Bus Inf Syst Eng 59(3):135–146 (2017)
DOI 10.1007/s12599-017-0466-4

RESEARCH PAPER

Integrating Data Collection Optimization into Pavement Management Systems

Gabriel Bazi · John El Khoury · F. Jordan Srour

Received: 27 June 2016 / Accepted: 12 January 2017 / Published online: 16 March 2017

Abstract This paper describes a method for using location data to optimize the routing of pavement data collection vehicles. In much of the developed world, pavement testing is performed on a regular basis; the pavement testing data, in turn, serves as input to Pavement Management Systems. Currently, in the United States of America, state departments of transportation plan this data collection work by providing the list of roads that must be tested and then leave the routing of the vehicles to the equipment operators who typically execute the work in an ad hoc manner. This study presents the processes required to code the list of roads for testing, select appropriate hotels in the region of testing, and apply a Traveling Salesman Problem with Hotel Stops model to derive a route. Applying the processes to a case study shows significant cost savings associated with this method of roadway testing, as opposed to the current ad hoc methods.

Keywords Optimized data collection · Route optimization · Pavement management system · Mixed integer program · Traveling salesman problem with hotel stops

1 Introduction

State departments of transportation (DOTs) and public works agencies have been performing road maintenance planning and operational activities since the 1880s (Dror 2000). It wasn't until the 1980s, however, that Information Systems began to play a significant role in the processes of maintenance and planning. Specifically, a foundational article, published in 1982 in the journal *Interfaces*, describes a Pavement Management System (PMS) developed by the State of Arizona that integrates management policy decisions, budgetary policies, environmental factors, and engineering decisions (Golabi et al. 1982). The authors claim savings of $14 million as a result of system use. At the heart of the system is an optimization module that recommends preservation policies to achieve long-term and short-term standards for road conditions at the lowest possible cost. This module relies on pavement condition data as its input and provides a maintenance and rehabilitation (M&R) plan as its output.

Overshadowed by the immense savings that a PMS can achieve for M&R planning, little attention has been paid to the development of pavement assessment plans to gather the input data. Over the years, surveys have shown that most agencies still rely primarily on field experience rather than systematic analysis to conduct roadway condition assessments (Jang 2011). For example, a survey in Minnesota showed that only one agency of 414 jurisdictions used computerized routing software for snow and ice control (Office of the Legislative Auditor 1995).

Accepted after two revisions by the editors of the special issue.

G. Bazi · J. El Khoury
Department of Civil Engineering, School of Engineering, Lebanese American University, P.O. Box 36, Byblos, Lebanon
e-mail: gabriel.bazi@lau.edu.lb

J. El Khoury
e-mail: john.khoury@lau.edu.lb

F. J. Srour (✉)
Department of Information Technology and Operations Management, Adnan Kassar School of Business, Lebanese American University,
P.O. Box 13-5053, Chouran Beirut 1102 2801, Lebanon
e-mail: jordan.srour@lau.edu.lb

The authors surveyed five state highway agencies regarding their use of computerized data driven processes for planning and executing pavement testing. The states of Alaska, California and Florida have no set process to route test vehicles during pavement data collection and they base their routing schemes on their personnel's experience. The state of Minnesota responded to our inquiry as follows: "MnDOT does not have any formal way to route for collecting pavement management data. Data collection is routed to ensure efficient collection during the times of the year that the geographic areas of the state are free of frost effects to the pavement International Roughness Index (IRI). Routing is also based on the locations to be collected that season on local county roads. Local road collection is based on a 3-year cycle". Along the same lines, the state of Wisconsin replied: "We don't use any program for routing our vans for the pavement condition data collection on our highways. The operator simply uses a map and highlights the routes. Two colors are used – one to identify the roads that need data collection in both directions and a different color for those that just need one direction. The operator then estimates distances to determine what can be done in a day. Or, if testing will take longer than a day, then the operator selects a city to stay in and then breaks up the areas all around it into subsections that can be tested in a day". This research seeks to illustrate the value of a holistic system for assimilating roadway data into a procedure for developing optimized route plans.

The remainder of the paper is organized as follows: a literature review describes the history of PMSs and the way that such systems work while highlighting the absence of a routing component in these systems. The following section, Sect. 3, then describes the proposed extension to existing PMS – a method to integrate roadway data with an operational route planner. In this section we also briefly comment on the algorithmic structure of the route planning tool. We then present the details of the case study instances used to validate the proposed system in Sect. 4. These instances originate from real-world data provided by a pavement testing firm in the Southwestern United States. The results of the proposed strategy relative to a manual solution are described in the second to last section. The article concludes with directions for future work.

2 Literature Review

Pavement management involves all activities regarding the planning, design, construction, maintenance, and rehabilitation of pavements. Pavement management systems (PMS) consist of a set of tools or methods to help pavement managers plan for constructing and maintaining pavements in a serviceable condition over a given period of time. In 2002, Tsai and Lai formalized the components of a PMS by putting forward a conceptual IT-based framework (Tsai and Lai 2002). Their framework relies on an operation component, a data component, and a decision support component. The data component is at the heart of the framework and relies heavily on the input of data from the operation component – chief among these inputs is pavement condition data. The data component, in turn, feeds the decision support component that utilizes optimization to plan multi-year M&R budgets. The decision support component then feeds back to the operation component in terms of the M&R plan.

While one of the functionalities of PMSs is to optimize funding choices via network analysis (Medury and Madanat 2013), current systems do not actually consider the operational routing of the pavement data collection vehicles. Highway agencies, such as state departments of transportation (DOTs) or public works agencies, update their PMSs every 3–5 years by performing data collection on their network. Depending on the size of the network and the available budget, the data collection is performed in 1 year or over several years. Furthermore, the highway system pavement inventory of the US is monitored as part of the Highway Performance Monitoring System (HPMS) program in a partnership between the US federal government and the various states.

In most cases, consultants are contracted by the agencies for the data collection. Such a task comprises sending one or more test-vehicles to drive over multiple predefined sections within the network. The number of test sections could be as few as a dozen sections in a small town or thousands of sections in a large city, state or a country. Nevertheless, routing through these test sections can have significant cost implications for the testing process.

While not currently recognized in testing, these cost implications are realized in other pavement management domains such as snow removal and pavement marking (Jang 2011; Perrier et al. 2007a, b; Office of the Legislative Auditor 1995). Problems of snow removal generally require that snow plows traverse all roadways in a given network. In contrast, pavement marking problems require traversal of isolated sections of roadway. In this regard, the domain of pavement marking is closest to our problem where specific, possibly disparate, sections within the network are tested.

In general, several types of testing equipment are typically used to assess the pavement's structural and functional performance. Such equipment includes, among others, falling weight deflectometers (FWDs), road surface profilers (RSPs), pavement imaging vehicles, friction testers, and ground penetrating radars (GPRs). The similarity of test routing to pavement marking is most readily seen with friction testing as the route is limited by the water tank

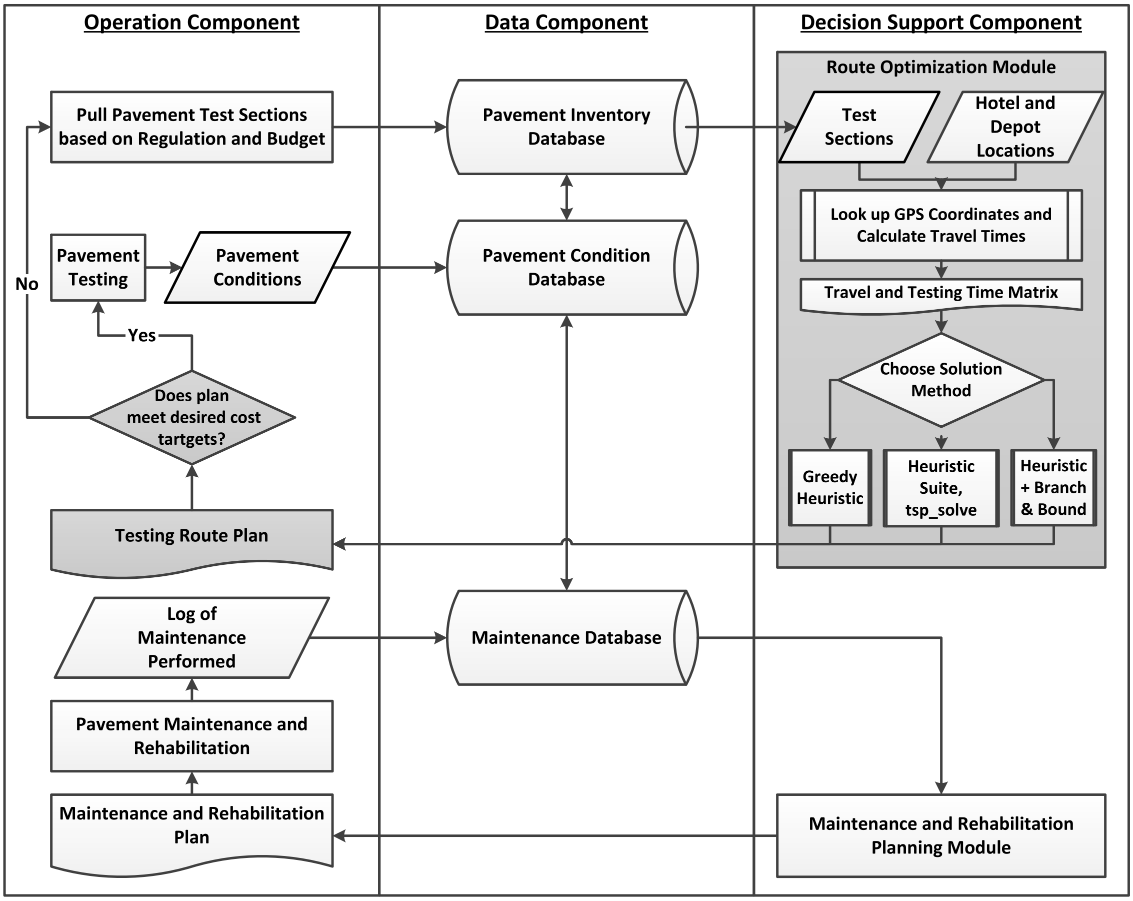
Operation Component
Data Component
Decision Support Component
Pull Pavement Test Sections based on Regulation and Budget
Pavement Inventory Database
Route Optimization Module
Test Sections
Hotel and Depot Locations
Look up GPS Coordinates and Calculate Travel Times
Travel and Testing Time Matrix
Pavement Testing
Pavement Conditions
Pavement Condition Database
No
Yes
Does plan meet desired cost tartgets?
Choose Solution Method
Greedy Heuristic
Heuristic Suite, tsp_solve
Heuristic + Branch & Bound
Testing Route Plan
Log of Maintenance Performed
Maintenance Database
Pavement Maintenance and Rehabilitation
Maintenance and Rehabilitation Plan
Maintenance and Rehabilitation Planning Module

Fig. 1 Flow chart depicting the Route Optimization Module within the PMS and the steps followed by the Route Optimization Module

capacity just as a marking vehicle is limited by the paint tank capacity. The pavement testing route module that we propose, however, focusses on the majority of testing equipment that is not limited by additional constraints such as the water tank capacity in friction testers.

Just as Ralyté et al. (2015) recommend a shift of Information Systems to Information Services Systems, the data integration process that we suggest fits well with existing pavement management information systems. Currently, PMSs rely on the process of pavement testing to generate data on pavement conditions. These conditions are then entered to the pavement condition database. The condition database interfaces with the maintenance database which in turn feeds the maintenance and rehabilitation module. The M&R module then produces, through optimization, an M&R plan. The plan is enacted and the maintenance, as performed, is logged. While this process works well within the current legislation that requires condition testing across the full network at fixed intervals, we believe that the pavement testing activity can be improved through the addition of a route optimization module.

Policies governing pavement testing along with available testing budgets can be used to pull a pavement section test set from the pavement inventory and condition databases. Once the test sections are selected, the route planning module informs the decision maker of the optimal route and the route costs that will be incurred with the selected road section test set. In this way, the decision maker may alter the road section test set to meet cost targets or can solicit bids from pavement testing contractors that abide by the expected cost levels using the optimized route provided by the route planning module. Figure 1 illustrates, by means of a cross functional flow chart, how the route optimization module fits within a general PMS structure; the new, proposed elements are shaded in grey.

3 Integrating Route Planning in PMS

A PMS typically provides a list of the sections to be tested that contains the roadway name, starting and ending post miles or cross streets, length, number of lanes to be tested, and direction of testing. The latitude and longitude coordinates for the starting and ending points are provided if the network is geo-referenced.

For some sections, data collection is required in both directions depending on the functional classification, number of lanes, or other parameters specific to the agency. For example, the data is collected on the slow lanes in both directions if the roadway is a multi-lane divided highway, and in the slow lane in one specified direction if it is a four lane undivided highway.

Once the PMS provides the road section test set, the location data is cleaned and a distance matrix generated. The distance matrix is fed into an optimization module that produces a route plan. The route is then presented to the decision maker in a graphical manner. Figure 1 details, within a larger PMS structure, the steps by which the optimization module works. These steps are also described in the following subsections.

3.1 Data Cleaning and Distance Matrix Generation

The distance matrix generation requires as input a list of the start and end GPS coordinates for every test section. These GPS points may be provided directly or generated using the cross streets or post miles. To obtain the GPS coordinates, if not provided, the cross streets must be entered into an online or offline mapping program. As noted by Zandbergen (2008) modern geocoding software can provide accurate location results for street networks. To expedite the process of geocoding the test section start and end points, one may also use a coded script to lookup sections via a mapping Application Program Interface (API) such as the Google® Maps Geocoding API (Google 2016).

An example of this data processing step is illustrated for a small area in Los Osos, CA that consists of 17 sections. The streets to be tested along with their cross street limits were provided. The first section is Santa Ysabel Avenue from 2nd Street to South Bay Boulevard. The latitude and longitude for the Santa Ysabel Avenue and the 2nd Street starting cross street are 35.330011 and −120.840864, respectively, whereas the latitude and longitude for the Santa Ysabel Avenue and the South Bay Boulevard ending cross street are 35.329901 and −120.823426, respectively. Table 1 provides a summary of the start and end coordinates of the 17 sections.

The route optimization model requires as input a matrix of the travel times between the sections and available hotels as well as the testing times of the sections. The matrix is a square matrix of order n, where n is equal to the total number of sections and hotels. The diagonal entries correspond to the testing time of each section. The matrix is asymmetric since the travel time from the end of one section – section “A” – to the start of another section – section “B” – is not the same as the travel time from the end of section “B” to the start of section “A”. Note if it is desired that testing commence and return to a depot, the depot should also be included as a location in the matrix.

Table 2 shows the travel time and testing time matrix for the 17 Los Osos, CA sections as generated using the Google® Maps Distance Matrix API (Google 2015). The diagonal entries show the testing time for every section. For example, the testing time of section ID 01 from start to end is 2.3 min excluding any equipment set up time. All other entries correspond to the travel times from the end of the sections shown in left column to the start of the

Table 1 GPS Coordinates for Start and End of the 17 Los Osos Sections

Section ID	Combined ID	Start latitude, longitude	End latitude, longitude
01	01	35.330011, −120.840864	35.329901, −120.823426
02	02	35.320657, −120.835452	35.329968, −120.835382
03	03	35.326836, −120.840804	35.330013, −120.840900
04	04	35.326921, −120.840889	35.326234, −120.829943
05	05	35.322597, −120.839650	35.326216, −120.839808
06	06A	35.316917, −120.833195	35.316499, −120.823404
07	06B	35.316499, −120.823404	35.313408, −120.817069
08	07	35.308884, −120.850654	35.308606, −120.833206
09	08	35.308759, −120.839943	35.312204, −120.839881
10	09	35.308844, −120.843170	35.312543, −120.843121
11	10A	35.299483, −120.863120	35.306641, −120.857797
12	10B	35.306641, −120.857797	35.312704, −120.852019
13	11	35.312090, −120.859145	35.311846, −120.853434
14	12	35.312793, −120.851976	35.318427, −120.852078
15	13	35.318472, −120.851962	35.318483, −120.844911
16	14	35.312730, −120.844814	35.320316, −120.844824
17	15	35.308140, −120.856645	35.300710, −120.847791

Table 2 Los Osos travel time/testing time matrix in minutes

Sect. ID	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
01	**2.3**	3.2	3.5	3.5	4.0	3.1	1.4	7.4	6.1	6.4	8.2	6.9	7.9	5.8	6.9	5.1	6.5
02	0.8	**1.8**	2.0	2.1	2.5	3.1	3.1	7.4	6.1	6.5	8.2	6.9	7.9	5.9	5.3	4.9	6.6
03	0.0	2.8	**0.7**	1.1	2.0	4.2	4.0	7.9	7.2	7.3	8.7	7.4	8.4	6.3	5.8	5.4	7.0
04	2.5	2.4	2.7	**2.1**	3.1	2.4	3.1	7.4	6.1	6.4	8.2	6.9	7.9	5.8	5.9	5.1	6.5
05	1.3	1.8	0.4	0.4	**0.9**	3.2	4.6	6.5	6.1	5.9	7.3	6.0	7.0	4.9	4.4	4.0	5.6
06	3.9	2.7	5.0	5.0	3.8	**2.1**	0.0	6.0	4.6	5.0	6.7	5.4	6.5	4.4	5.7	3.7	5.1
07	5.6	4.4	6.7	6.6	5.4	3.3	**1.0**	7.8	6.4	6.8	8.5	7.2	8.3	6.2	7.5	5.5	6.9
08	6.1	3.3	5.6	5.5	4.4	2.8	3.3	**2.6**	1.4	2.0	5.8	4.5	5.5	3.5	4.7	2.8	4.2
09	6.2	3.3	5.4	5.3	4.1	2.9	3.3	3.2	**0.7**	1.9	3.8	2.5	3.6	1.5	2.8	0.8	2.2
10	5.8	3.6	5.0	4.9	3.8	3.2	3.6	2.6	1.9	**0.6**	3.5	2.2	3.2	1.2	2.4	0.5	1.9
11	7.4	5.5	6.6	6.6	5.4	5.1	5.6	3.7	3.9	3.7	**1.5**	0.0	3.0	1.3	2.6	2.1	0.4
12	6.3	4.3	5.5	5.4	4.2	4.0	4.4	2.5	2.7	2.6	2.4	**1.1**	2.1	0.1	1.4	0.9	0.7
13	6.6	4.7	5.8	5.8	4.6	4.4	4.8	2.9	3.1	3.0	2.4	1.1	**0.7**	0.6	1.8	1.4	0.8
14	5.8	3.9	5.0	4.8	3.8	5.3	5.7	3.8	4.1	3.9	3.8	2.5	3.0	**1.1**	0.0	2.3	2.1
15	4.2	2.3	3.4	3.3	2.2	3.7	5.2	4.0	3.5	3.4	4.7	3.4	4.5	2.4	**1.1**	1.5	3.1
16	3.5	1.5	2.7	2.5	1.5	3.0	4.8	4.2	3.8	3.6	5.0	3.7	4.7	2.7	2.1	**1.5**	3.4
17	9.3	7.3	8.5	8.4	7.3	7.0	7.4	5.5	5.8	5.6	4.1	2.8	4.8	3.2	4.4	4.0	**2.1**

sections shown in the top row. For example, the travel time from the end of section ID 01 to the start of section ID 02 is 3.2 min.

A setup time of about 1 min is typically added to the testing time for every section. This time is spent by the operator to mainly create a new file for the data collection and to review the route. Whenever sections are combined for testing in one run, the equipment setup time is reduced to only one setup. This task is performed automatically in the optimization model.

For the purpose of comparison to the manual benchmark, Table 2 shows which sections are adjacent and can be combined. For example, the entry of row 06 and column 07 is zero indicating that sections 06 and 07 can be combined to form section 06A/06B (combined IDs 06A and 06B).

3.2 Route Optimization

After a day of work, the pavement testing vehicle operators must take breaks in a hotel or at the depot. Determining the least cost order of sections to test while balancing the cost of staying in a hotel versus returning to the depot adds a point of synchronization to our model. However, the need to traverse disparate arcs yields a routing problem similar to the rural post man problem (RPP) (Eiselt et al. 1995). The RPP is considered an arc routing problem. However, given the distance between the arcs that must be traversed, such routing problems can be reduced to an asymmetric traveling salesman problem (ATSP) (Srour and van de Velde 2013). Given this formulation, the problem may be modeled as a traveling salesperson problem with hotel selection (TSPHS) (Vansteenwegen et al. 2012; Castro et al. 2013, 2015). To avoid overwhelming the reader with notation, the model is described here only in words – the full mixed integer programming model is shown in Appendix A (available online at link.springer.com).

The objective of the model is to minimize the total cost associated with traveling between jobs, hotels, and the depot, the cost of setting up to test disparate road sections, the cost of staying in hotels, the cost of paying each day's regular wages and the cost of overtime wages. In this formulation a full day of wages are paid if any testing occurred on that day. However, if one prefers to charge for only the time involved in testing or mobilizing on any one day, this is easily done through a slight modification to the objective function.

This objective is subject to the following constraint sets: a set of routing constraints ensuring that all test sections are visited; the test vehicle must leave the depot at least once; no test sections may be left untested; arrival times to all test sections must be consistent in time; test sections must be served during the day; hotels or the depot can only be visited at night; following a trip to a hotel or a depot, the day increments by one.

The heart of this model is based on the well-known TSP, which depends on subtour elimination constraints and can take significant amounts of time to solve. As such, three solution methods are adopted each with different runtimes and tradeoffs in solution quality. First, a simple nearest-neighbor greedy heuristic is used to find a feasible solution quickly (Cormen 2009). This routing strategy begins at the depot and continues adding the closest section to the end of the route until approaching the working day's time limit. At that point, the nearest hotel is inserted. This process continues until all sections are accommodated and the return to the depot is added

Second, the heuristic suite included in Chad Hurwitz's GNU TSP Solver is applied along with a set of swapping and 4-opt improvement schemes (Hurwitz 1994). Specifically, the heuristics included in the suite relevant for the asymmetric TSP are: a nearest addition tour finder, a farthest insertion tour finder, a dispersion strategy, two strategies for patching the solution of the assignment problem into one tour, and a loss heuristic. Within the tsp_solve package, a tool called "heurbest" finds the best heuristic within the suite for the given problem. The solution is returned as a simple ordering of the jobs with no regard for which hotels should be used and when. A post-processing step is undertaken to place the nearest hotel on the route once each day's time limit is near.

Finally, the Gurobi v6.0 (Gurobi Optimization 2015) exact branch-and-bound solver is applied to the model formulation to derive an optimal solution. Since the problem is fundamentally a TSP, we provide the exact solver with a "warm-start" solution as input, which in our case, is the tsp_solve solution. At any time the exact solver is stopped and a solution is returned, that solution is known to be the best solution found up to that point in time

Given these three solution strategies, the user can make important decisions about the tradeoffs between solution time and solution quality. Such tradeoffs are apparent in Sect. 5 where the results of the case studies are presented.

3.3 Presentation of Resulting Route Plan

The optimization model then yields a list indicating the order in which the test sections should be traversed. The GPS coordinates of the sections or combined sections can then be exported to a mapping software for routing during data collection. The map waypoints for the tested sections are deleted at the end of the testing or end of the day. In this way, the operator can immediately see only those sections requiring testing; not those that are complete. Simultaneously, the pavement condition data that was collected can populate the data component of the PMS.

4 Validation Instances

To illustrate the capabilities of the proposed routing optimization, two instances based on real-world data were tested. The optimal routes were benchmarked against the routes generated using the manual process of the firm that collected the data.

The first instance is small and consists of the 17 Los Osos, CA sections. The total combined time to test all of the sections (excluding the drive time) is 23.9 min, which is equal to the sum of the diagonal entries from Table 2. The longest section requires 2.6 min of testing while the shortest section requires 0.6 min. Given the proximity of the sections to each other (the longest inter-section drive time is 9.3 min), all sections may be tested within 1 or 2 h, on one tour from the depot, without hotel stays. The scale of this instance allows for easy verification of the optimization solution.

The second instance is significantly larger, with 349 test sections. These sections were selected out of a larger project that consisted of 936 sections in California's Monterey, Santa Barbara, San Benito, Santa Cruz and San Luis Obispo Counties. A subset of 349 sections were selected for the purpose of this case as they are located in the same geographical area and were tested over 20 consecutive days. Therefore, this subset also serves to provide a manual benchmark for comparison. The total combined time to test all of the sections (excluding the drive time) is 1939 min or 32.3 h. The longest section requires about 1.7 h of testing while the shortest section requires about 3 s. The longest drive time is almost 5 h, which corresponds to the drive time from the depot (location of the equipment) in Ventura, CA to a test section in Santa Cruz, CA.

The 349 sections cannot be tested in one day and the operator needs to stay at hotels during testing. Thirteen potential hotels were selected in all counties with an assumed rate of $100 per night. The selection was based on where the operator typically stays during testing. Out of the 13 potential hotels, 8 were used during actual testing, and some of them were used up to 4 nights.

In this second, larger instance two scenarios were considered – one in which overtime is not allowed and one which allows for overtime. In the scenario without overtime, the daily testing for the optimal route was limited to a maximum of 8 h per day at a wage rate of $85 per hour; for a daily rate of $680. In the overtime scenario, the operator can work up to 12 h per day at a wage rate of $85 for the first 8 h followed by a rate of $127.5 per hour for time worked between 8 and 12 h. The $127.5 per hour overtime rate is one and a half the $85 regular rate.

5 Results

In this section, we compare the results of the manual benchmark from actual practice, the greedy heuristic, the tsp_solve heuristic, and the optimal route obtained using the model with the tsp_solve heuristic as a warm-start. In order to compare the routes on the basis of cost, a testing rate of $180 per hour and a mobilization rate of $110 per

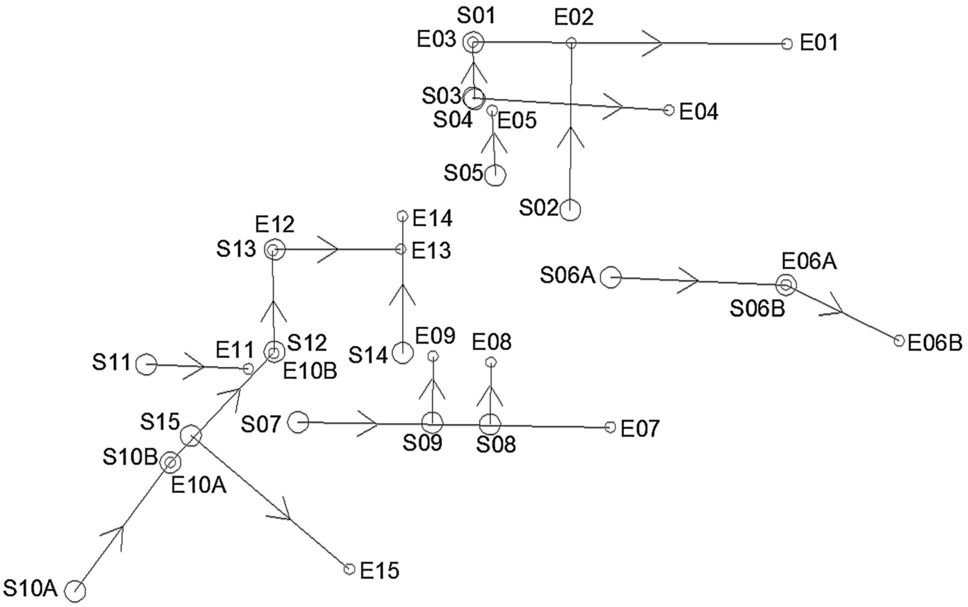

Fig. 2 Test Sections in Los Osos, CA numbered in actual testing order – the manual benchmark

hour were used, in addition to the operator wage rate. Note, that given the size of the instance, hotel stays were not necessary for the first instance, and thus the operator's wages were not charged daily, but were based on a prorated hourly wage of $85 per hour. Adjacent sections avoided an additional setup cost if the end of the first section is within 0.5 min of the start of the second section. The setup time at the beginning of every section or combined section was set to 1 min.

5.1 Instance One (17 Sections)

During the data collection, the operator selected Section 01 as the starting test section and then proceeded, regionally, through to Section 05, before moving to subsequent sections as shown in Fig. 2. The consecutive nature of labeling reflects the order of testing. The route resulting from the manual process placed the test sections in the following order: 01, 02, 03, 04, 05, 06A/06B, 07, 08, 09, 10A/10B, 11, 12, 13, 14 and 15.

In contrast, on a 64-bit server with an Intel Xeon 2.40 GHz processor and 128 GB RAM, both heuristic methods and the exact solver required less than 1 s to yield a route sequence. Given the small size of this instance, there was no depot. Thus, two open (as opposed to closed tour) routes were generated using the optimization – one in which the route must start at S01 (as in the manual solution) and one in which the route may begin at any of the sections; each may end without returning to the initial segment. The optimal route with a forced start at S01 places the test sections in the following order: 01, 11, 15, 10A, 10B, 12, 13, 07, 08, 09, 14, 02, 05, 03, 04, 06A and 06B. If the testing is not forced to start at Section 01, then the optimal route places the test sections in the following order: 11, 15, 10A, 10B, 12, 13, 07, 08, 09, 14, 05, 03, 01, 02, 04, 06A and 06B. The heuristic solutions, both greedy and that found using tsp_solve, follow nearly the same testing order with only a few permutations. Specifically, the greedy solution places section 15 between 10A and 10B in both the fixed and open start scenarios. The tsp_solve solution matches the optimal solution for the open start but permutes, at no additional cost, the single digit sections for the fixed start scenario.

The manual benchmark route required a total time of 73.55 min as shown in Table 3. The optimal solution for the 17 sections, starting at Section 01, required a total time of 63.45 min, for a total savings of 13–14% in both time and cost relative to the Manual Benchmark route. The optimal solution for the same instance when the start is not constrained to Section 01 had a total time of 56.25 min for a total savings of 20–23% in both time and cost relative to the Manual Benchmark. The heuristic solutions outperformed the manual solution, but in the case of the greedy heuristic was more costly than the optimal solution, for both the open and fixed start scenarios

Furthermore, in practice, the data obtained from the field for this instance indicates that testing, using the Manual Benchmark route, started at 13:38 and ended at 15:25 for total testing window of 1 h and 47 min. If the actual testing time of 23.87 min, the calculated mobilization time between the sections of 34.68 min, and the assumed setup time of 15 min are subtracted from 1 h and 47 min, then a

Table 3 Summary of time and cost required to execute the manual and optimal routes in test instance one – 17 Sections

Scenario	Testing		Mobilization		Setup		Total	
	Time (min)	Cost ($)	Time (min)	Cost ($)	Time (min)	Cost ($)	Time (min)	Cost ($)
Manual Benchmark	23.87	105.41	34.68	112.72	15.00	66.25	73.55	284.38
Greedy (S01 Start)	23.87	105.41	34.52	112.18	13.00	57.42	71.38	275.01
tsp_solve (S01 Start)	23.87	105.41	33.20	107.90	12.00	53.00	69.07	266.31
Optimal Route (S01 Start)	23.87	105.41	28.58	92.90	11.00	48.58	63.45	246.89
Savings Opt v. Manual (%)	0.0	0.0	17.6	17.6	26.7	26.7	13.7	13.2
Greedy (Open Start)	23.87	105.41	24.52	79.68	11.00	79.68	59.38	233.67
tsp_solve (Open Start)	23.87	105.41	22.38	72.75	10.00	44.17	56.25	222.32
Optimal Route (Open Start)	23.87	105.41	22.38	72.75	10.00	44.17	56.25	222.32
Savings Opt v. Manual (%)	0.0	0.0	35.5	35.5	33.3	33.3	23.5	21.8

total of about 33 "unaccounted for" minutes remains. This represents 31% of the testing window that was consumed in determining the route sequence, performing additional setup tasks, taking breaks, having longer mobilization times, or on other unknown factors. When the optimal route is pre-determined, there is no longer a need to spend time in the field determining where the test operator should go next. As such, these 33 min would be reduced representing greater data collection efficiency.

While savings of more than 20% is impressive, in such a small instance it represents only a $62.06 savings in absolute terms. Thus, illustrating the capabilities of the optimized routing solution on a larger instance is necessary.

5.2 Instance Two (349 Sections)

The results of the routing strategies on the instance with 349 sections are shown in Table 4. Due to the size of this instance, the solutions provided by the exact solver are not provably optimal. Yet, they are the best known solutions found after seeding the Gurobi branch-and-bound solver with the tsp_solve solution and allowing for 1 h of runtime. Since the route planning optimization module will be used for planning purposes rather than dynamic routing advice, it is acceptable to allow the optimization to run for 1 h It is infeasible to show the details of the routes given the size of this instance.

In contrast to the smaller instance, the costs in Instance Two include hotel costs and therefore require a slightly different wage structure for the purpose of studying the impact of overtime on costs. As such, a testing rate of $180 per hour and a mobilization rate of $110 per hour were used. Any day in which testing occurred, no matter how much or little, a cost of $680 is incurred as wages. Any testing and mobilization that ran beyond the standard 8 h was charged at a pro-rated cost of $127.5 per hour. As in the first instance, adjacent sections avoided an additional

Table 4 Summary of Time and Cost Required to Execute the Manual and Optimal Routes in Test Instance Two – 349 Sections

Scenario	Mobilization		Setup		Hotel Stays		Wages		Total	
	Time (min)	Cost ($)	Time (min)	Cost ($)	# of Nights	Cost ($)	Time (hrs)	Cost ($)	Prod. Time (hrs)	Cost ($)
Manual Benchmark	3778	6926	251	753	19	1900	160	13,600	99.5	28,996
Greedy, No OT	3443	6313	289	867	12	1200	104	8840	94.5	23,240
tsp_solve, No OT	2488	4562	247	741	10	1000	88	7480	77.9	19,600
Gurobi, No OT	2454	4499	234	702	10	1000	88	7480	77.1	19,498
Savings Gurobi v Manual (%)	35.0	35.0	6.8	6.8	47.4	47.4	45.0	45.0	22.5	32.8
Greedy, OT	3760	6893	287	1268	8	800	72	6120	99.8	38,888
tsp_solve, OT	2543	4663	248	744	7	700	64	5440	78.8	19,803
Gurobi, OT	2523	4626	240	720	7	700	64	5440	78.3	19,136
Savings Gurobi v Manual (%)	33.2	33.2	4.4	4.4	63.2	63.2	60.0	60.0	21.3	34.0

For all scenarios the testing time is 1939 min at a cost of $5817

Table 5 Sensitivity analysis of solution quality and solve time relative to problem size

Instance		Benchmarks		tsp_solve				tsp_solve + Gurobi, 1 h			
Jobs	Hotels	AP	Best	Soln.	% AP	% Best	Time (s)	Soln.	%AP	% Best	Time to best (s)
20	0	194	225*	225	16	0	0.01	225	16	0	0
25	1	267	396	467	75	18	0.00	396	48	0	19
30	1	332	436	437	32	0.23	0.01	436	31	0	112
40	2	413	535	543	31	1	0.02	535	30	0	126
80	4	630	840	936	49	11	0.24	840	33	0	462
160	6	949	1335	1415	49	6	4.68	1335	41	0	3132

Solutions reported expressed in time and include only setup and mobilization times

* Optimal solution

setup cost if the end of the first section is within 0.5 min of the start of the second section. The setup time at the beginning of every section or combined section was set to 1 min.

Just as in the smaller, 17-section instance, the cost savings are remarkable: more than 30% in both time and cost. Of note is the fact that both of the heuristic solutions with overtime performed worse, in terms of total costs, than their no overtime counterparts. This is due to the myopic placement of hotel stays which tend to force too much overtime relative to any potential savings in hotel costs. The Gruobi improved route with overtime, however, yields a nearly 2% improvement in total costs relative to the no overtime case

Furthermore, there are significant gains relative to productive time. Specifically in the Manual Benchmark, the labor paid for was 160 h, while the total mobilization and testing time amounted only to approximately 100 h. Thus, the employees were only engaged in productive work about 63% of the time. In contrast, in the Best Route with No Overtime the employees were paid for 88 h of work and were productive for 77.1 h or 88% of that time. When allowing overtime, the employees were paid for 78 h of work (64 regular and 14 overtime hours) and were productive for 78 h or 100% of that time.

5.3 Solution Quality Versus Solve Time

As the primary goal of this pavement testing route planner is to provide useable routes with less cost than current practice, the solution method should not require a long runtime. To gain more insight into the trade-off between the problem size, the solve time, and the solution quality, a set of six test instances were designed by randomly selecting test sections from the largest instance of 349 sections. These instances were then solved using both the tsp_solve strategy and Gurobi seeded with the tsp_solve solution. The solution strategies were each allowed to run for 1 h (although tsp_solve yields a solution much sooner) at which point the best solution was returned. This solution was subsequently compared to two benchmarks – the lowest bound established by solving an assignment problem (AP) version of the underlying ATSP and the best known solution. Table 5 presents the results of these tests

As shown in Table 5, the heuristic solution found by using the tsp_solve package does tremendously well achieving a solution over which Gurobi yields only moderate improvements, ranging from 0 to 18%, after 1 h of solve time. The gains found by applying Gruobi can be seen in the gap relative to the AP solution – ranging from an improvement of 27% over the gap found with the heuristic solution for the problem with 25 jobs to a more modest improvement of 0% in the smallest problem with 20 jobs. Interestingly, the gap relative to the AP solution is not strictly monotonic with regards to problem size; however, there is a generally increasing trend. Thus, as expected, the larger the problem, the larger the gap relative to the AP lower bound and the longer the solve time – for both strategies. Nevertheless, the heuristic strategy shows good results and can be used to gain reasonable solutions to this problem within reasonable times in practice

6 Discussion and Future Work

In this paper, a method for extending PMSs to include the design of pavement testing routes is presented. The proposed optimization model requires, as input, the cross street or mile post designations for the start and end points of each segment in the test set. This data is already contained in the data component of PMSs. For more advanced PMSs, the GPS coordinates may also be available. These data points along with a set of viable hotels serve as the input to the optimization. This method, based on the TSP, yields solutions that are around 33% better than solutions generated applying the best-practices currently used in the field.

The proposed model can be easily used by any highway agency or consulting firm to estimate the costs associated with pavement data collection. By standardizing the route selection with the proposed model, the differences in project costs during the bidding phase will be mainly dependent on the consulting firms' rates. Furthermore, the proposed model in contrast to the current strategy allows the "unaccounted for" time to become "accounted for". Specifically, the agency can now assume that the unaccounted time is not associated with the process of finding routes on-the-fly in the field, but is rather attributable to rest stops, setting-up equipment, and maintaining equipment.

Future work includes properly integrating the proposed route planning module into the decision support component of an existing PMS. Specifically, relying on the network analysis modules already included in most PMSs, a set of eligible test sections could be automatically identified and fed to the route planning module. These data services could all occur within the decision support component of the PMS framework as specified by Tsai and Lai (2002). The route planning module could then provide its output – the optimized route plan – to the operation component. The operation component, once the route had been executed, would then automatically feed its data back to the historical pavement condition data module.

In order for this integration to work smoothly, the route optimization module must be able to handle all problem sizes efficiently. Typical pavement data collection projects include as many as one or two thousand sections that are spread throughout the network. The largest instance that was solved using an off-the-shelf branch-and-bound algorithm seeded with a heuristic solution involved 349 sections and produced a good, but not necessarily optimal solution. Future work includes further developing the heuristics used here to warm start the exact solver with the goal of producing good solutions to the larger instances more efficiently. Given the improvement over the current practice seen in this work, the need for an optimal solution is not as critical as the need for rapidly arriving at a good solution. Rapid solution generation becomes even more important when the real-time dynamics of roadway congestion are considered.

The driving time matrix between the sections, the depot, and the hotels was populated using the Google® Maps distance matrix API. The generated matrix excludes any delays based on current traffic conditions or weather related events. During real-time data collection, traffic conditions could have a significant effect on the route selection and the overall project costs. As such, future work includes the development of a dynamic optimization technique that can be used during data collection as a means to incorporate real-time traffic updates.

Finally, certain pavement data collection equipment, such as the friction testers, require water refill after a certain number of test points or tested lane miles depending on whether the testing is discrete or continuous. The water refill locations are typically located at the DOT's maintenance yards. For other projects, a water tanker follows the friction tester or waits nearby the testing location for refill. In this regard, future work requires an extension of the model to simultaneously capture the need to visit fixed water refill locations based on the capacity of the friction testers and the need to select hotel stops based on hours of service.

Appendix A: Mathematical Formulation of Route Optimization Problem

In the language of graph theory, the authors define a graph G comprised of arcs, A, vertices V, and edges E. The arcs must be traversed, while the edges may, optionally, be used as necessary to travel between non-adjacent arcs. This relatively straight forward problem is complicated by the fact that drivers are bound by hours-of-service regulations that require a rest period after a period of 8 h of work. If overtime work is allowed, then rest is permitted after a minimum of 8 h and before a maximum of 12 h. Then, the drivers may either use a hotel or return to the home city of the testing agency to rest for the evening. Determining the tours through all arcs that minimize costs, which are a combination of drive time and hotel stays, is the objective of this problem.

To achieve this, the arcs (the sections that must be tested) are reduced to nodes. A node for the depot and nodes for the hotels that may be used within the route are also specified. Multiple copies of the depot and hotel nodes may be included in the model to allow for multiple trips to/from the depot and multiple stays at a specific hotel. Then the problem is formulated as a routing problem in which the goal is to construct a set of least cost cycles passing through all nodes. Given this problem description, the following notation for the parameters is designated.

D the set of depot copies included to permit as many trips home as necessary

H the set of hotels (including copies of hotels as necessary)

J the set of jobs (i.e. sections for testing)

N the set of all nodes, which is: $D \cup H \cup J$

t_{ij} the time required to travel from node i to node j

F the set of sections that are farther than a set threshold, θ, away. Mathematically, $F = \{(i,j) | t_{ij} \geq \theta\}$. Section pairs that are more than θ units apart require additional equipment setup time upon arrival

s_j the time required to setup the equipment to test a section $j \in J$
c_h the cost of staying in hotel $h \in H$
μ the cost of mobilization – that is the cost per unit time to travel to a node in the test equipment, excluding operator wages
τ the cost of testing – that is the cost per unit time to setup or test a segment, excluding operator wages
ω the operator's wages per unit time when not in overtime
Ω the operator's wages per unit time when in overtime
W the length of the working day; in our case, 8 h or 480 min
σ the allowable amount of overtime; may be 0 (indicating no overtime allowed) up to 4 h
M a large number set to be $W + 2 \cdot \max_{i,j}\{d_{ij}\}$

Given the problem of interest, the following three variables are specified:

x_{ij} a binary variable indicating whether arc (i, j) is used in the final routing; $i,j \in N$
δ_i a continuous variable designating the time of arrival at the location of node $i \in N$
η_i an integer variable designating the day on which a node $i \in N$ is used

Using the notation described above, a MIP is formulated as follows:

$$\min \mu \sum\nolimits_{i \in N} \sum\nolimits_{j \in N} t_{ij} x_{ij} + \tau \sum\nolimits_{(i,j) \in F} s_j x_{ij} + \sum\nolimits_{i \in J} \sum\nolimits_{h \in H} c_h x_{ih} + \omega W \sum\nolimits_{i \in J} \sum\nolimits_{j \in D \cup H} x_{ij} + \Omega \sum\nolimits_{i \in D \cup H} (\delta_i - W) \tag{1}$$

such that

$$\sum_{j \in N} x_{ij} = 1, \quad \forall i \in N \tag{2}$$

$$\sum_{i \in N} x_{ij} = 1, \quad \forall j \in N \tag{3}$$

$$\sum_{i \in J} x_{0i} = 1, \quad \text{where 0 represents the first depot node} \tag{4}$$

$$\sum_{i \in J} x_{ii} = 0 \tag{5}$$

$$\sum_{j \in N} t_{ij} x_{ij} \le \delta_j, \quad \forall i \in D \cup H \tag{6}$$

$$\delta_j - (\delta_i + d_{ii} + d_{ij}) - M x_{ij} \ge -M, \quad \forall i \in J, j \in N \text{ excluding } i,j \in F \tag{7}$$

$$\delta_j - (\delta_i + d_{ii} + d_{ij} - s_j) - M x_{ij} \ge -M, \quad \forall (i,j) \in F \tag{8}$$

$$0 \le \delta_j \le W + \sigma - d_{jj}, \quad \forall j \in J \tag{9}$$

$$W \le \delta_i \le W + \sigma, \quad \forall i \in H \cup D \tag{10}$$

$$\eta_i - \eta_j + M x_{ij} \le M, \quad \forall i \in N, j \in J \tag{11}$$

$$\eta_i - \eta_j + 1 + M x_{ij} \le M, \quad \forall i \in J, j \in H \cup D \setminus 0 \tag{12}$$

$$\eta_i - \eta_j + M x_{ij} \le M, \quad \forall i \in H \cup D \setminus 0, j \in H \cup D \setminus 0 \tag{13}$$

$$\delta_i \in \mathbb{R}^+, \quad \forall i \in N \tag{14}$$

$$\eta_i \in \mathbb{R}^+, \quad \forall i \in N \tag{15}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i,j \in N \tag{16}$$

In words, the objective of this model, in Eq. (1), is to minimize the total cost associated with traveling between jobs, hotels, and the depot, the cost of setting up to test disparate road sections, the cost of staying in hotels, the cost of paying each day's regular wages and the cost of overtime wages. In this formulation a full day of wages are paid if any testing occurred on that day. However, if one prefers to charge for only the time involved testing or mobilizing on any one day, the term ωW may be replaced by ωt_{ij} within the fourth summation of the objective. This objective is subject to the following constraints:

Equation (2) Each job and hotel/depot node must have one and only one arc leaving.
Equation (3) Each job and hotel/depot node must have one and only one arc entering.
Equation (4) The first copy of the depot node must have one route departing from it to a job. (All other copies of the depot node may be left unused.)
Equation (5) No jobs can be rejected.
Equation (6) If node j is the first node assigned following a stay in hotel/depot i, then the arrival time to j (δ_j) must be later than the time required to travel from the hotel/depot to the starting location of demand j.
Equation (7)–(8) If node j follows job i then the arrival time to node j must be later than the arrival time to job i plus the time required to serve job i plus the time required to travel between job i and node j; if, however, $x_{ij} = 0$, then the arrival time to job j is unconstrained. Furthermore, if the sections are within θ of each other, then the setup time may be ignored.
Equation (9) The arrival time to job j must be during the working day and at least before the time that serving job j would no longer be feasible with allowable overtime of σ
Equation (10) The arrival time at a hotel or back at the depot must be after the end of the working day and before a point where it would be considered overtime.

Equations (11)–(13) The day increases by one when a hotel or depot is used.
Equation (14) δ_i is a positive real number.
Equation (15) η_i is a positive real number.
Equation (16) x_{ij} is binary.

References

Castro M, Sörensen K, Vansteenwegen P, Goos P (2013) A memetic algorithm for the travelling salesperson problem with hotel selection. Comput Oper Res 40(7):1716–1728

Castro M, Sörensen K, Vansteenwegen P, Goos P (2015) A fast metaheuristic for the travelling salesperson problem with hotel selection. 4OR 13(1):15–34

Cormen TH (2009) Introduction to algorithms. MIT press, Cambridge

Dror M (2000) Arc routing: theory, solutions and applications. Springer, Heidelberg

Eiselt HA, Gendreau M, Laporte G (1995) Arc routing problems, part I: the Chinese postman problem. Oper Res 43(2):231–242

Golabi K, Kulkarni RB, Way GB (1982) A statewide pavement management system. Interfaces 12(6):5–21

Google (2015) Google® maps distance matrix api. [API] available from: https://developers.google.com/maps/documentation/distance-matrix/. Accessed 25 Jan 2017

Google (2016) Google® maps geocoding api. [API] available from: https://developers.google.com/maps/documentation/geocoding/start. Accessed 25 Jan 2017

Gurobi Optimization I (2015) Gurobi optimizer reference manual. http://www.gurobi.com. Accessed 13 Mar 2017

Hurwitz C (1994) Gnu tsp_solve v. 1.3.6. [software]. Available from: http://www.cs.sunysb.edu/~algorith/implement/tsp/implement.shtml. Accessed 25 Jan 2017

Jang W (2011) Optimizing winter/snow removal operations in MoDOT St. Louis District - includes outcome based evaluation of operations. Missouri Department of Transportation, Missouri

Medury A, Madanat S (2013) Simultaneous network optimization approach for pavement management systems. J Infrastruct Syst 20(3):04014010

Office of the Legislative Auditor (1995) Snow and ice control. a best practices review. Report no. 95-06. Tech. rep., Minnesota Department of Transportation http://www.auditor.leg.state.mn.us/ped/pedrep/9506-all.pdf. Accessed 13 Mar 2017

Perrier N, Langevin A, Campbell J (2007a) A survey of models and algorithms for winter road maintenance. Part III: vehicle routing and depot location for spreading. Comput Oper Res 34:211–257

Perrier N, Langevin A, Campbell J (2007b) A survey of models and algorithms for winter road maintenance. Part IV: vehicle routing and fleet sizing for plowing and snow disposal. Comput Oper Res 34:258–294

Ralyté J, Khadraoui A, Léonard M (2015) Designing the shift from information systems to information services systems. Bus Inf Syst Eng 57(1):37–49

Srour FJ, van de Velde S (2013) Are stacker crane problems easy? A statistical study. Comput Oper Res 40(3):674–690

Tsai YJ, Lai J (2002) Framework and strategy for implementing an information technology-based pavement management system. Transp Res Rec 1816:56–64. doi:10.3141/1816-07

Vansteenwegen P, Souffriau W, Sörensen K (2012) The travelling salesperson problem with hotel selection. J Oper Res Soc 63(2):207–217

Zandbergen PA (2008) A comparison of address point, parcel and street geocoding techniques. Comput Environ Urban Syst 32(3):214–232