



Parameterized Algorithms for the Traveling Purchaser Problem with Additional Constraints

Mingyu Xiao^(✉) , Jianan Zhang, and Weibo Lin

University of Electronic Science and Technology of China, Chengdu, China
myxiao@gmail.com, willjnzhang@tencent.com, lweb1688@gmail.com

Abstract. The traveling purchaser problem (TPP), a generalization of the traveling salesman problem, is to determine a tour of suppliers and purchase needed products from suppliers, while minimizing the traveling and purchasing cost. This problem finds applications in the routing and scheduling contexts and its variants with different constraints have been widely studied. Motivated by the phenomenon that most real-world instances of TPP have a small parameter (such as the number of suppliers, the number of products to purchase and others), we study TPP and its variants from the view of parameterized complexity. We show that TPP and some variants are fixed-parameter tractable by taking the number k of products or the number m of suppliers as the parameter, and $W[2]$ -hard by taking the number q of visited suppliers as the parameter. Furthermore, we implement some of our fixed-parameter tractable algorithms to show that they are practically effective when the parameters are not very large.

1 Introduction

The traveling purchaser problem (TPP) is a single vehicle routing problem that has been widely studied. In this problem, we need to buy several products from some suppliers and the objective is to minimize the total amount of traveling and purchasing costs. Let s_0 denote the home, which is the starting and ending point of the tour. We use $\mathcal{M} = \{s_0, s_1, s_2, \dots, s_{m-1}\}$ to denote the set of suppliers together with the home and $\mathcal{K} = \{g_1, g_2, \dots, g_k\}$ to denote the set of products to purchase, where $|\mathcal{M}| = m$ and $|\mathcal{K}| = k$. The input of the problem consists of an $m \times k$ matrix $\mathcal{P} = \{p_{ij}\}$ to indicate the price of product g_j at supplier s_i , where we may let $p_{i_0j_0}$ be ∞ or empty if a supplier s_{i_0} does not provide product g_{j_0} , and an $m \times m$ matrix $\mathcal{D} = \{d_{ij}\}$ to indicate the traveling costs (distances) from site s_i to site s_j . The goal is to find a tour (cycle) starting and ending at home s_0 , visiting a subset of the suppliers in \mathcal{M} to buy all products in \mathcal{K} , while minimizing the composed cost of traveling and purchasing. For the distances between sites, our problem does not require that $d_{ij} = d_{ji}$ (the symmetry assumption). However, we assume that the distances satisfy the triangle inequality, i.e., $d_{ij} \leq d_{il} + d_{lj}$ holds for any i, j, l . We also assume that each supplier has enough amount of

each provided product and then we do not buy a product from two different suppliers.

TPP is NP-hard, since it contains the well-known traveling salesman problem (TSP) as a special case, where each supplier provides only one different product. TPP combines the optimization of routing decisions and supplier selections together and fits well in many contexts, such as routing and scheduling problems. It can be straightforwardly interpreted to machine scheduling problems [9]. An application of the telecommunication network design was proposed in [17]. Many problems in location based services can be formulated as a traveling purchaser problem [12]. More applications of TPP can be found in [16] and [18].

Problem Variants. Due to the importance of TPP, many variants of it have been widely studied. Motivated by a scheduling problem (to assign some jobs to some machines), Gouveia *et al.* [9] considered TPP with two constraints: (I) the maximum number of suppliers to be visited is limited to q , where we can simply assume that $q \leq k$ since the distances satisfy the triangle inequality; (II) the maximum number of products can be bought from each supplier is limited to u . The two constraints are also called *side-constraints*. We will use TPP-S1 to denote TPP with only constraint (I) and TPP-S2 to denote TPP with both two constraints (I) and (II). To model a problem in telecommunication network designs, Ravi and Salman [17] introduced the traveling purchaser problem with *budget-constraint* (TPP-B). In TPP-B, a budget B on the purchasing cost is given, and the goal is to minimize the traveling cost such that we can buy all the products within the budget B . Two heuristic algorithms for this problem were studied in [14]. TPP with time windows can also be found in several real contexts [6, 11]. In this problem each supplier has a time window and it only serves in this time window. Recently, a multi-vehicle variant of TPP, called MVTTP, was introduced by Choi and Lee [5]. In MVTTP, the optimization has to be done over a fleet of homogeneous vehicles instead of a single vehicle, each vehicle in the fleet has the same capacity (the amount of product can be carried on). Several constraints on MVTTP have also been studied, such as the constraint on the traveling distance of each vehicle [4], the incompatibility constraint under which some products are not allowed to be loaded on the same vehicle [13], and so on. MVTTP is related to the vehicle routing problem [19].

In real-world instances of TPP, some values are small. Here are examples: In the model of stocking products from suppliers by a supermarket, the number of different products may be large while the number of suppliers may be small; In the model where a person wants to purchase something in a weekend, the number of potential shops in the city may be large while the number of things to purchase may be small; In some problems, the number of sites to be visited may also be small due to the time limitation and some other reasons; The model from some scheduling problems in [9] also assumes that the number of products is small compared to the number of suppliers. Motivated by the phenomenon of small values of parameters in these problems, we study TPP and its variants in parameterized complexity.

Parameterized Complexity. Parameterized complexity has attracted much attention in both theory and practice since the first introduction of it by Downey and Fellows [7]. An instance of a parameterized problem consists of an instance I of the original (NP-hard) problem and a parameter l . We want to design an algorithm for the problem with running time in the form of $f(l)poly(|I|)$, where $f(l)$ is a computable function on l only, and $poly(|I|)$ is a polynomial function on the input size. These kinds of algorithms are called *fixed-parameter tractable* (FPT) algorithms. A parameterized problem is *fixed-parameter tractable* (FPT) if and only if it has FPT algorithms. Under some reasonable assumptions, some parameterized problems do not allow FPT algorithms, which are called *W[1]-hard*. For FPT algorithms, the running time bound is exponential only on the parameter l and not related to the whole instance size. When the parameter l is small or a constant, FPT algorithms may run fast and solve practical problems exactly in a short time.

In this paper, we will study TPP, TPP with side-constraints (TPP-S1 and TPP-S2) and TPP with the budget-constraint (TPP-B) under three parameters: the number k of products, the number m of suppliers and the maximum number q of suppliers to be visited. For each parameterized problem, we will either design fast FPT algorithms for it or prove the W[1]-hardness or W[2]-hardness, where W[2]-hard problems may be harder than W[1]-hard problems under some reasonable assumptions.

Our Contributions. To the best of our knowledge, this is the first paper that contributes to the parameterized complexity of TPP and its variants. Our results are summarized in Table 1.

Table 1. Our results in parameterized complexity

Problems	Parameters		
	$k = \mathcal{K} $	$m = \mathcal{M} $	q
TPP	FPT (Theorem 1)	FPT (Theorem 4)	W[2]-hard (Theorem 7)
TPP-S1	FPT (Theorem 2)	FPT (Theorem 5)	W[2]-hard (Theorem 7)
TPP-S2	FPT (Theorem 3)	FPT (Theorem 6)	W[2]-hard (Theorem 7)
TPP-B	W[1]-hard (Theorem 8)	FPT (Theorem 5)	W[2]-hard (Theorem 7)

Our hardness results are obtained by reductions from two known hard problems: the set cover problem and the multi-subset sum problem. The main techniques used to design our FPT algorithms are dynamic programming and color coding. In fact, we will design two practical dynamic programming algorithms for TPP, which run in time $O(2^k(m^2 + mk))$ and $O(2^m(m^2 + mk))$ respectively and can be modified for TPP with several additional constraints without exponentially increasing the running time bound. For TPP-S2 parameterized by k , we need to use the color coding technique to design an FPT algorithm.

Our algorithms imply that TPP is polynomial-time solvable when $k = O(\log m)$ or $m = O(\log k)$. This is the reason why we can solve TPP quickly when one of k and m is small. Furthermore, the polynomial part of the running time of most algorithms is small, which is linear on the input size of the problem, because the input size of TPP is $O(m^2 + mk)$.

In practice, our dynamic programming algorithms are effective and easy to implement. Compared with previous algorithms for TPP with additional constraints, our algorithms can quickly solve instances with one of k and m being a small value. To show the advantage of our FPT algorithms in practice, we also implement some of our FPT algorithms to test their experimental performances. However, the experimental part is omitted due to the space limitation and it can be found in the full version of this paper.

2 Algorithms for Small Number of Products

In this section, we will design an $O(2^k(m^2 + mk))$ -time algorithm for TPP and then modify it to an $O(2^k q(m^2 + mk))$ -time algorithm for TPP-S1. Then we give an $O(u^2 m^{2+q} e^q q^{O(\log q)} \log m)$ -time algorithm for TPP-S2, where $e = 2.71828\dots$ is a constant and it holds that $q \leq k$. When we take k as the parameter, the three problems are FPT.

2.1 TPP with Parameter k

First we consider TPP. For each subset $K \subseteq \mathcal{K}$ of products and each supplier $s_i \in \mathcal{M}$, we consider the subproblem Sub-TPP(K, s_i): buy all the products in K from suppliers in a tour starting from home s_0 and ending at s_i , while minimizing the traveling and purchasing cost. Note that in this subproblem, we require that we finally arrive at s_i even we may not buy any product from s_i . If we let $K = \mathcal{K}$ and $s_i = s_0$, then this problem becomes the original TPP problem. Our idea is to solve Sub-TPP(K, s_i) for each $K \subseteq \mathcal{K}$ and $s_i \in \mathcal{M}$ in a dynamic programming method. To solve Sub-TPP(K, s_i) efficiently, we will also solve a variant of Sub-TPP(K, s_i), called Sub-TPP'(K, s_i): buy all the products in K from suppliers in a tour starting from home s_0 and ending at s_i such that at least one product in K is bought from s_i , while minimizing the traveling and purchasing cost. In Sub-TPP'(K, s_i), we have one more constraint that is to buy some product g_j from s_i . We can also assume that the product g_j was bought when visit s_i for the last time, i.e., we buy g_j after arriving the final site s_i of our tour.

Let $SOL[K, s_i]$ denote an optimal solution to Sub-TPP(K, s_i) (the information of the tour and where to buy each product) and $OPT[K, s_i]$ denote the total traveling and purchasing cost of $SOL[K, s_i]$. Let $SOL'[K, s_i]$ and $OPT'[K, s_i]$ denote an optimal solution and the optimal cost to Sub-TPP'(K, s_i), respectively. We have that $OPT[K, s_i] \leq OPT'[K, s_i]$.

There are two cases for an optimal solution $SOL[K, s_i]$: after arriving the final site s_i we buy at least one product $g_{i_0} \in K$ from s_i , and before arriving the final site s_i we have already bought all the products in K . For the former

case, we have that $OPT[K, s_i] = OPT'[K, s_i]$ and we can solve Sub-TPP(K, s_i) by solving Sub-TPP'(K, s_i). For the latter case, we can see that $OPT[K, s_i] = OPT[K, s_j] + d_{ji} \geq OPT[K, s_j]$ for some j , where s_j is the last but one site of the optimal tour in $SOL[K, s_i]$, and we can solve Sub-TPP(K, s_i) by solving Sub-TPP(K, s_j). Thus, we have

$$OPT[K, s_i] = \min\{OPT'[K, s_i], \min_{j \neq i}\{OPT[K, s_j] + d_{ji}\}\}. \tag{1}$$

We will solve Sub-TPP(K, s_i) in order of increasing cardinality of K . For each fixed K and all $s_i \in \mathcal{M}$, we first solve Sub-TPP'(K, s_i) by using the following recurrence relation

$$OPT'[K, s_i] = \min_{g_j \in K} \{OPT[K \setminus \{g_j\}, s_i] + p_{ij}\}, \tag{2}$$

and then compute $OPT[K, s_i]$ based on $OPT'[K, s_i]$ in a greedy method similar to Dijkstra's shortest path algorithm. Note that Eq. (2) allows many products in K to be bought in supplier s_i (not just g_j).

Assume that we have computed $OPT'[K, s_i]$ for a fixed K and all $s_i \in \mathcal{M}$. We are going to compute $OPT[K, s_i]$ for all $s_i \in \mathcal{M}$. Our algorithm will maintain two subsets $M_1, M_2 \subseteq \mathcal{M}$ such that $M_2 = \mathcal{M} \setminus M_1$, where for each $s_{i_0} \in M_1$ we have computed $OPT[K, s_{i_0}]$, and for each $s_{i_0} \in M_2$ we have not. Initially, we have $M_1 = \emptyset$. The algorithm iteratively selects an element $s_j \in M_2$, compute $OPT[K, s_j]$ and move it from M_2 to M_1 until M_2 becomes \emptyset . We select $s_j \in M_2$ such that

$$OPT'[K, s_j] \leq OPT'[K, s_r] \quad \text{for any } s_r \in M_2, \tag{3}$$

and compute $OPT[K, s_j]$ by

$$OPT[K, s_j] = \min\{OPT'[K, s_j], \min_{s_r \in M_1} \{OPT[K, s_r] + d_{rj}\}\}. \tag{4}$$

Next we prove the correctness of (4). Consider a supplier $s_{i_0} \in M_2$. If $OPT[K, s_{i_0}] \geq OPT'[K, s_j]$, then we get that $OPT[K, s_{i_0}] + d_{i_0j} \geq OPT'[K, s_j]$. Otherwise we assume that $OPT[K, s_{i_0}] < OPT'[K, s_j]$ and then $OPT[K, s_{i_0}] < OPT'[K, s_{i_0}]$. By (3) and (1), it holds that $OPT[K, s_{i_0}] = OPT[K, s_{i_l}] + d_{i_l i_{l-1}} + \dots + d_{i_2 i_1} + d_{i_1 i_0}$ for some $s_{i_l} \in M_1$ and $\{s_{i_{l-1}}, s_{i_{l-2}}, \dots, s_{i_0}\} \subseteq M_2$. By the triangle inequality, we get $OPT[K, s_{i_0}] \geq OPT[K, s_{i_l}] + d_{i_l i_0}$ and $OPT[K, s_{i_0}] + d_{i_0j} \geq OPT[K, s_{i_l}] + d_{i_l j}$. By (1) again, we get (4). After computing $OPT[K, s_j]$ according to (4), we move s_j from M_2 to M_1 .

We use A- k to denote the above algorithm for TPP. In this algorithm, we first let $OPT[\emptyset, s_j] = d_{0j}$ for each s_j since the length of the shortest path from s_0 to s_j is d_{0j} by the triangle inequality, and then compute $OPT[K, s_j]$ for $K \neq \emptyset$ in order of nondecreasing size by using the above method.

In this algorithm, we need to solve $O(2^k m)$ subproblems Sub-TPP(K, s_i). For each subproblem, we use $|K| \leq k$ basic computations to solve Sub-TPP'(K, s_i) and use $|\mathcal{M}| = m$ basic computations to compute $OPT[K, s_i]$ based on $OPT'[K, s_i]$. We have the following Theorem 1.

Theorem 1. *TPP can be solved in $O(2^k m(m+k))$ time and it is FPT by taking the number k of products as the parameter.*

2.2 TPP-S1 with Parameter k

Now we consider TPP-S1. In fact, the algorithm for TPP-S1 is modified from the above algorithm for TPP. In our algorithm for TPP-S1, the subproblem has one more input parameter (one more dimension) and the running time of it also increases. For each subset $K \subseteq \mathcal{K}$ of products, each supplier $s_i \in \mathcal{M}$ and each nonnegative integer $q^* \leq q$, we define the subproblem Sub-TPPS1(K, s_i, q^*): buy all the products in K from exactly q^* suppliers in a tour starting from home s_0 and ending at s_i , while minimizing the traveling and purchasing cost. The value $OPT[K, s_i, q^*]$ of an optimal solution to Sub-TPPS1(K, s_i, q^*), which is defined to be ∞ if no solution exists, can be computed by the following recurrence relation

$$OPT[K, s_i, q^*] = \min\{\min_{j \neq i}\{OPT[K, s_j, q^* - 1] + d_{ji}\}, \min_{g_i \in K}\{OPT[K \setminus \{g_i\}, s_i, q^*] + p_{ij}\}\}. \tag{5}$$

We can compute $OPT[K, s_i, q^*]$ in an order of increasing q^* and the size of K . The detailed steps of this algorithm are omitted since they are similar to these of the algorithm for TPP. We need to compute $O(2^k mq)$ subproblems and each subproblem takes $O(m+k)$ basic computations.

Theorem 2. *TPP-S1 can be solved in $O(2^k qm(m+k))$ time and it is FPT by taking the number k of products as the parameter.*

2.3 TPP-S2 with Parameter k

Now we consider TPP-S2. Compared with TPP-S1, TPP-S2 has one more restriction, which requires that at most u pieces of products can be bought from each supplier. The algorithm for TPP-S1 parameterized by the number k of products, can not be directly modified to an algorithm for TPP-S2, since it is hard to control the number of products purchased from each supplier. To get an FPT algorithm for TPP-S2 parameterized by k , we need to use the color coding technique [2] together with dynamic programming.

For a graph G , we say that G is q -colored if each vertex of G is colored by one of q different colors. For a q -colored graph G , if there is a path (resp., circle) such that all the vertices of the path are colored with pairwise distinct colors, we call it a *colorful path* (resp., *colorful circle*). We first solve a special case of TPP-S2, called Colored-TPP-S2, in which the input graph is q -colored, and we are asked to solve the TPP-S2 under the constraint that the traveling path (circle) is colorful.

We use χ to denote the set of q colors used in the graph G and $\chi(s_i)$ denote the color of supplier s_i . For each subset $K \subseteq \mathcal{K}$ of products, each supplier

$s_i \in \mathcal{M}$, each subset $X \subseteq \chi$ and each nonnegative integer $u^* \leq u$, we define the subproblem Sub-TPPS2(K, s_i, X, u^*): buy all the products in K by traveling from a colorful path starting from home s_0 and ending at s_i , the set of colors used in the colorful path is X , at most u products are bought from each supplier, and exactly u^* products are bought from supplier s_i , while minimizing the traveling and purchasing cost. The value of an optimal solution to Sub-TPPS2(K, s_i, X, u^*), which is defined to be ∞ if no solution exists, is denoted by $OPT[K, s_i, X, u^*]$.

It is not hard to see the result of Colored-TPP-S2 is equal to

$$\min_{s_i \in \mathcal{M}, X \subseteq \chi, u^* \leq u} \{OPT[\mathcal{K}, s_i, X, u^*] + d_{i0}\}. \tag{6}$$

Our idea is to compute all $OPT[K, s_i, X, u^*]$ in a dynamic programming way, in order of nondecreasing values of $|K|, |X|$ and u^* by using the following two state transition process.

When $u^* > 0$, it holds the *purchasing recurrence relation*

$$OPT[K, s_i, X, u^*] = \min_{g_j \in K} \{OPT[K \setminus \{g_j\}, s_i, X, u^* - 1] + p_{ij}\}. \tag{7}$$

When $u^* = 0$, it holds the *traveling recurrence relation*

$$OPT[K, s_i, X, 0] = \begin{cases} \infty, & \text{if } \chi(s_i) \notin X, \\ \min_{s_j \in \mathcal{M} \setminus \{s_i\} \& u^* \leq u} \{OPT[K, s_j, X \setminus \{\chi(s_i)\}, u^*] + d_{ji}\}, & \text{otherwise.} \end{cases} \tag{8}$$

Note that the number of sets K is 2^k , the number of sets X is 2^q , the number of possible values for u^* is $u + 1$, and s_i can be any candidate in \mathcal{M} . Thus, the number of subproblems Sub-TPPS2(K, s_i, X, u^*) is $O(um2^{k+q})$. When $u^* \neq 0$, we may use $|K| \leq k$ basic computations to compute $OPT[K, s_i, X, u^*]$ by (7). When $u^* = 0$, we may use $|\mathcal{M}|(u + 1) = m(u + 1)$ basic computations to compute $OPT[K, s_i, X, u^*]$ by (8). Note that $um \geq k$ otherwise the problem has no solution. So the total running time of the dynamic programming algorithm for Colored-TPP-S2 is $O(u^2m^22^{k+q})$.

Next we solve TPP-S2 by reducing it to Colored-TPP-S2. Consider an instance of TPP-S2. We use M^* to denote the set of suppliers visited in an optimal solution to it. We randomly color vertices of the instance graph by using q different colors with the same probability for each color. The probability such that all vertices in M^* get different colors is

$$\frac{|M^*|! \binom{q}{|M^*|}}{q^{|M^*|}}. \tag{9}$$

Since $|M^*| \leq q$, we know that

$$\frac{|M^*|! \binom{q}{|M^*|}}{q^{|M^*|}} \geq \frac{q!}{q^q}. \tag{10}$$

By using the well-known inequality $q! > (q/e)^q$, where e is the base of natural logs, we know that the probability of all vertices in M^* being colored with different colors is at least e^{-q} .

Thus, we can get a randomized algorithm of TPP-S2 by applying the random coloring operation and then solving each Colored-TPP-S2. We repeat the random coloring operation for e^q times and then get a colored instance such that vertices in M^* are colored with different colors with a constant probability.

There is also a technique to derandomize the above coloring operation with an additional running time factor of $q^{O(\log q)} \log m$ [15]. We have that

Theorem 3. *TPP-S2 can be solved in $O(u^2 m^2 2^{k+q} e^q q^{O(\log q)} \log m)$ time and it is FPT by taking the number k of products as the parameter.*

Note that we always assume $q \leq k$ due to the triangle inequality, and thus this algorithm is an FPT algorithm for the problem with parameter k .

3 Algorithms for Small Number of Suppliers

In this section, we will design an $O(2^m(m^2 + mk))$ -time algorithm for TPP, TPP-S1 and TPP-B, and then modify it to an $O(2^m(m^2 + mk\sqrt{k}))$ -time algorithm for TPP-S2. When we take m as the parameter, the four problems are FPT.

3.1 TPP with Parameter m

We still consider TPP first. In TPP, we have to decide two things, a tour of some suppliers and a purchasing plan of products. A tour of suppliers is a permutation of suppliers in the visiting order, and a purchasing plan of products is a decision that decides to buy each product from which supplier. This problem is hard because we need to optimize the traveling cost and purchasing cost at the same time. However, if we have decided the suppliers where we should visit (the set of which is M), then the problem can be reduced to the normal TSP problem.

In our algorithm, for each subset $M \subseteq \mathcal{M}$ of suppliers, we solve the subproblem: find a tour (circle) starting and ending at home s_0 , visiting all suppliers in M to buy all products in \mathcal{K} , while minimizing the total cost of purchasing and traveling. We use $c(M)$ to denote the cost of an optimal solution to the above subproblem, $d(M)$ to denote the cost of a minimum tour starting and ending at home s_0 and visiting all suppliers in M , and $p(M)$ to denote the optimal cost to buy all products in \mathcal{K} from suppliers only in M , where $p(M)$ may be ∞ if some product is not sold in any suppliers in M . Then for each $M \subseteq \mathcal{M}$,

$$c(M) = d(M) + p(M). \tag{11}$$

We compute $c(M)$ for all $M \subseteq \mathcal{M}$ using (11).

The next target is to compute $d(M)$ and $p(M)$. For each $M \subseteq \mathcal{M}$ and $s_i \in M$, we use $OPT[M, s_i]$ to denote the minimum distance of a tour which starts from home s_0 , visits all suppliers in M and ends at s_i . Then $OPT[M, s_i]$

can be computed in a dynamic programming method by the following recurrence relation (see [3, 10])

$$OPT[M, s_i] = \min_{s_j \in M \setminus \{s_i\}} \{OPT[M \setminus \{s_i\}, s_j] + d_{j_i}\}. \tag{12}$$

We get that

$$d(M) = \min_{s_j \in M} \{OPT[M, s_j] + d_{j_0}\}. \tag{13}$$

For a subset $M \subseteq \mathcal{M}$, we use $p_M(g_i)$ to denote the minimum price of product g_i in all suppliers in M , where $p_M(g_i) = \infty$ if no supplier in M provides g_i . Then

$$p(M) = \sum_{g_i \in \mathcal{K}} p_M(g_i). \tag{14}$$

The whole algorithm is denoted by A- m for TPP. In this algorithm, to compute $OPT[M, s_i]$ we use $|M| \leq m$ basic computations in (12). It takes $O(2^m m^2)$ time to compute all values of $OPT[M, s_i]$ for $M \subseteq \mathcal{M}$ and $s_i \in M$ in a dynamic programming method. For each fixed M , it takes at most m and mk basic computations to compute $d(M)$ in (13) and $p(M)$ in (14), respectively. The values of $c(M) = d(M) + p(M)$ for all M can be computed in $O(2^m mk)$ time. In total, this algorithm uses $O(2^m m^2 + 2^m mk)$ time.

Theorem 4. *TPP can be solved in $O(2^m m(m+k))$ time and it is FPT by taking the number m of suppliers as the parameter.*

3.2 TPP-S1 and TPP-B with Parameter m

The above algorithm can be easily modified for TPP-S1. We only need to compute $c(M)$ for $|M| \leq q$. Then TPP-S1 can be solved in $O(2^m m(m+k))$ time. For TPP-B, the goal is to find an M such that $d(M)$ is minimized under the budget constraint $p(M) \leq B$. We can also use the above algorithm to compute $d(M)$ and $p(M)$ and solve TPP-B in the same time.

Theorem 5. *TPP-S1 and TPP-B can be solved in $O(2^m m(m+k))$ time and they are FPT by taking the number m of suppliers as the parameter.*

3.3 TPP-S2 with Parameter m

We can also modify the algorithm for TPP to an algorithm for TPP-S2. However, we need one more technique to find a minimum cost matching on a bipartite graph. In TPP-S2, we have a part of input q to indicate the maximum number of suppliers can be visited and $\{u_i\}_{i=1}^m$ to indicate that at most u_i products can be bought from supplier s_i . For each $M \subseteq \mathcal{M}$ and $|M| \leq q$, we still use $d(M)$ to denote the cost of a minimum tour starting and ending at home s_0 and visiting all suppliers in M , and $p(M)$ to denote the optimal cost to buy all products in \mathcal{K} from suppliers only in M under the constraints in TPP-S2, where $p(M) = \infty$ if

we can not buy all the products from M under the constraints. To solve TPP-S2, we only need to find an M with $|M| \leq q$ such that the cost $c(M) = d(M) + p(M)$ is minimized. The above method to compute $d(M)$ is still suitable for TPP-S2. The hard part is to compute $p(M)$.

We construct a bipartite graph $H = (V_K \cup V_M, E)$ and compute $p(M)$ by finding a minimum cost matching in H . For each product $g_i \in \mathcal{K}$ we generate a vertex a_i in V_K . For each supplier $s_i \in M$ we generate u_i different vertices in V_M , each of which is adjacent to each $a_j \in V_K$ with the edge cost being the price p_{ij} of product g_j at supplier s_i . We can see that the minimum cost of a matching of size $|V_K| = k$ in H is equal to $p(M)$. By using the algorithm developed in [1], the minimum cost matching can be found in $O(k\sqrt{km})$ time.

Theorem 6. *TPP-S2 can be solved in $O(2^m m(m + k\sqrt{k}))$ time and it is FPT by taking the number m of suppliers as the parameter.*

4 Parameterized by the Number q of Suppliers to be Visited

In some real-world problems, usually the number q of suppliers to be visited is small. It is natural to consider q as the parameter. Different from the above two sections, this section will show that it is unlikely to have FPT algorithms by proving the W[2]-hardness of TPP parameterized by q . We will reduce from the well-known W[2]-hard problem: the set cover problem parameterized by the solution size.

An instance of the set cover problem is given by (U, \mathcal{C}) , where U is the universe of elements and \mathcal{C} is a collection of subsets of U . The target of the problem is to find a subset $\mathcal{A} \subseteq \mathcal{C}$ of minimum size such that $\cup_{A \in \mathcal{A}} A = U$.

For an instance $I = (U, \mathcal{C})$ of the set cover problem, we construct an instance I' of TPP. In I' , each product g is corresponding to an element g in U , and each supplier s is corresponding to a set s in \mathcal{C} . The price of a product g in a supplier s is 0 if the corresponding element g is contained in the corresponding set s and is ∞ otherwise. The distance between any two sites (home and suppliers) is 1. We can see that I has a set cover of size at most q if and only if I' has a solution with cost $q + 1$ (the purchasing cost is 0 and the traveling cost is $q + 1$).

Note that TPP-S1, TPP-S2 and TPP-B are general cases of TPP with more constraints. The above reduction also implies the hardness of TPP-S1, TPP-S2 and TPP-B.

Theorem 7. *TPP, TPP-S1, TPP-S2 and TPP-B are W[2]-hard by taking the number q of suppliers to be visited as the parameter.*

5 A Hardness Result for TPP-B

We have shown that TPP-B parameterized by m is FPT. The algorithm A- k for TPP can not be modified to TPP-B. In this section, we will show that TPP-B

parameterized by k is indeed $W[1]$ -hard. Our proof is based on a reduction from the multi-subset sum problem.

In the subset sum problem, we are given a set of integers $U = \{x_1, x_2, \dots, x_{|U|}\}$ and two integers w and k , and the task is to find a subset $S \subseteq U$ such that $|S| = k$ and the sum of the elements in S is equal to w . For the multi-subset sum problem, the input is the same and the task is to find a multi-subset S of U with k elements such that the sum of the elements in S is equal to w (i.e., an integer in U can appear more than one time in S). It is known that the subset sum problem with parameter k is $W[1]$ -hard [8]. The proof in [8], without any modification, can also prove the $W[1]$ -hardness of the multi-subset sum problem with parameter k .

For an instance $I = (U = \{x_1, x_2, \dots, x_{|U|}\}, w, k)$ of the multi-subset sum problem, we construct an instance I' of TPP-B. In I' , we have k different products to be bought and $k|U|$ suppliers. We partition the suppliers into k groups G_1, G_2, \dots, G_k , each group G_i has exactly $|U|$ suppliers.

We use $s_{i,j}$ ($j = 1, 2, \dots, |U|$) to denote the j th supplier in group G_i . Any supplier in the same group G_i only sales one product g_i . However, the price of g_i in supplier $s_{i,j}$ is x_j . Next we define the distance between each pair of sites. Let $X = \sum_{x \in U} x$. The distance from home s_0 to each supplier $s_{1,j}$ in group 1 is $X - x_j/2$. For any $i = 1, 2, \dots, k - 1$, the distance from each supplier s_{i,j_1} in group i to each supplier s_{i+1,j_2} in group $i + 1$ is $X - (x_{j_1} + x_{j_2})/2$. The distance from each supplier $s_{k,j}$ in group k to home s_0 is $X - x_j/2$. Any other distance between two sites not defined above is a very large number such that any optimal solution will not choose the path. The budget b is w . See Fig. 1 for an illustration for the construction.

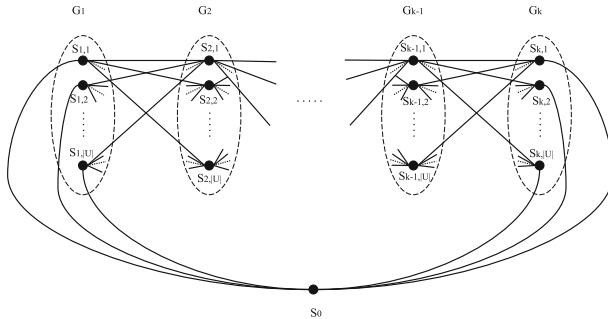


Fig. 1. Reduction from the multi-subset sum problem

In I' , an optimal solution will visit exact one supplier in each group. Assume that in an optimal solution, we buy product g_i with price y_i in group i ($i = 1, 2, \dots, k$). Then the traveling cost is $c_1 = (X - y_1/2) + (X - (y_1 + y_2)/2) + (X - (y_2 + y_3)/2) + \dots + (X - y_k/2) = (k + 1)X - \sum_{i=1}^k y_i$ and the purchasing cost is $c_2 = \sum_{i=1}^k y_i$. We can see that $c_1 + c_2 = (k + 1)X$ is a constant. When the

purchasing cost reaches the budget w , the traveling cost reaches the optimal value of $(k + 1)X - w$. So the instance I of the multi-subset sum problem has a solution of size k if and only if the instance I' of TPP-B has a solution with traveling cost $(k + 1)X - w$.

Theorem 8. *TPP-B is $W[1]$ -hard by taking the number k of products as the parameter.*

6 Conclusion

To deal with NP-hard problems, approximation algorithms relax the accuracy, heuristic methods loss the certainty (or accuracy), and parameterized algorithms find certain tractable ranges. Parameterized algorithms restrict the exponential part of the running time to parameters only. When the parameter is small, parameterized algorithms can optimally solve problems in a short time. Many real-world instances of TPP and its variants have the property of small parameters. In this paper, we establish the parameterized complexity of TPP with additional constraints and different parameters. In practice, the experimental results show the advantages of parameterized algorithms on instances with small parameters.

Acknowledgements. This work was supported by the National Natural Science Foundation of China, under grants 61772115 and 61370071.

References

1. Ahuja, R.K., Orlin, J.B., Stein, C.: Improved algorithms for bipartite network flow. *SIAM J. Comput.* **23**(5), 906–933 (1994)
2. Alon, N., Yuster, R., Zwick, U.: Color-coding. *J. ACM* **42**(4), 844–856 (1995)
3. Bellman, R.: Dynamic programming treatment of the travelling salesman problem. *J. ACM* **9**(1), 61–63 (1962)
4. Bianchessi, N., Mansini, R., Speranza, M.G.: The distance constrained multiple vehicle traveling purchaser problem. *Eur. J. Oper. Res.* **235**(1), 73–87 (2014)
5. Choi, M.J., Lee, S.H.: The multiple traveling purchaser problem for maximizing system's reliability with budget constraints. *Expert Syst. Appl.* **38**(8), 9848–9853 (2011)
6. Desaulniers, G.: Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Oper. Res.* **58**(1), 179–192 (2010)
7. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness. Cornell University, Mathematical Sciences Institute (1992)
8. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness II: on completeness for $W[1]$. *Theor. Comput. Sci.* **141**(1), 109–131 (1995)
9. Gouveia, L., Paiais, A., Voß, S.: Models for a traveling purchaser problem with additional side-constraints. *Comput. Oper. Res.* **38**(2), 550–558 (2011)
10. Held, M., Karp, R.M.: A dynamic programming approach to sequencing problems. *J. Soc. Ind. Appl. Math.* **10**, 196–210 (1962)

11. Ho, S.C., Haugland, D.: A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Comput. Oper. Res.* **31**(12), 1947–1964 (2004)
12. Kang, S., Ouyang, Y.: The traveling purchaser problem with stochastic prices: exact and approximate algorithms. *Eur. J. Oper. Res.* **209**(3), 265–272 (2011)
13. Manerba, D., Gendreau, M., Mansini, R.: The multi-vehicle traveling purchaser problem with pairwise incompatibility constraints and unitary demands: a branch-and-price approach. *Eur. J. Oper. Res.* **148**, 59–71 (2016)
14. Mansini, R., Tocchella, B.: The traveling purchaser problem with budget constraint. *Comput. Oper. Res.* **36**(7), 2263–2274 (2009)
15. Narayanaswamy, N., Raman, V., Ramanujan, M., Saurabh, S.: LP can be a cure for parameterized problems. In: 29th Symposium on Theoretical Aspects of Computer Science, STACS 2012, vol. 14, pp. 338–349. LIPIcs (2012)
16. Ramesh, T.: Traveling purchaser problem. *Opsearch* **18**(1–3), 78–91 (1981)
17. Ravi, R., Salman, F.S.: Approximation algorithms for the traveling purchaser problem and its variants in network design. In: *Algorithms-ESA 1999*, pp. 29–40 (1999)
18. Singh, K.N., van Oudheusden, D.L.: A branch and bound algorithm for the traveling purchaser problem. *Eur. J. Oper. Res.* **97**(3), 571–579 (1997)
19. Zhang, Z., He, H., Luo, Z., Qin, H., Guo, S.: An efficient forest-based tabu search algorithm for the split-delivery vehicle routing problem. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 3432–3438 (2015)