A Lagrangian Relaxation-based Heuristic for the Vehicle Routing with Full Container Load

Akio Imai
Faculty of Maritime Sciences
Kobe University
Fukae, Higashinada, Kobe 658-0022 Japan
&
World Maritime University
P.O. Box 500, S-201 24 Malmo, Sweden

Etsuko Nishimura
Faculty of Maritime Sciences
Kobe University
Fukae, Higashinada, Kobe 658-0022 Japan

and

John Current
Department of Management Sciences
The Ohio State University
Columbus, OH 43210-1399 USA

Abstract

We address a problem of vehicle routing that arises in picking up and delivering full container load from/to an intermodal terminal. The substantial cost and time savings are expected by efficient linkage between pickup and delivery tasks, if the time of tasks and the suitability of containers for cargo allow. As this problem is NP-hard, we develop a subgradient heuristic based on a Lagrangian relaxation which enables us to identify a near optimal solution. The heuristic consists of two sub-problems: the classical assignment problem and the generalized assignment problem. As generalized assignment problem is also NP-hard, we employ an efficient solution procedure for a bin packing based problem, which replaces the generalized assignment problem. The heuristic procedure is tested on a wide variety of problem examples. The test results demonstrate that the procedure developed here can efficiently solve large instances of the problem.

Keywords: Logistics, Container transportation, Vehicle routing, Heuristics, Lagrangian relaxation

1. Introduction

This paper aims to model and solve a complex routing problem associated with the pickup and delivery of containers. An intermodal transportation system comprises several means of transportation such as sea, rail and road. Intermodal shipments in containers are distributed, by trucks, to customers from interchange nodes (or intermodal terminals) such as ports and rail-yards. The container distribution is divided into two activities: pickup and delivery. These activities are essentially independent. A trailer-truck delivers a full (or loaded) container to a consignee from an intermodal terminal and hauls back empty to the terminal after unloading goods at the consignee site (delivery trip). Contrary to this, in a pickup activity a trailer-truck hauls an empty container to a shipper and takes a filled container back to the terminal after loading at the shipper (pickup trip).

Most shippers and consignees are located close together in a distribution area nearby a terminal, especially in Japan. Therefore, if criteria in compatibility, such as the time of shipment and the suitability of container to cargo in terms of type and size, are satisfied, a direct transfer of an empty container from a delivery point to a site for a subsequent shipment may be possible, reducing the trip length of empty trailer-truck (or deadhead). Such a direct transfer, which can be viewed as a merged route for delivery and pickup trips, also leads to the efficient use of empty containers.

A trailer-truck consists of a tractor and a trailer, and normally they can be uncoupled (in Japan most trailer-trucks designed for domestic container distribution cannot). Consequently, when a long loading or unloading process is expected, a tractor uncouples a trailer (or a chassis with a container on it) and leaves it at a customer site so that the truck can be assigned to the next shipment. However, this is not likely the case especially for intermodal container transportation within Japan, mainly because of the complexity of the tractor assignment to trailer and the physical feature of trailer-truck as mentioned above.

A firm usually has an option of either buying its own fleet of trucks or leasing it to carry out distribution. As the fleet size that satisfies the peak demand is not considered appropriate due to the capital and operating costs, the firm invests in a moderate size of fleet between the highest and lowest demands being estimated for the planning horizon, and in the operational phase the firm charters an additional fleet to cover the excess demand. Once the owned fleet is held, the capital costs including the driver's costs are regarded as the sunk costs, i.e., the costs are incurred for the truck's lifetime regardless of the amount of workload. The fuel costs are, however, typically incurred with revenue runs (and empty runs as well). From this point of view, the own fleet-related costs (mostly fuel costs) are less than the chartering costs, which are assumed to vary depending on the time length of charter to distribute cargoes. In this context, it is important to allocate as many distribution tasks to the own fleet as possible to reduce the entire distribution costs. Such an efficient task allocation plan, however, is not easy to be made because of the restrictive overwork of the own fleet. If the task allocation including own and outside fleets is not well planned, the firm needs to hire much more expensive outside trucks.

Motivated with the above-mentioned issues in the own fleet operation for container cargo delivery, we consider the problem called the vehicle routing problem with full container load (VRPFC) that is defined as follows: Assuming the homogeneous container type and size and own and chartered fleets with different hauling costs and working time lengths, find the optimal assignment of the own and chartered fleets to a set of delivery and pickup point pairs (hereafter referred to as *D-P pairs*), in order to minimize the total distribution cost. We treat the container distribution from a single terminal on a day. Note that as we focus on the domestic intermodal transportation, we do not assume a sophisticated and dynamic operation based on the separate use of tractor and trailer. As will be proven later, the VRPFC is an NP-hard problem. Thus, we propose a Lagrangian relaxation-based heuristic for that problem.

As mentioned above, this study assumes the homogeneous container type and size. This assumption is realistic in many cases for the following reasons: In the domestic container transportation system of Japan, there are normally three types of containers in size: 12-, 20- and 30-ft containers. If a shipment needs a 12-ft container, the shipment can be physically made up with a larger container. Container size compatibility can be seen in international container transportation systems, too. However, such a size substitution may apply a different freight rate and other transport-related agreements. The type substitution is more difficult than the size substitution. The problem we consider can be modified so that multiple kinds of containers are dealt and the same kind of containers are reused among specific customer sites. However, such a modification is a future research topic.

This paper is organized as follows. The next section describes the outline of the problem. Section 3 reviews the related literature. The problem formulation and its Lagrangian relaxation are described in Section 4. Section 5 outlines a subgradient method to yield a near optimal solution, highlighting an algorithm that obtains a feasible solution from a solution to the relaxation problem. Extensive numerical experiments are carried out in Section 6, followed by Section 7 that concludes the paper.

2. Problem description

The VRPFC is defined on a graph $G = (V, A)$ where V is a vertex set representing the terminal, pickup points (or shippers) and delivery points (or consignees) and A is the arc set. There are two types of trips, each relevant to delivery point or pickup point as illustrated in Fig. 1(a). A delivery trip and a pickup trip can be merged as shown in Fig. 1(b) if the compatibility of container and the time of the delivery and pickup are satisfied. The problem is to minimize the total distribution cost where container distribution trips are serviced by various trailer-trucks (hereafter referred to as trucks) with respective working time criterion being allowed (referred to as the time capacity in the remaining part of the paper). The trucks employed for the container distribution are categorized into own and chartered, while the distinction is not essential in the problem. As mentioned in Section 1, the chartered truck is more expensive than the owned one if the sunk cost of the own truck is ignored. The owned truck has a shorter time capacity than the chartered one, mainly because of the labor union of the firm. The merged trips are serviced by the own trucks as long as their time capacity allows. This task assignment policy may be interpreted as follows: The total cost of the own fleet is higher than the chartered ones if the sunk cost is included, thus the firm deploys the limited number of own trucks. The firm assigns as many tasks to the own fleet as possible because the expensive own fleet with the sunk cost should be effectively utilized with the time capacity.

Note that individual trips remain if the number of delivery points is not equal to the number of pickup points. Trips that are not served by the own trucks are covered by the chartered trucks. All the trips are serviced since it is assumed that the firm can charter as many outside trucks as possible.

Fig. 1

Fig. 2 illustrates a typical example of the VRPFC where there are 10 pickup points (points 1-10) and 10 delivery points (points 11-20). In the Lagrangian relaxation-based solution procedure that will be described in the subsequent sections, delivery points are paired in the first stage of each iteration and the fleets of trucks are assigned to the paired customers' sites (or the merged trips) in the second stage. In the solution procedure, as will be discussed in Section 4, the solution is improved through iterations. Fig. 2 reveals the feasible solution at the first iteration.

Depot-customer and customer-customer distances are shown in Table 1. Note that as a delivery point is followed by a pickup point in a merged trip, only distances from delivery points to pickup points are given in Table 1. Hauling times of merged trips of [the depot - a delivery point - a pickup point - the depot] are given in Table 2, where a hauling time covers a possible merged trip with the assumption that a truck runs at 15km/hour. Note that for simplicity, a hauling time does not include a loading/unloading time at a customer. The number of trucks, and associated costs and time capacities are shown in Table 3. Totally 20 points are paired to form 10 merged trips A-J. There are two different fleets of trucks: one is the own fleet and the other is a chartered fleet. The truck of the own fleet has the shorter time capacity. The right part of Fig. 2 is a Gantt chart that shows the assignment of the trucks to the merged trips A-J. One of the two chartered trucks is not employed at all. Note that in the problem setting, the cost is incurred in proportion to the travel time despite the truck time capacity. This implies that for the chartered fleets, the trucks are assumed to be sent back to the leasing companies as soon as they finish the distribution tasks on a day. For instance, Truck 3 is sent back to its company after finishing Trip F while 5 hours remain in its time capacity.

Fig. 2 and Tables 1 - 3

----

3. Related literature

Our problem is closely related to the vehicle routing problem with backhauls (VRPB) that finds an optimal set of orders (or routes) visiting deliveries (or linehauls) and pickups (or backhauls) after leaving a particular depot where in each route, any pickup loads are carried on the return trip after all deliveries. A heuristic method based on the saving algorithm of Clarke and Wright (1964) for the VRP, is proposed by Goetschalckx and Jacob-Blecha (1989). Anily (1996) describes a heuristic method for the VRPB that converges to the optimal solution, under mild probabilistic conditions and when there are no restrictions on the order in which the linehaul and backhaul customers are visited. A 1.5 approximation algorithm for the single vehicle version of the VRPB (i.e., TSP with backhauls) is developed by Gendreau et al. (1997). Osman and Wassan (2002) develop a tabu search algorithm for the VRPB where all backhaul customers are to be visited after serving all the linehaul customers.

Toth and Vigo (1997) propose an exact branch and bound method for the VRPB with a Lagrangian lower bound that is strengthened by adding valid inequalities in a cutting plane fashion. Mingozzi et al. (1999) describe another integer programming formulation of the VRPB based on a set-partitioning approach and develop an exact solution method that employs a valid lower bound to the optimal solution cost by combining different heuristic methods for solving the dual of the LP-relaxation of the exact formulation. All the above existing studies exploit detailed algorithms. Contrary to these, Daganzo and Hall (1993) discuss the pros and cons of various broad routing schemes and quantify their performance with simple distance formulas.

There are some variants of the VRPB. As one of them, Min et al. (1989, 1992) propose a multi-depot version of the VRPB. Their heuristic decomposes the problem into three steps: (1) aggregation of pickups and deliveries into clusters; (2) assignment of clusters to depots and routes; and (3) routing vehicles. Hall (1991) creates and evaluates spatial models for the VRPB with multiple depots. Jordan and Burns (1984) consider a backhaul problem with two depots, where only one backhaul load to a depot can be serviced after one linehaul from the other depot. In their study, each depot has multiple customers each of who must be serviced independently unlike the VRP. Thus, the structure of the routing can be referred to as the single stop route. They propose a greedy algorithm that optimally matches empty vehicles to form backhaul loops. Jordan (1987) extends this problem for a routing system with more than two depots. Nagy and Salhi (2005) propose heuristics for two backhaul problems: one with mixed pickups and deliveries where linehauls and backhauls can occur in any sequence and the other with simultaneous pickups and deliveries where customers may simultaneously receive and send loads. They consider single and multiple depot versions for those two variants. Ball et al. (1983) propose simple greedy heuristics especially tailored for bulk pickup and delivery routing by using a private fleet (actually long-leased fleet) and an outside carrier. Each pickup is coupled with its destination and no other pickup and/or delivery points are visited in between. Each route services a set of these D-P pairs. As each private vehicle has a time capacity for servicing the customers, additional private vehicles or the outside carrier undertake an excess workload if it emerges. The costly private truck requires an optimal fleet size and assignment of workload to the outside carrier. Their problem is not related to the backhauling in the strong sense because a pickup and a delivery in a pair is not separable; therefore by considering a D-P pair as an entity their problem could be referred to as the VRP with time capacity. Fisher et al. (1995) develop a network flow-based heuristic for a pickup-delivery problem that is similar to Ball et al. (1983).

Three studies of De Meulemeester et al. (1997), Bodin et al. (2000) and Jula et al. (2005) are closely related to this study. De Meulemeester et al. (1997) address the optimal route construction for skip collections and deliveries incident to disposal facilities. They assume two types of customers: domestic and industrial ones. Skip collection and delivery for a domestic customer exactly corresponds to container pickup and delivery of this study, while those for an industrial customer are more complex. Some of the individual skip collections and deliveries (they are all referred to as elementary routes) for both domestic and industrial customers for feasible working days are merged, so as to minimize the total distance traveled by vehicles. Note that in their study, each vehicle can perform the skip distribution within the maximal allowed length of a working day. Two simple heuristics and an exact enumerative algorithm are developed for the skip collection and delivery problem. Bodin et al. (2000) deal with a similar problem, which is a special case of the problem considered in De Meulemeester et al. in that while De Meulemeester et al. allow for some skip trips to be serviced at one of several disposal facilities, Bodin et al. assume all trips to be serviced at a single disposal facility. The VRPFC in this paper is a special case of the problem considered in Bodin et al., in that each elementary route in the VRPFC is a shuttle trip between the depot and a customer site (shipper or consignee). The VRPFC is also different from problems in De Meulemeester et al. and Bodin et al. from a viewpoint of the vehicle fleet. The VRPFC employs own and chartered fleets whereas the those studies deploy a single fleet. Jula et al. (2005) treat the multi-traveling salesman problem with time windows for container movements. In their study, a lot of container movements are performed by multiple trailers, each of which undertakes a series of two types of (loaded or empty) container movements: one from an intermodal facility (port, rail-yard, etc.) to a customer and the other vice versa. As each delivery is interrelated to a pickup as an opponent, pickup-delivery pairs are regarded as corresponding nodes so that the problem no longer has characteristics of the backhaul problem. Similar to this study, in their study the trailers have working shift and must come back to the depot before their shifts are over.

The problem we propose here is a single-stop backhaul problem and has already been addressed as a sub-problem for the multi-depot VRPB by Min et al. (1989, 1992). They employ a linear programming (LP) to solve this sub-problem and report that the LP yields pure integer solutions. However, the integral solution is not necessarily guaranteed since as will be shown later, that sub-problem is NP-Hard.

4. Problem formulation and Lagrangian relaxation

4.1. Problem formulation

The costs concerned with this problem are the ones incurred by revenue and empty truck hauling. For the own fleet, those costs substantially correspond to the fuel costs, which is proportional to the travel length or time. We assume that the outside truck chartering costs are also imposed in proportion to the travel length. In the formulation these costs are specified with each possible D-P pair.

The VRPFC may be formulated as follows:

[P] Minimize
$$\sum_{i \in V^D} \sum_{j \in V^P} \sum_{k \in K} CR_{ijk} x_{ijk}, \tag{1}$$

subject to
$$\sum_{i \in V^D} \sum_{k \in K} x_{ijk} = 1 \qquad \forall j \in V^P, \tag{2}$$

$$\sum_{j \in V^P} \sum_{k \in K} x_{ijk} = 1 \qquad \forall i \in V^D, \tag{3}$$

$$\sum_{i \in V^D} \sum_{j \in V^P} T_{ij} x_{ijk} \le L_k \qquad \forall k \in K, \tag{4}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i \in V^D, j \in V^P, k \in K, \tag{5}$$

where,

CR_{ijk} the cost for container haulage by truck *k* from the depot visiting delivery point *i* to pickup point *j* on the way back to the depot

T_{ij} the hauling time from the depot visiting delivery point *i* to pickup point *j* on the way back to the depot (including cargo handling time at customer points)

K the set of trucks

V^D the set of delivery points

V^P the set of pickup points

L_k the time capacity of truck *k*

x_{ijk} =1 if a container is hauled from *i* to *j* by truck *k*, =0 otherwise

The objective is the sum of the costs incurred with revenue and empty truck hauling. Constraint sets

(2) and (3) ensure that a delivery and a pickup are serviced by any truck, respectively. Constraint set (4) guarantees that each vehicle must service its assigned workload within its time capacity.

We assume $|V^D| = |V^P|$ where $|V^D|$ is the cardinality of V^D. Even if both values are not identical, i.e., if there are more delivery points than pickup points or vice versa, the problem is adaptable by introducing dummy customers and dummy arcs of null length associated with them. The transportation cost CR_{ijk} is assumed to be proportional to its hauling distance.

It is noteworthy that even if there are the same number of delivery points as pickup points, we may find a better solution by introducing dummy nodes and arcs for the following reasons: Suppose that there are five deliveries D1–D5 and five pickups P1–P5. In a solution without dummy nodes and arcs, we use two own trucks (one serving depot–D1–P1–depot–D2–P2–depot, and the other serving depot–D3–P3–depot–D4–P4–depot) and one expensive outside truck serving depot–D5–P5–depot, where both own trucks left an enough time to undertake one more customer (either delivery or pickup). By introducing the dummies to composite pairs of a customer and a dummy, this solution can be improved with an own truck for depot–D1–P1–depot–D2–P2–depot–D5–depot and the other own truck for depot–D3–P3–depot–D4–P4–depot–P5–depot. However, as we include the same number of dummy nodes as deliveries and pickups, the improvement must be offset by a considerably growing computational effort.

The VRPFC is closely related to several well-studied problems. Due to the travel cost assumption, if one has a considerable fleet size of the cheapest truck, the allocation of trucks to D-P pairs is not difficult to obtain. In such a case, the VRPFC reduces to the classical assignment problem (AP). If either V^D or V^P is empty, the question is which trucks are to be allocated to the set of customers subject to the time capacity, and the problem reduces to the generalized assignment problem (GAP). At first glance the association with AP is encouraging, as there exists an efficient procedure, the Hungarian method. Due to the truck assignment process, the VRPFC turns into an NP-hard problem that is difficult to solve.

Lemma 1. *The decision analog of VRPFC is NP-hard.*

Proof. Consider an instance of VRPFC where there are no delivery customers. Then, the decision analog of VRPFC is equivalent to the decision analog of GAP, which is known to be NP-hard (Garey and Johnson, 1979).□

4.2. Lagrangian relaxation

As shown above, the VRPFC is NP-hard. A lot of NP-hard problems are solved optimally by some exact methods such as a branch and bound procedure, but the solving process would be time-consuming especially for problems of practical size. As the VRPFC may be solved often by a trucking firm during a business hour due to frequent delivery information updates, fast computation is needed. This encourages us to develop a heuristic for the problem. This heuristic procedure employs a subgradient optimization procedure based on the following Lagrangian relaxation of the original formulation [P].

[PR] Minimize
$$\sum_{i\in V^D}\sum_{j\in V^P}\sum_{k\in K} CR_{ijk}x_{ijk} + \sum_{k\in K}\lambda_k\left(\sum_{i\in V^D}\sum_{j\in V^P} T_{ij}x_{ijk} - L_k\right)$$
$$= \sum_{i\in V^D}\sum_{j\in V^P}\sum_{k\in K}\left(CR_{ijk} + \lambda_k T_{ij}\right)x_{ijk} - \sum_{k\in K}\lambda_k L_k \tag{6}$$

subject to (2), (3) and (5),

where λ_k is a Lagrangian multiplier for truck *k*.

Substituting CR'_{ijk} for $CR_{ijk} + \lambda_k T_{ij}$, [PR] becomes the following.

$[PR']$ Minimize
$$\sum_{i\in V}\sum_{j\in V^P}\sum_{k\in K} CR'_{ijk}x_{ijk} \tag{7}$$

subject to (2), (3) and (5).

5. Solution procedure

5.1. Lower bound and feasible solution

As will be stated later, a proposed solution procedure is built around solving an AP and a GAP. $[PR']$ is no longer constrained by trucks *k*, i.e., the cheapest trucks can be employed as long as all the distribution tasks are finished; therefore it can be rewritten as follows:

[PL] Minimize
$$\sum_{i\in V^D}\sum_{j\in V^D}\hat{C}_{ij}y_{ij}, \tag{8}$$

subject to
$$\sum_{i\in V^D} y_{ij} = 1 \qquad \forall j \in V^P, \tag{9}$$

$$\sum_{j \in V^P} y_{ij} = 1 \qquad \forall i \in V^D, \tag{10}$$

$$y_{ij} \in \{0, 1\} \qquad \forall i \in V^D, j \in V^P, \tag{11}$$

where $\hat{C}_{ij}$ is defined as $\hat{C}_{ij} = \underset{k \in K}{\mathrm{Min}}\left(CR'_{ijk}\right)$ for all *i*, *j*; and y_{ij} =1 if a container is hauled from i to j, =0 otherwise.

For a given *i*, *j* pair, let k^* be the value of index that corresponds to the minimum cost CR'_{ijk} such that $\hat{C}_{ij} = CR'_{ijk^*}$. Then, the solution of problem [P] can be obtained from the solution of problem [PR] as follows:

For every *i*, *j* pair let $x_{ijk^*} = y_{ij}$; for all other *k* let $x_{ijk} = 0$.

Note [PL] is an AP that is easy to solve. In addition to providing us with a lower bound for problem [P], the solution to this problem forms a basis for a GAP which is used to identify a feasible solution for [P].

Given the optimal solution to [PL], the following GAP provides a feasible solution for [P]:

[PF] Minimize
$$\sum_{k \in K} \sum_{p \in P} \overline{C}_{kp} z_{kp}, \tag{12}$$

subject to
$$\sum_{k \in K} z_{kp} = 1 \qquad \forall p \in P, \tag{13}$$

$$\sum_{p \in P} T_p z_{kp} \le L_k \qquad \forall k \in K, \tag{14}$$

$$z_{kp} \in \{0, 1\} \qquad \forall k \in K, \forall p \in P, \tag{15}$$

where

$p \in P$ a D-P pair constructed by the problem [PR],

$\overline{C}_{kp}$ the transportation cost incurred in servicing D-P pair *p* by truck *k*,

T_p the hauling time of D-P pair *p*,

z_{kp} =1 if D-P pair *p* is serviced by truck *k*, =0 otherwise.

In essence, the solution to [PL] forms D-P pairs. In the solution to [PF] these D-P pairs are kept constant and the trucks are allocated to cover these pairs to minimize the total transportation cost. Constraint set (13) ensures that all of the D-P pairs determined in [PL] are satisfied. Truck time capacities are enforced through constraints (14). It is noted that [PR] provides a lower bound for [P] which can be used to judge the quality of the feasible solution obtained from [PF].

The problem [PF] is a reduced form of the GAP due to constraint set (14). The following modified constraint set:

$$\text{subject to} \quad \sum_{k \in K} z_{kp} = 1 \qquad \forall p \in P, \tag{16}$$

$$\sum_{p \in P} z_{kp} \le L_k \qquad \forall k \in K, \tag{17}$$

is completely equivalent to that of the GAP (Lee, 1992). Thus, [PF] cannot be solved in polynomially bounded time. A number of researches have been done over years to develop effective enumeration algorithms that solve problems of reasonable size to optimality (Cattrysse and Wassenhove, 1992). Most algorithms are implemented with using branch and bound techniques and relaxation of the assignment or knapsack constraints; therefore although they produce a good solution, their computational burdens are considerable. We employ a subgradient optimization procedure to find a good set of Lagrangian multipliers. A solution algorithm with heavy computational burden is restrictive for solving the GAP because the GAP is solved many times in the iterative computation of the subgradient method. In addition to this, through our computational experience, a sophisticated algorithm to find a better feasible solution in the subgradient optimization process does not always guarantee good multipliers and consequently does not identify a good final feasible solution in the entire subgradient procedure (Imai et al., 2001). For the above reason, we obtain a solution to the [PF] by solving not the [PF] but the [GBPP], which will be presented later, with a straightforward heuristic.

The GAP may be stated as follows. Given *m* integers $(T_1, T_2, ..., T_m)$ where T_p is the size of item *p*, and another *n* integers $(L_1, L_2, ..., L_n)$ where L_k is the size of bin *k*, the problem is to assign each item to a bin such that that sum of the item sizes in a bin does not exceed the bin size, while minimizing the sum of the item costs in bins, $\overline{C}_{kp}$. There is a similar packing problem well known as the Bin Packing Problem (BPP) which follows: Given *m* integers $(T_1, T_2, ..., T_m)$ where

T_p is the size of item *p*, and another integer *B*, the bin size, we are asked to assign each item to a bin such that the sum of the item sizes in a bin does not exceed *B*, while minimizing the number of bins used. The objective of BPP is the number of bins. This is also understandable as the minimization of the total bin cost to be used when every bin has an identical cost. Based on this, we here consider a generalization of BPP such that each bin has a unique size L_k and a unique cost per unit size C_k. Then, this problem may be formulated as follows:

[GBPP] Minimize $$\sum_{k \in K} C_k L_k y_k \,, \tag{18}$$

subject to $$\sum_{k \in K} z_{kp} = 1 \qquad \forall p \in P \,, \tag{19}$$

$$\sum_{p \in P} T_p z_{kp} \le L_k y_k \qquad \forall k \in K \,, \tag{20}$$

$$z_{kp} \in \{0, 1\} \qquad \forall k \in K, \forall p \in P \,, \tag{21}$$

$$y_k \in \{0, 1\} \qquad \forall k \in K \,, \tag{22}$$

where y_k =1 if bin *k* is used to pack items, =0 otherwise.

For the sake of simplicity in statement, the bin and item are hereafter referred to as the truck and workload. As will be described later, this problem can be approximately solved by simple greedy heuristics of the BPP with minor modifications. Furthermore, one has the following lemma.

Lemma 2. [GBPP] *yields the upper bound to* [PF].

Proof. Without loss of generality, constraint set (14) of [PF] can be rewritten as (20). Assuming for each truck the cost per unit hour is identical regardless of its state: loaded or empty, the total cost incurred with workloads carried out by truck *k* is no more than the truck cost for the entire time capacity. Consequently, there is obviously the following relationship between the objective functions (12) and (18).

$$\sum_{p \in P} \overline{C}_{kp} z_{kp} = \sum_{p \in P} \overline{C}_{kp} z_{kp} y_k \le C_k L_k y_k \qquad \forall k \in K \,. \tag{23}$$

Due to the above condition (23) and the same structure of constraints in both [GBPP] and [PF], the proof is completed.□

For [GBPP] we develop a BPP heuristic-based procedure. As the BPP is NP-hard, there are some heuristics developed. We employ the so-called ”First Fit Decreasing Order” heuristic which is reported to be one of the best heuristics (Bramel and Scimchi-Levi, 1997). This heuristic is slightly modified to be applicable to the problem [GBPP]. See the formal statement of the algorithm in Appendix A.

5.2. Entire solution procedure

The entire solution procedure can be formally summarized as follows:

Step 1. In a problem [P], relax truck time capacities to obtain a problem [PR]. To service delivery and pickup customers use the “lowest cost” truck to identify transportation cost $\hat{C}_{ij}$ in defining a problem [PL]. Note that the “lowest cost” is a Lagrangian cost.

Step 2. Solve the problem [PL]. For each delivery point the solution identifies the pickup point to be coupled.

Step 3. Define a problem [PF]. Use an actual distribution cost $\overline{C}_{kp}$, as a cost of delivery and pickup routing in this Generalized Assignment Problem [PF].

Step 4. Solve the problem [PF]. The solution indicates which trucks are used to perform D-P pairs, which are determined in Step 2. Solutions to the problems [PL] and [PF] together define a feasible solution for the problem [P].

The quality of the feasible solution by the above procedure strongly depends on one’s ability to determine good Lagrangian multipliers λ_k s. Essentially, with a zero multiplier in [PL] for each truck *k*, we assume that we can use the truck with the lowest transportation cost. This may lead to an infeasible solution to [P] as some trucks may remain under service beyond their time capacity. As the multipliers corresponding to the over-utilized trucks are increased, the “costs” of these trucks are increased in the Lagrangian problem, [PL], and the under-utilized trucks become more attractive. In other words, the solution to [PL] gets closer to a feasible solution to [P].

Good multipliers are also important as the quality of the lower bound, i.e., the objective function value of [PR], is a function of these multipliers. The best lower bound corresponding to the optimal multiplier vector λ^* is determined as

$$Z_{PL}\left(\lambda^*\right) = \underset{\lambda}{\mathrm{Max}}\left(Z_{PL}(\lambda)\right),$$

where $Z_{PL}(\lambda)$ is the value of the Lagrangian function with a multiplier set (vector) λ. There are a number of different approaches to finding a good, if not optimal, set of Lagrangian multipliers (Fisher, 1981). We have selected to use the subgradient optimization procedure. The subgradient method is an adaptation of the gradient method in which the subgradients replace the gradients (Bazaara and Goode, 1979; Held et al., 1974). Since the approach has been widely utilized and well understood we will not repeat details here. It suffices to note that as a termination criterion we fixed the maximum number of iterations at 200. The procedure is also terminated if the gap between the feasible solution value and the Lagrangian bound becomes less than 1. Given integer objective function coefficients, this condition is sufficient to detect an optimal solution.

[PL] is solved to obtain a lower bound for [P], while [PF] is solved to determine a feasible solution to [P]. Note that the objective function value of [PR] is equal to the objective function value of [PL] plus a constant. At the time of termination, the subgradient optimization procedure reports the best feasible solution and the best lower bound generated in all the iterations. A detailed statement of the procedure is included in Appendix B.

In the feasible solution at the initial iteration that is shown in Fig. 2, the merged trips are formed by solving the relaxation problem [PL]. The trips are supposed to be assigned to the cheapest trucks, whilst the assignment is not revealed in Fig. 2. The assignment of the trips to the trucks is determined by solving [GBPP] that follows the solving of [PR] at each iteration.

Fig. 3 illustrates the final feasible solution by the solution procedure where the feasible solution at the first iteration is shown in Fig. 2. As can be seen, the chartered time duration of truck 3 is reduced by one hour in the final solution.

Fig. 3

----

6. Numerical experiments

The solution procedure was coded in “C” language on a workstation with CPU: SPARC64 GP (275MHz). Computation times reported here are in CPU seconds on this computer. Problems used in these experiments were generated randomly, but systematically, in order to obtain problems with differing levels of cost or time capacity trade-offs. We provided five basic problems, four of which have the number of customer points (a half is pickup points and the other half is delivery points), v, ranged from 60-100 by 20. The other problem has $v = 200$, in order to examine the performance of the algorithm for an extremely huge size of problems.

The problems are expected to get more difficult with more widespread demand locations from the terminal because D-P pairs with a longer hauling distance are likely constructed, resulting in inefficient usage of the truck working time. The coordinates of customer points P_i were defined as

$$P_i = (R1_i \times 100 - 50) \times (r + R2_i \times 0.4),$$

where r is a parameter of 0.2, 0.4, or 0.6, and $R1_i$ and $R2_i$ are two series of random numbers both from a uniform distribution between 0 and 1. By changing r, we can determine the range of customer's location from the terminal. $R1_i$ defines a basic location for each point, while $R2_i$ gives a variance of the location for a particular range of a customer's location set. We provided 12 problem samples with a combination of v and r. For each problem sample, four variations were generated by placing pickup and delivery points randomly over a (100 x 100) plane with different seed sets of random numbers.

The truck fleet involves three kinds of trucks: A (own truck), B (chartered truck), and C (chartered truck with higher cost). The fleet size depends on the demand size as presented in Table 4. Note that since all the points must be serviced, we can charter as many trucks with the highest cost as possible when cheaper trucks are not sufficient. Three sets of truck cost variations are defined in Table 5. Besides the cost variation, a fleet is given three alternatives of the working time capacity as illustrated in Table 6.

Based on 12 problem samples, a total of 432 problems were created by varying (a) seed sets of random numbers, (b) the fluctuation in trucking costs, and (c) the fluctuation in the truck working time capacities.

Tables 4-6

While the 432 problems are solved, average values associated with problem solutions over the four variations are reported in Tables 7-9, each of which outlines solutions with a specific cost variation. Columns one to three in these tables indicate the index of the working time capacity, the problem size in terms of the number and the distribution of the customer points. The gap between the best feasible solution value and the lower bound is reported as a percent of the lower bound in column four. In column five, computing times are given in CPU seconds. The last two columns indicate in which iterations of the subgradient optimization algorithm the best feasible solution and the best lower bound were determined. The last two columns provide information regarding the convergence of the subgradient optimization algorithm. One observation which is not readily apparent from these columns, is that the objective function does not improve at every iteration but displays some fluctuation as we search through the solution space.

Tables 7-9

Except for $v = 200$, the maximum of the average gap between the feasible solution and the lower bound is 15 percent of the lower bound for cases with costs 1 and 2. Generally the average gap increases as the truck cost increases. In fact, most problems have the solution with less than a 5 percent gap for cost 1 and less than a 15 percent gap for cost 2, whilst most gaps are less than 50 percent for cost 3. The gap also increases with decreasing time capacity, whilst the tendency is not as prominent.

Furthermore, the procedure quickly identifies good feasible solutions, at less than the 30th iteration, with few exceptions. Later iterations only affect the quality of the lower bound, indicating that the subgradient procedure can be stopped sooner without significantly affecting the solution quality. Of course, doing so would result in larger gaps and one would have less confidence in the quality of the solution obtained.

There were no significant differences in the gap value among various problem sizes except for $v = 200$. As for the customer’s location spread out in the area, there is no obvious relationship between the gap and r. However, quite good feasible solutions are obtained as the optimal solutions (i.e., solutions with a null gap) are identified for almost all problems with $r = 0.2$ and $L_k \geq 2$ for all cost variations. The computing times range from one second for the smallest problems to nearly 130 seconds for the largest ones.

7. Concluding remarks

We have addressed the VRP with pickups and deliveries of full container load incident to an intermodal terminal. The substantial cost and time savings are expected by efficient linkage between pickup and delivery tasks, although this is possible only in the case that when the needs of shipment arise, suitable containers become available after deliveries to consignees. This, we call the VRPFC, is related to VRP with backhaul and had been already addressed as a sub-problem in an existing backhaul problem. It was reported that the VRPFC yielded an integer solution when solved by an LP. This, however, is not always the case since the problem is NP-hard as proven in this study. We have, then, developed a heuristic procedure for solving the VRPFC. The procedure is based upon a Lagrangian relaxation that enables us to identify an approximate solution via solving a series of sub-problems, which are the classical assignment problem and the generalized assignment problem. Due to the difficulty in solving the generalized assignment problem efficiently, feasible solutions to the VRPFC are obtained by a modified heuristic for the bin packing problem algorithm that is also NP-hard. The heuristic procedure was tested on 432 randomly generated problems and proved to be efficient in solving the problems. The gaps of most problems were under 10%. The results of the tests demonstrated that the procedure developed here was able to efficiently solve large instances of the VRPFC.

Acknowledgement

The authors express a sincere gratitude to Ms. Zhang Wei for her contribution in implementing the algorithm. The initial stage of this study was carried out while the first author of this paper, Akio Imai, was at the Ohio State University as a visiting researcher with a financial support by the Fulbright Program.

References

Anily, S., 1996. The vehicle-routing problem with delivery and back-haul options. *Naval Research Logistics* **43,** 415-434.

Ball, M.O., Golden, B.L., Assad, A.A., Bodin, L.D., 1983. Planning for truck fleet size in the presence of a common-carrier option. *Decision Science* **14,** 103-120.

Bazaara, M.S., Goode, J.J., 1979. A survey of various tactics for generating Lagrangian multipliers in the context of Lagrangian duality. *European Journal of Operational Research* **3,** 322-338.

Bodin L., Mingozzi, A., Baldacci, R., Ball, M., 2000. The rollon-rolloff vehicle routing problem. *Transportation Science* **34,** 271-288.

Bramel, J., Simchi-Levi, D., 1997. *The Logic of Logistics, Springer-Verlag*. New York.

Cattrysse, D.G., Van Wassenhove, L.N., 1992. A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research* **60,** 260-272.

Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* **12,** 568-581.

Daganzo, C.F., Hall, R.W., 1993. A routing model for pickups and deliveries: No capacity restrictions on the secondary items. *Transportation Science* **27,** 315-329.

De Meulemeester, L., Laporte, G., Louveaux, F.V., Semet, F., 1997. Optimal sequencing of skip collections and deliveries. *Journal of the Operational Research Society* **48,** 57-64.

Fisher, M.L., 1981. Lagrangian relaxation method for solving integer programming problems. *Management Sciences* **27,** 1-18.

Fisher, M.L., Tang, B., Zheng, Z., 1995. A network flow based heuristic for bulk pickup and delivery routing. *Transportation Science* **29,** 45-55.

Garey, M.R., Johnson, D.S., 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., San Francisco, CA.

Gendreau, M., Laporte, G., Hertz, A., 1997. An approximation algorithm for the traveling salesman problem with backhauls. *Operations Research* **45,** 639-641.

Goetschalckx, M., Jacobs-Blecha, C., 1989. The vehicle routing problem with backhauls. *European Journal of Operational Research* **42,** 39-51.

Hall, R.W., 1991. Characteristics of multi-stop/multi-terminal delivery routes, with backhauls and unique items. *Transportation Research* **25B,** 391-403.

Held, M., Wolfe, P., Crowder, H.P., 1974. Validation of subgradient optimization. *Mathematical Programming* **6,** 62-88.

Imai, A, Nishimura, E., Papadimitriou, S., 2001. The dynamic berth allocation problem for a container port. *Transportation Research* **35B,** 401-417.

Jordan, W.C., 1987. Truck backhauling on networks with many terminals. *Transportation Research* **21B,** 183-193.

Jordan, W.C., Burns, L.D., 1984. Truck backhauling on two terminal networks. *Transportation Research* **18B,** 487-503.

Jula, H., Dessouky, M., Ioannou, P., Chassiakos, A, 2005. Container movement by trucks in metropolitan networks: Modeling and optimization. *Transportation Research* **41E**, 235-259.

Lee, M.-K., 1992. A storage assignment policy in a man-on-board automated storage/retrieval system. *International Journal of Production Research* **30,** 2281-2292.

Min, H., Current, J., Schilling, D., 1989. The multiple depot vehicle routing problem with backhauling. Working Paper, The Ohio State University.

Min, H., Current, J., Schilling, D., 1992. The multiple depot vehicle routing problem with backhauling. *Journal of Business Logistics* **13,** 259-288.

Mingozzi, A., Giorgi, S., Baldacci, R., 1999. An exact method for the vehicle routing problem with backhauls. *Transportation Science* **33,** 315-329.

Nagy, G., Salhi, S., 2005. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research* **162,** 126-141.

Osman, I.H., Wassan, N.A., 2002. A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls. *Journal of Scheduling* **5,** 263-285.
Toth, P, Vigo, D., 1997. An exact algorithm for the vehicle routing problem with backhauls. *Transportation Science* **31,** 372-385.

APPENDIX A. HEURISTIC FOR THE GBPP

The following heuristic for the GBPP is a straightforward extension of "First Fit Decreasing Order" heuristic algorithm for the BPP.

Step 1. Sort the workloads in decreasing order of size T_p and make a list with sorted workloads.

Step 2. Sort the trucks in increasing order of cost per unit size C_k.

Step 3. Take the first truck.

Step 4. Put as many workloads as possible on the truck from the list.

Step 5. If no workloads are left, STOP.

Step 6. If no time capacity of the truck is available, take the next truck and go to step 4.

APPENDIX B. THE SUBGRADIENT PROCEDURE

In the subgradient optimization procedure, given a set of starting multipliers λ^0, a sequence of multipliers is generated using the following rule:

$$\lambda^{k+1} = \lambda^k + t_k\left(Ax^k - b\right),$$

where x^k is an optimal solution to the Lagrangian problem $PR\left(\lambda^k\right)$ and t_k is a positive scalar step size and $\left(Ax \le b\right)$ is the set of constraints being relaxed (i.e., constraint set 4). We use the following step size that has been used frequently in the past (Fisher, 1981):

$$t_k = d_k\left(\overline{Z} - Z_{PL}(\lambda)\right) / \left\|Ax^k - b\right\|^2,$$

where $\overline{Z}$ is the best known feasible solution value, $\left\|Ax^k - b\right\|$ is the Euclidean norm of $\left(Ax \le b\right)$, and d_k is a scalar satisfying the relation $0 \le d_k \le 2$. This scalar is set to 2 at the start of the procedure and is halved whenever the bound does not improve in 20 consecutive iterations. The following is a formal statement of the procedure:

Step 1. Maxiter=200, d =2, $\overline{Z}$ $=1\times10^8$, Iter=1, k =1, BestLB=0, $\lambda = \lambda^* = \{0\}$.

Step 2. Solve problem [PL], and calculate the objective function of [PR]. Let Z_{PR} be the solution value of [PR]. If Z_{PR}>BestLB, let BestLB=Z_{PR}, Iter=1, $\lambda^* = \lambda$, otherwise Iter=Iter+1.

Step 3. Solve problem [PF]. Let FEAS be the objective function value of the feasible solution. If FEAS<$\overline{Z}$, let $\overline{Z}$ =FEAS. If $\overline{Z}$ -BestLB<1, STOP.

Step 4. Let $k = k+1$. If k >Maxiter, STOP; otherwise continue.

Step 5. If Iter>20 let Iter=1, $\lambda = \lambda^*$, $d = d/2$, otherwise calculate step size t_k and update multipliers λ^k s.

Step 6. If λ^k <0, then set λ^k =0. Go to Step 2.

Table 1. Distances between customer points (km)

Points i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	7	23	7	14	2	9	7	26	7	10	13	13	19	13	6	21	20	10	8	29
11	32	72	34	36	30	44	40	60	26	45	-	-	-	-	-	-	-	-	-	-
12	37	47	34	55	28	28	28	73	41	38	-	-	-	-	-	-	-	-	-	-
13	45	85	47	43	42	56	52	68	38	56	-	-	-	-	-	-	-	-	-	-
14	41	66	40	42	28	37	33	78	37	30	-	-	-	-	-	-	-	-	-	-
15	22	59	23	29	16	30	26	56	15	31	-	-	-	-	-	-	-	-	-	-
16	57	63	54	67	44	45	44	95	57	45	-	-	-	-	-	-	-	-	-	-
17	44	61	42	68	42	47	46	71	51	58	-	-	-	-	-	-	-	-	-	-
18	32	69	33	28	25	39	34	65	25	37	-	-	-	-	-	-	-	-	-	-
19	30	59	30	33	18	29	25	67	26	25	-	-	-	-	-	-	-	-	-	-
20	72	80	70	77	58	61	60	110	71	58	-	-	-	-	-	-	-	-	-	-

Note: Point zero refers to the depot.

Table 2. Hauling times of merged trips of the depot-point i-point j-the depot (hours)

Points i \ j	1	2	3	4	5	6	7	8	9	10
11	2	4	2	2	2	2	2	4	1	3
12	2	3	2	3	1	1	1	4	2	2
13	3	5	3	2	2	3	3	4	2	3
14	2	4	2	2	1	2	2	5	2	2
15	1	3	1	1	1	2	1	3	1	2
16	3	4	3	4	2	3	2	6	3	3
17	2	4	2	4	2	3	3	4	3	3
18	2	4	2	1	1	2	2	4	1	2
19	2	3	2	2	1	1	1	4	1	1
20	4	5	4	5	3	4	4	7	4	3

Table 3. Truck cost

Truck type	Own	Chartered
Cost per km	30	60
Time capacity (hours)	8	10
# of Trucks	2	2

Table 4. Truck fleet size

v	Fleet size (# of trucks)		
	A (own)	B (chartered)	C (chartered)
60	6	2	∞
80	8	2	∞
100	10	2	∞
200	20	2	∞

Table 5. Truck cost per unit distance

Fleet	A (own)	B (chartered)	C (chartered)
Cost 1	3	4	5
Cost 2	3	5	7
Cost 3	3	6	9

Table 6. Truck working time capacity per day in hours (L_k)

Fleet	A (own)	B (leased)	C (leased)
Capacity 1	8	11	14
Capacity 2	10	12	14
Capacity 3	12	13	14

Table 7. Computational results for cost 1

L_k	v	r	Gap[a] (%)	CPU (second)	Best Feas. at Iter.	Best Bound at Iter.
1	60	0.2	5.33	2.5	10.5	65.5
		0.4	2.93	2.8	39.3	192.8
		0.6	2.81	2.8	11.8	197.3
	80	0.2	4.77	6.5	9.5	97.8
		0.4	3.50	6.6	21.5	198.0
		0.6	2.98	7.0	25.8	192.0
	100	0.2	5.24	13.4	18.3	46.8
		0.4	3.88	14.2	51.0	199.0
		0.6	4.46	14.0	29.5	158.8
	200	0.2	8.28	128.0	1.0	1.0
		0.4	27.98	128.0	21.0	1.0
		0.6	24.13	131.4	19.0	100.5
2	60	0.2	0.04	0.7	1.0	49.0
		0.4	2.46	2.7	36.8	188.0
		0.6	2.12	2.8	10.8	195.0
	80	0.2	0.00	0.0	1.0	1.0
		0.4	5.65	6.6	23.5	146.0
		0.6	2.23	7.0	18.0	191.0
	100	0.2	0.00	0.0	1.0	1.0
		0.4	15.47	13.4	17.5	48.8
		0.6	2.72	14.1	28.3	190.0
	200	0.2	0.00	0.5	1.0	1.0
		0.4	19.40	128.0	1.0	1.0
		0.6	31.48	128.0	21.0	1.0
3	60	0.2	0.00	0.0	1.0	1.0
		0.4	5.03	2.6	16.3	147.8
		0.6	2.48	2.8	22.5	194.5
	80	0.2	0.00	0.0	1.0	1.0
		0.4	6.83	6.5	14.0	94.0
		0.6	2.22	7.0	18.3	194.3
	100	0.2	0.00	0.0	1.0	1.0
		0.4	7.25	13.4	13.8	1.0
		0.6	7.12	13.8	36.0	148.5
	200	0.2	0.00	0.5	1.0	1.0
		0.4	9.84	127.9	1.0	1.0
		0.6	24.95	128.0	16.0	1.0

[a] Gap=(Feasible solution value-lower bound) * 100/lower bound.

Table 8. Computational results for cost 2

L_k	v	r	Gap[a] (%)	CPU (second)	Best Feas. at Iter.	Best Bound at Iter.
1	60	0.2	9.73	2.6	14.8	84.0
		0.4	5.16	2.8	23.3	194.0
		0.6	4.80	2.9	17.5	191.8
	80	0.2	9.92	6.6	8.8	91.5
		0.4	6.33	6.8	27.5	199.0
		0.6	4.64	7.2	23.8	196.0
	100	0.2	10.47	13.5	17.3	46.3
		0.4	7.10	14.4	24.8	194.8
		0.6	7.53	14.3	32.5	156.3
	200	0.2	16.57	128.0	1.0	1.0
		0.4	55.79	127.7	21.0	1.0
		0.6	46.53	133.1	22.8	59.3
2	60	0.2	0.35	0.7	1.0	11.8
		0.4	4.24	2.8	25.8	192.8
		0.6	3.84	2.9	17.3	195.3
	80	0.2	0.00	0.0	1.0	1.0
		0.4	14.63	6.7	64.5	185.3
		0.6	3.81	7.1	23.3	194.5
	100	0.2	0.00	0.0	1.0	1.0
		0.4	31.15	13.4	30.0	48.8
		0.6	4.54	14.3	33.5	197.0
	200	0.2	0.00	0.5	1.0	1.0
		0.4	38.80	127.8	1.0	1.0
		0.6	62.58	127.8	21.0	1.0
3	60	0.2	0.00	0.0	1.0	1.0
		0.4	9.96	2.6	29.3	148.5
		0.6	4.43	2.9	30.5	190.8
	80	0.2	0.00	0.0	1.0	1.0
		0.4	13.31	6.4	19.0	100.0
		0.6	3.85	7.1	14.3	194.5
	100	0.2	0.00	0.0	1.0	1.0
		0.4	13.95	13.5	24.5	1.0
		0.6	13.58	14.1	39.0	149.0
	200	0.2	0.00	0.5	1.0	1.0
		0.4	19.68	127.5	1.0	1.0
		0.6	49.93	127.9	16.0	1.0

[a] Gap=(Feasible solution value-lower bound) * 100/lower bound.

Table 9. Computational results for cost 3

L_k	v	r	Gap[a] (%)	CPU (second)	Best Feas. at Iter.	Best Bound at Iter.
1	60	0.2	14.57	2.5	12.3	100.5
		0.4	7.18	2.8	41.8	195.5
		0.6	5.75	3.0	29.0	198.8
	80	0.2	14.98	6.5	13.5	100.0
		0.4	8.37	6.9	19.0	190.8
		0.6	5.74	7.2	25.5	190.0
	100	0.2	15.12	13.4	19.8	46.8
		0.4	9.14	14.5	29.3	194.8
		0.6	9.54	14.6	12.8	154.0
	200	0.2	24.85	127.9	1.0	1.0
		0.4	82.87	127.6	21.0	1.0
		0.6	67.60	133.2	24.8	59.3
2	60	0.2	0.49	0.7	1.0	48.8
		0.4	5.84	2.8	62.0	184.8
		0.6	4.81	2.9	22.0	198.3
	80	0.2	0.00	0.0	1.0	1.0
		0.4	23.26	6.7	24.8	193.8
		0.6	5.08	7.2	19.0	199.0
	100	0.2	0.00	0.0	1.0	1.0
		0.4	46.66	13.5	25.0	48.8
		0.6	5.65	14.7	30.5	197.5
	200	0.2	0.00	0.5	1.0	1.0
		0.4	58.19	127.6	1.0	1.0
		0.6	93.18	128.1	21.0	1.0
3	60	0.2	0.00	0.0	1.0	1.0
		0.4	14.88	2.6	22.3	143.5
		0.6	5.79	2.9	13.0	195.5
	80	0.2	0.00	0.0	1.0	1.0
		0.4	19.68	6.4	19.5	98.5
		0.6	5.03	7.2	20.0	190.5
	100	0.2	0.00	0.0	1.0	1.0
		0.4	20.06	13.4	28.0	1.0
		0.6	49.81	13.6	31.8	50.0
	200	0.2	0.00	0.4	1.0	1.0
		0.4	29.53	125.2	1.0	1.0
		0.6	73.58	128.3	21.0	1.0

[a] Gap=(Feasible solution value-lower bound) * 100/lower bound.

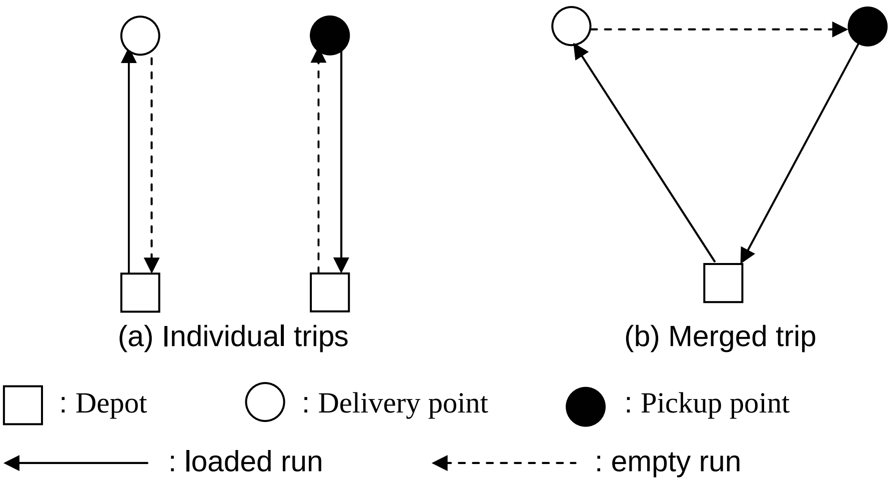

Fig. 1. Merging two individual trips of delivery and pickup.

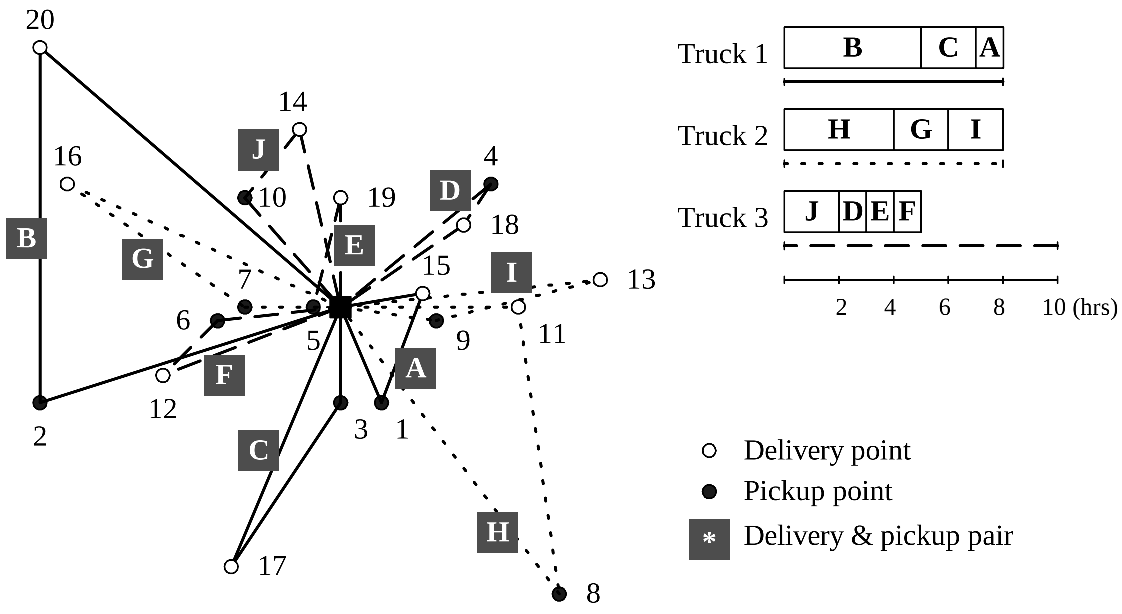

Fig. 2. The initial feasible solution.

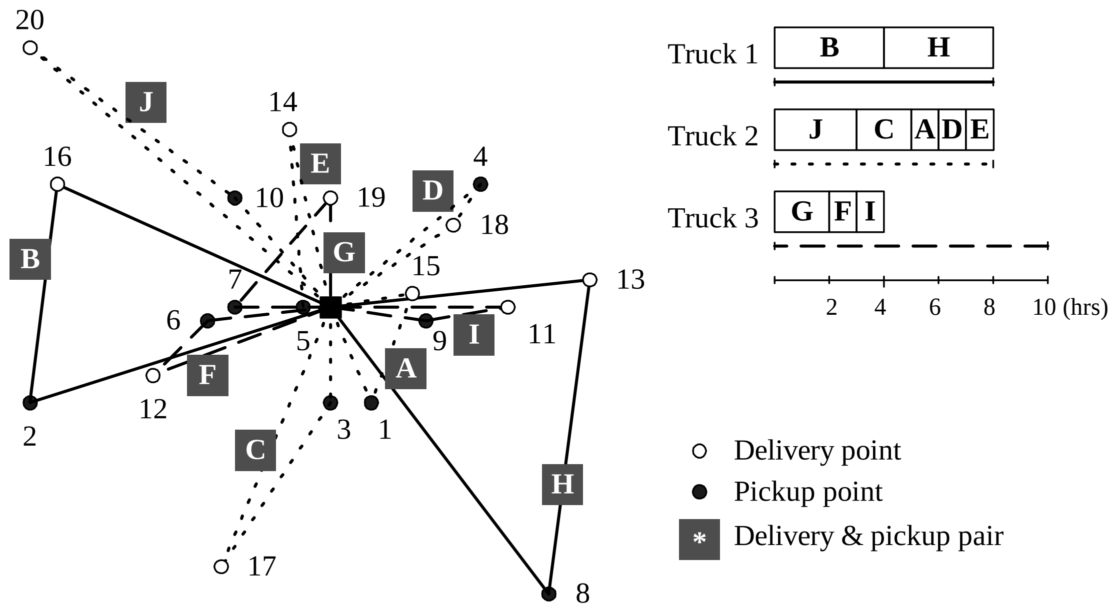

Fig. 3. The final feasible solution.