

The Attractive Traveling Salesman Problem

Güneş Erdoğan ^{*,†} Jean-François Cordeau ^{*} Gilbert Laporte [†]

August 14, 2007

Abstract

In the *Attractive Traveling Salesman Problem* the vertex set is partitioned into *facility vertices* and *customer vertices*. A maximum profit tour must be constructed on a subset of the facility vertices. Profit is computed through an attraction function: every visited facility vertex attracts a portion of the profit from the customer vertices based on the distance between the facility and customer vertices, and the attractiveness of the facility vertex. A gravity model is used for computing the profit attraction. The problem is formulated as an integer non-linear program. A linearization is proposed and is strengthened through the introduction of valid inequalities, and a branch-and-cut algorithm is developed. A tabu search algorithm is also implemented. Computational results are reported.

Keywords: traveling salesman problem, demand attraction, demand allocation, linearization, branch-and-cut, tabu search.

1 Introduction

The purpose of this paper is to introduce a new variant of the *Traveling Salesman Problem* (TSP), called the *Attractive Traveling Salesman Problem* (ATSP). The ATSP is defined on an undirected graph $G = (V \cup W, E)$, where $V \cup W$ is the vertex

^{*}Canada Research Chair in Logistics and Transportation, HEC Montréal, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[†]Canada Research Chair in Distribution Management, HEC Montréal, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

set and $E = \{(v_i, v_j) : v_i, v_j \in V \cup W, i < j\}$ is the edge set. The set V is a set of *facility vertices*, while the set W is a set of *customer vertices*. Let T be a subset of *compulsory vertices* of V , including a *depot* v_0 . A distance d_{ij} and a travel time t_{ij} are associated with each edge (v_i, v_j) . A profit p_k is associated with each customer vertex v_k . Each facility vertex v_i on the tour, apart from the depot, generates a profit derived from the customer vertices, which is measured by an attraction function to be defined later. Including vertex v_i in the cycle generates a dwell time r_i . The cycle length, including travel and dwell times, may not exceed an upper limit L . The aim of the ATSP is to design a cycle or tour of maximal profit, including all vertices of T and possibly some vertices of $V \setminus T$, subject to the length constraint.

An application of the ATSP arises in the planning of a tour of a mobile entertainment facility such as a circus or a theater company. The amount of time that can be spent by the mobile facility is limited. Visiting a facility vertex generates a profit. Facilities with extra services and closer to the larger population centers are assumed to be more attractive from the customers' point of view, and consequently visiting these facilities is more profitable. Another application arises in the routing of a military reconnaissance vehicle. In this application, the customer sites are the enemy installations or encampments, the facility sites are the possible observation points, and the travel distance limit is dictated by either the fuel capacity or the allowed duration of the mission. The attractiveness of a candidate observation point may be perceived as a function of its visibility range and concealment factor. The objective function becomes the maximization of information gathered. Yet another application is the design of a route for a mobile health care facility operating in an underdeveloped region. Typically, such a facility can only visit a subset of localities accessible by the main road network. Population centers located outside these localities access them on foot. The accessibility problem is exacerbated during the rainy seasons when only paved roads can be used by the mobile facility (Oppong and Hodgson, 1994; Hodgson, Laporte, and Semet, 1998).

There exist many studies on *Traveling Salesman Problems with Profits*. A recent survey by Feillet, Dejax, and Gendreau (2005) lists 95 references. This survey categorizes the problems into three classes, based on how the objectives of minimizing distance and maximizing profit are handled. The first class consists of problems in which both of the objectives are combined in the objective function. The second class is composed of problems in which the travel cost is a constraint and the objective is to maximize the profits collected. Finally, the problems in which the profit is a constraint and the objective is to minimize the travel cost constitute the third class. According to this categorization our problem is closest to the problems in the second

class, together with the *Orienteering Problem* (Golden, Levy, and Vohra, 1987), the *Maximum Collection Problem* (Kataoka and Morito, 1988), and the *Selective Traveling Salesman Problem* (Laporte and Martello, 1990). The most successful exact study regarding this class of problems is the one by Fischetti, Salazar González, and Toth (1998), in which the authors were able to solve instances involving up to 500 vertices within a few hours, using a branch-and-cut algorithm. The most successful heuristic for this class of problems is due to Gendreau, Laporte, and Semet (1998). These authors use a tabu search algorithm to solve instances involving up to 300 vertices in a few minutes, with an optimality gap typically less than 1%.

In all these studies profits are assumed to be collected from the visited vertices, implicitly implying that customer and facility vertices coincide, which may not be the case in all applications. As an example from the entertainment sector, race tracks are usually built outside the urban centers to avoid the noise and pollution generated by these facilities. Similarly, ski slopes are located in the mountainous areas, which are usually at a significant distance from the major population centers. In our study, we assume that the facility vertices and customer vertices do not coincide, although they can be arbitrarily close. This distinction also allows the division of the demand of a population center into smaller parts located around the facility, thereby permitting a finer aggregation of the demand data.

Because the sets of customer and facility vertices are disjoint, the questions of how much demand is captured and how the captured demand is allocated to facilities arise. To the best of our knowledge these questions have not been fully addressed in the literature on routing, perhaps because customer and facility vertices are usually not separated. There are a few notable exceptions. In Lee, Chiu, and Sanchez (1998), the authors define and study the *Steiner Ring Star Problem*, in which a tour over a subset of the facility vertices must be determined; the demand of a customer vertex is assumed to be assigned to the closest visited facility vertex. The objective is to minimize the sum of travel and assignment costs. In two closely related studies by Labbé, Laporte, Rodriguez-Martín, and Salazar González (2004, 2005), the authors solve the *Ring Star Problem* and the *Median Cycle Problem*. The difference between the problem studied in the former paper and the Steiner Ring Star Problem is that there is no distinction between facility and customer vertices, and each vertex can be visited. In the Median Cycle Problem, the objective is to minimize the routing cost, and there is an upper bound on the assignment cost. Another exception is the set of problems with covering aspect, such as the *Covering Tour Problem* introduced by Current in 1981, where a customer vertex is covered if it is within a prespecified distance of a visited facility vertex, and all the demand of

a customer vertex is assumed to be attracted to the visited facility vertices within the coverage range.

Demand attraction and allocation functions have been studied in the competitive location literature. The reader is referred to Drezner (1995) for a survey of competitive facility location models, and to Eiselt and Laporte (1998) for a critical review of demand allocation functions. It has been suggested by Hotelling (1929) that each customer patronizes the closest facility in a winner-takes-all manner, given that prices are identical. Much later, based on the gravity model of Reilly (1931), Huff (1964, 1966) has used a *gravity function* by which the probability P_{ki} that a customer at v_k patronizes a given facility v_i is proportional to the attractiveness of the facility, and inversely proportional to some power of the distance between the customer and the facility. In subsequent work, Hodgson (1981) has advocated using an exponential distance decay function rather than a polynomial one. Drezner and Drezner (2002) have tested and validated Huff's gravity based model using real world data, both for the polynomial and the exponential distance decay functions. One of the authors' conclusions is that the results are not sensitive to the choice of the distance decay function. The gravity based model has been used in the studies by Jain and Mahajan (1979), Drezner (1994), Bell, Ho, and Tang (1998), and more recently by Drezner and Drezner (2004, 2006, and 2007). Encouraged by these studies, we have chosen the gravity based model for determining the demand attraction and allocation.

The remainder of the paper is organized as follows. In Section 2, we provide a definition of the attraction function, we formally define the problem, we present a proof of NP-hardness, and we introduce a non-linear formulation. In Section 3, we propose a linearization scheme involving an infinite number of potential constraints, as well as valid inequalities. We describe a branch-and-cut algorithm for the problem in Section 4, and a tabu search algorithm in Section 5. Computational results for both the branch-and-cut and tabu search algorithms are presented in Section 6. In Section 7, we analyze an extension of the problem where there may be more than one option of service at the facility vertices. Conclusions follow in Section 8.

2 The Model

Based on Huff's gravity function, we derive a general formula for the probability P_{ki} that a customer from vertex v_k patronizes a facility at vertex v_i :

$$P_{ki} = \frac{\frac{a_i}{d_{ki}^q}}{\sum_{v_j \in V \setminus \{v_0\}} \frac{a_j}{d_{kj}^q}}, \quad (1)$$

where a_i is the attractiveness of the facility at vertex v_i , and $q \geq 1$ is a parameter. The attractiveness a_i of vertex v_i is based on the size, services, and other factors related to that facility.

One shortcoming of the demand attraction function (1) is the assumption that demand is never lost. In reality, some customers may choose not to get the service because substitute services are more attractive, or the visited facility vertices are too far away to be reached conveniently. To cope with this problem, we apply a minor modification to (1). Let b_k be the self-attraction of customer vertex v_k . Then the formula becomes:

$$P_{ki} = \frac{\frac{a_i}{d_{ki}^q}}{b_k + \sum_{v_j \in V \setminus \{v_0\}} \frac{a_j}{d_{kj}^q}}. \quad (2)$$

An interpretation of b_k is an estimate of substitute services available for vertex v_k .

While this modified allocation function suits our purposes better, it also violates the principle of insensitivity to scaling (Eiselt and Laporte, 1998) because of the constant in the denominator. That is, any scaling change in the distance or the attractiveness measurement will change the demand allocation. A simple way to overcome this problem is to redefine all parameters as the result of the division of the original parameters by its minimum, i.e. $d_{ij} := d_{ij} / \min\{d_{kl} | v_k, v_l \in V \cup W, v_k \neq v_l\}$, $a_i := a_i / \min\{a_j | v_j \in V \setminus \{v_0\}\}$ and modify the estimates for all b_k values accordingly.

We now prove that ATSP is NP-hard by a reduction from the Selective Traveling Salesman Problem.

Selective Traveling Salesman Problem (STSP): Given a graph $G = (V, E)$, a set $T \subset V$ of compulsory vertices, a distance d_{ij} associated with every edge in E , a limit on the total distance traveled L , and a positive integer profit p_i associated with each vertex in V , determine a maximal profit cycle whose length does not exceed L . STSP was proved to be NP-hard by Laporte and Martello (1990).

Proposition 1. ATSP is NP-hard.

Proof: We construct a reduction from the recognition form of the STSP. Take an arbitrary instance of the STSP. The recognition form of STSP is the question of the existence of a solution with objective function value greater than or equal to K . Construct an instance of the ATSP by adding a single customer vertex v_s , at an arbitrary distance d_{is} from every vertex $v_i \in V$, with a total profit of 1, and a self-attraction value of 1, i.e., $W = \{v_s\}$, $p_s = 1$, $b_s = 1$. Set the attractiveness of each vertex facility $v_i \in V$ to $a_i = p_i d_{is}^q$. Let $t_{ij} = d_{ij}$, $r_i = 0$. This setting of parameters results in an objective function $f(x)/(f(x) + 1)$ for the ATSP, where $f(x)$ is the objective function of the STSP. Since $f(x) \geq 0$, clearly $f(x) \geq K$ if and only if $f(x)/(f(x) + 1) \geq K/(K + 1)$. Solving this instance of the ATSP amounts to solving the STSP instance at hand, so ATSP is at least as hard as STSP. \square

We conclude this section by giving a non-linear integer programming formulation of the ATSP. Let x_{ij} ($i < j$) be equal to 1 if the vehicle traverses edge (i, j) and 0 otherwise, and let y_i be equal to 1 if vertex i is visited and 0 otherwise. The formulation is as follows:

(ATSP1)

$$\text{maximize } \sum_{v_k \in W} p_k \sum_{v_i \in V \setminus \{v_0\}} P_{ik} = \sum_{v_k \in W} p_k \sum_{v_i \in V \setminus \{v_0\}} \frac{\frac{a_i}{d_{ki}^q} y_i}{b_k + \sum_{v_j \in V \setminus \{v_0\}} \frac{a_j}{d_{kj}^q} y_j} \quad (3)$$

subject to

$$\sum_{v_i \in V, i < j} x_{ij} + \sum_{v_i \in V, i > j} x_{ji} = 2y_j \quad (v_j \in V) \quad (4)$$

$$\sum_{\substack{v_i \in S, v_j \in V \setminus S \\ \text{or } v_i \in V \setminus S, v_j \in V}} x_{ij} \geq 2y_t \quad (S \subset V : 2 \leq |S| \leq |V| - 2, T \setminus S \neq \emptyset, v_t \in S) \quad (5)$$

$$\sum_{v_i, v_j \in V} t_{ij} x_{ij} + \sum_{v_i \in V} r_i y_i \leq L \quad (6)$$

$$y_i = 1 \quad (v_i \in T) \quad (7)$$

$$y_i = 0 \text{ or } 1 \quad (v_i \in V \setminus T) \quad (8)$$

$$x_{ij} = 0 \text{ or } 1 \quad ((v_i, v_j) \in E). \quad (9)$$

The objective function (3) maximizes the profit generated from the customer vertices. Constraints (4) are degree constraints, and constraints (5) are connectivity constraints (Gendreau, Laporte, and Semet, 1997). Constraints (6) impose a maximal tour duration, while constraints (7) state that all vertices of T must be visited. Other integrality constraints are defined by (8) and (9).

3 Linearization Scheme and Valid Inequalities

In ATSP1, the objective function (3) consists of the sum of $|W|(|V| - 1)$ ratios of linear functions. This brings the problem into the domain of fractional programming, the elements of which are continuous optimization problems with sums of one or more ratios of functions. For a comprehensive survey of algorithms and literature on fractional programming, we refer to reader to Schaible (1995, 1996), Freund and Jarre (2001), and Benson (2004). The case of a single ratio of two linear functions and a polyhedron of feasible solutions is well solved, and can be handled through linear programming methods. However, the general fractional programming problem with more than a single ratio (also known as the sum-of-ratios problem) remains hard. It has been shown by Freund and Jarre (2001) that the sum-of-ratios problem is NP-complete. Most available algorithms for fractional programming with multiple ratios are based on sophisticated non-linear optimization algorithms, or on branch-and-bound. However, ATSP is a particular discrete optimization problem and standard methods are too time consuming to be used for solving the subproblems.

We first show that our problem possesses a special structure, which gives way to a relatively simple linearization scheme using valid inequalities. To the best of our knowledge linearizing fractional programming problems using valid inequalities is a new idea.

Notice that all terms in the second summation in (3) have the same denominator, allowing us to rewrite it as:

$$\text{maximize } \sum_{v_k \in W} p_k \frac{\sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} y_i}{b_k + \sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} y_i}. \quad (10)$$

Let $w_k = \sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} y_i$, so that (10) simplifies to

$$\text{maximize } \sum_{v_k \in W} p_k \frac{w_k}{b_k + w_k}. \quad (11)$$

Each of the functions $f_k(w_k) = w_k/(b_k + w_k)$ is concave in w_k for $w_k > -b_k$ and can be approximated by an upper envelope consisting of linear tangents. Formally, at point w_k^* :

$$\begin{aligned} f_k(w_k) &= \frac{w_k}{b_k + w_k} \leq \frac{b_k w_k}{(b_k + w_k^*)^2} + \left(\frac{w_k^*}{b_k + w_k^*} - \frac{b_k w_k^*}{(b_k + w_k^*)^2} \right) \\ &= \frac{b_k w_k}{(b_k + w_k^*)^2} + \frac{(w_k^*)^2}{(b_k + w_k^*)^2}. \end{aligned} \quad (12)$$

Let z_k denote the percent of profit captured from customer vertex v_k , i.e. $z_k = w_k/(b_k + w_k)$, and let y^* be the vector of y_i^* variables yielding w_k^* . Rewriting (12) in terms of z_k , w_k , and b_k gives

$$z_k \leq \frac{b_k \left(\sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} y_i \right)}{\left(b_k + \sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} y_i^* \right)^2} + \frac{\left(\sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} y_i^* \right)^2}{\left(b_k + \sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} y_i^* \right)^2}. \quad (13)$$

These constraints will be referred to as the *linearization constraints*. Using the same reasoning, this idea can be generalized and formalized as follows:

Proposition 2. Let $x \in X \subseteq R^n$, $a \in R^n$, and $b, c, d, e, f \in R$. Define $g(x) = (c(ax + b) + d)/(e(ax + b) + f)$. If $-2(cf - ed)/(e(ax + b) + f)^3 \leq 0, \forall x \in X$, then the following inequality is valid $\forall x^* \in X$:

$$g(x) \leq \frac{(cf - ed)(ax + b)}{(e(ax + b) + f)^2} + \frac{c(ax^* + b) + d}{e(ax^* + b) + f} - \frac{(cf - ed)(ax^* + b)}{(e(ax^* + b) + f)^2}. \quad (14)$$

We can now state our linearization:

(ATSP2)

$$\text{maximize } \sum_{v_k \in W} p_k z_k \quad (15)$$

subject to

$$0 \leq z_k \leq 1 \quad (v_k \in W). \quad (16)$$

and (4), (5), (6), (7), (8), (9), (13).

ATSP2 may be strengthened through the introduction of certain valid inequalities proposed for the Covering Tour Problem (Gendreau, Laporte, and Semet, 1997). The proofs of validity are identical for both problems.

1) Arc-vertex constraints :

Proposition 3. The inequalities

$$x_{ij} \leq y_i \quad (v_i, v_j \in V) \quad (17)$$

and

$$x_{ij} \leq y_j \quad (v_i, v_j \in V) \quad (18)$$

are valid for ATSP2.

2) Strong connectivity constraints :

Proposition 4. The inequalities

$$\sum_{\substack{v_i \in S, v_j \in V \setminus S \\ \text{or } v_i \in V \setminus S, v_j \in S}} x_{ij} \geq 2 \quad (S \subset V : 2 \leq |S| \leq |V| - 2, T \setminus S \neq \emptyset, S \cap T \neq \emptyset) \quad (19)$$

are valid for ATSP2.

3) Strong 2-matching constraints :

Proposition 5. The following inequalities are valid for ATSP2:

$$\sum_{v_i, v_j \in H} x_{ij} + \sum_{v_i, v_j \in E'} x_{ij} \leq \sum_{v_i \in H} y_i + \frac{1}{2}(|E' - 1|), \quad (20)$$

for all $H \subset V$ and $E' \subset E$ satisfying

- (i) $|\{v_i, v_j\} \cap H| = 1 \quad ((v_i, v_j) \in E')$,
- (ii) $\{v_i, v_j\} \cap \{v_k, v_l\} = \emptyset \quad ((v_i, v_j) \neq (v_k, v_l) \in E')$,
- (iii) $|E'| \geq 3$ and odd.

4 Branch-and-Cut Algorithm

We now describe a branch-and-cut algorithm using the linearization scheme and valid inequalities just introduced.

Step 1 (Lower bound). Compute a lower bound \underline{z} on the optimal solution of the problem, using a heuristic.

Step 2 (Root node). Construct the linear relaxation of ATSP2 without the connectivity constraints and the linearization constraints. Insert this subproblem in a list.

Step 3 (Node selection). If the list is empty, stop. Else select and remove a subproblem from the list according to a best-first criterion.

Step 4 (Subproblem solution). Solve the subproblem. Let z^* be the objective function value, and let x^* , y^* be the vector values taken by the x and y variables. If $z^* \leq \underline{z}$, go to Step 3.

Step 5 (Constraint generation). Generate all identified violated connectivity constraints, linearization constraints, arc-vertex constraints, strong connectivity constraints, and strong 2-matching constraints, and add them to the subproblem. If at least one constraint is generated, go to Step 4.

Step 6 (Integrality check). If the solution is integer, set $\underline{z} = z^*$, and go to Step 3.

Step 7 (Branching). Construct two subproblems by branching on a binary fractional variable with the highest pseudo-cost (Gauthier and Ribière, 1977). Add the subproblems to the list and go to Step 3.

The algorithm is handled by CPLEX, except for Step 5. The separation algorithms for the constraints to be added during the branch-and-cut algorithm are now outlined.

Linearization constraints: Although there are infinitely many members of this constraint set, any violated member can be identified in $O(|W||V|)$ time by simply plugging in the y^* and z^* values. This procedure is equivalent to generating the linear tangent at the current fractional point.

Arc-vertex constraints: Violated elements of this valid inequality set can be identified straightforwardly in $O(|V|^2)$ time.

Connectivity constraints: Violated members of the connectivity constraints can be identified by solving a *Maximum Flow Problem* from each $v_i \in T$ to each $v_j \in V \setminus T$, on a network where the upper bounds on the flows are given by the x^* values.

If the maximum flow is less than $2y_j^*$, then a violated inequality has been identified along the sets S and T separated by the minimum cut. The complexity of this procedure is $O(|T||V|^2|E|^2)$, assuming the Maximum Flow Problem is solved by the Edmonds-Karp algorithm (1972).

Strong connectivity constraints: Violated members of the strong connectivity constraints can be identified by solving a Maximum Flow Problem for each pair of elements from $|T|$, in $O(|T|^2|V||E|^2)$ time with the Edmonds-Karp algorithm. We solve a Maximum Flow Problem for each consecutive pair of vertices from any ordering of the elements of T , which reduces the complexity to $O(|T||V||E|^2)$ and does not result in a significant loss of quality in the lower bound.

Strong 2-matching constraints: To identify the violated strong 2-matching inequalities, we use the heuristic of Padberg and Rinaldi (1990). We first identify all blocks of the graph induced by the fractional flow variables. We then take each block to be H and try to find a violated inequality by adding edges to E' , which have an endpoint in H and another in $V \setminus H$, in a greedy manner.

5 Tabu Search Heuristic

Although the linearization scheme presented in the previous section gives way to a branch-and-cut algorithm, we find it necessary to devise a heuristic for the large instances that may exceed the memory and computing time requirements. We now give the outline of our tabu search heuristic for ATSP.

We define the insertion neighborhood to be the set of vertices not included in the tour. The deletion neighborhood is defined as the set of vertices included in the tour but not in the tabu list.

Parameters:

κ : Number of iterations since the last update of the best solution value.

η : The maximum number of iterations without updating the best solution.

λ : A parameter for determining the frequency of diversification.

θ : The number of iterations a vertex stays in the tabu list.

Step 1 (Initialization). Construct a random sequence of the vertices in $T \setminus \{v_0\}$, and append $\{v_0\}$ to the beginning and end of the sequence to construct a tour. Resequence the tour with the GENIUS algorithm of Gendreau, Hertz, and Laporte (1992) to minimize the total travel time. If the this time is larger than L , stop: the

algorithm has failed to find any feasible solution. Else, record the tour as the best solution. Set $\kappa = 1$.

Step 2 (Search type). If $\kappa = \eta$, stop. Else if κ is a multiple of $\lambda\theta$ go to Step 5.

Step 3 (Local search – insertion). For every vertex v in the insertion neighborhood, try to insert v in the tour and resequence the tour to minimize the total travel time. If the tour is feasible and has a solution value higher than the best solution, replace the best solution with the tour at hand, and set $\kappa = 1$. If v is not an element of the tabu list, the tour is feasible, and has a solution value higher than the best candidate, set v as the best insertion candidate. Insert the best insertion candidate in the tour, add it to the tabu list, and go to Step 6. If no such candidate is found, go to Step 4.

Step 4 (Local search – deletion). For every vertex v in the deletion neighborhood and not in the tabu list, try to delete v from the tour and resequence the tour to minimize the total travel time. If the resulting tour is feasible, and has a solution value higher than the best candidate, set v as the best deletion candidate. Delete the best deletion candidate from the tour, add it to the tabu list, and go to Step 6.

Step 5 (Diversification). Sort the elements $v \in V \setminus T$ that are not in the tour in ascending number of times each element has been added to the tour. For each element v in the sorted list, insert v in the tour and resequence the tour to minimize the total travel time. If the tour is feasible, add v to the tabu list, and go to Step 6. If no insertions are possible, identify a vertex $v \in V \setminus T$ in the tour that has been removed from the tour the least number of times. Remove v from the tour, add v to the tabu list, resequence the tour to minimize the total travel time, and go to Step 6.

Step 6 (Tabu list update). Increase the tabu tenure of each vertex in the tabu list by one. Remove from the tabu list the vertices having a tabu tenure greater than or equal to θ . Increase κ by 1. Go to Step 2.

6 Computational Results

We have implemented a branch-and-cut algorithm based on ATSP2 and on the valid inequalities presented in Propositions 3, 4, and 5. The algorithm was implemented using CPLEX 10.0.1 on a workstation with a 3.0 Ghz CPU and 1 GB of RAM. We have used $q = 2$ in (2). We have attempted to solve three sets of randomly created problem instances for $|V| = 50, 100, \text{ and } 150$. For each instance set, varying

values of $|W|$ and $|T|$ were imposed to analyze the effect of these parameters on the performance of the algorithm. Five instances for each setting have been created, resulting in a total of 180 instances. A computing time limit of one hour was imposed on the solution of any instance. The results are given in Tables 1, 2, and 3, respectively. The column headings are defined as follows:

Succ. : Number of instances successfully solved to optimality.

Avg. Dev. : Average deviation of the best solution found from the best upper bound.

Max. Dev. : Maximum deviation of the best solution found from the best upper bound.

Linearization : Average number of linearization constraints added.

Arc-vertex : Average number of arc-vertex constraints added.

Str. Conn. : Average number of strong connectivity constraints added.

Str. 2-Match : Average number of strong 2-matching constraints added.

Nodes : Average number of nodes generated in the branch-and-cut tree.

Tour size : Average number of vertices included in the best solution found.

Seconds : Average CPU time in seconds.

Opt. : Optimal objective function value.

Computational results presented in Tables 1, 2 and 3 offer insights in the internal mechanics of the problem. As expected, the number of linearization constraints increases with $|W|$. The number of arc-vertex inequalities decreases as $|T|$ increases, suggesting that the underlying LP cannot exploit the fractional variables for large values of $|T|$. Another conclusion we can draw from the tables is that the problem is harder for lower values of $|T|$. This can be explained by the fact that fewer y_i variables are fixed when $|T|$ is small and by the increase in the number of strong connectivity constraints. We note that CPLEX generated few cover and generalized upper bound cover inequalities. Finally, the problem is harder for smaller values of optimal tour size, and hence smaller values of L . This can be explained as follows. First observe that the slope of the curve in Figure 1 is larger in the region closer to the origin. This implies that even small changes in the value of w_k in this interval cause important changes in the overall objective function value. Hence, the underlying LP has more opportunities to exploit the fractional variables. As w_k increases beyond a certain point, the marginal changes do not affect the overall objective function value, and the problem becomes easier.

To ensure the repeatability of our results, we have also conducted experiments on instances adapted from TSPLIB (Reinelt, 1991). We have used the following scheme to convert the data for our problem. We take the first vertex in the data file to be the depot. We designate the next $|T| - 1$ vertices together with the first vertex to constitute T . The next $|V| - |T|$ vertices are used as elements of $V \setminus T$. The remaining vertices are the elements of W . The attractiveness values of the facility vertices and the profit values of the customer vertices are computed as $(\lfloor X + Y \rfloor \bmod 100) + 1$. Let $X_{max}, X_{min}, Y_{max}, Y_{min}$ denote the maximum X coordinate, minimum X coordinate, maximum Y coordinate, and minimum Y coordinate of all vertices, respectively. We compute d_{ij} using the Euclidean distance formula, and set $t_{ij} = d_{ij}$, $r_i = (X_{max} - X_{min} + Y_{max} - Y_{min})/10$ for all $v_i \in V$, and $L = r|T| + 2.5(X_{max} - X_{min} + Y_{max} - Y_{min})$. The self-attraction value of each customer vertex is set to the value that will ensure that 70% of the profit is captured if only the closest facility vertex is visited. The results of the instance files constructed in this manner are presented in Table 4.

We have also implemented the tabu search heuristic described in Section 5. In our computational experiments, we have set the tabu tenure limit θ to 10, the diversification parameter λ to 3, and the iteration limit η to 300. The results of this algorithm are given in Table 5. Note that in all of the 180 instances tested, the overall maximum deviation from the upper bound is about 3.25% and the overall average deviation is 0.28%. The computing times were no more than four minutes. We have also applied the tabu search algorithm to the instances adapted from the TSPLIB. The longest of those runs took about 15 seconds. The maximum deviation was 0.74% and the average deviation was 0.04%. We were able to find an optimal solution in 31 out of the 36 instances.

7 Extension: More Than One Service Option

In closing, we define an interesting extension of the ATSP. In real world applications, there may be more than one option for visiting a facility vertex. These options may differ in service time length and the attractiveness values they yield. For example, a circus may be given the opportunity to select a stay of one week or two weeks at a certain facility. In the military example, the reconnaissance vehicle may opt to stay at an observation point for a longer period to gather more information. Similarly, a mobile medical team may opt for various lengths of stay. Let r_{il} denote the service time for vertex i and option l and a_i^l denote the attractiveness of facility

at vertex i for option l . It is clear that for service options α and β at vertex i , if $S_i^\alpha \geq S_i^\beta$ and $r_{i\alpha} \leq r_{i\beta}$, then option α *dominates* option β . This implies that we can safely assume without loss of generality, that a longer service time results in a higher attractiveness.

There are two possible ways of constructing a model for the extension. The first is the insertion of a copy of the vertex in the graph for each different service option. This path of implementation would also require that no more than one copy of a facility vertex may be visited. The second possibility consists of splitting the variable y_i into m_i parts, where m_i denotes the number of options available at vertex v_i . Namely, let \bar{y}_{il} be equal to 1 if vertex v_i is visited with option l , and 0 otherwise. Note that $\sum_{l=1}^{m_i} \bar{y}_{il} = y_i$. This modeling option is more convenient in terms of implementation and causes a smaller increase in the problem size. We now give the updated linearization for the extension.

(ATSP3)

$$\text{maximize } \sum_{v_k \in W} P_k z_k \quad (21)$$

subject to

$$\sum_{v_i \in V, i < j} x_{ij} + \sum_{v_i \in V, i > j} x_{ji} = 2 \sum_{l=1}^{m_j} \bar{y}_{jl} \quad (v_j \in V) \quad (22)$$

$$\sum_{\substack{v_i \in S, v_j \in V \setminus S \\ \text{or } v_i \in V \setminus S, v_j \in V}} x_{ij} \geq 2 \sum_{l=1}^{m_t} \bar{y}_{tl} \quad (S \subset V : 2 \leq |S| \leq |V| - 2, T \setminus S \neq \emptyset, v_t \in S) \quad (23)$$

$$\sum_{v_i, v_j \in V} t_{ij} x_{ij} + \sum_{v_i \in V} \sum_{l=1}^{m_i} \bar{y}_{il} \leq L \quad (24)$$

$$\sum_{l=1}^{m_i} r_{il} \bar{y}_{il} = 1 \quad (v_i \in T) \quad (25)$$

$$\sum_{l=1}^{m_i} r_{il} \bar{y}_{il} \leq 1 \quad (v_i \in V \setminus T) \quad (26)$$

$$z_k \leq \frac{b_k \left(\sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} \left(\sum_{l=1}^{m_i} \bar{y}_{il} \right) \right)}{\left(b_k + \sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} \left(\sum_{l=1}^{m_i} \bar{y}_{il}^* \right) \right)^2} + \frac{\left(\sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} \left(\sum_{l=1}^{m_i} \bar{y}_{il}^* \right) \right)^2}{\left(b_k + \sum_{v_i \in V \setminus \{v_0\}} \frac{a_i}{d_{ki}^q} \left(\sum_{l=1}^{m_i} \bar{y}_{il}^* \right) \right)^2} \quad (27)$$

$$y_{il} = 0 \text{ or } 1 \quad (v_i \in V, l \in \{1, 2, \dots, m_i\}) \quad (28)$$

$$x_i = 0 \text{ or } 1 \quad ((v_i, v_j) \in E). \quad (29)$$

Note that the strong connectivity constraints are not affected by this extension since they do not involve y variables; similarly, the arc-vertex constraints and strong 2-matching constraints are still applicable after the transformation $\sum_{l=1}^{m_i} \bar{y}_{il} = y_i$.

For the computational experiments with ATSP3, we have constructed a new set of instances using the TSPLIB instances. The self-attraction of each customer vertex is set to the value that will ensure that 40% of the profit is captured if only the closest facility vertex is visited for the first (shortest) option, computed using (2). Facility vertices can be visited for one, two, or three time units. We have assumed that the unit k of the stay has 0.9^{k-1} times the attractiveness of the facility. The results of the computational experiments with ATSP3 are given in Table 6. The new column headings are defined as follows:

Best Sol. : Objective value of the best solution found by the branch-and-cut algorithm.

B & C Dev. : Deviation of the best solution found by the branch-and-cut algorithm from the best upper bound.

TS Dev. : Deviation of the best solution found by the tabu search algorithm from the best upper bound.

Seconds : CPU time for the branch-and-cut algorithm in seconds.

The problem becomes harder as the number of possible choices increase. As a result the computation times increase a hundredfold on average when compared with Table 4. As with the original problem, the hardest instances occur when $|T| = 1$. We were able to solve 31 instances out of 36 to optimality within one hour of computing time. The maximum observed optimality gap is 12.34% for one of the instances.

We have adapted the tabu search algorithm described in Section 5 to cope with the extended problem, by simply adding a copy of each vertex to the graph for each extra unit of stay, with the associated attraction value modified appropriately. We would like to comment here that this approach is not feasible for the case when extra units of stay are actually more attractive than the first ones, i.e., the marginal gain is increasing. However, this case is not likely to be observed in practice. The deviations of the results of the tabu search algorithm are also included in Table 6. The average deviation is 1.92%, whereas the maximum run time is about a minute.

8 Conclusion

We have defined, analyzed, and solved a variant of the TSP, where the profit is attracted from customer vertices by visiting facility vertices, rather than collected by visiting customer vertices. We have used a non-linear gravity demand allocation function to formulate the problem. A linearization scheme was devised using the linear tangents of the concave portions of the objective function as valid inequalities, and a branch-and-cut algorithm, as well as a tabu search heuristic were implemented. We have also analyzed an extension of the original problem where more than one service option is allowed. The solution methods we have developed for the original problem can be adapted to the extended version in a straightforward manner. We have conducted computational experiments on randomly generated instances and on instances derived from TSPLIB. The solution values of our algorithms do not deviate on the average by more than a few percents from the best upper bound value.

Acknowledgments: This work was partially funded by the Canadian Natural Sciences and Engineering Research Council under grants 227837-04 and 39682-05. This support is gratefully acknowledged. The authors thank Stefan Ropke for providing the code for an implementation of the Edmonds-Karp algorithm for the Maximum Flow Problem.

Bibliography

- [1] D.R. Bell, T.-H. Ho, and C.S. Tang, Determining Where To Shop: Fixed and Variable Costs of Shopping, *Journal of Marketing Research* 35 (1998), 352-370.
- [2] H.P. Benson, On the Global Optimization of Sums of Linear Fractional Functions over a Convex Set, *Journal of Optimization Theory and Applications* 121 (2004), 19-39.
- [3] T. Drezner, Competitive Facility Location in the Plane, *Facility Location: A Survey of Applications and Methods*, Z. Drezner (Editor), Springer, Berlin (1995), pp 285-300.
- [4] T. Drezner and Z. Drezner, A Note on Applying the Gravity Rule to the Airline Hub Problem, *Journal of Regional Science* 41 (2001), 67-73.
- [5] T. Drezner and Z. Drezner, Validating the Gravity-Based Competitive Location Model Using Inferred Attractiveness, *Annals of Operations Research* 111 (2002), 227-237.
- [6] T. Drezner and Z. Drezner, Finding the Optimal Solution to the Huff Based Competitive Location Model, *Computational Management Science* 1 (2004), 193-208.
- [7] T. Drezner and Z. Drezner, Multiple Facilities Location in the Plane Using the Gravity Model, *Geographical Analysis* 38 (2006), 391-406.
- [8] T. Drezner and Z. Drezner, The Gravity p -Median Model, *European Journal of Operational Research* 179 (2007), 1239-1251.
- [9] J. Edmonds and R.M. Karp, Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems, *Journal of the Association for Computing Machinery* 19 (1972), 248-264.
- [10] H.A. Eiselt and G. Laporte, Demand Allocation Functions, *Location Science* 6 (1998), 175-187.
- [11] D. Feillet, P. Dejax, M. Gendreau, Traveling Salesman Problems With Profits, *Transportation Science* 39 (2005), 188-205.

- [12] M. Fischetti, J. J. Salazar González, and P. Toth, Solving the Orienteering Problem through Branch-And-Cut. *INFORMS Journal on Computing* 10 (1998), 133-148.
- [13] R.W. Freund and F. Jarre, Solving the Sum-of-Ratios Problem by an Interior-Point Method, *Journal of Global Optimization* 19 (2001), 83–102.
- [14] J.M. Gauthier and G. Ribière, Experiments in Mixed-Integer Linear Programming Using Pseudo-Costs, *Mathematical Programming* 12 (1977), 26-47.
- [15] M. Gendreau, G. Laporte, and F. Semet, The Covering Tour Problem, *Operations Research* 45 (1997), 568-576.
- [16] M. Gendreau, G. Laporte, and F. Semet, A Tabu Search Heuristic for the Undirected Selective Traveling Salesman Problem, *European Journal of Operational Research* 106 (1998), 539-545.
- [17] M. Gendreau, A. Hertz, and G. Laporte, New Insertion and Postoptimization Procedures for the Traveling Salesman Problem, *Operations Research* 40 (1992), 1086–1094.
- [18] B.L. Golden, L. Levy, and R. Vohra, The Orienteering Problem, *Naval Research Logistics* 34 (1987), 307–318.
- [19] M.J. Hodgson, G. Laporte, and F. Semet, A Covering Tour Model for Planning Mobile Health Care Facilities in Suhum District, Ghana, *Journal of Regional Science* 38 (1998), 621–638.
- [20] D.L. Huff, Defining and Estimating A Trade Area, *Journal of Marketing* 28 (1964), 34-38.
- [21] D.L. Huff, A Programmed Solution for Approximating an Optimum Retail Location, *Land Economics* 42 (1966) 293-303.
- [22] A.K. Jain and V. Mahajan, Evaluating the Competitive Environment in Retailing Using Multiplicative Competitive Interactive Models, *Research in Marketing* 2, J. Sheth (Editor), JAI Press, Greenwich, 1979, pp. 217–235.
- [23] S. Kataoka and S. Morito, An Algorithm for the Single Constraint Maximum Collection Problem, *Journal of Operations Research Society of Japan* 31 (1988), 515-530.

- [24] M. Labbé, G. Laporte, I. Rodriguez-Martín, and J. J. Salazar González, The Ring Star Problem: Polyhedral Analysis and Exact Algorithm, *Networks* 43 (2004), 177–189.
- [25] M. Labbé, G. Laporte, I. Rodriguez-Martín, and J. J. Salazar González, Locating median cycles in networks, *European Journal of Operational Research* 160 (2005), 457–470.
- [26] G. Laporte and S. Martello, The Selective Traveling Salesman Problem, *Discrete Applied Mathematics* 26 (1990), 193–207.
- [27] Y. Lee, S.Y. Chiu, and J. Sanchez, A Branch and Cut Algorithm for the Steiner Ring Star Problem, *International Journal of Management Science* 4 (1998), 21–34.
- [28] J.R. Oppong and M. J. Hodgson, Spatial Accessibility to Health Care Facilities in Suhum District, Ghana, *The Professional Geographer* 46 (1994), 199–209.
- [29] M.W. Padberg and G. Rinaldi, Facet Identification for the Symmetric Traveling Salesman Polytope, *Mathematical Programming* 47 (1990), 219–257.
- [30] W.J. Reilly, *The Law of Retail Gravitation*, Knickerbocker Press, New York, 1931.
- [31] G. Reinelt, TSPLIB - A Traveling Salesman Problem Library, *ORSA Journal on Computing* 3 (1991), 376–384.
- [32] S. Schaible, Fractional Programming, *Handbook of Global Optimization*, R. Horst and P. M. Pardalos (Editors), Kluwer, Dordrecht, 1995, pp. 495–608.
- [33] S. Schaible, Fractional Programming with Sums of Ratios, *Scalar and Vector Optimization in Economic and Financial Problems*, E. Castagnoli, and G. Giorgi (Editors), Elioprint, Milano, 1996, pp. 163–175.

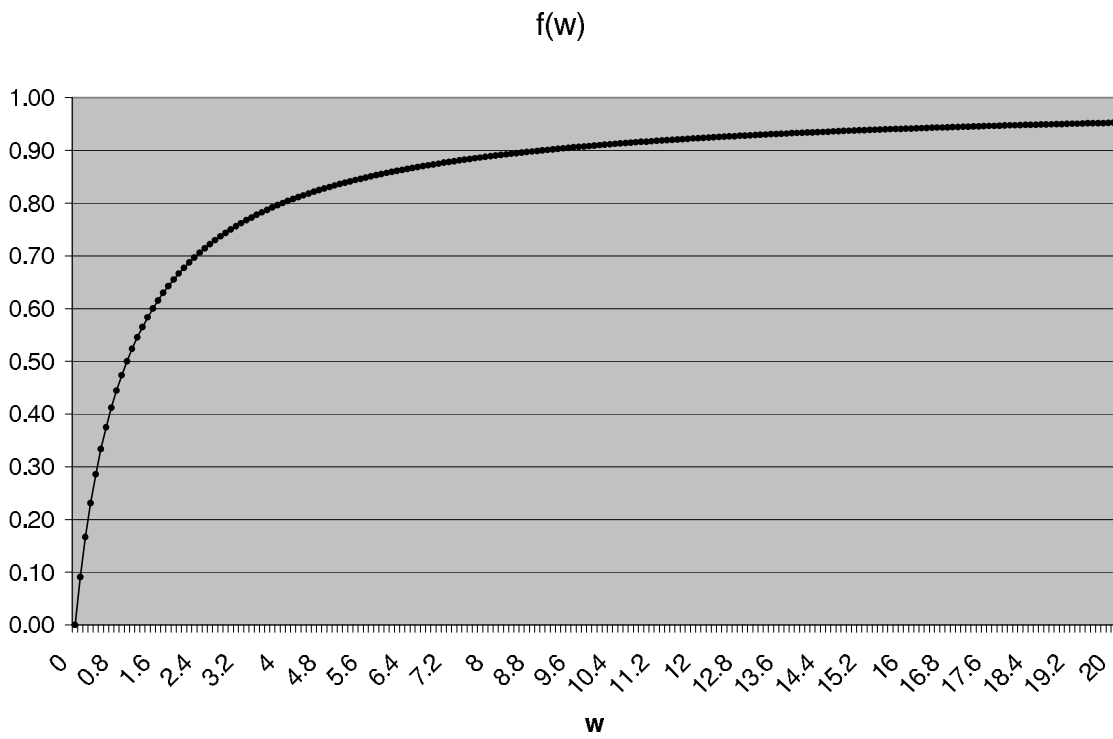


Figure 1: The graph of $f(w_k) = \frac{w_k}{b_k + w_k}$

$ T $	$ W $	Succ.	Avg. Dev.	Max. Dev.	Arc-vertex	Str. Conn.	Str. 2-Match	Lin.	Nodes	Tour size	Seconds
1	50	5	0.00%	0.00%	119.20	0.00	86.20	479.80	384.00	23.00	42.41
1	100	5	0.00%	0.00%	115.40	0.00	81.80	1356.60	614.60	22.80	114.94
1	150	5	0.00%	0.00%	125.60	0.00	182.60	2936.00	1792.60	23.40	668.07
12	50	5	0.00%	0.00%	52.40	182.80	36.00	294.40	260.60	31.60	7.49
12	100	5	0.00%	0.00%	103.15	45.70	96.65	795.40	463.20	31.20	22.26
12	150	5	0.00%	0.00%	81.40	380.60	226.80	1743.00	1619.00	30.60	193.13
25	50	5	0.00%	0.00%	25.60	350.20	201.60	155.20	413.60	41.20	17.39
25	100	5	0.00%	0.00%	31.20	353.00	170.80	422.20	251.80	41.80	12.94
25	150	5	0.00%	0.00%	58.75	262.46	146.37	814.40	244.60	40.80	21.63
37	50	5	0.00%	0.00%	13.40	414.40	332.20	147.20	248.80	44.80	14.12
37	100	5	0.00%	0.00%	15.80	513.40	254.80	356.00	683.80	44.80	44.27
37	150	5	0.00%	0.00%	28.95	378.69	221.15	522.40	52.60	45.20	5.68

Table 1: Average computational results for $|V| = 50$.

$ T $	$ W $	Succ.	Avg. Dev.	Max. Dev.	Arc-vertex	Str. Conn.	Str. 2-Match	Lin.	Nodes	Tour size	Seconds
1	75	4	0.07%	0.34%	291.40	0.00	456.20	1607.60	1596.20	22.20	1738.68
1	150	5	0.00%	0.00%	252.00	0.00	196.60	3399.60	2680.20	23.20	2293.77
1	225	1	0.60%	1.14%	282.80	0.00	203.60	5982.80	1397.00	23.40	3303.87
18	75	5	0.00%	0.00%	115.80	947.00	377.80	738.20	755.00	35.60	193.39
18	150	5	0.00%	0.00%	115.40	817.00	349.40	1913.20	1347.80	35.60	340.45
18	225	5	0.00%	0.00%	118.40	623.40	168.20	2817.20	1598.20	35.40	490.71
37	75	5	0.00%	0.00%	67.00	1105.60	590.80	577.60	2598.80	51.60	323.07
37	150	5	0.00%	0.00%	68.60	1055.60	396.40	1076.80	1142.40	51.00	112.34
37	225	5	0.00%	0.00%	77.00	1389.00	718.00	1747.80	1615.80	50.20	295.21
56	75	5	0.00%	0.00%	10.60	341.60	223.00	166.40	26.20	69.00	4.75
56	150	5	0.00%	0.00%	21.00	780.20	414.40	467.80	188.40	68.40	26.53
56	225	5	0.00%	0.00%	22.40	986.80	558.60	844.60	460.00	66.20	60.07

Table 2: Average computational results for $|V| = 75$.

$ T $	$ W $	Succ.	Avg. Dev.	Max. Dev.	Arc-vertex	Str. Conn.	Str. 2-Match	Lin.	Nodes	Tour size	Seconds
1	100	1	0.69%	2.47%	402.60	0.00	304.20	2077.80	775.60	26.60	3382.10
1	200	0	1.28%	2.02%	360.00	0.00	120.60	4607.60	537.80	28.00	3602.68
1	300	0	2.09%	4.11%	367.40	0.00	101.40	6741.20	236.20	25.20	3602.01
25	100	4	0.02%	0.06%	138.40	1348.80	461.60	1375.40	1128.80	40.40	875.25
25	200	3	0.22%	0.87%	199.40	2448.00	645.40	2959.80	1117.60	41.20	2582.54
25	300	0	0.78%	1.13%	189.60	2577.80	647.20	5424.40	1360.20	41.00	3601.46
50	100	4	0.03%	0.09%	109.20	4104.80	873.80	917.40	2217.40	61.60	1135.70
50	200	5	0.00%	0.00%	81.20	1560.00	456.40	1634.20	778.40	61.40	297.32
50	300	5	0.00%	0.00%	85.00	2160.40	991.00	3166.40	3107.80	61.20	1279.34
75	100	3	0.01%	0.02%	49.00	4928.80	2865.80	404.20	6408.40	82.80	1846.41
75	200	5	0.00%	0.00%	40.20	2526.20	1401.40	832.80	2054.80	82.40	740.44
75	300	5	0.00%	0.00%	35.80	1933.40	858.00	1260.60	810.60	82.00	228.47

Table 3: Average computational results for $|V| = 100$.

Data file	$ V $	$ T $	$ W $	Opt.	Tour size	Seconds	Data file	$ V $	$ T $	$ W $	Opt.	Tour size	Seconds
kroA100.tsp	25	1	75	3208.87	11	34.06	kroB150.tsp	37	18	113	4775.70	23	2.12
kroA100.tsp	25	6	75	3415.33	15	8.16	kroB150.tsp	37	27	113	4907.78	30	0.99
kroA100.tsp	25	12	75	3491.10	19	1.72	kroB200.tsp	50	1	150	5158.18	13	147.86
kroA100.tsp	25	18	75	3562.48	24	0.06	kroB200.tsp	50	12	150	5122.27	19	627.50
kroA150.tsp	37	1	113	4781.18	12	381.00	kroB200.tsp	50	25	150	5725.12	30	10.31
kroA150.tsp	37	9	113	5083.13	17	106.79	kroB200.tsp	50	37	150	6001.81	39	2.25
kroA150.tsp	37	18	113	5319.77	24	25.27	kroC100.tsp	25	1	75	2910.12	11	10.48
kroA150.tsp	37	27	113	5417.65	31	1.42	kroC100.tsp	25	6	75	3055.04	15	2.86
kroA200.tsp	50	1	150	5571.03	13	2047.76	kroC100.tsp	25	12	75	3119.39	17	2.84
kroA200.tsp	50	12	150	5777.59	18	83.27	kroC100.tsp	25	18	75	3237.36	22	0.32
kroA200.tsp	50	25	150	6334.22	28	11.13	kroD100.tsp	25	1	75	3002.30	13	8.15
kroA200.tsp	50	37	150	6584.07	38	3.75	kroD100.tsp	25	6	75	3094.04	16	0.89
kroB100.tsp	25	1	75	3074.84	12	1.35	kroD100.tsp	25	12	75	3146.34	19	0.14
kroB100.tsp	25	6	75	3046.43	14	5.08	kroD100.tsp	25	18	75	3170.25	23	0.04
kroB100.tsp	25	12	75	3149.02	19	0.32	kroE100.tsp	25	1	75	3311.27	12	3.77
kroB100.tsp	25	18	75	3194.25	23	1.15	kroE100.tsp	25	6	75	3313.15	14	0.91
kroB150.tsp	37	1	113	4278.50	12	513.81	kroE100.tsp	25	12	75	3393.93	19	2.18
kroB150.tsp	37	9	113	4553.95	17	42.83	kroE100.tsp	25	18	75	3474.43	24	0.11

Table 4: Computational results for the instances adapted from TSPLIB.

$ V $	$ T $	$ W $	Succ.	Avg. Dev.	Max. Dev.	Tour size	Seconds
50	1	50	2	0.09%	0.23%	22.60	6.08
50	1	100	3	0.08%	0.24%	22.40	8.43
50	1	150	2	0.11%	0.36%	22.20	11.85
50	12	50	2	0.03%	0.09%	31.20	9.56
50	12	100	0	0.08%	0.25%	30.80	10.94
50	12	150	1	0.10%	0.18%	30.20	10.89
50	25	50	3	0.03%	0.07%	41.20	6.73
50	25	100	2	0.06%	0.11%	41.60	8.13
50	25	150	3	0.06%	0.16%	40.80	10.77
50	37	50	4	0.01%	0.02%	44.60	4.36
50	37	100	3	0.06%	0.20%	44.60	6.35
50	37	150	4	0.01%	0.03%	45.20	5.32
75	1	75	3	0.14%	0.51%	21.80	15.21
75	1	150	2	0.24%	0.41%	22.40	22.51
75	1	225	0	1.09%	1.98%	21.80	25.05
75	18	75	0	0.08%	0.19%	35.20	23.38
75	18	150	3	0.07%	0.18%	35.20	30.07
75	18	225	2	0.12%	0.37%	35.20	30.91
75	37	75	1	0.06%	0.08%	51.60	28.52
75	37	150	3	0.06%	0.24%	51.00	26.98
75	37	225	1	0.06%	0.18%	50.20	30.81
75	56	75	2	0.07%	0.20%	68.60	13.38
75	56	150	2	0.03%	0.05%	68.20	15.79
75	56	225	0	0.11%	0.17%	65.80	14.82
100	1	100	0	0.73%	1.96%	25.60	27.97
100	1	200	0	1.66%	2.66%	25.40	29.42
100	1	300	0	2.16%	3.25%	23.80	60.86
100	25	100	0	0.26%	0.37%	40.00	41.98
100	25	200	0	0.39%	1.14%	40.60	57.40
100	25	300	0	0.95%	1.44%	40.20	93.39
100	50	100	0	0.27%	0.53%	61.00	73.59
100	50	200	0	0.19%	0.34%	61.00	88.28
100	50	300	0	0.17%	0.40%	61.00	100.93
100	75	100	0	0.14%	0.40%	82.60	36.21
100	75	200	1	0.16%	0.43%	82.20	60.87
100	75	300	0	0.12%	0.20%	82.00	62.63

Table 5: Average computational results for the tabu search heuristic.

Data file	$ V $	$ T $	$ W $	Best Sol.	B&C Dev.	TS Dev.	Tour size	Seconds
kroA100.tsp	25	1	75	2356.94	3.38%	4.84%	9	3600.07
kroA100.tsp	25	6	75	2588.61	0.00%	0.45%	10	550.52
kroA100.tsp	25	12	75	2725.50	0.00%	0.06%	16	264.75
kroA100.tsp	25	18	75	2879.12	0.00%	0.04%	20	385.60
kroA150.tsp	37	1	113	3507.38	6.79%	7.24%	7	3600.16
kroA150.tsp	37	9	113	3882.47	0.00%	0.74%	14	418.71
kroA150.tsp	37	18	113	4166.33	0.00%	0.00%	20	37.11
kroA150.tsp	37	27	113	4268.37	0.00%	0.00%	30	100.44
kroA200.tsp	50	1	150	3695.43	12.34%	13.73%	10	3600.49
kroA200.tsp	50	12	150	3938.35	0.00%	0.71%	17	1508.07
kroA200.tsp	50	25	150	4545.33	0.00%	0.00%	28	748.35
kroA200.tsp	50	37	150	4914.69	0.00%	1.32%	38	4.50
kroB100.tsp	25	1	75	2574.24	0.00%	5.13%	7	5.26
kroB100.tsp	25	6	75	2392.91	0.00%	0.00%	11	17.62
kroB100.tsp	25	12	75	2507.46	0.00%	0.00%	15	19.84
kroB100.tsp	25	18	75	2599.71	0.00%	0.00%	20	209.94
kroB150.tsp	37	1	113	3041.24	7.64%	10.47%	8	3600.16
kroB150.tsp	37	9	113	3282.68	0.00%	0.31%	15	320.64
kroB150.tsp	37	18	113	3525.57	0.00%	0.00%	22	88.78
kroB150.tsp	37	27	113	3700.55	0.00%	0.00%	29	8.97
kroB200.tsp	50	1	150	3589.77	5.31%	7.28%	10	3600.30
kroB200.tsp	50	12	150	3537.65	0.00%	0.27%	20	1984.54
kroB200.tsp	50	25	150	4209.82	0.00%	0.01%	30	1700.38
kroB200.tsp	50	37	150	4597.72	0.00%	0.00%	38	26.76
kroC100.tsp	25	1	75	2114.73	0.00%	3.56%	8	1341.87
kroC100.tsp	25	6	75	2287.62	0.00%	2.32%	10	310.89
kroC100.tsp	25	12	75	2303.60	0.00%	0.00%	15	43.10
kroC100.tsp	25	18	75	2524.49	0.00%	1.31%	18	0.87
kroD100.tsp	25	1	75	2335.89	0.00%	1.25%	8	788.75
kroD100.tsp	25	6	75	2435.53	0.00%	0.55%	12	190.42
kroD100.tsp	25	12	75	2522.17	0.00%	1.10%	15	1.24
kroD100.tsp	25	18	75	2642.82	0.00%	0.00%	19	6.66
kroE100.tsp	25	1	75	2618.69	0.00%	3.53%	7	90.49
kroE100.tsp	25	6	75	2561.26	0.00%	2.70%	10	9.03
kroE100.tsp	25	12	75	2659.52	0.00%	0.31%	16	2.85
kroE100.tsp	25	18	75	2766.34	0.00%	0.00%	22	46.59

Table 6: Computational results for the extended problem.