Transactions in GIS

Research Article

Transactions in GIS, 2014, 18(2): 286–301

A Comparative Analysis of Traveling Salesman Solutions from Geographic Information Systems

Kevin M. Curtin,* Gabriela Voicu,† Matthew T. Rice* and Anthony Stefanidis*

**George Mason University*

†City of McKinney Engineering Department

Abstract

The Traveling Salesman Problem is one of the most prominent problems in combinatorial optimization, and is regularly employed in a wide variety of applications. The objective of this article is to demonstrate the extent of sub-optimality produced by Traveling Salesman solution procedures implemented in the context of Geographic Information Systems and to discuss the consequences that such solutions have for practice. Toward that end, an analysis is made of Traveling Salesman solutions from implementations in four Geographic Information System packages. These implementations are tested against the optimal solution for a range of problem sizes. Computational results are presented in the context of a school bus routing application. This analysis concludes that no Traveling Salesman implementation in GIS is likely to find the optimal solution when problems exceed 10 stops. In contrast, optimal solutions can be generated with desktop linear programming software for up to 25 cities. Moreover, one GIS implementation consistently found solutions that were closer to optimal than its competitors. This research strongly suggests that for applications with fewer than 25 stops, the use of an optimal solution procedure is advised, and that GIS implementations can benefit from the integration of more robust optimization techniques.

1 Introduction

The Traveling Salesman Problem (TSP) is one of the most prominent problems in combinatorial optimization, and at the same time a quintessential applied spatial-analytic challenge. The straightforward way in which the problem is defined in combination with its notorious difficulty has stimulated many efforts to find efficient solution procedures (Hoffman and Padberg 2001). The TSP is a classic tour problem in which a hypothetical salesman must find the most efficient sequence of destinations in a territory, stopping only once at each, while returning at the end of the tour to the initial starting location. Due to the combinatorial complexity of the TSP, approximate or heuristic solution procedures are frequently employed in practice.

Since so many problems can be structured as a TSP on a network, and since Geographic Information Systems (GIS) have proven to be extremely efficient for modeling networks (Curtin 2007) and for transportation applications (Simkowitz 1988; Niemeier and Beard 1993; Miller and Shaw 2001), heuristics for the TSP have been widely implemented in GIS. However, there are many heuristics for the TSP with different characteristics, and – understandably – software vendors do not specify the details of the heuristics that they have implemented. Given that heuristic methods will sometimes give sub-optimal solutions are the norm, vendors may not even mention that a heuristic is employed at all. When this is the case, casual GIS users may be completely unaware of the consequences of employing heuristic solution procedures, and the possibility that they will misinterpret the nature of the solutions is significant.

Address for correspondence: Kevin M. Curtin, Geography and GeoInformation Science, George Mason University, 4400 University Drive (MS6C3), Fairfax, VA 22030, USA. E-mail: curtin@gmu.edu

doi: 10.1111/tgis.12045

This research seeks to reveal the quality of the solutions provided by the heuristics employed by various GIS software products, and to perform a comparative analysis of those implementations. In seeking the answers to these research questions this article reflects an attempt to provide a better understanding of the nature of TSP heuristics in GIS, and to allow for more informed decision making regarding the selection of solution procedures for the TSP. In order to do so, the following section briefly reviews the TSP itself (including a mathematical formulation of the problem), solution procedures for the TSP, and the implementations of TSP solution procedures within GIS. The remainder of the article is devoted to a test of GIS-TSP implementations found in several popular GIS software packages. The data for this test is based on a case study of the school bus routing needs for Kimball Elementary School, in Mesquite, TX as well as a larger street network surrounding this school for larger instances of the problem. Computational results are presented and the research concludes with a discussion of the performance of the varying TSP implementations, the consequences of using GIS for solving the TSP, and the potential for improvements to GIS implementations.

2 Theoretical Background

The literature pertinent to this research exists in three related areas: the TSP itself, solution procedures for the TSP, and implementations of the TSP in GIS software. Each of these areas is reviewed in turn. However, the literature surrounding the TSP with its applications and solution procedures is enormous and this is not the appropriate forum for reviewing the entirety of that literature. Rather, this review focuses on the TSP as a problem that can be solved in the context of GIS.

2.1 The Traveling Salesman Problem

2.1.1 A brief history of the TSP

The TSP is, in one sense, the most pedestrian of optimization problems in that travelers solve this problem for themselves every day. Anyone with more than a single stop from home to work and home again (e.g. school, shopping, bank, restaurant, entertainment, etc.) must decide which stops to visit, in which order, in such a way as to minimize their total cost, however they choose to measure that cost. Frequently cost is measured by distance or time. The TSP is pedestrian in another sense, in that a micro-geographic version of the problem must be solved by anyone lacing their shoes (Punnen 2002). For those whose livelihoods depend (at least in part) on efficiently conducting tours of locations with least cost – the traveling salesmen themselves – the problem is of particular interest. In the early 1800s this interest engendered a manual for salesmen providing suggested tours through Germany (Schrijver 2005). Modern delivery systems must solve some version of this same problem on a range of geographic scales.

Beyond the colloquial use of this problem, the TSP has its mathematical origins in the Knight's Tour problem (versions of which date to antiquity) formally discussed by L. Euler and A.T. Vandermonde in the mid-1700s (Hoffman and Wolfe 1985). In the mid-1800s, the problem was identified as an element of graph theory and was studied by the Irish mathematician, Sir William Rowan Hamilton. The problem of visiting every vertex in a graph only once in a closed cycle (a Hamiltonian cycle) retains his name (Wilson 1996). At about the same time, the problem was also identified by the mathematician Thomas Penyngton Kirkman (Hoffman and Wolfe 1985).

The more modern history of the TSP begins with the mathematician and economist Karl Menger (Applegate et al. 1998), who recognized both the combinatorial complexity of the problem, and that a nearest neighbor solution would not generally result in the optimal solution to the problem (Gutin and Holloway 2004). It is believed that Menger introduced the problem to Hassler Whitney at Harvard (Schrijver 2005), who a few years later, presented the problem at Princeton University. It was at Princeton where A.W. Tucker and Merrill Flood discussed the problem in the context of Flood's New Jersey school-bus routing study (Flood 1956). Flood went on to popularize the TSP at the RAND Corporation in Santa Monica, California in late 1940s and beyond (Hoffman and Wolfe 1985; Schrijver 2005).

Significant progress on the TSP was made in 1954 when Dantzig et al. (1954) introduced a new method for solving the TSP, the cutting plane method, which became a prototype in integer linear programming. Since that time the TSP has been considered one of the classic models in combinatorial optimization, and is used as a test case for virtually all advancements in solution procedures. For further reading on the background of the TSP the reader is directed to an annotated bibliography of TSP surveys, applications, and solution procedures produced by Junger et al. (1997).

2.1.2 A mathematical formulation of the TSP

There are many mathematical formulations for variants of the TSP, employing a variety of constraints that enforce the requirements of the problem (Gutin and Punnen 2002). Although any of the common optimization formulations would suffice for this article, one has been chosen (Vajda 1961) in order to demonstrate how such a formulation is specified, and for use in the comparative analysis that follows. The following notation is used:

n = the number of stops to be visited; the number of nodes in the network
i, j, k = indices of stops that can take integer values from 1 to n
t = the time period, or step in the route between the stops
x_{ijt} = 1 if the edge of the network from i to j is used in step t of the route, 0 otherwise
d_{ij} = the distance or cost from stop i to stop j

A standard version of the problem requires starting from a given place, visiting subsequent stops, and returning to the starting place. The optimal solution is one that minimizes the total distance traveled. The objective function (Z) is then to minimize the sum of all costs (distances) of all of the selected elements of the tour:

$$\text{Minimize} \quad Z = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{t=1}^{n} d_{ij} x_{ijt} \tag{1}$$

The tour is subject to the following constraints.

Since the traveler cannot travel between more than one pair of stops at one time, for all values of t, exactly one arc must be traversed, hence:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ijt} = 1 \text{ for all } t \tag{2}$$

For each stop, i, there is just one other stop which is being reached from it, at some time, hence:

$$\sum_{j=1}^{n}\sum_{t=1}^{n} x_{ijt} = 1 \text{ for all } i \tag{3}$$

For all stops, there is some other stop from which it is being reached, at some time, hence:

$$\sum_{i=1}^{n}\sum_{t=1}^{n} x_{ijt} = 1 \text{ for all } j \tag{4}$$

When a stop is reached at time t, it must be left at time $t + 1$, in order to exclude disconnected sub-tours that would otherwise meet all of the above constraints. These sub-tour elimination constraints are formulated as:

$$\sum_{i=1}^{n} x_{ijt} = \sum_{k=1}^{n} x_{jkt+1} \text{ for all } j \text{ and } t \tag{5}$$

In addition to the above constraints the decision variables must be integers taking only the value 0 or 1:

$$x_{ijt} = 0, 1 \text{ for all } i, j, \text{ and } t \tag{6}$$

If n is the number of stops to be visited, there are $(n\text{-}1)!$ possible routes. As the number of stops increases the computational time one would need to evaluate all possible tours among the stops quickly creates a problem that is difficult to solve. The TSP has been proven to be an NP-complete combinatorial optimization problem (Bodin et al. 1983; Hoffman and Padberg 2001). Although specific large instances of the TSP have been solved with a combination of robust solution procedures and a great deal of computing power, the general difficulty in determining optimal solutions to even modest sized instances of the TSP is the reason why heuristic solution procedures are employed in GIS applications. The serious consequences of employing those heuristics (as shown below) are a primary motivation for this article.

2.2 Solution Procedures for the TSP

Given the combinatorial complexity of the TSP and the ubiquitous nature of the problem itself, efforts to develop solution procedures for the problem are ongoing. These efforts fall broadly into two categories: (1) exact; and (2) approximate approaches.

2.2.1 Exact solution procedures

Exact approaches to solving the TSP guarantee optimal solutions, but – due to the combinatorial complexity of the problem – they can generally only be successfully used for modestly sized problem instances. The most obvious exact solution procedure is to evaluate all possible combinations of network elements that meet the constraints and to choose the set that performs best. This method is termed complete enumeration and a program designed to perform such an enumeration was developed for this research for comparative purposes. It was found that the complete enumeration of all possible solutions to determine the optimal solution could not solve the TSP for more than 10 cities (Table 2).

Exact linear programming solution procedures generally use some form of the Simplex algorithm to find optimal solutions to relaxations of the problem that do not require integer solutions. Since integrality is required in practice, a search procedure uses the non-integer solution as an upper bound and proceeds to search through possible integer solutions to define the lower bound. If the upper and lower bound can be made to coincide, optimality is achieved. The branch and bound search procedure is perhaps the most common method to determine integer optimality (Balas and Guinard 1979; Evans and Minieka 1992; Skiena 2008). For this research, optimal solutions were determined through integer programming with the mathematical formulation of the TSP given above. Optimal solutions to problems of up to 25 cities could be determined with industry standard linear programming software on a desktop computer (Table 2). Research is ongoing to determine ways of optimally solving larger TSP instances (Applegate et al. 2003). For those who wish to use code designed specifically for the TSP, software is available that can solve significantly larger problems. However, in the current user environment, there are many GIS users who are choosing to solve the TSP within the GIS environment. It is likely that some – if not most – casual GIS users are unaware that the TSP solutions generated by their software are heuristic and are therefore susceptible to providing sub-optimal solutions. Moreover, no research to date has documented the extent to which the solutions produced through heuristic, GIS-based, solution procedures are sub-optimal.

2.2.2 Approximate solution procedures – heuristics

In this context heuristic solution procedures are approximate approaches that construct feasible solutions within a reasonable amount of computing time (Gendreau 2003). Although heuristic approaches may find the optimal solution for particular problem instances – perhaps even frequently – they cannot guarantee optimality. The two most important criteria when evaluating heuristics are: (1) speed, meaning the total computational time, or in some cases the number of iterations required to reach a solution; and (2) performance with respect to the optimal solution. This latter criterion can be measured in several ways, but is frequently expressed as a percentage over (or under for maximization problems) the optimal solution. This can be complemented by a statement of how frequently the heuristic finds the optimal solution, or with measures of the best case, worst case, and average performance of the heuristic. A heuristic is considered "good" if the number of elementary computational steps is bounded by a polynomial in the size of the problem (Lawler 2001).

Although there is a wide variety of heuristics that can be applied to the TSP, and while new variants are regularly developed (Marinakis et al. 2005, 2008), they cannot all be reviewed here. Instead, three heuristics are discussed: the Nearest Neighbor heuristic as an example of a construction heuristic, a generic description of improvement heuristics, and the TABU search method as a metaheuristic superimposed on an improvement heuristic.

The Nearest Neighbor heuristic is the simplest and most intuitive type of heuristic for determining solutions to the TSP. This heuristic is considered to be "greedy" since it gradually constructs a tour by repeatedly selecting the least-cost edge to cities not already in the tour, and adding that edge to the tour until all cities are reached. The Nearest Neighbor heuristic is extremely fast and easy to implement, but it does not provide particularly good solutions for even modest sized problems.

Improvement heuristics start with an initial feasible solution and successively improve it through a sequence of exchanges (Gutin and Holloway 2004). The initial solution for an improvement heuristic may in fact be chosen by means of the Nearest Neighbor heuristic, since its solution would not violate any of the problem's constraints and since it very quickly determines this feasible solution. Once an initial feasible solution is generated an improved solution is sought by some transformation, generally an exchange of edges in the tour. The most common ways to improve an initial tour generated by construction heuristics are the two-optimal (2-opt) and three-optimal (3-opt) local searches. The 2-opt algorithm exchanges two edges within the tour and tests to see if the solution is improved. The 3-opt algorithm works similarly with exchanges of three edges. The Lin-Kernighan heuristic uses a more complex edge exchange procedure where the number k of the edges to be exchanged is variable (Voudouris and Tsang 1999).

Improvement heuristics are universally plagued by local optima. That is, from either the initial solution or from some subsequently improved solution it may be the case that no permitted exchange can improve the solution, yet the global optimum has still not been reached. If an improvement heuristic becomes trapped in such a local optimum, it will not be able to continue searching for the global optimum unless additional intelligence is provided. Such intelligence comes in the form of a metaheuristic. A metaheuristic is a strategy that guides other heuristics in the search for improved solutions (Black 2009). Although there are many metaheuristics (e.g. simulated annealing, genetic algorithms) this research is concerned primarily with Tabu search.

Tabu Search is widely considered to be the best approach to solving large vehicle routing problems (Osman 1993; Fiechter 1994; Knox 1994; Xu and Kelly 1996; Voudouris and Tsang 1999; Gamboa et al. 2006; Niizuma et al. 2006; Vogt et al. 2007; Gribkovskaia et al. 2008). Specialized versions of Tabu heuristics are frequently designed to approach variant TSP and other routing problems (Gendreau et al. 1996, 1999; Cordeau et al. 1997, 2001; Archetti et al. 2006; Wassan et al. 2008; Cote and Potvin 2009; Zachariadis et al. 2009). Tabu performs a best improvement local search by selecting the best move in the neighborhood. However that exchange can only be accepted if it is not excluded by the Tabu list which contains forbidden solutions (Glover 1990). Tabus may sometimes prohibit attractive moves, even when there is no danger of cycling. It is imperative therefore, to add some criteria that will allow the search to override the Tabu list. These are called aspiration criteria. The simplest and most commonly used aspiration criterion consists of allowing a move, if that move results in a solution having an objective value better than that of the current best-known solution. Other criteria set the length of time (number of iterations) a move can be relegated to the Tabu list (Tsubakitani and Evans 1998). Given the evidence in the literature regarding Tabu search being the most commonly implemented TSP heuristic, and complemented by some statements from GIS vendors, this research presumes that Tabu search heuristics dominate the implementations of TSP solvers integrated into GIS.

2.3 The Implementation of the TSP in the Context of Geographic Information Systems

The ability of GIS to solve a wide range of spatially related problems for an extraordinary number of applications has been well documented. Moreover, GIS is one of the primary tools used to support the basic research undertaken in quantitative geography or geographic information science (Goodchild 1992). GIS has been particularly useful in the modeling of networks (Curtin 2008) and network-based phenomena (Miller and Shaw 2001). However, GIS are not designed to solve combinatorially complex location problems optimally, thus any implementation must employ heuristic solution procedures (Church and Sorenson 1994). While comparative analyses of heuristic solution procedures for the TSP have been made in the past (Golden et al. 1980), this research is concerned with the instantiation of those procedures in GIS. More specifically, it is common to distinguish between general-purpose GIS and GIS-T software designed specifically for transportation applications (Waters 1999; Slavin 2004); this research considers the former. Early efforts at comparing routing heuristics examined "microcomputer-based vehicle routing software" prior to the widespread availability of GIS (Golden et al. 1986). The need for

Table 1 Software packages tested with solver names and heuristics for solving the TSP

Software Package	Solver Name	Heuristic for TSP Implementation
ArcView 3.2	Network Analyst 3.0	Tabu (presumed)
ArcGIS 9.1	Network Analyst 9.1	Tabu (presumed)
ArcLogistics Route	Create Routes	Tabu (known)
Intergraph GeoMedia	Transportation Manager	Tabu (presumed)

testing location science heuristics as they are implemented in GIS has been clearly stated in the literature (Church 2002).

There is no question that heuristic approaches to routing problems in the context of GIS can provide substantial benefit to users who must quickly generate good (albeit not guaranteed optimal) solutions (Weigel and Cao 1999), although it has been recognized those benefits have historically come with the added costs associated with maintaining a GIS (Bodin 1990). Moreover, it is recognized the GIS is becoming the platform of choice for an increasing number of routing and logistics practitioners (Sutton and Visser 2004) including those who wish to solve instances of the TSP (Fischer 2004). Unfortunately for these users, a review of GIS documentation made for this research has found that GIS software vendors almost never specify the details of the heuristics they employ, they occasionally do not admit to using heuristics at all, and they sometimes imply – and occasionally state – that optimal solutions will be determined with their software, even though this is demonstrably false.

For this research, four commercial off-the-shelf GIS software implementations of the TSP were analyzed. The four packages, their solver names, and the known or presumed heuristics used for the TSP implementations are given in Table 1.

Although these products can perform several types of network analyses, the focus here is on the TSP solution. Each package contains many possible options for solving the TSP including, but not limited to, specifying impedance values, allowing or disallowing U-turns, specifying curb approach restrictions, identifying time windows for arrival or departure, adding barriers, and specifying stop order. In the analysis below every effort was made to hold all options constant across software platforms, and this was largely successful. The one exception was ArcLogistics Route which is a standalone software product that requires the use of its own proprietary network dataset. Due to this restriction network locations and distances were not identical to those used by the other products, but were approximated as well as was possible.

Of these four products only the documentation for ArcLogistics Route specifies that Tabu search is used to find the best route. Conversely, the documentation for ArcView 3.2 and ArcGIS 9.1 repeatedly specify that the TSP will be solved by finding the "best route" or the "shortest route", or even that the procedure will "optimize the solution". While Esri and Intergraph maintain that the specifics of their implementations are proprietary (and that is certainly reasonable), public GIS forum discussions have suggested that ArcView 3.2 and ArcGIS 9.1 use a Tabu Search heuristic (Sandhu 2001, 2006). Nothing is publicly known about the heuristic implementation in Intergraph's GeoMedia Transportation Manager. This research compares the performance of these GIS implementations of the TSP against each other and against the optimal solution when available.

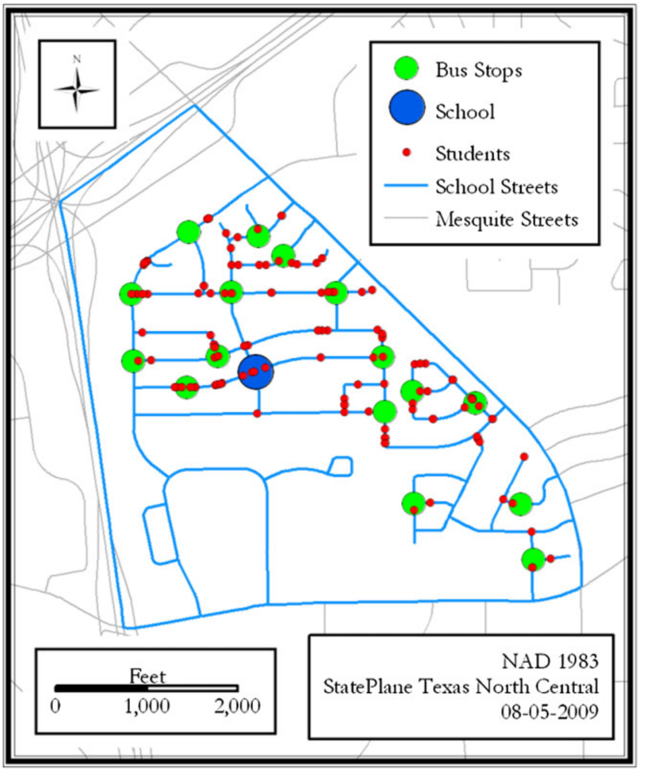

Figure 1 Example problem instance (16 stops) and associated dataset in Mesquite, TX

3 Comparative Analysis and Computational Experience

3.1 Data Sources

A case study dataset was developed on which the heuristic and exact solution approaches were tested. The case study was based on a school bus routing application, a classic application for the TSP that has historically been modeled as a multiple traveling salesman problem (Angel et al. 1972; Bowerman et al. 1995; Li and Fu 2002; Bektas and Elmastas 2007; Fugenschuh 2009) although the research in this test is restricted to single tours. An example of the data for an instance of the problem with 16 stops is illustrated in Figure 1. The underlying network dataset consists of a portion of the street centerline network for the City of Mesquite, TX. Twenty-five problem instances were generated, with the number of stops ranging from four to 1,500. For small instances of the TSP (fewer than 10 stops) the tests were performed using geocoded student location information and attendance school zones from Mesquite Independent School District (MISD). For larger instances of the problem the bus stops were chosen by randomly selecting records from the table of attributes for network junctions. Figure 1 shows the portion of the network dataset surrounding Kimball Elementary School.

3.2 Tests of Solution Procedures

The primary results of the comparative analysis are presented in Table 2. Two exact solution procedures and five heuristic solution procedures were implemented in order to test the ability

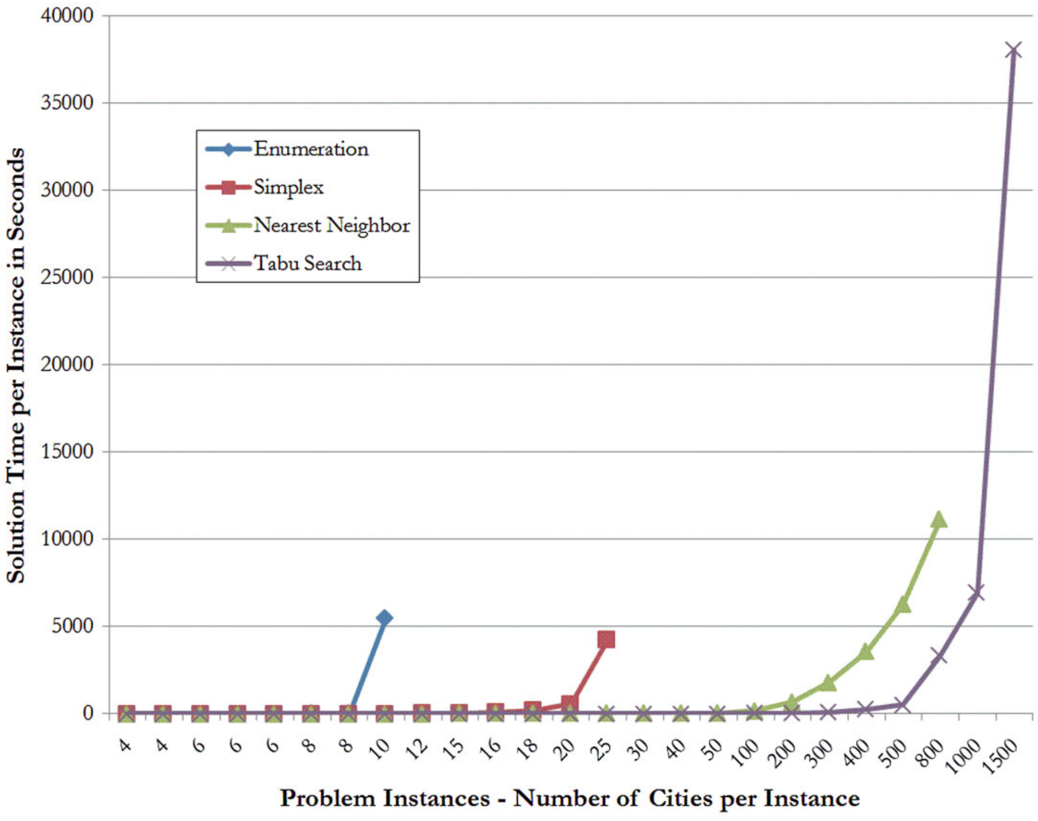

Figure 2 Exact and heuristic TSP solution times for all problem instances

of GIS to solve instances of the TSP, and to make comparisons across software platforms. For each of the seven solution procedures, the procedure was timed on each of the 25 problem instances (Figure 2), and – if a solution could be generated by that method on that problem instance – the objective function value (total traveled distance in feet) was captured.

First, a complete enumeration procedure was coded as a standalone Visual Basic 6.0 Windows application. The complete enumeration method was able to determine the optimal solution for a maximum of 10 cities before the memory resources of the host computer were exhausted. Second, the linear programming formulation given in Section 2.1.2 was used to determine the optimal solution for 14 problem instances ranging from four to 25 cities. The formulation itself was coded in the ILOG Optimization Programming Language Studio, the data files were generated from the GIS dataset, and the optimal solution was determined with CPLEX 8.1.0 running on a desktop computer with a 2.4 GHz XEON processor with 1 GB of RAM. This same platform was used for the GIS-based heuristic TSP implementations.

As a base from which to begin evaluating heuristic GIS implementations, a nearest neighbor solution procedure was implemented through the development of a standalone Visual Basic 6.0 Windows application. This solution procedure could determine a feasible solution for up to 800 stops in a reasonable amount of time (Figure 2). The four GIS implementations were tested on the same problem instances as the exact solution procedures and the nearest neighbor heuristic. Network Analyst 3.2, Network Analyst 9.1, and Transportation Manager all found the optimal solution for each of the first eight problem instances with 10 cities or less. Please note that due to the proprietary network required by ArcLogistics Route the objective function values occasionally differ slightly from those calculated for the other solution procedures even when the same route has been found. Therefore when ArcLogistics route found the optimal solution for small problems the objective function value appears to be less than optimal (which is impossible). This is simply an artifact of the different networks, and in each case this was checked to confirm that the same route was found, only with slightly different lengths associated with the network elements.

3.3 Comparative Results

Table 2 shows the objective function values for each problem instance and TSP implementation combination. Table 2 compares the optimal solution to the heuristic solutions for those problem instances where we have both values available. The Nearest Neighbor heuristic found the optimal solution only twice out of the 23 problem instances that it could solve., and these were two of the smallest problem instances with only four and six stops. In all other instances the solution was suboptimal, and in several cases, far above optimal (since the sense of optimization is to minimize, all sub-optimal solutions are above optimal). In two cases, the solution was 40% or more above optimal, and the average sub-optimal result was 26.4% greater than optimal (Table 2).

While the poor performance of the nearest neighbor heuristic is not surprising given the longstanding studies of this method in the literature, the performance of the GIS implementations is much more revealing. The heuristic GIS implementations were only able to determine optimal solutions to problems with 10 or fewer stops, with one exception. This is essentially the same performance (in terms of ability to reach optimality) that was achieved through the complete enumeration procedure. There was a single problem instance with 25 stops that was solved optimally by three of the four GIS implementations. Given that this route was shorter than the heuristically derived routes for instances with between 12 and 20 stops, we believe that a particularly fortuitous configuration of stops was generated through the random selection process, allowing the heuristics to tackle an "easy" problem. Although 175 problem instance/solution procedure combinations were examined here, additional research is required to determine how often such anomalies may occur.

For those problem instances with between 12 and 20 stops we are able to compare the optimal solution generated by the linear programming procedure with the heuristic GIS results. Although the GIS heuristics perform better than the nearest neighbor heuristic (which is, in all likelihood, the starting solution for a version of the TABU heuristic) the objective function values for these problem instances were from 14.4% to 19% suboptimal (Table 3).

Of course all of the heuristic solution procedures could determine solutions for far larger problems, albeit with no presumption of optimality. For the instances where we do not have optimal solutions, we can only examine across the heuristic implementations. In terms of problem size, the Esri Network Analyst 3.2 and 9.1 implementations were able to solve the largest instances (1,000 and 1,500 stops, respectively). The ArcLogistics Route software, which is marketed as one that solves practical routing problems, consistently found solutions that were longer than those solutions found by the other three heuristics. More specifically, for problems with between 12 and 50 cities, ArcLogistics Route found the solutions longer than any other heuristic. Moreover, ArcLogistics Route could not solve problems larger than 100 cities. For all instances where Intergraph Transportation Manager could determine a solution, it always determined a solution that was equal to or better than (shorter than) any of the Esri products.

Table 2 Objective function values for seven solution procedures on 25 problem instances

Total distance traveled (feet)
Best routes found with 7 solution procedure/problem instance combinations
Proven Optimal Solutions shaded in gray
* denotes solutions from ArcLogistics Route that vary due to differences in default road dataset
*** signifies the solution could not be determined with that method on that instance

Instance #	Number of Stops (n)							
	Solution Method	Complete Enumeration	Simplex and Search	Nearest Neighbor	Tabu Search I	Tabu Search II	Tabu Search III	Unknown
	Software	Visual Basic 6.0	CPlex 8.1.0	Visual Basic 6.0	Esri Network Analyst 3.2	Esri Network Analyst ArcGIS 9.1	Esri Arc Logistics Route	Intergraph Transportation Manager
1	4	4,276	4,276	4,276	4,276	4,276	4,244*	4,276
2	4	9,561	9,561	10,259	9,561	9,561	8,504*	9,561
3	6	8,344	8,344	8,344	8,344	8,344	8,448	8,344
4	6	10,036	10,036	11,828	10,036	10,036	16,368	10,036
5	6	12,677	12,677	13,154	12,677	12,677	12,672*	12,677
6	8	11,362	11,362	12,591	11,362	11,362	10,032*	11,362
7	8	15,325	15,325	21,762	15,325	15,325	15,312*	15,325
8	10	16,954	16,954	17,143	16,954	16,954	13,728*	16,954
9	12	***	50,950	71,056	62,274	62,274	62,832	62,274
10	15	***	58,981	67,386	66,285	66,285	68,112	66,285
11	16	***	58,908	71,468	66,870	67,679	68,112	66,870
12	18	***	58,939	75,471	68,063	68,063	69,168	68,063
13	20	***	65,022	84,228	71,486	71,005	80,784	71,005
14	25	***	59,958	64,876	59,958	59,958	66,000	59,958
15	30	***	***	78,327	63,610	63,610	64,944	63,610
16	40	***	***	128,344	106,117	106,117	117,744	97,522
17	50	***	***	129,617	115,360	115,360	118,800	109,861
18	100	***	***	223,001	202,750	205,865	204,864	191,489
19	200	***	***	***	218,830	205,164	***	***
20	300	***	***	294,266	247,723	250,168	***	236,401
21	400	***	***	***	296,766	283,638	***	***
22	500	***	***	415,759	372,252	355,539	***	341,497
23	800	***	***	611,686	578,613	548,199	***	***
24	1,000	***	***	***	623,920	581,997	***	***
25	1,500	***	***	***	***	880,133	***	***

Table 3 Sub-optimality of GIS solutions

Route #	Number of Stops n	Nearest Neighbor (%)	Tabu Search I ArcView (%)	Tabu Search II ArcGIS (%)	Tabu Search III ArcLogistics (%)	Unknown Intergraph (%)
9	12	40	22	22	23	22
10	15	14	12	12	15	12
11	16	21	14	15	16	14
12	18	28	15	15	17	15
13	20	29	10	9	24	9
Average when Sub-Optimal:		26.4	14.6	14.6	19	14.4

4 Conclusions, Future Research, and Recommendations for GIS Implementations of the TSP

The results of the comparative test are clear. It is unlikely that any of the heuristic GIS implementations will be able to determine the optimal solution for a TSP instance with more than 10 stops. In the context of solving even the most routine delivery or service problem instance in an urban setting it is likely that dozens, if not hundreds, of stops must be made by a single vehicle. Certainly more than 10 stops is the norm for many urban application areas, e.g. postal delivery (Hollis et al. 1985), or garbage pickup (Bhargava and Tettelbach 1997).

The potential consequences of these results are equally clear. Most users who wish to solve TSP instances cannot be expected to fully understand the nature of combinatorially complex problems, and the difficulties in determining optimal solutions to those problems. These users are simply given a task and asked to solve it. A review of the documentation of these methods in the GIS software could easily give a naïve user the impression that they will obtain the optimal solution. If they convey to their clients or superiors that they have determined an optimal solution to the problem, and then at some later time a solution that is up to 14% lower in cost is determined, there may be serious consequences for those users. When transportation costs for businesses and government agencies run into the millions of dollars per year, solutions that are 14% sub-optimal represent a major additional cost. Given the results above, it is clearly the responsibility of the GIS research community to inform the user community which employs GIS heuristic solutions about the potential consequences of using these methods. At a minimum those users should know the following facts: (1) that solutions to problems of 10 stops or fewer may be optimal, but that there is no guarantee of optimality and there is no way within the GIS to test for optimality; (2) that their solutions to problem instances with greater than 10 stops are very likely to be suboptimal, and those solutions could be suboptimal by 14% or more; and (3) that they could find guaranteed optimal solutions to those problem instances of 25 stops or less through the implementation of a linear programming solution procedure. It is only with this information that users can make a reasoned judgment about the importance of an optimal solution versus the importance of convenience or speed in determining a solution.

Moreover, beyond the clear need to communicate the potential sub-optimality of results to typical users, these results present several future research challenges to the applied spatial analysis community. First, although four GIS implementations were available to be tested in this research, there are indeed other GIS implementations (e.g. TransCAD) and the authors would welcome further tests to determine if other implementations can perform better than those tested here. The GIS research community should encourage GIS software producers to publish tests of their own implementations against optimal solutions to demonstrate the ability of their heuristics to produce optimal or near-optimal solutions under different conditions. We recognize that larger TSP instances could be solved optimally either with more powerful hardware, or with specialized solution procedures. Doing so would allow for a larger set of optimal solutions against which to compare GIS-heuristic solutions. Second, a single TSP instance with 25 stops was solved optimally by three of the four GIS implementations while no other problem with more than 10 stops could be solved optimally. This may be a reflection of a particularly tractable data configuration, but more research is needed to determine if this is the case, and if so, what is the frequency of such data anomalies. More importantly, if there is something unique about this particular data configuration that permits optimal solution of larger problems, the potential to exploit that data structure for solution procedures deserves attention. In order to test for the prevalence of such solutions future experiments should utilize a large number of different network datasets, perhaps randomly generated. Such experiments would also speak to the generalizability of the results presented here. Finally, since network analysis in GIS is a very active applied research area, it is critical that advances in TSP solution procedures be implemented in GIS software packages as quickly as possible.

In conclusion, there is no discredit in using heuristic solutions to solve large instances of the TSP for which optimal solutions cannot be determined. When that is the case, the GIS solution procedures are perfectly acceptable as long as the users are made aware that it is very unlikely that the GIS will provide the optimal solution. If they know of this limitation they will be in a position to provide appropriate caveats with their results. GIS were not originally designed to solve combinatorially complex problems optimally, and these problems can become intractable so quickly that even specialized software can often only solve moderately sized problems optimally. It is hoped that the research presented here will encourage GIS providers to include performance results of their heuristics so that their users can know the performance they can expect when solving the TSP. Given the results above, it appears there is both a need and an opportunity to improve GIS through the integration of GIS with linear programming software. Some recent attempts have been made to integrate GIS and integer programming software in order to increase the number and type of problems that can be solved (Jung et al. 2006; Curtin et al. 2010). The advances in cloud computing for geographic applications suggest that the success of such integration is more likely. The ability to simultaneously take advantage of the strengths of two disciplines – geography and operations research – holds promise as a research area for the immediate future.

References

Angel R, Caudle W, Noonan R, and Whinston A 1972 Computer assisted school bus scheduling. *Management Science* 18: 279–88

Applegate D, Bixby R, Chvatal V, and Cook W 1998 On the solution of traveling salesman problems. *Documenta Mathematica* 3: 645–56

Applegate D, Bixby R, Chvátal V, and Cook W 2003 Implementing the Dantzig-Fulkerson-Johnson algorithm for large traveling salesman problems. *Mathematical Programming* 97: 91–153

Archetti C, Speranza M, and Hertz A 2006 A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science* 40: 64–73

Balas E and Guinard M 1979 Branch and bound/implicit enumeration. *Annual Discrete Mathemathics* 5: 185–91

Bektas T 2006 The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega* 34: 209–19

Bektas T and Elmastas S 2007 Solving school bus routing problems through integer programming. *Journal of the Operational Research Society* 58: 1599–604

Bhargava H and Tettelbach C 1997 A web-based decision support system for waste disposal and recycling. *Computers, Environment and Urban Systems* 21: 47–65

Black P 2009 Metaheuristic. WWW document, http://xlinux.nist.gov/div897/sqg/dads/HTML/metaheuristic.html

Bodin L 1990 Twenty years of routing and scheduling. *Operations Research* 38: 571–79

Bodin L, Golden B, Assad A, and Ball M 1983 Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research* 10(2): 65–211

Bowerman R, Hall B, and Calamai P 1995 A multi-objective optimization approach to urban school bus routing: Formulation and solution method. *Transportation Research: Part A, Policy and Practice* 29: 107–23

Church R L 2002 Geographical information systems and location science. *Computers and Operations Research* 29: 541–62

Church R L and Sorenson P 1994 *Integrating Normative Location Models into GIS: Problems and Prospects with p-median Model*. Santa Barbara, CA, National Center for Geographic Information and Analysis Technical Report No: 94–5

Cordeau J, Gendreau M, and Laporte G 1997 A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 30: 105–19

Cordeau J, Laporte G, and Mercier A 2001 A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* 52: 928–36

Cote J and Potvin J 2009 A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. *European Journal of Operational Research* 198: 464–69

Curtin K M 2007 Network analysis in geographic information science: Review, assessment, and projections. *Cartography and Geographic Information Science* 34: 103–11

Curtin K M 2008 Network data structures. In Kemp K (ed.) *The Encyclopedia of Geographic Information Science*. Thousand Oaks, CA, Sage: 314–17

Curtin K M, Hayslett-McCall K, and Qiu F 2010 Determining optimal police patrol areas with maximal covering and backup covering location models. *Networks and Spatial Economics* 10: 145–65

Dantzig G, Fulkerson D, and Johnson S 1954 Solution of a large scale traveling salesman problem. *Journal of the Operations Research Society of America* 2: 393–410

Evans J and Minieka E 1992 *Optimization Algorithms for Networks and Graphs* (Second Edition). New York, Marcel Dekker

Fiechter C 1994 A parallel tabu search algorithm for large traveling salesman problems. *Discrete Applied Mathematics* 51: 243–67

Fischer M 2004 GIS and network analysis. In Hensher D, Button K, Haynes K, and Stopher P (eds) *Handbook of Transport Geography and Spatial Systems*. Amsterdam, Elsevier: 391–408

Flood M 1956 The traveling-salesman problem. *Operations Research* 4: 61–75

Fugenschuh A 2009 Solving a school bus scheduling problem with Integer programming. *European Journal of Operational Research* 193: 867–84

Gamboa D, Rego C, and Glover F 2006 Implementation analysis of efficient heuristic algorithms for the traveling salesman problem. *Computers and Operations Research* 33: 1154–72

Gendreau M 2003 An introduction to tabu search. In Glover F (ed.) *Handbook of Metaheuristics*. Dordrecht, Germany, Springer: 37–54

Gendreau M, Guertin F, Potvin J, and Taillard E 1999 Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science* 33: 381–90

Gendreau M, Laporte G, and Seguin R 1996 A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research* 44: 469–77

Glover F 1990 Tabu search: A tutorial. *Interfaces* 20: 74–94

Golden B, Bodin L, Doyle T, and Stewart J 1980 Approximate traveling salesman algorithms. *Operations Research* 28: 694–711

Golden B, Bodin L, and Goodwin T 1986 Microcomputer-based vehicle routing and scheduling software. *Computers and Operations Research* 13: 277–85

Goodchild M F 1992 Geographical information science. *International Journal of Geographical Information Systems* 6: 31–45

Gribkovskaia I, Laporte G, and Shlopak A 2008 A tabu search heuristic for a routing problem arising in servicing of offshore oil and gas platforms. *Journal of the Operational Research Society* 59: 1449–59

Gutin G and Holloway R 2004 Traveling salesman problems. In Gross J and Yellen J (eds) *Handbook of Graph Theory*. Boca Raton, FL, CRC Press: 279–99

Gutin G and Punnen A 2002 *The Traveling Salesman Problem and Its Variations* (Volume 12). Dordrecht, Germany, Kluwer

Hoffman A and Wolfe P 1985 History. In Lawler E L, Lenstra J L, Rinnooy Kan A H G, and Shmoys D B (eds) *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Chichester, John Wiley and Sons: 1–17

Hoffman K and Padberg M 2001 Traveling salesman problem. In Gass S L and Harris C M (eds) *Encyclopedia of Operations Research and Management Science*. Boston, MA, Kluwer: 849–52

Hollis M, Thompson R, and Baillie K 1985 Heuristic determination of postal delivery routes. *Computers, Environment and Urban Systems* 10: 57–67

Jung H, Lee K, and Chun W 2006 Integration of GIS, GPS, and optimization technologies for the effective control of parcel delivery service. *Computers and Industrial Engineering* 51: 154–62

Junger M, Reinelt G, and Rinaldi G 1997 The traveling salesman problem. In Martello S, Dell'Amico M, and Maffioli F (eds) *Annotated Bibliographies in Combinatorial Optimization*. Chichester, John Wiley and Sons: 199–221

Knox J 1994 Tabu search performance on the symmetrical traveling salesman problem. *Computers and Operations Research* 21: 867–76

Lawler E 2001 *Combinatorial Optimization: Networks and Matroids*. Mineola, NY, Dover Publications

Li L and Fu Z 2002 The school bus routing problem: A case study. *Journal of the Operational Research Society* 53: 552–58

Marinakis Y, Migdalas A, and Pardalos P 2005 A hybrid genetic-GRASP algorithm using Lagrangean relaxation for the traveling salesman problem. *Journal of Combinatorial Optimization* 10: 311–26

Marinakis Y, Migdalas A, and Pardalos P 2008 Expanding neighborhood search-GRASP for the probabilistic traveling salesman problem. *Optimization Letters* 2: 351–61

Miller H and Shaw S 2001 *Geographic Information Systems for Transportation*. New York, Oxford University Press

Niemeier D and Beard M 1993 GIS and transportation planning: A case study. *Computers, Environment and Urban Systems* 17: 31–43

Niizuma D, Yasuda K, and Ishigame A 2006 Multi-point tabu search for traveling salesman problems. *IEEJ Transactions on Electrical and Electronic Engineering* 1: 126–29

Osman I 1993 Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 41: 421–51

Punnen A 2002 The traveling salesman problem: Applications, formulations and variations. In Gutin G and Punnen A (eds) *The Traveling Salesman Problem and Its Variations* (Volume 12). Dordrecht, Germany, Kluwer: 1–28

Sandhu J 2001 GISList: Network Analyst (Vol. 2006). WWW document, http://spatialnews.geocomm.com/community/lists/viewmessage.phtml?id=9242

Sandhu J 2006 TSP Algorithm (Vol. 2006). Redlands, CA, Esri

Schrijver A 2005 On the history of combinatorial optimization (Till 1960). *Discrete Optimization* 12: 1–68

Simkowitz H 1988 Transportation applications of geographic information systems. *Computers, Environment and Urban Systems* 12: 253–71

Skiena S 2008 *The Algorithm Design Manual* (Second Edition). London, Springer-Verlag

Slavin H 2004 The role of GIS in land use and transport planning. In Hensher D, Button K, Haynes K, and Stopher P (eds) *Handbook of Transport Geography and Spatial Systems*. Amsterdam, Elsevier: 329–56

Sutton J and Visser J 2004 The role of GIS in routing and logistics. In Hensher D, Button K, Haynes K, and Stopher P (eds) *Handbook of Transport Geography and Spatial Systems*. Amsterdam, Elsevier: 357–74

Tsubakitani S and Evans J 1998 Optimizing tabu list size for the traveling salesman problem. *Computers and Operations Research* 25: 91–7

Vajda S 1961 *Mathematical Programming*. Reading, MA, Addison-Wesley

Vogt L, Poojari C, and Beasley J 2007 A tabu search algorithm for the single vehicle routing allocation problem. *Journal of the Operational Research Society* 58: 467–80

Voudouris C and Tsang E 1999 Guided local search and its application to the traveling salesman problem. *European Journal of Operational Research* 113: 469–99

Wassan N, Wassan A, and Nagy G 2008 A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. *Journal of Combinatorial Optimization* 15: 368–86

Waters N 1999 Transportation GIS: GIS-T. In Longley P A, Goodchild M F, Maguire D J, and Rhind D W (eds) *Geographical Information Systems: Principles, Techniques, Management and Applications* (Second Edition). Hoboken, NJ, John Wiley and Sons: 827–44

Weigel D and Cao B 1999 Applying GIS and OR techniques to solve Sears technician-dispatching and home delivery problems. *Interfaces* 29: 112–30
Wilson R 1996 *Introduction to Graph Theory*. Harlow, Essex, Longman
Xu J and Kelly J 1996 A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science* 30: 379–93
Zachariadis E, Tarantilis C, and Kiranoudis C 2009 A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research* 195: 729–43

