

Worst Case Analysis for a Pickup and Delivery Problem with Single Transfer

YOSHITAKA NAKAO

Department of Applied Mathematics and
Physics, Graduate School of Informatics,
Kyoto University

HIROSHI NAGAMOCHI

Department of Applied Mathematics and
Physics, Graduate School of Informatics,
Kyoto University

1 Introduction

The vehicle routing problem (abbreviated as VRP) is one of the well studied combinatorial problems with important applications to logistics and transportation industries. VRP can be stated as follows. Given a set of vehicles starting and ending at a depot and a set of customers with their demand, the problem asks to find a route for each vehicle such that the total travel cost is minimized under the restrictions that all customer demands are met and each customer demand is not split. Numerous algorithms for the VRP have been proposed in these decades [8, 16, 17]. There exist many variants of the VRP, such as VRP with time windows (VRPTW) [6, 14, 15], the multiple depot VRP (MDVRP) and the split delivery VRP (SDVRP) [1, 2, 4, 5]. Most of them are known to be NP-hard.

The pickup and delivery problem (abbreviated as PDP) is an extension of the VRP that handles pickup and delivery of loads between customers. Each transportation request must be picked up at a predetermined customer and delivered to an another predetermined customer. The PDP introduces two side constraints. One is a *coupling constraint* that the actions of pickup and delivery must be done by the same vehicle. The other is a *precedence constraint* that the action of pickup must be done before that of delivery. The total quantity of loads cannot exceed the vehicle capacity any time. Note that the VRP is a special case where all pickup customers or all delivery customers are located on the same place (i.e., depot). One of the most popular variant of the PDP is the pickup and delivery problem with time windows (PDPTW). Since it is very difficult to solve exactly the PDP and PDPTW problems, many heuristics and metaheuristic algorithms have been developed for them [9, 11, 12].

The pickup and delivery problem with transfer (PDPT) is a variant of the PDP such that each request can be served by more than one vehicle by dropping a load at a transshipment point and picking it up by another vehicle [3, 10, 13].

The split delivery vehicle routing problem (SDVRP) is a problem such that each customer can be visited more than once, that is, a quantity for a customer is split into several parts, each of which is allowed to be delivered by a different vehicle. Archetti et al. studied lower bounds of travel cost saved by introducing split deliveries to the original VRP [2]. Let $z(VRP)$ be an optimal travel cost to the VRP and $z(SDVRP)$ be an optimal travel cost to the SDVRP. They showed that $z(VRP) \leq 2z(SDVRP)$ holds by converting an optimal SDVRP solution into a VRP solution whose travel cost is at most $2z(SDVRP)$. Furthermore, they introduced instances which show that the bound is tight.

In this paper, we analyze lower bounds of travel cost saved by introducing a transshipment point to the PDP. We suppose that the number of transshipment points is one and each vehicle can visit the transshipment point at most once. Let $z(PDP)$ be an optimal travel cost to the PDP, and $z(PDPT)$ be an optimal travel cost to the PDPT. We denote by p the number of

requests, and denote by m the number of routes in an optimal PDPT solution. In this paper, we show that $z(PDP) < (6\lceil\sqrt{m}\rceil + 1) \cdot z(PDPT)$ and $z(PDP) < (6\lceil\sqrt{p}\rceil + 1) \cdot z(PDPT)$ hold. This indicates that travel cost saved by transferring requests at a transshipment point can be in proportion to square root of the number of requests, while the bound for the SDVRP is constant 2 as shown in [2].

The rest of this paper is organized as follows. In Section 2 we give some notations and define problems. In Section 3, we analyze lower bounds of travel cost saved by introducing a transshipment point to the PDP. Finally in Section 4, we make concluding remarks.

2 Preliminaries

This section formulates problems PDP and PDPT. We first introduce the PDP. We are given a vertex set $V = C \cup P$, where C denotes a set of *customers*, and P denotes a set of *depots*. For simplicity, we assume that $C \cap P = \emptyset$ (by duplicating the same vertex and giving them different indices if necessary). Let $R = \{r_1, r_2, \dots, r_p\}$ be a set of *requests*. Each request $r = \{r^+, r^-\} \in R$ consists of a pickup location $r^+ \in C$ and a delivery location $r^- \in C$. Let $q(r)$ stand for *quantity* of loads for request r . Each vehicle can pick up request r at r^+ and deliver r to r^- . The entire amount $q(r)$ of loads cannot be split, i.e., each request is serviced exactly by one vehicle. We denote by $d(j, j')$ *travel cost* from j to j' for $j, j' \in V$. Travel cost $d(j, j')$ is a nonnegative real number, and in general asymmetric, i.e., $d(j, j') \neq d(j', j)$ may hold. Every vehicle has capacity c , where c is a nonnegative real number, and each vehicle must start from its predetermined depot, and return to the depot after serving requests assigned to the vehicle. We assume that any number of vehicles is available at each depot.

Given $v_0, v_1, v_2, \dots, v_u \in V$, *path* σ is a sequence of vertices in V , and its travel cost $d(\sigma)$ is defined to be

$$d(\sigma) = \sum_{0 \leq i \leq u-1} d(v_i, v_{i+1}).$$

If $v_0 = v_u$ and $v_i \neq v_j$ for $i \neq j$, then we call σ a *cycle*, and in addition if $v_0 \in P$, i.e., a vehicle starts and ends at a depot, then we call σ a *route*. For simplicity, we may treat σ as an ordered subset of V .

The PDP asks to determine a route for a vehicle such that the total travel cost of vehicles are minimized under restrictions that all requests are serviced, the load of a vehicle does not exceed vehicle capacity c any time. Furthermore, vertices r^+ and r^- must be visited by the same vehicle (*coupling constraint*) and r^+ must be visited before r^- (*precedence constraint*).

We next introduce a pick and delivery problem with transfer (PDPT). In the PDPT, we are given a vertex set $V = C \cup P \cup T$, where T denotes a set of *transshipment points*. For simplicity, we assume that $C \cap T = \emptyset$ and $P \cap T = \emptyset$. Vehicles are allowed to temporarily drop a load and pick it up later (A vehicle that drops a load can be different from a vehicle that picks it up). The precedence constraint holds for the PDPT, i.e., if request r is serviced by more than one vehicle by visiting a transshipment point, then r_i^+ must be visited before the transshipment point, which must be visited before r_i^- . We assume without loss of generality that no transshipment of loads at any vertices of $V \setminus T$ is allowed. This paper assumes that $|T| = 1$, and we denote the transshipment point by t . This paper also assumes that each vehicle can visit t at most once.

3 Worst-Case Analysis for the PDPT

Given an optimal PDPT solution S , let m be the number of routes in S . We show the next theorem.

Theorem 1 *Suppose that each vehicle can visit the transshipment point at most once. Let m denote the number of routes in an optimal PDPT solution. Then*

$$z(PDP) < (6\lceil\sqrt{m}\rceil + 1) \cdot z(PDPT).$$

To show this, we convert S to a PDP solution in which no vehicles visit t , and we will show that the travel cost for the constructed solution is less than $(6\lceil\sqrt{m}\rceil + 1) \cdot z(PDPT)$. For simplicity, we assume that every route visits t since if there exists a route that does not visit t , the route need not to be converted, which does not lead to increase travel costs.

3.1 Division of PDPT cycles

We first make some preliminaries for the PDPT solution. Let π be a set of cycles in a PDPT solution. Given $\sigma_i, \sigma_j \in \pi$, let $R(\sigma_i, \sigma_j)$ stand for the set of requests that are picked up at customers on σ_i and delivered to customers on σ_j and let $q(\sigma_i, \sigma_j) = \sum_{r \in R(\sigma_i, \sigma_j)} q(r)$. Let σ^+ be the subpath of σ from the next customer of depot $p \in \sigma$ to the customer before t on σ , and σ^- be the subpath from the next customer of t to the customer before p on σ for $i = 1, 2, \dots, m$. We denote by $\sigma = \sigma_1 \cup \sigma_2$ that route σ follows σ_2 after σ_1 . Then, σ is expressed by $\sigma = p \cup \sigma^+ \cup t \cup \sigma^- \cup p$. For $\sigma \in \pi$, since all loads in $R(\sigma, \pi) - \{(R(\sigma^+, \sigma^+) \cup R(\sigma^-, \sigma^-))\}$ are on a vehicle when σ reaches t in an PDPT solution,

$$q(\sigma, \pi) - \{q(\sigma^+, \sigma^+) + q(\sigma^-, \sigma^-)\} \leq c \quad (1)$$

holds.

We divide set π of routes into $\lceil\sqrt{m}\rceil$ subsets π_i , $i = 1, \dots, \lceil\sqrt{m}\rceil$, so that each subset π_i , $i = 1, \dots, \lceil\sqrt{m}\rceil$, include at most $\lceil\sqrt{m}\rceil$ routes. Let $\pi_i = \{\sigma_{i,1}, \sigma_{i,2}, \dots, \sigma_{i,\lceil\sqrt{m}\rceil}\}$. If $\lceil\sqrt{m}\rceil \cdot \lceil\sqrt{m}\rceil > m$, then we assume that $\sigma_{i,j} = \emptyset$ for some $i, j \in [1, \lceil\sqrt{m}\rceil]$. Let $R(\pi_i, \pi_j)$ stand for the set of requests that are picked up at customers on π_i and delivered to customers on π_j . Let $q(\pi_i, \pi_j) = \sum_{r \in R(\pi_i, \pi_j)} q(r)$ and $d(\pi_i) = \sum_{\sigma \in \pi_i} d(\sigma)$. It is trivial to see that

$$z(PDPT) = \sum_{i=1}^{\lceil\sqrt{m}\rceil} d(\pi_i) \quad (2)$$

holds. We introduce the following Lemma.

Lemma 2 *Let m denote the number of routes in a PDPT solution. Given π_i , $i = 1, \dots, \lceil\sqrt{m}\rceil$, it holds*

$$\begin{aligned} q(\pi_i, \pi) - \sum_{\sigma \in \pi_i} \{q(\sigma^+, \sigma^+) + q(\sigma^-, \sigma^-)\} \\ \leq \lceil\sqrt{m}\rceil \cdot c \text{ for } i = 1, \dots, \lceil\sqrt{m}\rceil. \end{aligned} \quad (3)$$

proof: Inequality (1) and the assumption that $|\pi_i| \leq \lceil\sqrt{m}\rceil$ ensure the lemma. ■

3.2 First-Fit Procedure

In this subsection, we introduce a well-known First-Fit procedure that is used in conversion algorithms in Subsection 3.3. We are given n bins with capacity c , and a set I of items such that item $i \in I$ has quantity $q(i)$. Procedure FIRSTFIT inserts each item into one of the bins so that each item is not split and total quantity of items for each bin is not beyond c . We denote by I_k a set of items inserted into k -th bin for $k = 1, \dots, n$.

Procedure FIRSTFIT(I, c, n)

Input: A set I of items, capacity c and n bins, where item $i \in I$ has quantity $q(i)$.

Output: Sets I_k , $k = 1, \dots, n'$ ($n' \leq n$) of items such that $I_1 \cup I_2 \cup \dots \cup I_{n'} = I$ and $I_k \neq \emptyset$ for $k = 1, \dots, n'$.

```

1:  $I_k := \emptyset$  for  $k = 1, \dots, n$ .
2: for  $i = 1, \dots, |I|$  do
3:   Search  $I_1, \dots, I_n$  and select  $I_k \in \{I_1, \dots, I_n\}$ 
      such that  $q(I_k \cup i) \leq c$  and  $q(I_j \cup i) > c$  for  $j = 1, \dots, k - 1$ .
4:    $I_k := I_k \cup i$ .
5: end /* for */

```

The next theorem for the First-Fit procedure is known.

Theorem 3 [7] *Given n bins with capacity c and a set I of items where item $i \in I$ has quantity $q(i)$. Let n' be the number of bins that at least one of the items are inserted in FIRSTFIT. Then it holds*

$$n' \leq \frac{2}{c} \sum_{i=1}^{|I|} q(i).$$

proof: We show the theorem by contradiction. Suppose that $n' \cdot c/2 > \sum_{i=1}^{|I|} q(i)$. Then there exists a bin X , where total quantity of items is less than $c/2$. If there exists a bin Y other than X , where total quantity of items is less than $c/2$, then the items in X and Y must be inserted in the same bin, which is a contradiction. If X is the only bin whose total quantity is less than $c/2$, then there exists a bin Z such that the sum of quantity of X and Z less than or equal to c , which is also a contradiction. ■

3.3 Analysis of lower bounds

This subsection gives a proof of Theorem 1. The algorithm to convert a PDPT solution to a PDP solution is described as follows. In a PDP solution, route $\sigma'_{i,j}$ first follows π_i , and then follows π_j to pick up and delivery requests in $R(\pi_i, \pi_j)$ if $i \neq j$ and $R(\pi_i, \pi_j) - \bigcup_{\sigma \in \pi} \{R(\sigma^+, \sigma^+) \cup R(\sigma^-, \sigma^-)\}$ if $i = j$ for $i, j = 1, \dots, \lceil \sqrt{m} \rceil$. Another route σ' follows all routes to service requests in $\bigcup_{\sigma \in \pi} \{R(\sigma^+, \sigma^+) \cup R(\sigma^-, \sigma^-)\}$.

Algorithm CONVERT

Input: A set R of requests and a set π of PDPT routes, each of which visits t at most once.

Output: A set $\{\sigma'_{i,j} \mid i, j = 1, 2, \dots, \lceil \sqrt{m} \rceil\} \cup \{\sigma'\}$ of PDP routes.

```

1: for  $i = 1, 2, \dots, \lceil \sqrt{m} \rceil$  do
2:   for  $j = 1, 2, \dots, \lceil \sqrt{m} \rceil$  do
3:     if  $i \neq j$  then
4:        $\{R_1, \dots, R_{n'}\} := FirstFit(R(\pi_i, \pi_j), c, m)$ .
5:     else
6:        $\{R_1, \dots, R_{n'}\} := FirstFit(R(\pi_i, \pi_i), c, m) - \bigcup_{\sigma \in \pi_i} \{R(\sigma^+, \sigma^+) \cup R(\sigma^-, \sigma^-)\}, c, n)$ .
7:     end /* if */
8:     for  $k = 1, 2, \dots, n'$  do
9:        $\sigma'_{i,j}$  follows  $\pi_i$  to pick up requests in  $I_k$ .
10:       $\sigma'_{i,j}$  follows  $\pi_j$  to delivery requests in  $I_k$ .
11:    end /* for */
12:  end /* for */

```

13: end /* for */

14: σ' follows all routes to service $\bigcup_{\sigma \in \pi} \{R(\sigma^+, \sigma^+) \cup R(\sigma^-, \sigma^-)\}$.

In Line 9, route $\sigma'_{i,j}$ follows π_i by $\sigma'_{i,j} = \sigma_{i,1}^+ \cup t \cup \sigma_{i,2}^- \cup t \cup \sigma_{i,2}^+ \cup \dots \cup \sigma_{i,\lceil\sqrt{m}\rceil}^- \cup t \cup \sigma_{i,\lceil\sqrt{m}\rceil}^+ \cup t \cup \sigma_{i,1}^-$. In Line 10, $\sigma'_{i,j}$ follows π_j in the same way. In Line 14, route σ' follow all routes by $\sigma' = \sigma_{1,1}^+ \cup t \cup \sigma_{1,2}^- \cup t \cup \sigma_{1,2}^+ \cup \dots \cup \sigma_{\lceil\sqrt{m}\rceil,\lceil\sqrt{m}\rceil}^- \cup t \cup \sigma_{\lceil\sqrt{m}\rceil,\lceil\sqrt{m}\rceil}^+ \cup t \cup \sigma_{1,1}^-$. We show the following lemma.

Lemma 4 *PDP routes obtained by CONVERT services all requests in R .*

proof: By iterating Line 3-11 for $i, j = 1, \dots, \lceil\sqrt{m}\rceil$, all requests in $R - \bigcup_{\sigma \in \pi} \{R(\sigma^+, \sigma^+) \cup R(\sigma^-, \sigma^-)\}$ are serviced, and all requests in $\bigcup_{\sigma \in \pi} \{R(\sigma^+, \sigma^+) \cup R(\sigma^-, \sigma^-)\}$ are serviced in Line 16. Thus, we have the lemma. ■

We now analyze the travel cost of routes constructed by CONVERT. The following lemma is used to prove Theorem 1.

Lemma 5 *For given $i \in [1, \lceil\sqrt{m}\rceil]$, let d'_i be travel cost to follow π_i in Line 9 of CONVERT for all $j = 1, \dots, \lceil\sqrt{m}\rceil$. Then, it holds*

$$d'_i < 3\lceil\sqrt{m}\rceil \cdot d(\pi_i).$$

proof: For given integers $i, j \in [1, \lceil\sqrt{m}\rceil]$, Theorem 3 gives $n' \leq \lceil 2q(\pi_i, \pi_j)/c \rceil$ for $i \neq j$, and $n' \leq \lceil 2(q(\pi_i, \pi_j) - \sum_{\sigma \in \pi_i} \{q(\sigma^+, \sigma^+) + q(\sigma^-, \sigma^-)\})/c \rceil$ for $i = j$. Let $q'(\pi_i, \pi_j) = q(\pi_i, \pi_j) - \sum_{\sigma \in \pi_i} \{q(\sigma^+, \sigma^+) + q(\sigma^-, \sigma^-)\}$. Then, we obtain

$$\begin{aligned} d'_i &= \sum_{j=1}^{\lceil\sqrt{m}\rceil} \lceil 2q'(\pi_i, \pi_j)/c \rceil \cdot d(\pi_i) \\ &< \sum_{j=1}^{\lceil\sqrt{m}\rceil} (2q'(\pi_i, \pi_j)/c + 1) \cdot d(\pi_i) \\ &= \left(\sum_{j=1}^{\lceil\sqrt{m}\rceil} 2q'(\pi_i, \pi_j)/c + \lceil\sqrt{m}\rceil \right) \cdot d(\pi_i). \end{aligned}$$

By applying (3),

$$\begin{aligned} d' &< (2\lceil\sqrt{m}\rceil + \lceil\sqrt{m}\rceil) \cdot d(\pi_i) \\ &= 3\lceil\sqrt{m}\rceil \cdot d(\pi_i). \end{aligned}$$

We now show the proof of Theorem 1.

proof of Theorem 1: For given $j \in [1, \lceil\sqrt{m}\rceil]$, let d''_j be travel cost to follow π_j in Line 10 of CONVERT for all $i = 1, \dots, \lceil\sqrt{m}\rceil$. Lemma 5 is easily extended to show that

$$d''_j < 3\lceil\sqrt{m}\rceil \cdot d(\pi_j).$$

The travel cost for Line 14 is $z(PDPT)$. Thus, we obtain

$$\begin{aligned} z(PDP) &\leq \sum_{i=1}^{\lceil\sqrt{m}\rceil} d'_i + \sum_{j=1}^{\lceil\sqrt{m}\rceil} d''_j + z(PDPT) \\ &= 6\lceil\sqrt{m}\rceil \cdot \sum_{i=1}^{\lceil\sqrt{m}\rceil} d(\pi_i) + z(PDPT). \end{aligned}$$

From (2), it holds

$$z(PDP) < (6\lceil\sqrt{m}\rceil + 1) \cdot z(PDPT).$$

If we use the number p of requests, the next theorem holds by using $m \leq p$.

Theorem 6 *Suppose that each vehicle can visit the transshipment point at most once. Let p denote the number of requests. Then*

$$z(PDP) < (6\lceil\sqrt{p}\rceil + 1) \cdot z(PDPT).$$

4 Conclusion

In this paper, we analyzed lower bounds of travel cost saved by introducing a transshipment point to the PDP. We showed that the bounds are in proportion to square root of the number of routes in an optimal PDPT solution and also square root of the number of requests. Since the effectiveness on reducing travel cost by transferring requests at a transshipment point is high comparing to admitting split delivery to the VRP, developing algorithms for constructing PDPT routes would be practically helpful for real world logistics.

References

- [1] C. Archetti, M. G. Speranza, and A. Hertz, A tabu search algorithm for the split delivery vehicle routing problem, *Transportation Science*, Vol.40, No.1, pp. 64-73, 2006.
- [2] C. Archetti, M. W. P. Savelsbergh, and M. G. Speranza, Worst-case analysis for split delivery vehicle routing problems, *Transportation Science*, Vol.40, No.2, pp.226-234, 2006.
- [3] C. E. Cortés, M. Matamala, and C. Contardo, The pickup-and-delivery problem with transfers: Formulation and solution approaches, *Submitted for publication*, 2006.
- [4] M. Dror and P. Trudeau, Savings by split delivery routing, *Transportation Science*, Vol.23, pp. 141-145, 1989.
- [5] M. Dror and P. Trudeau, Split delivery routing, *Naval Research Logistics*, Vol.37, pp. 383-402, 1990.
- [6] H. Hashimoto, T. Ibaraki, S. Imahori, and M. Yagiura, The vehicle routing problem with flexible time windows and traveling times, *Discrete Applied Mathematics*, Vol.154, pp. 2271-2290, 2006.

- [7] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, R. L. Graham, Worst-case performance bounds for simple one-dimensional packing algorithms, *SIAM Journal on Computing*, Vol.3, No.4, pp. 299-325, 1974.
- [8] G. Laporte, M. Gendreau, J. Y. Potvin, and F. Semet, Classical and modern heuristics for the vehicle routing problem, *International Transactions in Operational Research*, Vol.7, pp. 285-300, 2000.
- [9] H. Li and A. Lim, A metaheuristic for the pickup and delivery problem with time windows, *International Journal on Artificial Intelligence Tools*, Vol. 12, No.2, pp.173-186, 2003.
- [10] S. Mitrović-Minić and G. Laporte, The pickup and delivery problem with time windows and transshipment, *INFOR*, Vol.44, No.3, pp.217-227, 2006.
- [11] W. P. Nanry and J. W. Barnes, Solving the pickup and delivery problem with time windows using reactive tabu search, *Transportation Research Part B*, Vol.34, pp. 107-121, 2000.
- [12] G. Pankratz, A group genetic algorithm for the pickup and delivery problem with time windows, *OR Spectrum*, Vol.27, pp.21-41, 2005.
- [13] J. S. Shang and C. K. Cuff, Multicriteria pickup and delivery problem with transfer opportunity, *Computers and Industrial Engineering*, Vol.30, No.4, pp. 631-645, 1996.
- [14] M. M. Solomon, Algorithms for the vehicle routing and scheduling problem with time window constraints, *Operations Research*, Vol.35, No. 2, pp. 254-265, 1987.
- [15] E. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J. Y. Potvin, A tabu search heuristic for the vehicle routing problem with soft time windows, *Transportation Science*, Vol.31, pp.170-186, 1997.
- [16] P. Toth and D. Vigo (eds.), The vehicle routing problem, *Society for Industrial and Applied Mathematics*, 2002.
- [17] P. Toth and D. Vigo, The granular tabu search and its application to the vehicle-routing problem, *INFORMS Journal on Computing*, Vol.15, No.4, pp.333-346, 2003.