single *dominant constraint*:

$$\begin{aligned} & \sum_{p_1 \in P_{k_1} | (i,j) \in p_1} f_{p_1} + \sum_{p_2 \in P_{k_2} | (i,j) \in p_2} f_{p_2} \\ & + \sum_{p_3 \in P_{k_3} | (i,j) \in p_3} f_{p_3} \leq 1. \end{aligned} \quad (32)$$

One approach to finding such constraints is to construct an *incompatibility* network over which completely connected subgraphs, called cliques, are identified. To construct the incompatibility network, we create one node for each path in the Flow Master Problem solution. An arc connects a pair of nodes if the associated paths are contained in at least one constraint that is added to the Flow Master Problem as a result of the Pre-processing or Scheduling Sub-Problem solution steps. Each completely connected subgraph in the incompatibility network is a clique that corresponds to set of paths (the nodes of the clique), of which at most

one can exist in a solution. We find dominant, or strong cuts, by identifying *maximally connected components*, or *cliques*. For the constraints (28) - (30), Figure 3 is the incompatibility network, giving rise to the dominant constraint (31).

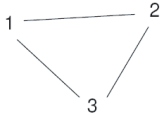


Fig. 3 Incompatibility Network: Cliques

Cliques have been well-studied in the literature. Tarjan (1972) presents one of the earliest and best (asymptotically efficient) to find cliques in a graph. More efficient algorithms that build upon the above have been proposed in Nuutila and Soisalon-Soininen (1994) and Wood (1997). Though identifying maximally connected components in a network is NP-hard (Garey and Johnson 1979), because we expect to have only a subset of shipments incurring infeasibilities at a particular node, we expect that the size of our incompatibility network will be small and therefore tractability should not be an issue. Also, we need not identify all cliques in the graph to improve the algorithmic efficiency; adding even a few clique constraints can improve algorithmic performance.

Identification of dominant cuts minimizes the number of cuts that must be identified and added to the Flow Master Problem, thus potentially reducing the number of iterations of the decomposition algorithm and leading to faster solution times. Note that identifying cliques and stronger constraints requires additional computation time. It is necessary, then, to find appropriate trade-offs between increased time to identify stronger constraints and the corresponding reduction in overall solution time.

3.10 Identifying multiple cuts per iteration

Suppose we identify, in a specific iteration, a set of m paths p_1, p_2, \dots, p_m , belonging to commodities k_1, k_2, \dots, k_m respectively, as schedule-incompatible. That is, in steps (vii) and (viii) of the Scheduling Sub-Problem, we identify a set of arcs $(i, j) \in G$ for which $LDT_j - EAT_i < tt_{ij}$ and the labels on the nodes indicate that paths p_1, p_2, \dots, p_m are the ones that cause the infeasibility. Then, the constraint $p_1 + p_2 + \dots + p_m \leq m - 1$ is to be added to eliminate the infeasibility.

Proposition: Now suppose, without loss of generality, that path $p_1 \in P^{k_1}$ (of commodity k_1) contains arcs $(i_1, j_1), \dots, (i_p, j_p) \in G$, all of which have negative time-windows with $LDT_j - EAT_i < tt_{ij}$, as shown in the Figure. Let $\tilde{p} \in P^{k_1}$ be another path of commodity k_1 that has tighter time-windows than p_1 on arcs $(i_1, j_1), \dots, (i_p, j_p) \in G$, as shown in Figure 4.

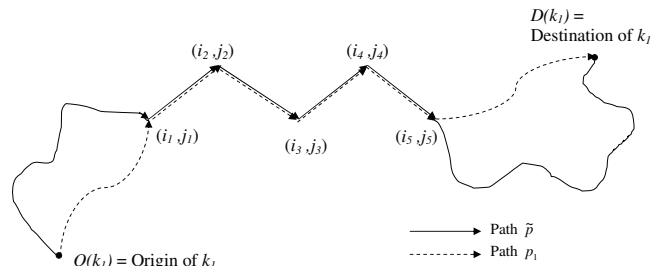


Fig. 4 Multiple paths for commodity k_1

Then, $\tilde{p} + p_2 + \dots + p_m \leq m - 1$ can also be added to the Flow Master Problem in the same iteration.

Proof. We are given that the time-windows of \tilde{p} when propagated as described in Step (i) of the Scheduling Sub-Problem (CRTW-UU-SP) are tighter than the corresponding time-windows of p_1 on arcs $(i_1, j_1), \dots, (i_p, j_p) \in G$. Then, when time-windows on $\tilde{p}, p_2, \dots, p_m$ are propagated in the following steps of the Scheduling Sub-Problem to check for compatibility, the movement time windows and the final time windows in Step (vii) will be at least as tight as those for p_1, p_2, \dots, p_m . Because p_1 causes infeasibility, \tilde{p} will also cause infeasibility and the constraint $\tilde{p} + p_2 + \dots + p_m \leq m - 1$ can also be added to the Flow Master Problem in the same iteration.

Paths of the type \tilde{p} can be identified by examining the network G_{k_1} and performing a network modification to combine arcs $(i_1, j_1), \dots, (i_p, j_p)$ into a single arc, and examining the other paths of commodity k_1 on this network. They may also be found by detecting ‘longer paths’, for example, by finding the shortest path (with all distances made negative) between $O(k_1)$ and (i_1, j_1) , and (i_5, j_5) and $D(k_1)$ as shown in the figure. \square

Similarly, such paths may be found for k_2 as well, keeping p_1, p_3, \dots, p_m the same and identifying those paths that will cause infeasibility with p_1, p_3, \dots, p_m on G_{k_2} , etc. This is most useful when there are several paths per shipment which differ by a few arcs. In such a case, it is possible that the algorithm will try each and every alternate path in every iteration, causing the number of iterations to grow significantly. Then, adding multiple such cuts by examining the existing paths and deter-

mining more than one path of type \tilde{p} that can cause infeasibility can decrease the number of iterations of the algorithm.

3.11 Advantages and Disadvantages of the Decomposition Modeling Approach

Our decomposition methodology involves solving a multi-commodity Flow Master Problem and a series of network-based, easy-to-solve sub-problems. Because we are breaking a large optimization problem into two smaller parts, one involving finding an optimal solution and the other simply finding a feasible solution, each iteration is fairly tractable. Moreover, the Scheduling Sub-Problems to ascertain whether or not a feasible schedule exists for the Flow Master Problem solution are very efficient. Polynomial-time network labeling algorithms are used, which not only identify if a feasible schedule exists, but also identify one or more constraints that can be added to the master problem to guide it towards schedule-feasible solutions. Also, specific types of additional constraints, such as those restricting time-windows of a vehicle at a particular location, can be added without increasing algorithmic complexity.

The decomposition approach explicitly captures the fact that associated with any feasible solution to the CRTW-UU, there is not only one feasible schedule but in fact, a set of time-windows associated with movements. Upon termination of the algorithm, the scheduling sub-problem would have identified a set of feasible time-windows corresponding to those movements. Therefore, for a solution obtained from any approach (traditional, decomposition, or other), the scheduling sub-problem can be used as a post-processing step, in order to generate windows of schedules and characterize the solution's sensitivity to uncertainty.

Modeling uncertainty in the demand and travel time parameters using the decomposition approach does not incur much additional complexity relative to solving the CRTW for the nominal case. However, there are limitations in the modeling capabilities. One such is that correlations between uncertain parameters (such as between travel times of adjacent links on a network) are not modeled. A second is that the choice of 'protection levels' for travel times on the network has to be made a priori, and involves repeated solution of instances of our model. It is possible to apply a sequential process to try increased protection levels in later iterations of the Scheduling Sub-Problem, however, it would involve a trial-and-error process on the part of the user. As in the case of the ECCP formulation of the Flow Master Problem that maximizes protection level within a budget, it would be useful to develop a mechanism to

automate the choice of path protection levels in the Scheduling Sub-Problem.

Mathematically, the presented approach always works in the feasible domain, because the set of paths for each commodity include an artificial path with high cost and zero travel time. Note that there is always a feasible but cost-maximizing solution (all commodities take their artificial paths). In practice, the use of an artificial path indicates infeasibility in the sense that the related shipment(s) cannot be served within the specified time-window, and these shipments will be delayed in service. They can either be delivered with an expanded time window, during the next day of service, or subcontracted to another carrier. The appropriate cost can be assigned for the delay, and incorporated into the Master Problem as the 'penalty cost of not serving within the specified time window'.

Computationally, the number of iterations of the decomposition approach is sensitive to the starting solution to the Flow Master Problem in the first iteration. To speed it up, a seed solution such as one from a traditional modeling approach, or one that is being implemented by the carrier, may be provided. In case of a high degree of infeasibility in the optimal solution (specifically, if several shipments cannot be served and take on artificial paths in the optimal solution) the decomposition approach can consume a lot of time iterating between the Flow Master Problem and the Scheduling Sub-Problem. This is because the Flow master Problem minimizes the penalties and maximizes the number of shipments served. It will therefore examine all possible combinations before assigning a shipment to an artificial path.

When the algorithm terminates, we always find a feasible solution. The solution contains feasible schedules for the routes; and moreover, all possible feasible solutions are found. To find feasible solutions more easily, warm start procedures with solutions used by the carrier (even if partially infeasible) can be used.

However, if the algorithm is stopped before completion, it is possible that in the current solution, a set of commodity routes have incompatible schedules with each other. This solution could be made feasible by assigning artificial paths to those shipments with incompatible schedules that is, in practice, expand their time window and serve them late, or use an alternate (subcontracted) carrier. Another possible way to obtain a feasible solution is to use insertion heuristics to add these commodities into existing route in the solution, or create new route(s) for these commodities. Thus, the solution from the algorithm does not leave one with no solution at all, though it could be far from optimal. In

practice, this is particularly true in the case of a warm start solution.

This algorithm can be applicable for any cost structure in which the objective function can be decomposed between the master Problem and the Scheduling Sub-Problem. Specifically, if the costs related to the choice of paths can be captured in the Master Problem alone, and the Scheduling problem can be cast purely as a feasibility problem, the decomposition algorithm can be used. This means that the decomposition algorithm can be used if the objective is based on cost structures that price routes based on length of routes, number of routes chosen, shipment type, etc.

4 Case Study: Truckload shipment routing and scheduling

4.1 Problem Description

Our proof-of-concept case study considers the planning of region-wide ground movements of a large pickup and delivery carrier. We consider truckload movements from a set of depots to a number of locations. The carrier owns a fleet of trucks whose movements (routes) over the network are pre-determined. However, schedules for these routes are not determined. Demands arise in the form of trailers that need to be moved from their origins to their destinations on the network of vehicles. Time-windows within which the trailers should move are also specified. Shipments that are not served within the specified time-windows incur a non-service penalty. The data instance upon which most of our experiments are performed has 6 depots, 28 locations, 41 vehicles and 87 shipments in a daily schedule, and is described further in §4.4. This data set represents a medium-sized operation for this carrier. Our goal is to find a set of routes for the trailers, and schedules for truck and trailer moves; which minimize non-service costs, and are robust to uncertainty in travel times and demands.

We solve this problem using the decomposition approach algorithm presented in §3 and the traditional model presented in §4.2. We measure the performance of our algorithm based on various metrics: (i) number of shipments with infeasible schedules, (ii) running time of the algorithm, (ii) Percentage of scenarios requiring delivery deadline extensions of 0, 15, 30 and 60 minutes respectively to be feasible; (iii) percentage of scenarios infeasible with 60 minute extension in latest delivery time; and iv) level of robustness of solutions. To evaluate solution robustness, we use the simulator described in §4.3.

4.2 Traditional Approach

In this section, we present a traditional modeling approach for CRTW, in which we model time as a continuous variable Cordeau et al. (2007). Let t_i be continuous variable representing the time at which vehicle v departs from node $i \in N(v) \forall v \in V$ and f_0^k represent the artificial path for shipment k . Borrowing notation from §2.2, the formulation of the traditional model is:

$$\min \sum_{k \in K} \sum_{p \in P^k} d_k c_p f_p \quad (33)$$

$$\text{s.t. } t_i + t t_{ij}^k \delta_{ij}^p f_p - M(1 - f_p) \leq t_j \quad \forall (i, j) \in A, \\ \forall p \in P_k, \forall k = 0, 1, \dots, |K| \quad (34)$$

$$t_{O(k)} \geq EAT_{O(k)}^k (1 - f_0^k) \quad \forall k = 0, 1, \dots, |K| \quad (35)$$

$$t_{D(k)} \leq LDT_{D(k)}^k + M(f_0^k) \quad \forall k = 0, 1, \dots, |K| \quad (36)$$

$$\sum_{p \in P^k} f_p = 1 \quad \forall k = 0, 1, \dots, |K| \quad (37)$$

$$\sum_{k \in K} \sum_{p \in P^k} d_k \delta_{ij}^p f_p \leq u_{ij} \quad \forall (i, j) \in A \quad (38)$$

$$f_p \in \{0, 1\} \quad \forall p \in P^k, \forall k = 0, 1, \dots, |K| \quad (39)$$

$$t_i \geq 0 \quad \forall i \in N. \quad (40)$$

Constraints (37) - (39) form the path-based multi-commodity flow formulation that assigns a path to each shipment. Schedule related constraints (34) constrain the differences in the departure time on adjacent nodes of a vehicle or shipment path to the travel time on the arc. (35) and (36) constrain pickup and delivery of a shipment to be within the time-windows of pickup and delivery of the shipment at its origin and destination respectively.

4.3 Simulator

The simulator examines if a given solution is feasible under a set of scenarios, where each scenario comprises of a set of realized demands for shipments and a set of travel times on arcs. The simulator (41) - (49) is run once for each scenario and $t t_{ij} \forall (i, j) \in A$ and $d_k \forall k \in K$ represent the realizations in that scenario. Let p_k^{sol} be the path for shipment (trailer) k , for all shipments (trailers) k output by the optimization (decomposition or other) approach. Let t_i be the time of departure of the vehicle (whose path i belongs to) at each node $i \in N$. Let $l \in L$ represent the allowable levels of extension to the shipment delivery time, to measure degree of schedule infeasibility; and e_l be the extension in minutes allowed for level l . In our experiments we set $L = 3$, and $e_1 = 15$ minutes, $e_2 = 30$ minutes, $e_3 = 60$ minutes. Let

$y_{lk} = 1$ if shipment $k \in K$ requires an extension of up to e_l in order to be feasible.

Using the same notation used in §2 and §4.2, the simulator solves the following problem. The goal is to minimize costs of extending delivery deadlines while constraining the problem to use only the shipment paths from the solution to the optimization problem.

$$\min \sum_{l \in L} \sum_{k \in K} e_l y_{lk} \quad (41)$$

$$\text{s.t. } t_i + t_{ij}^k \delta_{ij}^p f_p - M(1 - f_p) \leq t_j \quad \forall (i, j) \in A, \\ \forall p \in P_k, \forall k = 0, 1, \dots, |K| \quad (42)$$

$$t_{O(k)} \geq EAT_{O(k)}^k (1 - f_0^k) \quad \forall k = 0, 1, \dots, |K| \quad (43)$$

$$t_{D(k)} \leq LDT_{D(k)}^k + M(f_0^k) + \sum_{l \in L} e_l y_{lk} \\ \forall k = 0, 1, \dots, |K| \quad (44)$$

$$\sum_{p \in P^k} f_p = 1 \quad \forall k = 0, 1, \dots, |K| \quad (45)$$

$$\sum_{k \in K} \sum_{p \in P^k} d_k \delta_{ij}^p f_p \leq u_{ij} \quad \forall (i, j) \in A \quad (46)$$

$$f_p \in \{0, 1\} \quad \forall p \in P^k, \forall k = 0, 1, \dots, |K| \quad (47)$$

$$t_i \geq 0 \quad \forall i \in N \quad (48)$$

$$p_k^{sol} = 1 \quad \forall k = 0, 1, \dots, |K| \quad (49)$$

4.4 Computational Experience

We test our approach on proprietary data provided by the pickup and delivery carrier of interest. The data instance contains movements of 41 vehicles spanning 28 locations, on which a set of 87 shipments has to be picked up and delivered. The vehicles are required to operate a daily schedule. This data set represents a medium-size daily operation of the carrier. Our experiments evaluate the performance of our decomposition approach as against a traditional approach for two types of uncertainty (occurring simultaneously) in each scenario: (i) travel time uncertainty and (ii) demand uncertainty. Details of the scenarios of uncertainty are provided in the following sections.

We solve the problem instances using (a) a traditional approach by solving the equations (33) - (40) and (b) using our decomposition approach algorithm described in §3. We use ILOG CPLEX 12.2 interfaced with Java via Eclipse 3.7.0. Computational experiments are conducted on a Dell PowerEdge R510 computer with dual Xeon X5670 2.93 Ghz processors and 4GB RAM.

4.5 Random scenario generation

Uncertainty in travel and service times, and uncertainty in demands are two of the most common sources of lack of robustness in network schedules. In our scenarios, travel time uncertainty manifests independently of demand uncertainty, that is, their realizations are assumed independent.

We generate scenarios of uncertain travel times to reflect two types of underlying service distributions - uniform and gamma. Our first set of 5,000 scenarios is drawn from uniform distributions centered around the deterministic values of travel times provided by the carrier, with ranges of $\pm 10\%$ around the mean. The second set of 5,000 scenarios is drawn from the Gamma distribution, which has been seen in practice to reflect travel time uncertainty well (Dessouky et al. 1999). Gamma distributions are skewed to have higher probability of delays rather than early arrivals. In our experiments, we use a truncated gamma distribution with the mean equal to the input travel time provided by the carrier. The distribution is truncated to avoid overly long travel times as well as overly short travel times compared to the mean, as such occurrences are impractical. The Gamma distribution samples are broader in range than the uniform distribution samples drawn.

Scenarios of demand realizations are sampled with a 3% uncertainty in trailer demand, which is representative of the uncertainty realistically experienced by the carrier company of interest. This means that the number of trailers (truckloads) to be picked up and delivered can potentially increase by 3%. We generate scenarios in which, for about 3% of the time, the carrier sees increased demands from some locations in the form of an additional trailer to be delivered (between the same origin and destination and with the same time windows). Our set of 5,000 scenarios randomly chooses 3% of all shipments to have two trailers instead of one to be delivered. (From discussions with the company, it is rare that the uncertainty decreases, that is, the demand between an origin-destination pair has zero trailer demand. Moreover, the decrease in demand is not as critical a scenario as an increase. Therefore we do not consider the scenario of zero demand between an origin-destination pair.)

The scenarios of demand and travel times are randomly paired to generate combined uncertainty. We present results of simulation under the traditional approach and from the robust decomposition approach.

4.6 Applying the decomposition approach

We model demand uncertainty using our decomposition approach using the *CRTW-UU-MP* (with a budget $\delta=4$). The budget δ is chosen as a fraction of the infeasibility that is experienced by the non-robust solution from the traditional approach. After simulating the solution from the traditional approach under several scenarios and finding its percent infeasibility, we set the robustness budget δ as equivalent to a small fraction of the infeasibility of the traditional approach solution. That is, this is the number of commodities we allow to not be shipped in order that the entire solution be more robust under uncertainty. We set 5% of the total number of shipments (that is 4 shipments) to be allowed to be infeasible. Note here, that we use the ECCP model that protects against the greatest decrease in vehicle capacity (in the right-hand-side of the constraint (8)) as a proxy for increase in demand (in the left-hand-side of the constraint). We model travel time uncertainty by *a priori* assuming a percentage protection for shipment paths. The quantiles of protection for the paths chosen in the Master Problem are assumed to be about 70th percentile protection. This is chosen based on a priori estimates of degree of protection required.

We compare performance of the decomposition approach solutions to those from the traditional approach (33) - (40) in which neither ECCP is used to protect against demand uncertainty, nor are arcs protected against travel time uncertainty. Note that in the traditional model, a subset of arcs may be protected up to a certain quantile, however, it is difficult to decide *which* arcs to protect without knowing the paths that are present in the solution. Protecting all arcs against uncertainty is very expensive as it results in several schedule infeasibilities.

4.7 Results and Discussion

The three solutions we discuss in Table 1 are the following.

- Solution 1 represents the solution using a traditional deterministic approach, obtained by solving (33) - (40), and not modeling travel time or demand uncertainty.
 - Solution 2 is the decomposition approach solution, obtained using the 70th quantile of uniform distribution travel times, and protected against demand uncertainty using the ECCP formulation in CRTW-UU-MP. The distribution used in choosing the quantile is the same as the distribution that the scenarios are sampled from.
 - Solution 3 is the robust decomposition approach, obtained using the 70th quantile of gamma distributed travel times, and protected against demand uncertainty using the ECCP formulation in CRTW-UU-MP.
- We compare the performance of these solutions using the following metrics:
- Ships. Infeasible = Number of shipments with no path assigned (in the problem solution, they are assigned an ‘artificial’ path with high cost)
 - Solution time = time to run the approach, in seconds
 - Iterations = number of iterations needed for the decomposition approach
 - ‘0 min’ Feasibility = Percentage of scenarios requiring no extension to the latest delivery time of shipments in order to be feasible;
 - ‘15 min’ Feasibility = Percentage of scenarios requiring an extension up to a max of 15 minutes after the latest delivery time of shipments in order to be feasible;
 - ‘30 min’ Feasibility = Percentage of scenarios requiring an extension greater than 15 minutes and up to a maximum of 30 minutes after the latest delivery time of shipments in order to be feasible;
 - ‘60 min’ Feasibility = Percentage of scenarios requiring an extension greater than 30 minutes and up to a maximum of 60 minutes after the latest delivery time of shipments in order to be feasible;
 - Infeasible % = percentage scenarios infeasible even with a deadline of 60 minutes to the latest delivery time of shipments

Num.	Approach	Ships. Infeasible	Sol. Time(sec)	Iterations
1	Traditional	0	5	-
2	Decomposition	4	60	5
3	Decomposition	4	58	4

Table 1 Characteristics of solutions from the traditional approach and decomposition approach

Nr.	Solution Approach	Percent feasibility under extensions				Infeasible %
		0 min	15 min	30 min	60 min	
1	Traditional	27.40	13.34	0.44	0.00	58.82
2	Decomposition	46.04	19.28	0.20	0.00	34.48

Table 2 Percentage feasibility of solutions under uniformly distributed travel time uncertainty with 10% range and 3% demand uncertainty

From Table 1, we see that our decomposition approach solves realistic problems in reasonable time scales.

Solution		Percent feasibility under extensions				Infeasible %
Nr.	Approach	0 min	15 min	30 min	60 min	
1	Traditional	4.84	8.30	19.44	5.62	61.8
3	Decomposition	19.80	15.82	25.92	6.96	31.50

Table 3 Percentage feasibility of solutions under Gamma distributed travel time uncertainty (shape = 9) and 3% demand uncertainty

The model takes longer to solve than the traditional approaches but the solutions are more robust when compared with respect to multiple metrics. Because the decomposition approach captures higher quantiles of time, 4 shipments are schedule infeasible in solutions 2 and 3 (reported in column 3 of Table 1).

Traditional approaches result in a set of paths and point schedules, however, the paths flows output in the solution have more flexibility in their movements. Our decomposition recognizes the existence of time windows for a given feasible set of movements, computes the time windows and exploits the existence of such time windows to capture the extent of schedule flexibility and arrive at robust solutions.

If instead of a higher quantile of protection, the mean travel time were used in the decomposition approach and the (1) - (4) were used instead of CRTW-UU-MP, the decomposition approach gives solutions with the same objective as the decomposition approach, empirically confirming the correctness of our approach.

Tables 2 and 3 compare the performance of solutions from traditional and decomposition frameworks with respect to feasibility metrics of delivery times, and need for deadline extensions. The solutions from the decomposition approach (solutions 2 and 3) are more robust in that they need fewer extensions (both in quantity and degree) on delivery deadlines. This is because the decomposition approach protects against both travel time and demand uncertainty. The number of scenarios in which decomposition approach solutions need deadline extensions are fewer than those required by solutions from traditional approaches. The solutions from the robust decomposition approach also exhibit less infeasibility, where infeasibility is defined as the inability to deliver shipments more than 60 minutes after the deadline.

The solution quality differences in the traditional and decomposition approaches are due to the following. To deal with demand uncertainty, compared with a traditional model, the ECCP formulation embedded within the decomposition approach (CRTW-UU-MP) distributes loads more equitably among vehicles. The traditional model may couple together several shipments on one vehicle and when higher demands are realized, this may lead to infeasibility. Because of the objective function of maximizing protection levels, the ECCP-

based decomposition model distributes loads more evenly across vehicles, allowing for possible extra capacity if demand is higher than expected. The higher quantiles of protection applied to specific paths in the solutions also allows for greater slack in the schedule, thus decreasing the probability of connections breaking for shipments and delivery times exceeding deadlines.

5 Other Applications

Our decomposition approach also allows further flexibility in modeling. We illustrate this with the example of priority shipment scheduling and dynamic airline scheduling.

5.1 Priority Shipment Scheduling

In specific instances, carriers would like to be able to guarantee service for some shipments even in the worst-case of uncertainty. For the priority network (or sub-network) of the carrier the question might be: how many shipments or trailers can be served even in the worst-case?

In such cases, a worst-case-based approach might be useful. If the bounds of uncertainty of travel time are known (or assumed), then, such uncertainty can be modeled in the decomposition approach by making the following changes. While performing forward propagation in the scheduling sub-problem, the most optimistic (smallest) value of travel time is considered; and when performing backward propagation, the most pessimistic (largest) value of travel time is used. This will generate time-windows that are feasible under *all* realizations of travel times within the bounds assumed, and provide a more conservative estimate for the number of shipments that can be served; at the same time, allowing a high service level for all shipments that are served.

5.2 Dynamic Airline Scheduling

Dynamic airline scheduling (Jiang and Barnhart 2009) addresses the problem of demand stochasticity faced by airlines. Because airlines determine their flight schedules a year to six months in advance, when demand forecasts are highly uncertain, they face the issue of matching capacity to demand during operations. Demand forecasts for flights become more accurate as the date of flight approaches, giving the airline an opportunity to match capacity to demand better by adopting a dynamic scheduling approach. In this context, demand uncertainty is typically addressed not through

fleet/aircraft size adjustments, but by re-scheduling of flights, using the same aircraft, in a small (typically 15-minute) time window so as to capture as much demand as possible. The decision to be made is how much re-timing to perform for each flight in order to maximize the capture of demand (which is uncertain). This was first proposed by Jiang and Barnhart (2009), who proposed an integer programming based approach for this problem.

Our decomposition approach can be applied to this setting by creating static networks for aircraft movements, and finding the revenue-maximizing assignments in the Flow Master Problem (Marla and Barnhart 2011). The Scheduling Sub-problem traces passenger itineraries and if some are found infeasible, indicates to the Flow Master Problem that time-expanded networks be created for those specific itineraries. We present the detailed model, the formulation of the Master Problem and the Scheduling Sub-Problem, in Marla (2007). We also present in Marla (2007) computational experiments in the case of a medium-sized US airline, compare the performance of the traditional integer programming approach and the decomposition approach, and demonstrate tractability improvements.

6 Summary

We described a new decomposition modeling approach for commodity routing with time windows under uncertainty (CRTW-UU). At the core of this approach is our new Extended Chance-Constrained Programming (ECCP) approach, that maximizes constraint protection under a budget. The CRTW-UU problem is decomposed into routing and scheduling modules and solved by iterating repeatedly between them. In spite of multiple iterations being required, the model is tractable because of the Scheduling Sub-Problem consisting of simple labeling and propagation algorithms and being polynomial in complexity. The algorithm is also made more efficient due to Pre-processing and eliminating a large set of infeasible solutions. Our decomposition approach is flexible in modeling different types of uncertainty as well as in modeling uncertainty using partial knowledge of the underlying uncertainty distribution in the form of quantiles. We model uncertainty in vehicle capacities and shipment demands in the routing module using the ECCP and uncertainty in travel or service times in the scheduling module by using a priori chosen higher quantiles of travel times. Computational results on routing and scheduling for a large ground carrier indicate that the output of our decomposition algorithm generates results that are more robust with respect to several metrics, and under uncer-

tainty in both demands and travel times. Our model distributes shipments among vehicles in such a way as to maximize spare capacity, and allocates routes that include buffers in shipment paths to allow for transfers and increased travel time realizations. Moreover, our approach recognizes that movements or flows have associated time-windows of feasibility, and not simply point schedules. Our approach provides a way to efficiently compute these time-windows, which indicate sensitivity of the solution, and to exploit the existence of time-windows to generate robust solutions.

Acknowledgements We sincerely thank the anonymous reviewers for their time in reading our paper and for their valuable comments and suggestions, which significantly contributed to improving the quality of this manuscript.

References

- R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, New York, 1993.
- C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998a.
- Cynthia Barnhart, Christopher A. Hane, Ellis L. Johnson, and Gabriele Sigismondi. A column generation and partitioning approach for multi-commodity flow problems. *Telecommunication Systems*, 3:239–258, 1994. ISSN 1018-4864. URL <http://dx.doi.org/10.1007/BF02110307>.
- A. Ben-Tal and A. Nemirovski. Robust solutions to uncertain programs. *Operations Research Letters*, 25: 1–13, 1999.
- A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424, 2000.
- D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming Series B*, 98:49–71, 2003.
- D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.
- D. Bertsimas and D. Simchi-Levi. A new generation of vehicle routing research: Robust algorithms, addressing uncertainty. *Operations Research*, 44(2):286–304, 1996.
- J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, 1997.

- A. Charnes and W. W. Cooper. Chance constrained programming. *Management Science*, 6(1):73–79, 1959.
- A. Charnes and W. W. Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11(1):18–39, 1963.
- J. Cordeau, G. Laporte, J. Potvin, and M. Savelsbergh. Transportation on demand. *Handbook of Operations Research*, Section 2, 2006.
- J.-F. Cordeau, G. Laporte, J.-Y. Potvin, and M.W.P. Savelsbergh. Transportation on demand. In C. Barnhart and G. Laporte, editors, *Transportation, Handbooks in Operations Research and Management Science, Volume 14*, pages 429–466. Elsevier, Amsterdam, 2007.
- J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. *Time Constrained Routing and Scheduling, Handbook in Operations Research/ Management Science, Network Routing*. M. Ball, North-Holland, Amsterdam, 1995.
- M. Dessouky, R. Hall, A. Nowroozi, and K. Mourikas. Bus dispatching at timed transfer transit stations using bus tracking technology. *Transportation Research Part C: Emerging Technologies*, 7:187–208, 1999.
- M. Dror, G. Laporte, and P. Trudeau. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation Science*, 23(3):166–176, 1989.
- Y. Dumas, J. Desrosiers, and F. Soumis. The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54(1):7–22, 1991.
- M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.
- H. Jiang and C. Barnhart. Dynamic airline scheduling. *Transportation Science*, 43(3):336–354, 2009.
- L.S. Kang and J.P. Clarke. Degradable airline scheduling. Working paper, Operations Research Center, Massachusetts Institute of Technology, 2002.
- S. Lan, J.P. Clarke, and C. Barnhart. Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science*, 40(1):15–28, 2006.
- T. Mahr. Vehicle routing under uncertainty. Master’s thesis, Budapest University of Technology and Economics, 2011.
- L. Marla. Robust optimization for network-based resource allocation problems. Masters thesis, Massachusetts Institute of Technology, 2007.
- L. Marla. *Airline Schedule Planning and Operations: Optimization-based Approaches for Delay Mitigation*. PhD thesis, Massachusetts Institute of Technology, 2010.
- L. Marla and C. Barnhart. A decomposition approach to dynamic airline scheduling. Working paper, 2011.
- J. M. Mulvey, R. J. Vanderbei, and S. A. Zeinos. Robust optimization of large-scale systems. *Operations Research*, 43:34–56, 1981.
- E. Nuutila and E. Soisalon-Soininen. On finding the strongly connected components in a directed graph. *Information Processing Letters*, 49:9–14, 1994.
- F. Ordonez and J. Zhao. Robust capacity expansion of network flows. Technical report, University of Southern California, 2011.
- D. Paraskevopoulos, E. Karakitsos, and B. Rustem. Robust capacity planning under uncertainty. *Management Science*, 37(7):787–800, 1991.
- R.T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41, 2000.
- S. Shebalov and D. Klabjan. Robust airline crew pairing: Move-up crews. Working paper, the University of Illinois at Urbana-Champaign, Available from <http://netfiles.uiuc.edu/klabjan/www>, 2004.
- A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21, 1973.
- I. Sungur, F. Ordonez, M. M. Dessouky, Y. Ren, and H. Zhong. A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science*, 44:193–205, 2010.
- R. Tarjan. Depth first search and linear graph algorithms. *Journal of Computing*, 1(2):146–160, 1972.
- Richard D. Wollmer. Investment in stochastic minimum cost generalized multicommodity networks with application to coal transport. *Networks*, 10(4):351–362, 1980. ISSN 1097-0037. doi: 10.1002/net.3230100406. URL <http://dx.doi.org/10.1002/net.3230100406>.
- D. Wood. An algorithm for finding a maximum clique in a graph. *Operations Research Letters*, 21:211–217, 1997.
- J. Yang, P. Jaillet, and H. Mahmassani. Real-time multivehicle truckload pickup and delivery problems. *Transportation Science*, 38(2):135–148, 2004.