# Branch and Bound Algorithm for a Single Vehicle Routing Problem with Toll-by-Weight Scheme

Zizhen Zhang[1], Hu Qin[1,⋆], Andrew Lim[1], and Songshan Guo[2]

[1] Department of Management Sciences, City University of Hong Kong, Tat Chee Ave, Kowloon Tong, Hong Kong
{zizzhang,tigerqin,lim.andrew}@cityu.edu.hk

[2] Department of Computer Science, School of Information Science and Technology, Sun Yat-Sen University, Guangzhou, Guangdong, PR China
issgssh@mail.sysu.edu.cn

**Abstract.** Most of previous studies on vehicle routing problems assume that traversal cost of each edge is simply equivalent to a constant number. Unfortunately, the models of this kind can not be applied in China because toll per kilometer of Chinese expressways varies with vehicle's weight. Motivated by rapidly increasing market of expressway transportation in China, we address a new and special vehicle routing problem that takes a single vehicle and its weight into account. To solve this problem practically, we provide a branch-and-bound algorithm with a well-designed lower bound. This algorithm can deal with any toll scheme in which toll per unit distance monotonically increases with weight. Computational results show that test instances with up to 42 vertices can be solved in reasonable computing time.

**Keywords:** vehicle routing problem; toll-by-weight; branch and bound.

## 1 Introduction

Expressway transportation has been playing an important role in promoting rapid growth of Chinese economy. By the end of 2008, China's expressway network has a total mileage of over 60,000 kilometers and its overall construction cost exceeds 240 billion US dollars. In some countries, such as Australia and United States, the majority of expressways are state-owned and toll-free. But in China, almost all of expressways are owned by for-profit corporations that raise construction funds from securities markets or banks, and recoup investments through tolls.

Tolls of Chinese expressways have been levied on basis of vehicle's type and traveling distance for a long time. This toll scheme drives transportation companies or individuals to overload their vehicles for more economic benefits and

⋆ Corresponding author.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part III, LNAI 6098, pp. 179–188, 2010.

accordingly leads to serious damages of expressways, dangerous driving or illegal modification of vehicles. To present a fairer method and prevent overloading, since 2003, over twenty Chinese provinces have implemented a new toll scheme, referred to as toll-by-weight, whereby expressway tolls are collected based on vehicle's weight and traveling distance. Despite the existence of various toll schemes all over the world, apart from China, few countries adopt the toll-by-weight scheme.

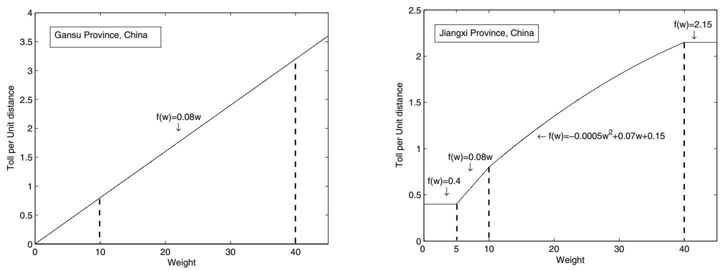
Most of previous studies on vehicle routing problems assume that traversal cost of each edge is simply equivalent to edge length (constant number), irrelevant to vehicle's weight [1]. Thus, one objective of these problems is to minimize the overall travel distance of single or multiple vehicles. Obviously, in China, this assumption is not reasonable because vehicle's traversal cost varies with its weight. In this paper, we investigate a new and special vehicle routing problem that takes a single vehicle and its weight into account, called the single vehicle routing problem with toll-by-weight scheme (SVRPTS). Since in more than twenty Chinese provinces all expressway transportation plans have to consider toll-by-weight schemes, the SVRPTS has innumerable applications.

## 2 Problem Description and Literature

The SVRPTS models the following scenario. Let $G = (V, E)$ be a complete and undirected graph, where $V = \{v_0, v_1, \ldots, v_n, v_{n+1}\}$ is the vertex set and $E = \{e_{i,j} = (v_i, v_j) : v_i, v_j \in V, i \neq j\}$ is the edge set. Vertex $v_0$ and $v_{n+1}$ represent the exit from and the entrance to the depot, and $V_C = \{v_1, \ldots, v_n\}$ denotes the set of $n$ customers. Each customer $i$ has a demand with weight of $q_i$ ($q_0 = q_{n+1} = 0$) to be delivered from the depot and each edge $e_{i,j}$ has a travel distance $d_{i,j}$, where distance matrix $[d_{i,j}]$ satisfies the triangularity condition. A vehicle with empty weight of $Q_0$ and unlimited capacity is loaded with all customer demands, $Q = \sum_{i=1}^{n} q_i$, at the depot and then successively visits each of vertices exactly once. On arriving vertex $v_i \in V_C$, the vehicle's weight is decreased by $q_i$ to fulfill customer demand. When the vehicle passes through edge $e_{i,j}$ from $v_i$ to $v_j$, its weight is denoted by $w_{i,j}$. Note that $w_{i,j} = 0$ and $w_{j,i} = 0$ if the vehicle's route does not contain edge $e_{i,j}$. Toll function $f(w)$, where $w$ is the vehicle's weight, is applied to all graph edges for calculating transportation cost per unit distance. The traversal cost of edge $e_{i,j}$ paid by the vehicle with weight $w_{i,j}$ is calculated by $d_{i,j} f(w_{i,j})$. The objective of the problem is to find a Hamiltonian path on $G$, starting from $v_0$ and ending at vertex $v_{n+1}$, while minimizing the vehicle's total transportation cost.

Fig. 1 gives two typical toll functions (or schemes) in which the unit of $f(w)$ is Chinese Yuan (RMB) per kilometer and the weight unit is ton. As shown in Fig. 1(a), the tolls in Gansu province are charged directly proportional to the vehicle's weight, namely, the toll per ton per kilometer is a fixed value, 0.08 RMB. Jiangxi province uses a relatively more complex toll scheme (see Fig. 1(b)), which is expressed by the following piecewise nonlinear function:

$$f(w) = \begin{cases} 0 & \text{if } w = 0 \\ 0.4 & \text{if } 0 < w \le 5 \\ 0.08w & \text{if } 5 < w \le 10 \\ -0.0005w^2 + 0.07w + 0.15 & \text{if } 10 < w \le 40 \\ 2.15 & \text{if } w > 40 \end{cases}$$




(a) (b)

**Fig. 1.** (a) Linear toll function. (b) Piecewise nonlinear toll function.

Let us indicate with $r = (v_{r(0)}, v_{r(1)}, \ldots, v_{r(n)}, v_{r(n+1)})$, where $r(0) = 0$, $r(n+1) = n+1$, a SVRPTS route on which vertex $v_{r(i)}$ $(1 \le i \le n)$ appears in position $i$. From now on, $d_{r(i),r(i+1)}$ and $w_{r(i),r(i+1)}$ represent the length and weight of edge $(v_{r(i)}, v_{r(i+1)})$, and $q_{r(i)}$ is the weight of demand at $v_{r(i)}$. The total transportation cost $z(r)$ of a given route $r$ can be calculated by:

$$z(r) = \sum_{i=0}^{n} d_{r(i),r(i+1)} f(w_{r(i),r(i+1)}) = \sum_{i=0}^{n} d_{r(i),r(i+1)} f\Big(Q_0 + \sum_{j=i+1}^{n+1} q_{r(j)}\Big) \quad (1)$$

Toll function $f(w)$ can be linear, that is, $f(w) = \alpha w + \beta$, where $\alpha$ and $\beta$ are non-negative constants. If $\alpha = 0$ and $\beta > 0$, the SVRPTS reduces to the traditional traveling salesman problem (TSP) [2], one of the most intensively studied NP-complete problems. If $\alpha > 0$ and $\beta \ge 0$, according to Proposition 1, the SVRPTS can be transformed to the minimum latency problem (MLP) [3,4,5,6,7], which is also termed the traveling repairman problem [8,9], the traveling deliveryman problem [10,11,12], and the cumulative traveling salesman problem [13]. Three types of solution approaches have been applied to the minimum latency problem, which are exact algorithms [6,8,10,11,12,13], approximation algorithms [3,4,5,7] and meta-heuristics [9].

**Proposition 1.** *The SVRPTS with $f(w) = \alpha w + \beta$ $(\alpha > 0, \beta \ge 0)$ is equivalent to the weighted and unweighted MLP.*

*Proof.* According to expression (1), for any route $r$, we have:

$$\begin{aligned} z(r) &= \sum_{i=0}^{n} d_{r(i),r(i+1)} \Big( \alpha \Big( Q_0 + \sum_{j=i+1}^{n+1} q_{r(j)} \Big) + \beta \Big) \\ &= (\alpha Q_0 + \beta) \sum_{i=0}^{n} d_{r(i),r(i+1)} + \alpha \sum_{i=0}^{n} d_{r(i),r(i+1)} \sum_{j=i+1}^{n+1} q_{r(j)} \\ &= (\alpha Q_0 + \beta) \sum_{i=0}^{n} d_{r(i),r(i+1)} + \alpha \sum_{j=1}^{n+1} q_{r(j)} \sum_{i=0}^{j-1} d_{r(i),r(i+1)} \\ &= (\alpha Q_0 + \beta + \alpha q_{r(n+1)}) \sum_{i=0}^{n} d_{r(i),r(i+1)} + \sum_{j=1}^{n} \alpha q_{r(j)} \sum_{i=0}^{j-1} d_{r(i),r(i+1)} \end{aligned} \quad (2)$$

Letting $q'_{r(i)} = \alpha q_{r(i)}$ for $0 \le i \le n$ and $q'_{r(n+1)} = \alpha Q_0 + \beta + \alpha q_{r(n+1)}$, we rewrite $z(r)$ as:

$$z(r) = \sum_{j=1}^{n+1} q'_{r(j)} \sum_{i=0}^{j-1} d_{r(i),r(i+1)} \quad (3)$$

According to the definition of MLP, $z(r)$ can be viewed as the total weighted latency associated with route $r$, where $q'_{r(j)}$ is the weight of vertex $v_{r(j)}$. So the SVRPTS with such toll function is equivalent to the weighted MLP. As stated by [7], the weighted MLP can be further transformed to the unweighted MLP. We have therefore proved this proposition. □

We can only find two previous articles, [14,15], that study the toll-by-weight scheme in the vehicle routing problem. Both of them apply heuristics to solve their models. As for the SVRPTS, its cases with linear toll functions have been widely investigated since they can be transformed to the TSP and the MLP. However, we can not find literature papers tackling the SVRPTS cases with nonlinear toll functions.

After in-depth analysis, we know that all Chinese provinces that have implemented toll-by-weight schemes have adopted monotonically increasing toll functions. The branch and bound algorithm proposed in this paper is the first exact approach that is able to handle the SVRPTS with any monotonically increasing toll function $f(w)$. The remainder of the paper is structured as follows. Sect. 3 introduces a lower bound of the problem, followed by the description of the branch and bound algorithm in Sect. 4. Computational results are reported in Sect. 5 and concluding remarks are discussed in Sect. 6.

## 3 Lower Bound

The quality of lower bound can influence the performance of the branch and bound greatly. To derive lower bound for the SVRPTS, we first present several preliminary results.

**Lemma 1.** *Let* $(d_0^r, \ldots, d_n^r)$*, where* $d_i^r \le d_{i+1}^r$ *for* $0 \le i \le n-1$*, be increasingly sorted edge lengths of any SVRPTS route* $r = (v_{r(0)}, v_{r(1)}, \ldots, v_{r(n)}, v_{r(n+1)})$*. Given a monotonically increasing toll function* $f(w)$*, we have:*

$$z(r) = \sum_{i=0}^{n} d_{r(i),r(i+1)} f(w_{r(i),r(i+1)}) \ge \sum_{i=0}^{n} d_i^r f(w_{r(i),r(i+1)}) \tag{4}$$

*Proof.* For any route $r$, it is obvious that $w_{r(i),r(i+1)} \ge w_{r(i+1),r(i+2)}$. And monotonically increasing toll function $f(w)$ ensures that $f(w_{r(i),r(i+1)}) \ge f(w_{r(i+1),r(i+2)})$. Thus, we have $f(w_{r(0),r(1)}) \ge f(w_{r(1),r(2)}) \ge \cdots \ge f(w_{r(n),r(n+1)})$. By the well-known *rearrangement inequality* [16], we can deduce this lemma straightforwardly. □

**Lemma 2.** *On a complete and undirected graph* $G = (V, E)$*, where* $V$ *contains* $n$ *vertices,* $T$ *is a minimum spanning tree (MST) and* $T'$ *is an arbitrary spanning tree. Sequence* $(d_1^T, \ldots, d_{n-1}^T)$ *and* $(d_1^{T'}, \ldots, d_{n-1}^{T'})$ *are increasingly sorted edge lengths of* $T$ *and* $T'$*, i.e.,* $d_i^T \le d_{i+1}^T$ *and* $d_i^{T'} \le d_{i+1}^{T'}$ *for* $1 \le i \le n-2$*. There exists* $d_i^T \le d_i^{T'}$ *for* $1 \le i \le n-1$*.*

*Proof.* Letting $e_i^T$ and $e_i^{T'}$ be the edges corresponding to $d_i^T$ and $d_i^{T'}$, the edge sets of $T$ and $T'$ can be denoted by $E(T) = \{e_1^T, \ldots, e_{n-1}^T\}$ and $E(T') = \{e_1^{T'}, \ldots, e_{n-1}^{T'}\}$. According to the property of spanning tree, we know that if any $e_i^{T'} \in E(T')$ is added to $T$, there must exist a unique cycle $(e_i^{T'}, e_1^i, \ldots, e_{k_i}^i, e_i^{T'})$, where $k_i$ is the number of cycle edges and $e_j^i \in E(T)$ for $1 \le j \le k_i$, and the length of $e_j^i$ must be less than or equal to $d_i^{T'}$.

Construct a bipartite graph $G_B = (X, Y, E_B)$, where each vertex in $X$ (or $Y$) corresponds to one edge in $E(T')$ (or $E(T)$). If $e_j^T$ is on the cycle associated with $e_i^{T'}$, i.e., $e_j^T \in \{e_1^i, \ldots, e_{k_i}^i\}$, $E_B$ contains an edge connecting $e_j^T$ and $e_i^{T'}$. Take the example shown in Fig. 2 to illustrate this construction process.

Next, we will show that there exists a perfect bipartite matching on $G_B$. To this end, according to Hall's Theorem [17], we need to show $|N(S)| \ge |S|$ for every $S \subseteq X$, where $N(S)$ is the edge set including all elements of $Y$ adjacent to elements of $S$ For example, if $e_i^{T'} \in S$, $\{e_1^i, \ldots, e_{k_i}^i\} \subseteq N(S)$. Taking $V(S)$ and $V(N(S))$ to indicate the vertex sets associated with $S$ and $N(S)$, we construct a graph $G' = (V', S)$, where $V' = V(S) \cup V(N(S))$. Obviously, this graph is a forest with $c(S)$ components, each of which may be subset of $S$ or an isolated vertex. Since $S$ does not contain cycles, by Corollary 3.1.8 of [18], we can deduce that $|V'| = |S| + c(S)$. In the same way, we can also construct $G'' = (V', N(S))$ and get $|V'| = |N(S)| + c(N(S))$. From the definition of $N(S)$, we can easily find that any pair of vertices connected by one edge in $S$ must be connected by one or a couple of edges in $N(S)$ and thus $c(S) \ge c(N(S))$. Hence, we can prove $|S| \le |N(S)|$ and each $e_i^{T'}$ can exclusively select one edge $e_j^i \in \{e_1^i, \ldots, e_{k_i}^i\}$ such that $d_i^{T'}$ is greater than or equal to the length of $e_j^i$.

Now we prove $d_i^T \le d_i^{T'}$ for $1 \le i \le n-1$ by contradiction. Suppose $d_i^T > d_i^{T'}$. Apparently, $d_i^T$ is at least greater than $i$ elements in $\{d_1^{T'}, \ldots, d_{n-1}^{T'}\}$, i.e., $d_i^T >$

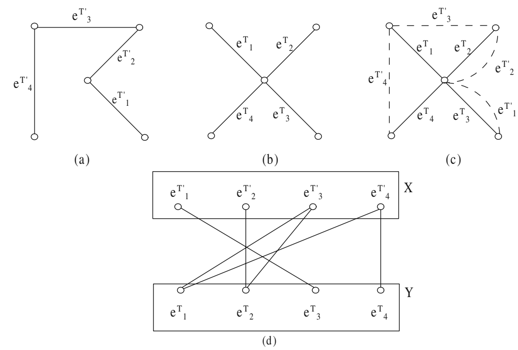



**Fig. 2.** (a) Spanning tree. (b) Minimum spanning tree. (c) Spanning tree (dashed lines) and minimum spanning tree. (d) Bipartite graph.

$d_i^{T'} \geq \ldots \geq d_1^{T'}$. As proved previously, each $e_i^{T'} \in E(T')$ with length $d_i^{T'}$ must exclusively have one edge $e_j^T \in E(T)$ such that $d_i^{T'} \geq d_j^T$. As a result, we can deduce that at least $i$ elements in $\{d_1^T, \ldots, d_{n-1}^T\}$ are less than $d_i^T$. Nevertheless, $d_i^T$ can only be greater than at most $i-1$ elements of $\{d_1^T, \ldots, d_{n-1}^T\}$, leading to a contradiction. Thus, $d_i^T$ must be less than or equal to $d_i^{T'}$ for $1 \leq i \leq n-1$. □

Before continuing to the theorem, we need to define a special tree, called *minimum restricted spanning tree* (MRST).

**Definition 1 (MRST).** *Let $G = (V, E)$ be a complete and undirected graph, where $V = \{v_0, v_1, \ldots, v_n, v_{n+1}\}$. The MRST is constructed by first identifying the MST of $\{v_1, \ldots, v_n\}$ and then connecting vertex $v_0$ and $v_{n+1}$ to their nearest neighbors on the MST.*

In MRST, the vertex $v_0$ and $v_{n+1}$ are restricted to have degree of one. The reader can distinguish the MRST and MST by Fig. 3.

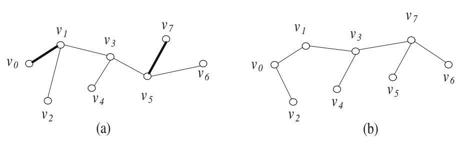



**Fig. 3.** (a) Minimum restricted spanning tree. (b) Minimum spanning tree.

Using the above two lemmas and the MRST, we can obtain a lower bound of the SVRPTS, which is given by the following theorem.

**Theorem 1.** *Let $G = (V, E)$ be a complete and undirected graph with vertex set $\{v_0, v_1, \ldots, v_n, v_{n+1}\}$. $\bar{T}$ is a MRST on $G$. The lengths of $\bar{T}$'s edges incident to $v_0$ and $v_{n+1}$ are indicated with $d_0^{\bar{T}}$ and $d_n^{\bar{T}}$. First, sort the lengths of* *the other edges on $\bar{T}$ increasingly, thereby generating a sequence $(d_1^{\bar{T}}, \ldots, d_{n-1}^{\bar{T}})$. Then, sort the weights of customer demands in descending order and obtain a sequence $(p_1, \ldots, p_n)$, where $p_i \in \{q_1, \ldots, q_n\}$ and $p_i \geq p_{i+1}$ for $1 \leq i \leq n-1$. A polynomially computable lower bound for the SVRPTS is:*

$$LB = d_0^{\bar{T}} f(Q + Q_0) + d_n^{\bar{T}} f(Q_0) + \sum_{i=1}^{n-1} d_i^{\bar{T}} f\Big( \sum_{j=i+1}^{n} p_j + Q_0 \Big) \tag{5}$$

*Proof.* Assume $r$, starting from $v_0$ and ending at $v_{n+1}$, is an optimal SVRPTS route on $G$. After sorting all edge lengths of $r$, except $d_{r(0),r(1)}$ and $d_{r(n),r(n+1)}$, in ascending order, we can obtain a sequence $(d_1^r, d_2^r, \ldots, d_{n-1}^r)$. By Lemma 1, we can get:

$$\begin{aligned} z(r) &= \sum_{i=0}^{n} d_{r(i),r(i+1)} f(w_{r(i),r(i+1)}) \\ &\geq d_{r(0),r(1)} f(w_{r(0),r(1)}) + \sum_{i=1}^{n-1} d_i^r f(w_{r(i),r(i+1)}) + d_{r(n),r(n+1)} f(w_{r(n),r(n+1)}) \end{aligned}$$

According to Definition 1, we know that $d_{r(0),r(1)} \geq d_0^{\bar{T}}$ and $d_{r(n),r(n+1)} \geq d_n^{\bar{T}}$. Since subroute $(v_{r(1)}, v_{r(2)}, \ldots, v_{r(n)})$ is one of spanning trees covering vertex set $\{v_1, v_2, \ldots, v_n\}$, by Lemma 2, we have $d_i^r \geq d_i^{\bar{T}}$ for $1 \leq i \leq n-1$. Subsequently, we can deduce that:

$$z(r) \geq d_0^{\bar{T}} f(w_{r(0),r(1)}) + \sum_{i=1}^{n-1} d_i^{\bar{T}} f(w_{r(i),r(i+1)}) + d_n^{\bar{T}} f(w_{r(n),r(n+1)}) \tag{6}$$

One can easily obtain that $w_{r(0),r(1)} = Q + Q_0$, $w_{r(n),r(n+1)} = Q_0$ and $w_{r(i),r(i+1)} = \sum_{j=i+1}^{n} q_j + Q_0 \geq \sum_{j=i+1}^{n} p_j + Q_0$. Since $f(w)$ is a monotonically increasing function, $f(\sum_{j=i+1}^{n} q_j + Q_0) \geq f(\sum_{j=i+1}^{n} p_j + Q_0)$. Based on the above inequalities, we can complete this proof. □

## 4 Branch-and-Bound Algorithm

Branch-and-bound is a general technique for finding optimal solutions of various combinatorial optimization problems. In our branch-and-bound algorithm, the upper bound of the SVRPTS is firstly generated by the meta-heuristics described in [15] and then updated during the search process. Fig. 4 shows the branch-and-bound search tree. Usually, "vertex" and "node" have the same meaning and are interchangeable. But, in this paper, we specify that "vertex" represents joint point in graph $G$ and "node" is used for search tree. The search tree is explored according to a depth-first policy while the best-first policy is applied to the unexplored nodes at the same tree level.

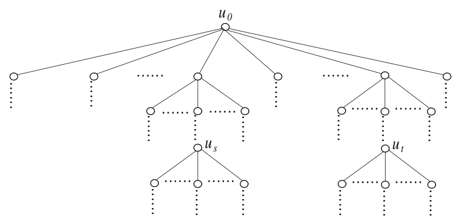



**Fig. 4.** Branch and bound search tree

Given a node $u_s$ at level $m$ of the tree, we can easily identify its path to the root. This path is essentially a partial route of the SVRPTS, denoted by $r_s = (v_{r_s(0)}, \ldots, v_{r_s(m-1)}, v_{r_s(m)})$, where $u_0 = v_{r_s(0)}$ and $u_s = v_{r_s(m)}$. The subproblem related to node $u_s$ is also a SVRPTS with vertex set $V - \{v_{r_s(0)}, \ldots, v_{r_s(m-1)}\}$, starting from $v_{r_s(m)}$ and ending at $v_{n+1}$. The lower bound $LB_s$ associated with node $u_s$ is the sum of the transportation cost on $r_s$ and the lower bound of its subproblem. If $LB_s$ is greater than or equal to the current upper bound, all complete routes sharing $r_s$ can be fathomed, that is, node $u_s$ can be pruned, and the search procedure can backtrack to another part of the tree. Otherwise, we continue to branch and choose the children node of $u_s$ that has the best lower bound.

In addition to lower bound pruning, one tree node can also be eliminated from consideration by dominance pruning. Let us consider two nodes $u_s$ and $u_t$ at a certain level of the tree as well as their paths $r_s$ and $r_t$. Assuming $r_s$ and $r_t$ contain the same set of customers and end at the same customer, it is well-known that if the transportation cost on $r_s$ is less than that on $r_t$, $u_t$ is dominated by $u_s$ and can be eliminated from the search tree. Therefore, before calculating the lower bound of each tree node $u_s$, we fix $u_0$, $u_s$ and employ an $O(n^2)$ heuristics to check permutations of other customers on $r_s$. Therefore, before calculating the lower bound of each tree node $u_s$, we employ an $O(n^2)$ heuristics to check whether this node can be dominated by other nodes.

## 5 Computational Experiments

The algorithm described in this paper was coded in C++ and all experiments were run on a DELL laptop with 2.40 GHz Intel Duo CPU, 4Gb memory and Windows Vista. Computational times reported here were in CPU seconds on this machine.

We generated 15 test instances for the SVRPTS from TSPLIB and VRPLIB files: *ulysses16, ulysses22, gr24, fri26, bayg29, bays29, dantzig42, swiss42, E021, E022, E023, E030, E033, E036, E041*, which can be accessed at: `http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/` and `http://www.or.deis.unibo.it/research_pages/ORinstances/VRPLIB/VRPLIB.html`. Our test instances only use the vertex location information in these files and the first vertex is designated to be the depot. The distance unit and weight unit are assumed to be kilometer and ton, respectively. The weight $Q_0$ of empty vehicle was set to 5 and the demanding weight $q_i$ of each vertex was randomly generated from the uniform distribution in the interval $[0.8, 1.2]$. Experiments were conducted under both toll schemes of Gansu and Jiangxi province that have been described in Sect. 2.

In Table 1, we report computational results for our branch-and-bound algorithm. The first column contains names of the TSPLIB and VRPLIB files. The other columns are divided into two blocks, corresponding to two toll schemes. Each block contains the following information: the optimal objective value (Value), the computing time (Time), the number of explored nodes in the search tree (Nodes), and the ratio of the lower bound to the optimal objective value (Ratio). From this table, we can see that instances with up to 33 vertices can be easily handled. For the instances with over 40 vertices, the computing time and the number of explored nodes increase explosively. The values in column "Time" of Gansu toll scheme is less than that of Jiangxi toll scheme. The reason of this phenomenon is highly probable that the non-linear toll function implemented by Jiangxi province makes instances harder.

**Table 1.** Experimental results for the branch-and-bound algorithm

| | Gansu Toll Scheme | | | | Jiangxi Toll Scheme | | | |
|---|---|---|---|---|---|---|---|---|
| Cases | Value (RMB) | Time(s) | Nodes | Ratio (%) | Value (RMB) | Time(s) | Nodes | Ratio (%) |
| ulysses16 | 57.57 | 0.09 | 14,556 | 61.32 | 54.36 | 0.10 | 23,727 | 62.79 |
| ulysses22 | 68.20 | 2.87 | 229,467 | 58.82 | 62.01 | 4.03 | 351,701 | 61.00 |
| gr24 | 1,517.33 | 1.17 | 37,514 | 76.37 | 1,333.97 | 1.48 | 47,932 | 77.08 |
| fri26 | 1,150.24 | 0.75 | 18,127 | 80.16 | 978.71 | 0.86 | 16,015 | 82.14 |
| bayg29 | 2,295.62 | 25.19 | 574,736 | 76.34 | 1,949.36 | 50.30 | 1,205,874 | 76.88 |
| bays29 | 2,857.27 | 76.11 | 1,925,966 | 71.18 | 2,411.19 | 108.84 | 2,780,570 | 72.34 |
| dantzig42 | 1,176.87 | 644.90 | 4,263,222 | 74.95 | 925.85 | 1,728.64 | 12,361,548 | 77.16 |
| swiss42 | 2,227.24 | 649.66 | 4,737,059 | 75.54 | 1,749.90 | 2,437.59 | 20,244,906 | 77.36 |
| E021 | 298.50 | 0.06 | 2,308 | 82.69 | 265.95 | 0.07 | 2,886 | 83.89 |
| E022 | 329.46 | 0.14 | 5,496 | 75.76 | 290.40 | 0.14 | 5,869 | 78.06 |
| E023 | 512.93 | 0.10 | 3,370 | 72.95 | 454.74 | 0.11 | 3,545 | 75.05 |
| E030 | 525.44 | 4.81 | 134,385 | 64.18 | 439.98 | 5.81 | 170,468 | 67.33 |
| E033 | 582.85 | 3.58 | 37,158 | 75.51 | 481.72 | 7.40 | 81,044 | 76.81 |
| E036 | 575.21 | 179.77 | 2,176,065 | 82.18 | 465.95 | 308.09 | 4,011,228 | 83.07 |
| E041 | 657.87 | 476.92 | 3,913,325 | 83.72 | 512.49 | 972.57 | 8,714,554 | 85.06 |

## 6 Conclusions

Motivated by the toll-by-weight schemes implemented by over twenty Chinese provinces, we propose a single vehicle routing problem in which transportation cost per unit distance monotonically increases with the vehicle's weight. This problem is a new variant of traditional vehicle routing problem and has been studied by few researchers. The branch-and-bound algorithm described in this paper is the first exact algorithm for the problem and is capable of solving to optimality all instances with up to 42 vertices. The study of toll-by-weight scheme in the vehicle routing problem has huge growing space because of its scarceness in the literature and also its wide application in China.

## References


1. Toth, P., Vigo, D. (eds.): The vehicle routing problem. SIAM, Philadelphia (2002)
2. Gutin, G., Punnen, A. (eds.): The Traveling Salesman Problem and Its Variations. Kluwer Academic Publishers, Dordrecht (2002)
3. Blum, A., Chalasani, P., Coppersmith, D., Pulleyblank, B., Raghavan, P., Sudan, M.: The minimum latency problem. In: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing (STOC 1994), Montreal, Quebec, Canada, pp. 163–171 (1994)
4. Goemans, M., Kleinberg, J.: An improved approximation ratio for the minimum latency problem. Mathematical Programming 82(1-2), 111–124 (1998)
5. Arora, S., Karakostas, G.: Approximation schemes for minimum latency problems. SIAM Journal on Computing 32(5), 1317–1337 (2003)
6. Wu, B., Huang, Z., Zhan, F.: Exact algorithms for the minimum latency problem. Information Processing Letter 92(6), 303–309 (2004)
7. Archer, A., Levin, A., Williamson, D.: A faster, better approximation algorithm for the minimum latency problem. SIAM Journal on Computing 37(5), 1472–1498 (2008)
8. García, A., Jodrá, P., Tejel, J.: A note on the traveling repairman problem. Networks 40(1), 27–31 (2002)
9. Salehipour, A., Sörensen, K., Goos, P., Bräysy, O.: An efficient grasp+vnd metaheuristic for the traveling repairman problem. Working papers, University of Antwerp, Faculty of Applied Economics (2008)
10. Minieka, E.: The delivery man problem on a tree network. Annals of Operations Research 18(1-4), 261–266 (1989)
11. Fischetti, M., Laporte, G., Martello, S.: The delivery man problem and cumulative matroids. Operations Research 41(6), 1055–1064 (1993)
12. Méndez-Díaz, I., Zabala, P., Lucena, A.: A new formulation for the traveling deliveryman problem. Discrete Applied Mathematics 156(17), 3223–3237 (2008)
13. Bianco, L., Mingozzi, A., Ricciardelli, S.: The traveling salesman problem with cumulative costs. Networks 23(2), 81–91 (1993)
14. Chen, X., Li, J., Fan, Y., Wu, Y., Ding, L.: Design and implementation of vehicle routing optimization system based on toll-by-weight. Manufacture Information Engineering of China 17, 69–72 (2007)
15. Shen, C., Qin, H., Lim, A.: A capacitated vehicle routing problem with toll-by-weight rule. In: Chien, B., Hong, T. (eds.) Opportunities and Challenges for Next-Generation Applied Intelligence. Studies in Computational Intelligence, vol. 214, pp. 311–316. Springer, Heidelberg (2009)
16. Hardy, G., Littlewood, J., Pólya, G.: Inequalities. Cambridge University Press, Cambridge (1988)
17. Hall, P.: On representatives of subsets. Journal of London Mathematical Society 10, 26–30 (1935)
18. Gross, J., Yellen, J. (eds.): Graph Theorey and Its Applications. Chapman and Hall/CRC (2005)