

# A Scalable Heuristic for Pickup-and-Delivery of Splittable Loads and Its Application to Military Cargo-Plane Routing

**Myoung-Ju Park**

Department of Industrial and Management Systems Engineering, Kyung Hee University

**Moon-Gul Lee\***

Logistics Staff Center, Republic of Korea Air Force HQ

(Received: March 26, 2012 / Accepted: April 30, 2012)

---

## ABSTRACT

This paper is motivated by a military cargo-plane routing problem which is a pickup-and-delivery problem in which load splits and node revisits are allowed (PDPLS). Although this recent evolution of a VRP-model enhances the efficiency of routing, a solution method is more of a challenge since the node revisits entail closed walks in modeling vehicle routes. For such a case, even a compact IP-formulation is not available and an effective method had been lacking until Nowak *et al.* (2008b) proposed a heuristic based on a tabu search. Their method provides very reasonable solutions as demonstrated by the experiments not only in their paper (Nowak *et al.*, 2008b) but also in ours. However, the computation time seems intensive especially for the class of problems with dynamic transportation requests, including the military cargo-plane routing problem. This paper proposes a more scalable algorithm hybridizing a tabu search for pricing subproblem paused as a single-vehicle routing problem, with a column generation approach based on Dantzig-Wolfe decomposition. As tested on a wide variety of instances, our algorithm produces, in average, a solution of an equivalent quality in 10~20% of the computation time of the previous method.

Keywords: Military Cargo Plane Routing, Pickup and Delivery, Split Loads, Decomposition, Column-Generation

\* Corresponding Author, E-mail: mlee@snu.ac.kr

---

## 1. INTRODUCTION

Currently, Republic of Korea Air Force (ROKAF) operates an air-logistics system of military cargo-planes on the network of major air-bases to meet weekly transportation demands. The flight schedule is, however, updated semi-annually based on the transport results of the previous half year. Due to the intractability of problem, the changes usually involve minimal adjustments in flight frequencies of existing lines. And the cost from the disparity between the weekly demands and the semi-annual schedule becomes significant. Ideally, the schedule is pre-

pared weekly with a minimal computation time to take into account the last-minute transportation requests.

The weekly transportation request between a pair of air-bases is a collection of numerous and heterogeneous demands and therefore can be split and serviced by more than one planes. Also, nothing prohibits a cargo-plane from revisiting the same air-base as long as the typical rules, such as the maximum number of take-offs, are observed. Therefore, the military cargo-plane routing problem is formulated best as the *pickup-and-delivery with load splits and node revisits*, or PDPLS in Nowak *et al.* (2008). This recent evolution of a pickup-and-delivery

model certainly enhances routing efficiency. However, a solution method is more of a challenge since the node revisits entail closed walks in modeling the routes of vehicles. For such cases, even a compact IP formulation is not available unless the number of revisits to a node by a plane is bounded by a small constant number, a priori. Nowak *et al.* (2008b) proposed a heuristic based on tabu search. Their method provides solutions of very reasonable quality, tested on the instances not only from (Nowak *et al.*, 2008a, 2008b) but also from this paper, as shown subsequently. However, the computation times seem intensive when the size of a problem gets larger. That is especially the case for the military cargo-plane problem in which we need to accommodate the weekly transportation requests rushing or modified in the last moments.

This paper pursues a scalability of method for solving PDPLS by combining two components, column generation and tabu search. PDPLS can be viewed as a collection of single-vehicle pickup-and-delivery problems conjoined to meet each of the transportation requests along the pairs of nodes. Relying on this separability, we can reformulate it, using the Dantzig-Wolfe decomposition, as a mixed integer program in which each route and the amounts of commodities transported along the route of a vehicle are represented by binary and continuous variables, respectively. For a fruitful utilization of this MIP-formulation, of course, we need an effective algorithm for solving the column generation subproblem which is a single-vehicle pickup-and-delivery problem. Although only a single vehicle is involved, closed walks are still inevitable, and even this subproblem has been lacking an effective solution method. To cope with this, we adopt a pragmatic approach of solving this pricing subproblem by a tabu search. Thus, in sum, we pursue scalability by applying tabu search to the decomposed subproblems but maintain solution quality by coordinating the solutions of subproblems via dual values of the LP-relaxation. For a minimal computation, we take a further pragmatic step: based on the solution of LP-relaxation, we fix a promising routing of a vehicle at each iteration.

This fix-and-regenerate algorithm was tested on a wide variety of instances adopted from (Nowak *et al.*, 2008b) as well as from the military cargo-plane routing problem. It turns out that the fix-and-regenerate algorithm produces, in average, a solution of an equivalent quality only in 10~20% of the computation time of the previous algorithm.

This paper is organized as follows. Section 2 reviews the previous VRP models that allow load splits. In Section 3.1, PDPLS is formally formulated as a mixed integer program. Section 3 is devoted to the discussion of our column-generation based heuristic and tabu search method for its pricing subproblem. The heuristic is then tested on the real data from the Korea military cargo-plane routing problem and a set of instances for a comparison with the previous heuristic in Section 4. Section 5 provides some concluding remarks and further studies.

## 2. RELATED WORKS

### 2.1 Split Delivery Vehicle Routing Problem

Previously, the load splits was considered in the delivery from a single depot. The *split delivery vehicle routing problem* or SDVRP, proposed by Dror and Trudeau (1989, 1990) was the first such a model, to the authors' best knowledge: a single delivery request can be split over more than one vehicle. They observed, under the assumption of the triangle inequality satisfied among the edge costs, that no more than a single requests can be split into the same pair of vehicles in an optimal solution of SDVRP. Also, such pairs, connected with edges, can not induce a circuit. Relying on these properties, they developed a local search heuristic which provides, from their experiment, a solution whose cost is 13% less than the one produced by the same heuristic but without a load split.

Frizzell and Giffin (1992) considered the SDVRP on a grid network. They allowed load splits to the request whose size falls in a predetermined interval. They developed a heuristic employing the notion of splitting cost accrued when a request is split into more than one vehicle.

Sierksma and Tijssen (1998) modeled helicopter routing problem for the crew exchange at a neutral gas platform as an SDVRP and formulated it as an IP. They solved its LP relaxation using column generation and applied a rounding method to get a feasible IP solution.

Belenguer *et al.* (2000) studied the polyhedron of SDVRP and identified some facets of it. Using corresponding valid inequalities, they developed a cutting-plane algorithm to generate the lower bound of the optimal value. Archetti *et al.* (2006) developed a tabu search method for SDVRP which extends the method of (Dror and Trudeau, 1989) with a *k*-split interchange. Archetti *et al.* (2008) studied the effects of the size and location of the transportation demand on the benefit of load splits using the tabu search methods from (Archetti *et al.*, 2006) and (Toth and Vigo, 2003), respectively, for SDVRP and VRP. Archetti, Savlesbergh, and Speranza (2008) developed a heuristic combining a tabu search of (Archetti *et al.*, 2006) and IP optimization. Using a set of promising routes derived from a tabu search of (Archetti *et al.*, 2006), they formulated the SDVRP as a route-based IP and solved it with a commercial code.

Chen *et al.* (2007) considered an improvement step by reassigning the transportation demands serviced by vehicles in an initial unsplit solution, which is formulated as an MIP-formulation. In their approach, an initial unsplit solution is obtained by the Clarke-Wright method (Clarke and Wright, 1964). Then, they applied commercial IP-code for a predetermined time and applied a heuristic, called *record-and-record* developed by Li *et al.* (2005). Jin *et al.* (2008) developed a heuristic based on column generation for large SDVRP's. After a column generation they fix a promising route using a priority rule instead of a branching. This *fix-and-regenerate* idea, re-

current in literature to find a solution with a minimal computation, will be also used in our heuristic for PDPLS.

Some exact algorithms also have been proposed for SDVRP. Dror *et al.* (1994) formulated the SDVRP as an MIP along with some valid inequalities. They developed a brand-and-bound algorithm in which upper bounds are provided by their previous heuristic (Dror and Trudeau, 1989) and lower bounds by the MIP formulation with some constraints relaxed. In their experiment, however, this algorithm is implemented only at the root node of branching tree to demonstrate the efficiency of the heuristic (Dror and Trudeau, 1989).

Lee *et al.* (2006) formulated SDVRP as a dynamic program with finite state and action spaces. They also represented the dynamic program as network whose node and edge sets, respectively, correspond to the state and action spaces. They developed an exact algorithm based on the observation such that solving the dynamic program is equivalent to find a shortest path in the network.

Jin *et al.* (2007) developed a two-stage algorithm. The first-stage creates the clusters each of which is the set of demand points that a vehicle visits. For each cluster from the first-stage, the second-stage finds a minimum distance route whose distance is, in turn, used to add a valid inequality in the first-stage. To improve the efficiency, they hired additional valid inequalities to the first-stage. They proved that the algorithm finds an optimal solution in finite steps. Experiments were performed in comparison with a commercial solver and the dynamic program based algorithm (Lee *et al.*, 2006).

Some studies extended SDVRP to the SDVRP with *time windows*, or SDVRPTW. Frizzell and Giffin (1995) considered the SDVRPTW on a grid network. They developed a heuristic using a look-ahead approach improved by a local search hiring node insertions and exchanges. Mullaseril *et al.* (1997) applied the heuristic of (Dror and Trudeau, 1989) to a livestock feed distribution problem formulated as an SDVRPTW. Ho *et al.* (2004) and Bolduc *et al.* (2010) developed a tabu search for SDVRPTW. Gendreau *et al.* (2003) developed an exact algorithm for SDVRPTW based on column generation. Belfiore and Yoshizaki (2009) proposed a scatter search algorithm, an evolutionary methods, to solve a real application of SDVRPTW.

## 2.2 Pickup and Delivery Problem with Load Splits and Node Revisits, PDPLS

This recent evolution of a VRP-model was first considered by Nowak *et al.* (2008b). They proved that the benefit of load splits is maximized when the load of the each request is little bigger than the half of the vehicle capacity  $C$ , namely  $C/2+\varepsilon$  for small  $\varepsilon > 0$ . They developed a tabu search heuristic based on the Clarke-Wright method and additional local search rules. They also analyzed the benefits of load splits in terms of various factors such as load size, number of requests, number of pickup nodes relative to the number of delivery nodes, and others

(Nowak *et al.*, 2008a).

Moreno *et al.* (2006) considered a helicopter routing problem for commuting the workers over offshore drilling wells. Similar with our heuristic, their method is based on a column generation method in which a tabu search is used to generate columns. As developed for a problem with the additional complications, such as time windows, multiple vehicle type, and lunch break hours for pilots, their heuristic does not appear elaborated well enough for the usual PDPLS. To maintain computation minimal, they restricted the length of a route to a small number without considering node revisits. As each column corresponds to the transportation request of a single worker, the decomposition does not seem suitable for the PDPLS with relatively large transportation demands. A feasible integral solution is obtained by applying a commercial code, for a predetermined time, to the IP formulation restricted to the generated columns.

In maritime routing literature, Andersson *et al.* (2011) presented two path flow formulation to solve the maritime pickup and delivery problem *with time windows and split loads problem* or PDPTWSL. They considered that all feasible routes or columns are generated a priori and only time window constraints are considered to generate the feasible routes. Korsvik *et al.* (2011) also proposed a large neighborhood search heuristic which can be solved PDPTWSL for ship routing and schedule. Their methods perform simple node-insertion in current route, a descent local search using split and merge operator until a local optimum is reached, and then destroy and repair algorithm finally. They shown that introducing split loads can be of great benefit as the shipping company can better utilize the ship capacity and additional shipping.

## 2.3 Column Generation Via Heuristics

The column generation method has been widely used in the pickup and delivery problem (Dell'Amico *et al.*, 2006; Desrochers *et al.*, 1992; Jin *et al.*, 2008; Moreno *et al.*, 2006; Savelsbergh and Sol, 1998; Xu *et al.*, 2003). Its performance largely depends on the efficiency of the method for a pricing subproblem especially when the subproblem is NP-hard. In Section 3 for a detailed discussion of our algorithm, we also offer a comparison of such methods in which the pricing subproblem is solved by a tabu search.

# 3. THE ALGORITHM

## 3.1 Mathematical Formulation

Consider a complete digraph  $G = (V, E)$  on the node set  $V = \{0, \dots, n\}$  where 0 is reserved to denote the single depot. Each edge  $ij$  is assigned a positive distance  $c_{ij} > 0$  between node  $i$  and  $j$ . Also  $H$  is the set of transportation requests, each of which is represented by a triple  $(o_h, d_h,$

$l_h$ ), where  $o_h$ ,  $d_h$ , and  $l_h$ , respectively, are the pickup node, delivery node, and load of the request  $h$ . There are  $K$  identical vehicles with the same capacity  $Q$ . Then the PDPLS is to find the route of each of the vehicles, a closed walk, along with the loads picked up and delivered at the nodes of the route that satisfy the followings:

1. all transportation requests are met,
2. each vehicle, starting and ending at node  $0$ , should not travel more than a distance  $L$ , and
3. the load of a vehicle should not exceed  $Q$  at any time.

The objective is to minimize the total distance travelled by vehicles. Let  $W$  be the set of closed walks starting and ending at the depot node  $0$ . For each walk  $w \in W$ , let us denote by  $c_w$  and  $E_w$ , respectively, the travelling distance and the set of edges of  $w$ . For each request  $h \in H$ , let  $I_{wh}$  be the binary variable indicating if  $w$  contains at least one  $o_h$ - $d_h$  subwalk (hence  $I_{wh} = 1$ , if so, and  $0$ , otherwise). Notice that if a vehicle picks up a positive load  $y_h$  from  $o_h$  then it would be suboptimal to visit  $o_h$  again before delivering it to  $d_h$  because we can pick up the load  $y_h$  at the second visit of  $o_h$  without increasing the distance. Similarly, an optimal solution would not delay a deliver at  $d_h$  to the next visit.

#### Mixed Integer Programming for PDPLS

Min	$\sum_{k=1}^K \sum_{w \in W} c_w x_w^k$	(1)
subject to	$\sum_{k=1}^K \sum_{w \in W} y_{wh}^k = l_h, \quad \forall h \in H$	(2)
	$c_w x_w^k \leq L,$	(3)
	$k = 1, \dots, K, \quad \forall w \in W$	
	$\sum_{h, i: e \in E_{wh}} y_{whi}^k \leq Q,$	(4)
	$k = 1, \dots, K, \quad \forall w \in W, \quad \forall e \in E_w$	
	$y_{whi}^k \leq I_{wh} l_h,$	(5)
	$k = 1, \dots, K, \quad \forall w \in W, \quad \forall h \in H, \quad i = 1, \dots, h_w$	
	$\sum_{i=1}^{h_w} y_{whi}^k = y_{wh}^k,$	(6)
	$k = 1, \dots, K, \quad \forall h \in H$	
	$x_w^k \in \{0, 1\},$	(7)
	$k = 1, \dots, K, \quad \forall w \in W$	
	$y_{whi}^k \geq 0,$	(8)
	$k = 1, \dots, K, \quad \forall w \in W, \quad \forall h \in H, \quad i = 1, \dots, h_w.$	

Therefore, in  $w$ , the  $o_h$ - $d_h$  subwalks to carry load for request  $h$  are not overlapped and hence can be numbered uniquely in their order of appearance. We will denote, by  $h_w$ , the number of such subwalks in  $w$ . Also, let  $E_{whi}$  be the set of edges of the  $i$  th  $o_h$ - $d_h$  subwalk in  $w$ . For each given triple of request  $h$ , walk  $w$ , and vehicle  $k$ , we have two groups of decision variables:

$$x_w^k = \begin{cases} 1, & \text{if vehicle } k \text{ chooses } w \text{ as route,} \\ 0, & \text{otherwise.} \end{cases}$$

Denote by  $y_{whi}^k$  the demand of request  $h$  carried by vehicle  $k$  via the  $i$  th  $o_h$ - $d_h$  subwalk in  $w$ . Define  $y_{wh}^k = \sum_{i=1}^{h_w} y_{whi}^k$ . Then PDPLS can be formulated as above MIP.

The objective (1) is to minimize the total distance of routes taken by vehicles. Constraints (2), (3), and (4), respectively, correspond to the condition 1, 2, and 3. Constraint (5) enforces that a vehicle services the request  $h$  only if its route has an  $o_h$ - $d_h$  subwalk.

Notice that without (2) the above MIP formulation is decomposed into  $k$  independent single vehicle subproblems. Such separability is usually exploited by a decomposition technique such as the Lagrangian relaxation method and the column generation method.

But, the former does not seem suitable for this particular problem. The reason is that the dual problem is decomposed into the subproblems which are identical for every vehicle. Thus in the dual optimal solution of the Lagrangian relaxation, every vehicle uses exactly the same route. Thus, it is not likely to provide a tight lower bound let alone a feasible solution.

On the other hand, in the latter, each vehicle, endowed with its own dual values, has a different set of preferable routes in an optimal solution of LP-master in every pricing step. Furthermore, dealing directly with the vehicle routes as the columns of LP-master enables us to adopt various heuristic ideas to expedite the solution procedure as discussed subsequently.

### 3.2 Dantzig-Wolfe Decomposition

To implement the column generation based approach, we reformulate the MIP-model (1~8) using the Dantzig-Wolfe decomposition considering (2) as coupling constraints. Notice that the subproblem defined by (3~8) has a feasible solution set that is identical for every vehicle  $k$  and denoted by  $\Omega = \{\omega_1, \dots, \omega_T\}$ . Notice that each  $\omega_t$  consists of a route and the loads carried along the route. Then, expressing each feasible solution of vehicle  $k$  as  $\omega^k = \sum_{t=1}^T \omega_t \lambda_t^k$  using binary variables  $\lambda_t^k \in \{0, 1\}$  with  $\sum_{t=1}^T \lambda_t^k = 1$ , the MIP-model (1~8) can be restated as the following binary IP, called IP-master, or simply IPM.

$$\min \sum_{k=1}^K \sum_{t=1}^T c_{\omega_t} \lambda_t^k \quad (9)$$

$$\text{s.t. } \sum_{k=1}^K \sum_{t=1}^T y_{\omega_t h} \lambda_t^k = l_h, \quad \forall h \in H \quad (10)$$

$$\sum_t \lambda_t^k = 1, \quad k = 1, \dots, K, \quad t = 1, \dots, T \quad (11)$$

$$\lambda_t^k \in \{0, 1\}, \quad k = 1, \dots, K, \quad t = 1, \dots, T \quad (12)$$

Here,  $c_{\omega_t}$  and  $y_{\omega_t h}$ , respectively, are the traveling distance of route and the load of request  $h$  carried along the  $t$  th route of  $\omega_t$ .

Now we consider the LP-relaxation obtained from the IPM, (9-12), by replacing binary restrictions  $\lambda_t^k \in \{0, 1\}$  with  $0 \leq \lambda_t^k$ . The column generation algorithm applied to this primal LP-relaxation, called LP-master, or LPM consists of the following steps.



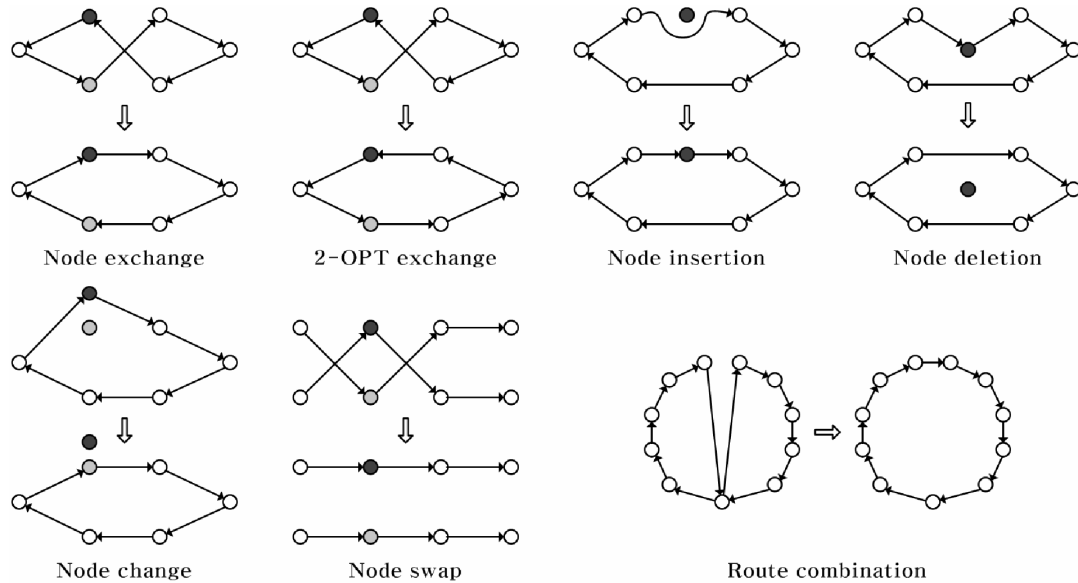


Figure 1. The Neighbourhood Search Operations

summarizes such neighborhood operations adopted by some previous studies employing a tabu search. Figure 1 illustrates these operations. Among them, we employ node exchange, 2-OPT exchange, node insertion, node deletion, and node change as they seem suitable for a single vehicle routing.

It is easy to check that we can exhaust a neighborhood search for a current solution  $w$  in  $(|V_w|^2 + 2|V||V_w|)$  steps where  $|V|$  and  $|V_w|$ , respectively, are the number of nodes of  $G$  and  $w$ .

A complete tabu search procedure is as followings. For the initial solution of the  $i$  th repetition, we use the route generated for the  $i$  th column of the RLPM. We use a tabu list constructed in the before search as in Caricato *et al.* (2003) while we reduce its size to  $\kappa_3 = 10$  instead of  $\kappa_1 = 30$  after the first search.

Note that a tabu search's bottleneck is where all routes included in a neighboring value are accurately estimated. To reduce computation time, we only need to evaluate the solution output of a tabu search and the rest requires approximate evaluation by using Fix-and-regenerate algorithm shown in next chapter. Also, a local search precedes a tabu search for the same purpose after conducting pricing by a tabu search.

When a solution found by a local search shows that all vehicles get non-negative reduced costs, executed a tabu search. We intend to find column with negative reduced costs easily using generated  $\omega$  columns. In other words,  $\omega$ -added RLPM's value of dual solution  $(\pi, \mu)$  and  $\omega$  are used to execute a local search to compute new  $\omega'$ . And, several columns are generated after checking whether  $\omega'$  a negative reduced cost has. That is, enter columns which can be computed through a local search, employing new RLPM dual solution when they have negative reduced cost for certain vehicles.

### 3.4 Fix-and-Regenerate Algorithm

Our algorithm derives a feasible solution of IPM, (9-12) by rounding the fractional solution  $\lambda^*$  of LPM computed by the column generation algorithm using the tabu search pricing heuristic. Among the routes corresponding to the positive values in  $\lambda^*$ , we fix a feasible route of a vehicle that is most efficient according to the following measure:

$$\text{efficiency}_\omega = \frac{\hat{\lambda}_\omega \times |H_\omega| \times \sum_{h \in H} y_{\omega h}}{c_\omega \times (|E_\omega| - 2)}, \quad (14)$$

where  $H_\omega = \{h | y_{\omega h} > 0\}$  and (recall that)  $E_\omega$  is the set of edges of  $\omega$ . This measure considers a route better if it has a larger  $\lambda$ -value, serves more requests, carries larger loads with a smaller cost and less edges.

After fixing a route with the maximum efficiency, we regenerate the problem, that is, we remove the fixed vehicle and the loads carried by the vehicle from the problem. The procedure is repeated to the reduced problem while there is an uncovered request. As a post-process, we improve our routes by the operation of route combination under satisfying maximum distance constraint. By this step, it can be reduced the number of vehicles. Algorithm 1 is the summary of the above procedure.

**Remark 1:** To further enhance quality and speed of solution, include route generated in a previous phase when creating initial RLPM while fixing to a single vehicle. For feasibility, as to route created in previous phase, eliminate the amount of transportation loads as large as the amount the fixed vehicle transports.

Algorithm 1. Fix-and-Regenerate Algorithm

- 1: **While** there is an uncovered request
- 2: **Step 1** Solve the primal relaxation from Dantzig-Wolfe decomposition using column generation with the pricing heuristic to obtain a solution  $\lambda = \hat{\lambda}$ .
- 3: **Step 2** Among the routes  $\omega$  with  $\hat{\lambda}_\omega > 0$ , choose  $\omega^*$  with the maximum efficiency (14).
- 4: **Step 3** Fix  $\lambda_{\omega^*} = 1$  and regenerate the problem.
- 5: **End-While**
- 6: **Step 4** Improve the solution by the operation of route combination.

4. EXPERIMENTAL RESULTS

The fix-and-regenerate algorithm (FRA) is coded in C++ using the ILOG Concert 2.4 and the callable library of CPLEX 10.2. For computation, a desktop with specifications of 2.4GHz Intel core quad CPU processor and 2Gb memory was used. On two instances, FRA was tested and compared to the benchmark algorithm of Nowak *et al.* (2008b), which will be hereafter referred to as NA: (i) real instances from the Korean military cargo-plane routing and (ii) random instances generated for checking FRA performance (Nowak *et al.*, 2008b).

4.1 Real Instances

As illustrated in Figure 2, the Korean military cargo-plane network is a complete digraph on 15 nodes of air bases. Node 1 functions both as a pickup-and-delivery post and depot where all cargo-planes are stored and maintained. When applying real instances to the FRA, node 1 is duplicated and set as Node 0 for the purpose of convenient computation and easier comparison to existing routing problems whose nodes' numbers start from zero.

The employed data comes from average weekly demands among pairs of several bases in the semi annual of 2006 and 2007 respectively. The loads for each request are scaled to allow each cargo plane to have a unit capacity (Q) of 1. One request is also scaled to be split as small

as 0.01. As a constraint factor, the number of hops are restricted to the maximum of six times for in reality, Korean cargo planes' takeoffs and landings are restricted to six times per day before coming back to the depot.

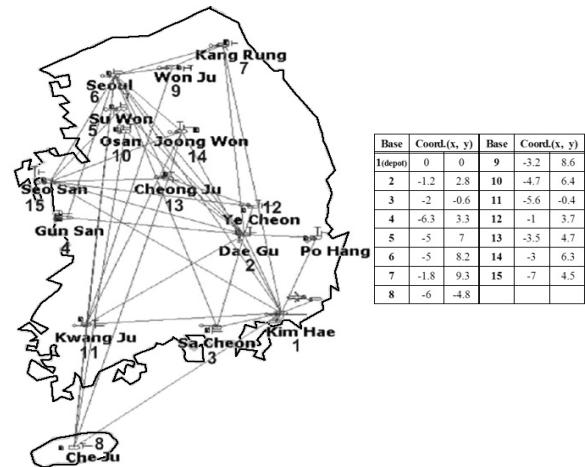


Figure 2. The Korean Military Cargo-Plane Network and Geo-Coordination

Table 2 shows the results of our experiment. FRA, compared to the semiannual schedule, reduced the total traveling distance by 48.6~55.1% and number of cargo-planes by 47.2~52.6%. Takeoffs were decreased by 40~43.3%. The costs of fuels were estimated to be saved by at least about \$2 million per year. A weekly-updated flight schedule of FRA as shown enables more flexible and efficient operation of cargo-planes than NA.

In our comparison, another constraint was the maximum value for distance (L), set at 35. In this case, our experiment also shows that FRA outperforms NA. As will be discussed later, FRA is more advantageous when more numbers of nodes receive pickup and delivery demands. Of particular note is significant reduction in computation time.

**Remark 2:** Above result demonstrates that FRA performs far better than that of experiment random instances. This is because in military cargo-plane instances, pickup and

Table 2. Experimental Results of Korean Military Cargo-Plane Routing Problem

constraint	type of algorithm	2006(74.5 requests)				2007(77.7 requests)			
		total dist.	CPU time(sec)	# of routes	total takeoffs	total dist.	CPU time(sec)	# of routes	total takeoffs
# of hops ≤ 6	Manual	507.8	-	19	92	451.6	-	18	91
	FRA	228.0	33.4	9	52.2	232.1	36.8	9	54.7
	Reduction of FRA(%)	55.1	-	52.6	43.3	48.6	-	47.2	40.0
dist.(L) ≤ 35	NA	232.2	1585.5	9	71.8	229.6	1783.5	8.3	73.2
	FRA	182.1	110.7	6	54.1	173.1	153	5.8	53
	Reduction of FRA(%)	21.6	93.0	33.3	24.6	24.6	91.4	30.0	27.6

delivery requests exist at most nodes but not at Nowak's instances and random instances' depot has none. To confirm this outcome, we tested the same network with two different settings: one with randomly-generated requests and load sizes, and another only with load size generated in the similar fashion. The former shows the qualities of both FRA and NA almost same while the latter FRA was about six percent superior to NA.

## 4.2 Random Instances

### 4.2.1 Instance Design

Again, the data are scaled to make the vehicle capacity equal to 1. The depot is located at the origin and other nodes are generated to be uniformly distributed over  $[-40, 40]^2$  in the two dimensional plane. We set the maximum distance L at 500. This Table 3 shows other parameters of an instance. Node is called pure if it is exclusively for pickup or delivery location, and otherwise combined. The first column indicates different types of an instance. An instance is defined as pure if its every single node is pure;

combined if every node is combined; and as mixed when both pure and combined nodes co-exist. Each of three types creates its own pickup-and-delivery model. For instance, mixed-type models include the delivery system consisting of factories, major warehouses, and retail stores. Nowak *et al.* (2008b) only considered pure types, which is incapable of coping with combined types in the Korean military cargo-plane routing problem. The first column also contains a vector  $(p, d, c, r)$ , whose elements are, in their order, the number of pickup nodes, delivery nodes, combined nodes, and the maximum possible number

Table 3. Instance Parameters

Instance type ( $p, d, c, r$ )	Load range	Request density
Pure (5, 25, 0, 125)	[0.1~0.5]	30%
Combined (0, 0, 15, 210)	[0.3~0.7]	50%
Mixed (5, 10, 5, 215)	[0.5~1.0]	70%
	[0.1~1.0]	100%

Table 4. Performance of FRA and NA over random instances

Pure												
Instance	Request density(from maximum possible 125 requests)											
Load range	30%			50%			70%			100%		
	Obj Gap	CPU time(sec)		Obj Gap	CPU time(sec)		Obj Gap	CPU time(sec)		Obj Gap	CPU time(sec)	
		FRA	Ratio*		FRA	Ratio		FRA	Ratio		FRA	Ratio
[0.1~0.5]	-11.4	46.0	23.2	-11.0	142.9	20.0	-3.6	284.2	17.3	-4.6	592.5	15.4
[0.3~0.7]	-8.8	58.6	18.6	-5.1	135.8	11.9	-3.9	271.4	10.2	-1.3	596.3	9.1
[0.5~1.0]	-7.2	46.8	12.4	-2.5	116.3	7.7	0.0	223.2	5.3	1.8	515.6	4.5
[0.1~1.0]	-9.8	46.4	20.1	-5.3	122.3	14.0	-3.9	235.2	13.1	-2.5	514.9	12.9
Avg.	-9.3	49.5	18.6	-6.0	129.3	13.4	-2.8	253.5	11.4	-1.7	554.8	10.5
Combined												
Instance	Request density(from maximum possible 210 requests)											
Load range	30%			50%			70%			100%		
	Obj Gap	CPU time(sec)		Obj Gap	CPU time(sec)		Obj Gap	CPU time(sec)		Obj Gap	CPU time(sec)	
		FRA	Ratio		FRA	Ratio		FRA	Ratio		FRA	Ratio
[0.1~0.5]	2.8	110.4	14.3	6.3	546.8	26.3	8.4	837.4	17.3	10.7	2035.4	22.8
[0.3~0.7]	-1.0	113.6	11.4	4.2	375.4	12.2	5.0	726.4	13.6	9.7	1588.3	12.1
[0.5~1.0]	0.7	116.9	14.2	4.5	321.8	11.1	8.1	603.2	8.7	9.8	1856.1	15.2
[0.1~1.0]	-1.1	114.0	16.7	2.7	287.8	14.9	4.6	636.3	16.3	5.6	1538.1	20.1
Avg.	0.3	113.7	14.1	4.4	383.0	16.1	6.5	700.8	14.0	8.9	1754.5	17.6
Mixed												
Instance	Request density(from maximum possible 215 requests)											
Load range	30%			50%			70%			100%		
	Obj Gap	CPU time(sec)		Obj Gap	CPU time(sec)		Obj Gap	CPU time(sec)		Obj Gap	CPU time(sec)	
		FRA	Ratio		FRA	Ratio		FRA	Ratio		FRA	Ratio
[0.1~0.5]	-3.0	134.2	22.9	2.2	438.3	19.6	7.2	919.7	19.3	7.7	2060.4	23.7
[0.3~0.7]	-3.0	157.7	16.9	2.7	470.2	14.3	5.4	1003.3	17.4	6.5	2336.1	13.8
[0.5~1.0]	0.5	160.1	12.7	4.8	473.9	9.8	5.3	756.7	9.2	8.8	2244.8	9.2
[0.1~1.0]	-4.4	154.3	20.6	1.1	403.2	18.8	2.5	779.4	14.5	5.0	2058.3	19.1
Avg.	-2.5	151.6	18.3	2.7	446.4	15.6	5.1	864.8	15.1	7.0	2174.9	16.4

Note) \* Ratio is set to  $100 \times t_{FRA}/t_{NA}$ , where  $t_{FRA}$  and  $t_{NA}$ , respectively, are the computation times of FRA and NA.



of origin-destination pairs of nodes. The second column represents the range in which the loads of requests are uniformly distributed. The third column represents the request density, i.e., the ratio of the number of actual requests to  $r$ . For testing, each of the 48 combinations of parameters from the three columns was used to create 10 random instances.

4.2.2 Analysis of Experimental Results

Table 4 shows the performance of FRA in comparison with NA over the random instances. Let  $z_{FRA}$  and  $z_{NA}$ , respectively, be the objective value of FRA and NA. By  $Obj\_Gap$  we mean the percentage gap of  $z_{FRA}$  to  $z_{NA}$ :

$$Obj\_Gap = 100 \times \frac{z_{NA} - z_{FRA}}{z_{NA}}$$

Each algorithm produces the solutions of equivalent qualities. More specifically, for the pure type instances, NA outperforms FRA by up to 9.3% at low density (30%). On the other hand, for the combined type instances, FRA was better than NA by 8.9% at high density (100%). For the mixed type instance, NA performs better in the instances of lower request density while FRA is stronger in the ones of higher request density.

FRA shows, however, a significant reduction in computation time. The computation time of FRA is, in average, about 15.1% of the one required by NA. The more significantly time gets reduced, the larger instance size becomes, as shown in Figure 4. Figure 3 plots the objective values and computation times of 10 random instances of pure type with request density of 100%. For all load ranges, FRA shows a smaller variance in compu-

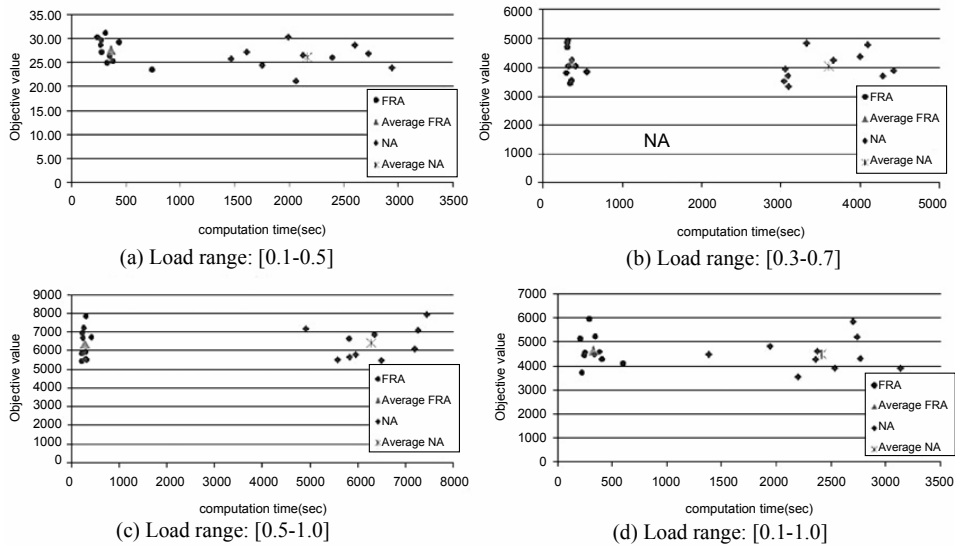


Figure 3. Objective Values and Computation Times of Pure Type Instances with 100% Request Density

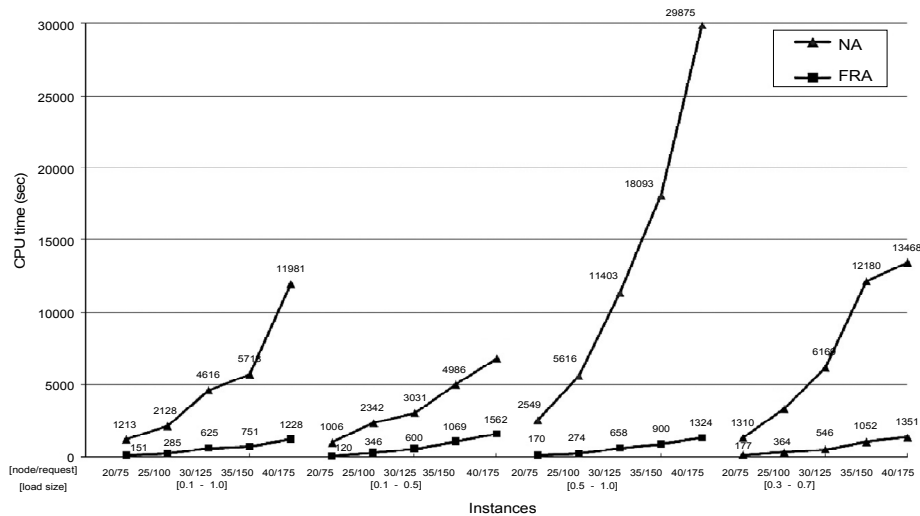


Figure 4. Computation Times of Two Algorithms Depending on the Size of the Instance

tation time; the variation of NA is about 63 times of that of FRA. To show the effect of the size of instance on the computation time, we construct the networks with 5 pickup nodes and 15, 20, 25, 30, and 35 delivery nodes, respectively.

For each network, we generate the requests on every origin-destination pair of nodes. Figure 4 shows the results on these instances that the computation time of NA gets more intensive than that of FRA as the size of instance gets larger.

## 5. CONCLUSION AND FURTHER RESEARCH

In this paper, we have developed a more scalable algorithm for PDPLS. A Korean military cargo planes routing problem has thus far been formulated as an MIP with constraints of revisit, splittable loads and closed-walk routing solution. Then, a fast and effective column-generation method-based heuristic algorithm is proposed which can be applied to PDPLS problem in various conditions. As for the developed algorithm, its quality as well as performance is analyzed through various random instances. It is confirmed that our proposal can significantly reduce transportation routes and their accompanied costs. It would be interesting to study how far it can be maintained when multiple types of vehicle and/or multiple depots are used.

On the other hand, researches on PDPLS have rarely proposed any of methods for solving the issue thus far. Though Nowak *et al.* (2008b) firstly introduced the topic and its methods, they had apparent weaknesses. For instance, when pickup and delivery demands are being concentrated at one point or vehicles with limited hops must visit nodes multiple times, the method cannot guarantee good qualities and result in undesirable outcomes.

Further studies, though, enabled the aforementioned column-generation method to be widely used for solving VRP as well as PDP. The bottom line is how to solve a pricing problem fast and effectively. So, it is necessary for more studies to be conducted especially because a pricing problem is NP-hard. And further researches are required for various PDPLS with additional constraints. Among them include the supplement of existing heuristics, required for solving pricing-subproblem, constraint condition of heterogeneous vehicles as well as multiple depots existing at several locations, and long-distance pickups and delivery service in which a vehicle's starting and destination points differ.

## ACKNOWLEDGMENT

The authors are grateful to Nowak, Ergun, and White for providing us the code of their algorithm (Nowak, 2008b).

## REFERENCES

- Aldaihani, M. and M. M. Dessouky, "Hybrid scheduling methods for paratransit operations," *Computers and Industrial Engineering* 45 (2003), 75-96.
- Andersson, H., M. Christiansen, and K. Fagerholt, "The maritime pickup and delivery problem with time windows and split loads," *INFOR* 49 (2011), 79-91.
- Archetti, C., A. Hertz, and M. G. Speranza, "A tabu search algorithm for the split delivery vehicle routing problems," *Transportation Science* 40, 11 (2006), 64-73.
- Archetti, C., M. W. P. Savelsbergh, and M. G. Speranza, "An optimization-based heuristic for the split delivery vehicle routing problem," *Transportation Science* 42, 1 (2008), 22-31.
- Archetti, C., M. W. P. Savelsbergh, and M. G. Speranza, "To split or not to split: That is the question," *Transportation Research Part E* 44 (2008), 114-123.
- Belenguer, J., M. Martinez, and E. Mota, "A lower bound for the split delivery vehicle routing problem," *Operations Research* 48, 5 (2000), 801-810.
- Belfiofre, P. and H. T. Y. Yoshizaki, "Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil," *European Journal of Ops. Res.* (2009), 750-758.
- Bolduc, M.-C., G. Laporte, J. Renaud, and F. F. Bocktor, "A tabu search heuristic for the split delivery vehicle routing problem with production and demand calendars," *European Journal of Ops. Res.* 202 (2010), 122-130.
- Caricato, P., G. Ghiani, A. Grieco, and E. Guerriero, "Parallel tabu search for a pickup and delivery problem under track contention," *Parallel Computing* 29 (2003), 631-639.
- Chen, S., B. Golden, and E. Wasil, "The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results," *Networks* 49, 4 (2007), 318-329.
- Clarke, C. and J. Q. Wright, "Scheduling of vehicle from a central depot to a number of delivery points," *Operations Res.* 12 (1964), 568-581.
- Cordeau, J. and G. Laporte, "A tabu search heuristic for the static multi-vehicle dial-a-ride problem," *Transportation Res. B* 37 (2003), 579-594.
- Dell'Amico, M. D., G. Righini, and M. Salani, "A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection," *Transportation Science* 40, 2 (2006), 235-247.
- Desaulniers, G., F. Lessard, and A. Hadjar, "Tabu Search, Partial Elementarity, and Generalized k-Path Inequalities for the Vehicle Routing Problem with Time Windows," *Transportation Science* 42, 3 (2008), 387-404.
- Desrochers, M., G. Desaulniers, and M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," *Operations Research*

- 40 (1992), 342-354.
- Dror, M., G. Laporte, and P. Trudeau, "Vehicle routing with split deliveries," *Discrete Appl. Math.* 50 (1994), 239-254.
- Dror, M. and P. Trudeau, "Saving by split delivery routing," *Transportation Science* 23 (1989), 141-145.
- Dror, M. and P. Trudeau, "Split delivery routing," *Naval Research Logistics* 37, 3 (1990), 383-402.
- Frizzell, P. W. and J. W. Giffin, "The bounded split delivery vehicle routing problem with grid networks distances," *Asia Pacific Journal of Operational Research* 9 (1992), 101-116.
- Frizzell, P. W. and J. W. Giffin, "The split delivery vehicle scheduling problem with time windows and grid network distances," *Computers Ops. Res.* 22, 6 (1995), 655-667.
- Gendreau, M., P. Dejax, D. Feillet, and C. Gueguen, "Vehicle routing with time windows and split deliveries," *working paper*, 2003.
- Ho, S. C. and D. Haugland, "A tabu search heuristic for the vehicle routing problem with time windows and split deliveries," *Computers and Operations Research* 31 (2004), 1947-1964.
- Indra-Payoong, N., A. Sumalee, K. Vanitchakornpong, and W.Y. Szeto, "A hybrid column generation and local search algorithm for pickup and delivery problem," *Transportation Research Boards 88<sup>th</sup> Annual Meeting*, 2009.
- Jin, M., K. Liu, and R. B. Bowden, "A two-state algorithm with valid inequalities for the split delivery vehicle routing problem," *Int. J. Production Economics* 105 (2007), 228-242.
- Jin, M., K. Liu, and B. Eksioglu, "A column generation approach for the split delivery vehicle routing problem," *OR Letters* 36 (2008), 265-270.
- Korsvik, J. E., K. Fagerholt, and G. Laporte, "A large neighbourhood search heuristic for ship routing and scheduling with split loads," *Computers Ops. Res.* 38 (2011), 474-483.
- Landrieu, A., Y. Mati, and Z. Binder, "A tabu search heuristic for the single vehicle pickup and delivery problem with time window," *Journal of Intelligent Manufacturing* 12 (2001), 497-508.
- Lee, C., M. A. Epelman, C. C. White, and Y. A. Bozer, "A shortest path approach to the multiple-vehicle routing problem with split pick-ups," *Transportation Research Part B* 40, 4 (2006), 265-284.
- Li, F., B. Golden, and E. Wasil, "Very large-scale vehicle routing: New test problems, algorithms, and results," *Computers Ops. Res.* 32 (2005), 1197-1212.
- Moreno, L., M. Poggi de Aragao, and E. Uchoa, "Column generation based heuristic for a helicopter routing problem," *LNCS 4007* (2006), 219-230.
- Mullaseril, P., M. Dror, and J. Leung, "Split delivery routing heuristics in livestock feed distribution," *J. Oper. Res. Soc.* 48, 2 (1997), 107-116.
- Nanry, W. P. and J. W. Barnes, "Solving the pickup and delivery problem with time windows using reactive tabu search," *Transportation Research Part B* 34 (2000), 107-121.
- Nowak, M., O. Ergun, and C. White, "An empirical study on the benefit of split loads with the pickup and delivery problem," *European Journal of Operational Research*, 2008.
- Nowak, M., O. Ergun, and C. White, "Pickup and delivery with split loads," *Transportation science* 42, 1 (2008), 32-43.
- Savelsbergh, M. W. P. and M. Sol, "Drive: Dynamic routing of independent vehicles," *Oper. Res.* 46 (1998), 474-490.
- Sierksma, G. and G. Tijssen, "Routing helicopters for crew exchanges on off-shore locations," *Annals of Oper. Res.* 76 (1998), 261-286.
- Subramanian, A., L. M. A. Drummond, C. Bentes, L. S. Ochi, and R. Farias, "A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery," *Computers Ops. Res.* 37 (2010), 1899-1911.
- Toth, P. and D. Vigo, "Heuristic algorithm for the handicapped person transportation problem," *Transportation Science* 31, 1 (1997), 60-71.
- Toth, P. and D. Vigo, "The granular tabu search and its application to the vehicle-routing problem," *INFORMS Journal on Computing* 15 (2003), 333-346.
- Verweij, B. and K. Aardal, "The merchant subtour problem," *Mathematical Programming Ser. B*, 94 (2003), 295-322.
- Xu, H., Z. Chen, S. Rajagopal, and S. Arunapuram, "Solving a practical pickup and delivery problem," *Transportation Science* 37, 3 (2003), 347-364.