A Tabu Search Heuristic for the Heterogeneous Vehicle Routing Problem on a Multi-graph

David S. W. Lai[1] and Ozgun C. Demirag[2]
[1]*Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong*
[2]*Black School of Business, Penn State Erie, The Behrend College, Erie, PA, 16563, U.S.A.*

1 STAGE OF THE RESEARCH

I am currently a graduate student, under the supervision of Professor Janny Leung, at the Chinese University of Hong Kong. I have been working on practical problems arising in transportation science and logistics. These research projects, which integrate optimization theories and practice, use mathematical programming and metaheuristics techniques extensively.

During the first part of my Ph.D. study, we have been working on the Shift Rostering Problem (SRP) — the assignment of staff to shifts over a planning horizon such that work rules are observed. SRP arises in hospital nurse-scheduling, call center operations, airlines, urban transportation, and supply chain industries. Effective scheduling of staff can generate considerable saving where unnecessary costs due to misallocation of staff to the demand are reduced. In this project, mathematical programming exact approaches and optimization-based heuristics were developed.

Afterwards, we investigate the Heterogeneous Vehicle Routing Problem (HVRP) where a mixed fleet of vehicles, having distinct vehicle capacities, fixed costs and travel costs, is used to serve a set of customers, minimizing the total costs, subject to the service duration constraint and the capacity constraint. HVRP has been important in many application fields, including transportation, logistics, manufacturing, service industries, etc.

Lastly, starting in 2014, we collaborate with a public transit company in Hong Kong to work on a real-time stochastic integrated vehicle and crew scheduling problem. Vehicles and drivers are scheduled to deliver customers with stochastic travel times. Disruptions due to uncertainties are managed. The problem can be viewed as an integrated version of SRP and HVRP that closely resembles a realistic situation. We are considering algorithms in disruption management and simulation optimization.

2 OUTLINE OF OBJECTIVES

The objectives of my Ph.D. studies include

- modelling and solving shift rostering problems;
- efficient heuristics for HVRP that generates near-optimal feasible solutions in a short time;
- modelling and solving the integrated vehicle and crew scheduling problem with stochastic travel times;
- efficient heuristics that repair and re-optimize disrupted vehicle and crew schedules.

This paper describes a tabu search heuristic that solves the HVRP for near-optimal solutions in a short time.

3 RESEARCH PROBLEM

3.1 Introduction

This paper addresses the Heterogeneous Vehicle Routing Problem (HVRP) where a mixed fleet of vehicles, having distinct vehicle capacities, fixed costs and travel costs, is used to serve a set of customers. The problem is also known as the Mixed Fleet Vehicle Routing Problem or the Heterogeneous Fleet Vehicle Routing Problem. As pointed out by (Baldacci and Mingozzi, 2009), HVRP is a generalization of several important variants of the Vehicle Routing Problem (VRP).

We study an extended version of HVRP where parallel arcs are allowed in the underlying graph. As described by (Dai and Zhou, 2008) in a study of China market, toll charges may contribute a large portion of total transportation cost. It is necessary to consider alternative paths connecting two cities. e.g. a cheaper and longer path without tolls should be selected when

S. W. Lai D. and C. Demirag O..
A Tabu Search Heuristic for the Heterogeneous Vehicle Routing Problem on a Multi-graph.
Copyright © 2014 SCITEPRESS (Science and Technology Publications, Lda.)

time is not restrictive. The potential saving is significant as justified by (Garaix et al., 2010).

Although there are many economic incentives for solving HVRP, handling the problem effectively remains an interesting question. It is possible to solve HVRP to optimality only for relatively small instances using mathematical programming techniques. For larger instances, heuristics and meta-heuristics are more effective ways for obtaining near-optimal solutions. However, with the presence of parallel arcs in the underlying graph, simple operations in many classical heuristics and meta-heuristics often become difficult optimisation sub-problems. Solving sub-problems frequently would be time-consuming. Furthermore, to support sensitivity analysis for decision-making at a strategy level, large number of instances of HVRP have to be solved quickly.

An insertion heuristic and a tabu search heuristic are developed for obtaining good feasible solutions in a short time.

3.2 Formulation

The HVRP we study is described below. Let $G(V,E)$ be a directed multigraph where V is a vertex set and E is an arc set. Vertex $v_0 \in V$ denotes a depot at which all the vehicles are based. The remaining vertices represent n customers. For all customers $i \in V \setminus \{v_0\}$, there is a service time $s_i \in \mathbb{R}^+$ and a demand $d_i \in \mathbb{Z}^+$ to be delivered by a vehicle. Parallel arcs between two vertices represent alternative paths connecting the two locations.

There are heterogeneous vehicles with distinct capacities, fixed costs and travel costs. Vehicles are categorized into types, indexed by $\mathcal{K}$, so that vehicles of the same type are identical. For all vehicle types $k \in \mathcal{K}$, let $Q_k \in \mathbb{Z}^+$ denote the vehicle capacity, $f_k \in \mathbb{R}^+$ denote the fixed cost, and $m_k \in \mathbb{Z}^+$ denote the number of vehicles available. When a vehicle of type $k \in \mathcal{K}$ travels through arc $e \in E$, there is a travel cost $c_e^k \in \mathbb{R}^+$ and a travel time $t_e \in \mathbb{R}^+$.

The objective is to determine the least-cost vehicle routes subject to the following requirements.

1. every route starts and ends at the depot;
2. every customer is visited exactly once by exactly one vehicle;
3. the total demand of any vehicle route of type k may not exceed the vehicle capacity Q_k;
4. all vehicles should return to the depot within a time limit L;
5. the number of type-k vehicles in use should not exceed the vehicles available m_k.

All problem parameters are assumed to be known with certainty. Moreover, arcs represent non-dominated paths only, since paths with a higher cost and a higher travel time can be ignored.

We formulate HVRP as a mixed integer linear programming model.

For all arc $e \in E$ and vehicle type $k \in \mathcal{K}$, let

$$x_e^k = \begin{cases} 1, & \text{if a vehicle of type } k \text{ travels on arc } e; \\ 0, & \text{otherwise.} \end{cases}$$

For all $i, j \in V$ with $i \neq j$, let $y_{ij} \in \mathbb{R}^+$ denote the total demand delivered when the vehicle leaves customer i to serve customer j; Similarly, let $w_{ij} \in \mathbb{R}^+$ denote the cumulative service and travel time when the vehicle leaves customer i to serve customer j. When there is no vehicle travelling from i to j, both y_{ij} and w_{ij} are set to 0.

For notational simplicity, let $E_{ij} \subset E$ denote the set of arcs that incident from vertex i to vertex j; $\delta^+(i) \subset E$ denote the arcs that incident from vertex i; and $\delta^-(i) \subset E$ denote the arcs that incident to vertex i. Furthermore, the depot demand d_{v_0} and service time s_{v_0} are both set to zero.

The problem is formulated as the following mixed integer programming model.

$$\min \sum_{k\in\mathcal{K}} f_k \sum_{e\in\delta^+(v_0)} x_e^k + \sum_{k\in\mathcal{K}} \sum_{e\in E} c_e^k x_e^k, \tag{1}$$

$$s.t. \sum_{k\in\mathcal{K}} \sum_{e\in\delta^+(i)} x_e^k = 1, \forall i \in V \setminus \{v_0\}, \tag{2}$$

$$\sum_{e\in\delta^+(i)} x_e^k - \sum_{e\in\delta^-(i)} x_e^k = 0, \forall k \in \mathcal{K}, i \in V, \tag{3}$$

$$\sum_{e\in\delta^+(v_0)} x_e^k \leq m_k, \forall k \in \mathcal{K}, \tag{4}$$

$$\sum_{j\in V: j\neq i} y_{ij} - \sum_{j\in V: j\neq i} y_{ji} = d_i, \forall i \in V \setminus \{v_0\}, \tag{5}$$

$$y_{ij} \leq \sum_{k\in\mathcal{K}} \sum_{e\in E_{ij}} (Q_k - d_j) x_e^k, \forall i, j \in V : i \neq j, \tag{6}$$

$$\sum_{j\in V: j\neq i} w_{ij} - \sum_{j\in V: j\neq i} w_{ji} = s_i + \sum_{k\in\mathcal{K}} \sum_{e\in\delta^-(i)} t_e x_e^k, \forall i \in V \setminus \{v_0\}, \tag{7}$$

$$w_{ij} \leq \sum_{k\in\mathcal{K}} \sum_{e\in E_{ij}} (L - s_j - t_e) x_e^k, \forall i, j \in V : i \neq j, \tag{8}$$

$$w_{ij} \in \mathbb{R}^+, \forall i, j \in V : i \neq j, \tag{9}$$

$$y_{ij} \in \mathbb{R}^+, \forall i, j \in V : i \neq j, \tag{10}$$

$$x_e^k \in \{0,1\}, \forall e \in E, k \in \mathcal{K}. \tag{11}$$

There are $|E||\mathcal{K}|$ binary variables and $2|V|^2$ non-negative continuous variables. The objective is to

minimize the total fixed costs and travel costs. Constraints (2) ensure that every customer is visited exactly once by exactly one vehicle. Constraints (3) balance the number of vehicles entering and leaving a vertex. Constraints (5) - (6) ensure that all vehicle routes satisfy the capacity constraints. Constraints (7) - (8) ensure that all vehicle routes satisfy the duration constraints. Due to the capacity constraints (or the duration constraints), no subtour would appear in any vehicle route. Constraints (4) limits the number of vehicles in use. If constraints (4) are removed, the model would determine the optimal fleet size for each vehicle type simultaneously.

Constraints (12) - (14) are introduced as needed.

$$\sum_{j\in V\setminus\{v_0\}} y_{ij} \geq \sum_{k\in\mathcal{K}} \sum_{j\in V\setminus\{v_0\}} \sum_{e\in E_{ij}} d_i x_e^k, \forall i \in V\setminus\{v_0\}, \tag{12}$$

$$\sum_{i\in V\setminus\{v_0\}} y_{iv_0} = \sum_{i\in V\setminus\{v_0\}} d_i, \tag{13}$$

$$\sum_{j\in V\setminus\{v_0\}} y_{v_0 j} + \sum_{j\in V\setminus\{v_0\}} w_{v_0 j} = 0. \tag{14}$$

(Yaman, 2006) presented a number of MIP models for a HVRP. Our model is most similar to the disaggregated flow formulation in the sense that the duration and capacity constraints are handled using variables associated on the arcs and vehicle types.

(Baldacci et al., 2008) presented a model for a HVRP where the service duration are not considered and parallel arcs are not allowed. The model described in (Baldacci et al., 2008) corresponds to one with constraint sets (1) - (5), (10), (11), (15) and (16).

$$y_{ij} \leq \sum_{e\in E_{ij}} (Q_k - d_j) x_e^k, \quad \forall i,j \in V : i \neq j, k \in \mathcal{K}, \tag{15}$$

$$y_{ij} \geq \sum_{e\in E_{ij}} d_j x_e^k, \quad \forall i,j \in V : i \neq j, k \in \mathcal{K}. \tag{16}$$

Note that Constraints (6) and (12) are aggregated versions of (15) and (16).

3.3 Literature Review

As compared to VRP, HVRP is less well studied. Since the work of (Golden et al., 1984), a number of heuristics and meta-heuristics have been developed for HVRP. Recent surveys of VRP and HVRP can be found in (Cordeau et al., 2007) and (Baldacci et al., 2008) respectively. Studies of VRPs on multigraphs are scarce. In this section, we give a brief review on a few construction heuristics.

Insertion heuristics (or insertion-based construction heuristics) are widely used because they are able to produce good feasible solutions with a low computational effort. They serve as a major component of meta-heuristics, for constructing initial solutions and generating good vehicle routes. Also, they are useful for repairing interrupted schedules quickly. An extensive survey of construction heuristics for VRPs is presented in (Bräysy and Gendreau, 2005).

Insertion heuristics generate feasible solutions by inserting unassigned customers into vehicle routes, one by one, at a location that minimizes the *insertion cost* (changes in objective value), until all customers are inserted. The sequential version handles one vehicle route at a time whereas the parallel version considers multiple vehicle routes for each insertion. The effectiveness of insertion heuristics depend on the selection of the next insertion customer and the selection of the next insertion location.

A classical sequential insertion heuristic begins with a route that contains a seed customer. The remaining unassigned customers are inserted one by one into the current route at a location with the least insertion cost. When no unassigned customer can be inserted into the current route without violating the capacity constraint or the duration constraint, the procedure repeats with a new vehicle route. Seed customers are selected among the unassigned customers with the farthest distance from the depot. The procedure stops when all customers are assigned.

Another popular class of sequential insertion heuristic is known as sweep heuristic. Customers are inserted one by one following a predetermined order based on the polar angles from the depot to the customers. (Renaud and Boctor, 2002) developed a sophisticated version for generating good vehicle routes. By solving a set-partitioning model, near-optimal solutions of a HVRP are obtained effectively.

4 STATE OF THE ART

While the mixed integer programming model can solve small instances effectively, heuristics are proposed for large instances. We contribute a novel insertion heuristic that generates feasible solutions quickly.

Insertion heuristics generate feasible solutions by inserting unassigned customers into vehicle routes, one by one, at a location that minimizes the *insertion cost* (changes in objective value), until all customers are inserted. For classical insertion heuristics, insertion costs are determined with fixed vertex orders in existing vehicle routes. When parallel arcs presence, there are more flexibility and higher chance of improving a solution through vertex sequencing and arc selection. Therefore, insertion costs would be poorly

estimated if vertex sequencing and arc selection are fixed. This may lead to a poor solution because customers are assigned to undesirable vehicles.

With this in mind, we propose an insertion heuristic where insertion costs are efficiently estimated through vertex sequencing and arc selection. Furthermore, the proposed insertion heuristic could be incorporated in a tabu search heuristic for obtaining near-optimal solutions. Extensive computational tests have been performed for sensitivity analysis.

5 METHODOLOGY

We present an efficient insertion-based heuristic for the HVRP. The insertion costs are estimated with customer sequencing and arc selection using an efficient algorithm. The proposed insertion heuristic is incorporated in a tabu search heuristic for obtaining near-optimal solutions. Extensive computational tests have been performed for sensitivity analysis.

5.1 Penalized Objective Function

Heuristics often need to examine solutions that may be infeasible. The penalized objective function used in the heuristics described in Section 5.2 and Section 5.3 is defined as follows.

Let $\mathcal{X}$ denote the set of solutions that satisfy requirements 1 and 2: every route starts and ends at the depot; every customer is visited exactly once by exactly one vehicle. For any solution $x \in \mathcal{X}$, let $\mathcal{R}(x)$ denote the vehicle routes that contain at least one customer. For any vehicle route $P \in \mathcal{R}(x)$, let $V(P)$ and $E(P)$ be the vertices and arcs in the route. If a vehicle of type $k \in \mathcal{K}$ is assigned to route P, the travel cost $c(P)$, overload $q(P)$ and overtime $t(P)$ can be written as follows.

$$c(P) = f_k + \sum_{e \in E(P)} c_e^k,$$

$$t(P) = \Big[\sum_{e \in E(P)} t_e + \sum_{i \in V(P)} s_i - L \Big]^+,$$

$$q(P) = \Big[\sum_{i \in V(P)} d_i - Q_k \Big]^+.$$

To penalize the infeasible solutions, a solution $x \in \mathcal{X}$ is evaluated using the *penalized objective function* $z(x) = \sum_{r \in \mathcal{R}(x)} (c(r) + \alpha\, q(r) + \beta\, t(r))$ where $\alpha \in \mathbb{R}^+$ and $\beta \in \mathbb{R}^+$ are *penalty weights*.

5.2 Insertion Heuristic

A heuristic is proposed to construct feasible solutions by iteratively inserting unassigned customers into vehicle routes. Initially, each available vehicle route contains exactly one customer that is randomly picked. The remaining customers are inserted one by one, following a randomized order, into a vehicle route that minimizes the *insertion cost* (changes in penalized objective value). We estimate insertion costs using an efficient heuristic where vertex sequencing and arc selection are performed.

Insertion is an elementary operation that is performed frequently — a customer is inserted into a vehicle route at a position that minimizes the penalized objective function. It is an easy operation for a simple graph. However, when multiple arcs presence, it is a difficult problem even when vertex sequence is fixed. Suppose that $\mathcal{V} = (v_0, v_1, ..., v_l, v_{l+1})$ is the

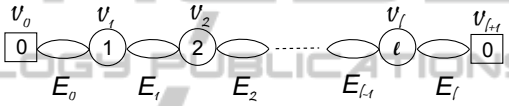

Figure 1: Notations for the FSASP.

vertex sequence (after a customer is inserted) where v_0 and v_{l+1} denote the depot. For all $i \in \{0, 1, 2, .., l\}$, let E_i be the set of arcs between vertices v_i and v_{i+1}. Let $\widehat{c_e}$ and $\widehat{t_e}$ be the travel cost and travel time of arc e for the corresponding vehicle. The problem is to select an arc-combination $\mathcal{E} \in E_0 \times ... \times E_l$ minimizing $K_1 + \sum_{e \in \mathcal{E}} \widehat{c_e} + \beta \big[\sum_{e \in \mathcal{E}} \widehat{t_e} - K_2 \big]^+$ where K_1 and K_2 are non-negative constants given by $K_1 = f_k + \alpha \big[\sum_{i \in \mathcal{V}} d_i - Q_k \big]^+$ and $K_2 = L - \sum_{i \in \mathcal{V}} s_i$ for the corresponding vehicle type $k \in \mathcal{K}$. Essentially, the problem corresponds to a Multiple Choice Knapsack problem which is $\mathcal{NP}$-hard. (Garaix et al., 2010) introduced the problem as the Fixed Sequence Arc Selection Problem (FSASP) which is solved using dynamic programming.

We develop an efficient heuristic for FSASP. The heuristic is described below.

Step 1. If the longest path (with largest time and smallest cost) satisfies the duration constraint, return the longest path.

Step 2. If the duration of the shortest path is greater than or equal to the time limit, return the path with the following arcs.

$$e_i^* = arg \min_{e \in E_i} \widehat{c_e} + \beta \widehat{t_e}, \qquad \forall i \in \{0, 1, ..., l\}.$$

Step 3. Otherwise, start with the shortest path and then pick the arcs, one by one, following a prede-

termined order, until the penalized objective function is no longer improving.

(a) Denote the currently selected arc in E_i as $\bar{e}_i$.

(b) For all $i \in \{0,1,...,l\}$, determine arc $e_i^* \in E_i$ minimizing the ratio as follows.

$$e_i^* = arg\min_{e \in E_i} \frac{\widehat{c_e} - \widehat{c_{\bar{e}_i}}}{\widehat{t_{\bar{e}_i}} - \widehat{t_e}}, \qquad \forall i \in \{0,1,...,l\}.$$

(c) Arc e_i^* is selected one by one following a non-decreasing order of the ratio.

(d) Stop when the penalized objective function is no longer improving.

The algorithm is implemented with time complexity $O(nlog(n))$ where n is the number of customers.

Vertex sequencing is achieved based on solving FSASP iteratively. Each iteration, a customer is picked and is reinserted into the same vehicle route at a position that minimizes the penalized objective function. Stop when the penalized objective function is no longer improving.

5.3 Tabu Search Heuristic

Tabu search has been one of the most widely applied meta-heuristics for obtaining near optimal solutions. The search begins with an initial solution. Successive neighbors of a solution are examined iteratively. To avoid poor local minima, the search always moves to the best neighborhood solution, even if it is worse than the current solution. Recently examined solutions are forbidden for a number of iterations to prevent cycling.

The following paragraphs describe the tabu search heuristic used in our prototype. Additional features developed for tabu search can be applied to improve the search. See, e.g. (Rochat and Taillard, 1995), (Gendreau et al., 1999), (Taillard, 1999) and (Ho and Gendreau, 2006).

Neighborhood Structure. Neighborhood solutions of $x \in \mathcal{X}$, denoted as $\mathcal{N}(x)$, are defined as the resulting solutions after applying a move operation on the current solution x — moving a customer to another vehicle route at a location that minimizes the penalized objective function.

Tabu Operations and Aspiration Criterion. To prevent cycling, if customer v has been moved from vehicle R to vehicle S in the i iteration, then moving customer v back to vehicle R is forbidden until the $i+\theta$ iteration. These tabu solutions are allowed only when their objective values are better than the best feasible solution found by the search, which is often referred as the *aspiration criterion*.

The Best Neighborhood Solution. Operations that have been performed frequently should be penalized for diversification purpose. As described in (Ho and Gendreau, 2006), the penalty $\phi(x) = \lambda c(x)\sqrt{n}\vartheta_{ik}$ is used where n is the number of customers, ϑ_{ik} counts the number of times customer i has been moved to vehicle k, and λ is an user defined positive parameter that controls the intensity of diversification. The number of non-empty vehicles is not included since it has been reflected in the fixed costs in the objective function.

The best neighborhood solution $\bar{x} \in \mathcal{N}(x)$ is selected that minimizes $z(\bar{x}) + \phi(\bar{x})$ while $\bar{x}$ is non-tabu solution (unless it satisfies the aspiration criterion).

Self Adjusting Penalty Weights. To avoid poor local minima, the penalty weights α and β are self-adjusting in the search. As described in (Ho and Gendreau, 2006), the penalty weights are initially set to 1. If the next solution satisfies the constraint, the corresponding penalty weight is divided by $\delta + 1$ where $\delta \in \mathbb{R}^+$ is a user defined parameter. Otherwise, it is multiplied by $\delta + 1$.

6 COMPUTATIONAL RESULT

Although real-world datasets are interesting from a practical point of view, the randomly generated datasets allow us more freedom to construct different scenarios for sensitivity analysis. In this section, the following approaches are tested on some randomly generated instances that simulate practical scenarios described in (Dai and Zhou, 2008).

- **MIP:** The mixed integer programming model (1) - (14) described in Section 3.2 is solved using CPLEX 12.5 with the default setting. The best feasible solution obtained within two hours is reported. This serves as a reference to the other approaches.
- **IH:** The insertion heuristic described in Section 5.2 is used to generate 1000 solutions with various insertion order and number of vehicles used. The best solution obtained is reported. To encourage feasible solutions, the penalty weights are set to a large positive number.
- **TS:** Initially, 5 solutions are constructed by **IH**. To encourage diversified structure, the penalty weights are set to 1. A tabu search is carried out on each of 5 solutions for 100 iterations. The best solution found so far is selected as a starting point for the main search for 1000 iterations.

The parameters used in the tabu search heuristic are shown in Table 1.

Table 1: Parameters for the tabu search heuristic.

Parameter	Value
Diversification intensity (λ)	0.0001
Penalty update factor (δ)	0.5
No. of tabu iterations (θ)	$\lceil 5log_{10}(n) \rceil$

Note: n is the number of customers.

6.1 Instances

HVRP instances are randomly generated to reflect different practical considerations with the parameters shown in Table 2. Vehicles with a larger capacity have higher dispatch cost and fuel cost. Most customers are located in the region near the depot. Some customers are located in more remote area. Figure 2 shows an example.

For easier illustration, we simplify the problem as follows. Firstly, there are exactly two parallel arcs in every vertex-pair. Secondly, for a vehicle of a given type, the travel times and travel costs going from vertex i to vertex j are same as going from vertex j to vertex i (i.e. symmetric times and costs).

Table 2: Characteristics of the instances.

Type k	Fuel cost δ_k	Toll charge η_{ij}^k
Small Vehicle	Uniform[0.5,1.1]	Uniform[0.2,0.3] (50% of the arcs)
Large Vehicle	Uniform[1.4,2.0]	Uniform[0.2,0.3] (50% of the arcs)

Demand	d_i	Uniform[5,35], integer
Service time	s_i	Uniform$[1+0.2d_i, 2+0.2d_i]$
Location coordinates $(r \cdot cos\theta,\ r \cdot sin\theta)$		$r \sim$ Uniform[0,25] (80% of customers) $r \sim$ Uniform[25,100] (20% of customers) $\theta \sim$ Uniform[0,2π]
Travel time	t_{ij}	Manhattan distance between i and j
Travel cost	c_{ij}^k	$(\delta_k + \eta_{ij}^k)\, t_{ij}$
Time limit	L	250 units
No. of Vehicles	m_k	Small vehicles: $max(3, \lceil \sum_{i \in V \setminus \{v_0\}} d_i/150 \rceil)$ Large vehicles: $max(3, \lceil \sum_{i \in V \setminus \{v_0\}} d_i/300 \rceil)$
Vehicle Capacity	Q_k	Small vehicles: 150 Large vehicles: 300
Dispatch cost	f_k	Small vehicles: Uniform[95,105] Large vehicles: Uniform[145,155]

Algorithmic parameters, if any, are tuned using a set of instances (*training set*). All the algorithms are evaluated using another set of instances (*test set*). To avoid over-fitting, no algorithmic parameter were tuned using the test set.

The performance of different algorithms are compared using the best solution values found within a fixed maximum solving time. Large number of instances are used so that similar result could be reproduced with a different test set.

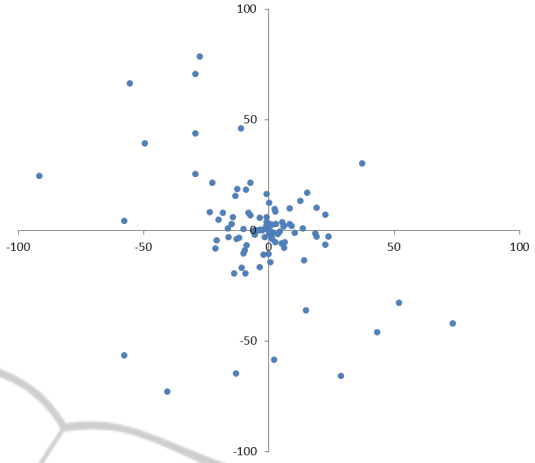

Figure 2: Customer distribution.

All experiments were conducted on a desktop personal computer running Windows 7 with an Intel Core i7-2600 processor 3.4 GHz and 4 GB of main memory. All algorithms are implemented in C++ and have been compiled using Visual Studio 2012.

6.2 Solution Quality

In order to provide accurate sensitivity analysis, we need to evaluate the quality of the solutions obtained by the proposed approaches.

The solution quality of IH and TS are evaluated using a test set with 100 small instances. The number of customers is selected from the set {14,15,16,17}, with 25 instances generated for each case. Optimal solutions are obtained using MIP. Figure 3 illustrates the performance of IH and TS as compared to the optimal solutions. The optimality gap is calculated by $\frac{100(z-r)}{r}$ where z is the solution found by TS or IH, and r is the optimal solution. As shown in Figure 3, for small instances, IH is able to produce good feasible solutions quickly while TS produces near-optimal solutions.

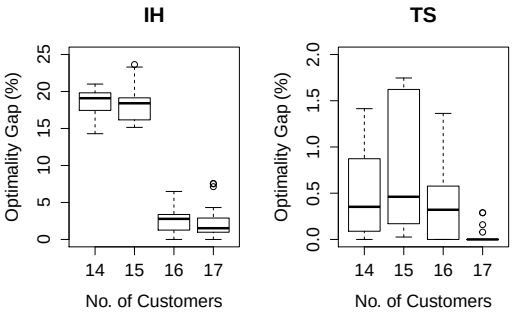

Figure 3: Optimality gap on small instances.

Optimal solutions for larger instances could not be obtained in a reasonable time. Therefore, we evaluate the solution quality of IH and TS by comparing to the best solutions obtained by MIP at a time limit of two hours. There are 15 instances tested. There are 3 small vehicles and 3 large vehicles. The number of customers is selected from the set $\{20, 25, 30\}$ with 5 instances generated for each case. When there are more customers, MIP may not be unable to find a good feasible solution within the computational time limit. The result is shown in Table 3.

Table 3: Performance of MIP, IH and TS.

Instance	No. of Customers	MIP obj.	IH obj.	IH time(s)	TS obj	TS time(s)
1	20	822.39	856.91	0.218	822.39	1.575
2	20	783.6	825.84	0.218	786.94	1.623
3	20	814.39	877.77	0.218	816.3	1.591
4	20	823.12	905.35	0.219	824.25	1.575
5	20	959.92	832.11	0.203	814.28	1.592
6	25	1024.59	1117.11	0.343	1024.59	2.215
7	25	1077.89	1073.45	0.343	1052.51	2.465
8	25	1069.79	1061.88	0.343	999.74	2.152
9	25	1092.2	1043.69	0.343	996.98	2.169
10	25	1073.74	1020.98	0.344	1030.58	2.652
11	30	1074.77	1121.29	0.483	1041.56	2.621
12	30	1127.1	1125.27	0.468	1021.4	2.62
13	30	1048.39	1186.97	0.468	1044.61	2.621
14	30	1090.07	1158.13	0.484	1038.03	2.559
15	30	1126.36	1224.11	0.468	1060.05	2.496

As shown in Table 3, IH and TS tend to produce better solutions than MIP for larger instances. While IH generates a large number of feasible solutions in a short time, TS is able to find much better solution than IH and MIP within a few seconds.

6.3 Sensitivity Analysis

Through sensitivity analysis, we try to develop managerial insights for better decisions and hopefully generate more effective heuristics. The result could be reproduced on a different test set without significant deviations.

6.3.1 Duration Constraint

Instances with different time limit is tested. Initially, 10 scenarios are generated with 100 customers. By varying the service time-limit L to values of $\{200, 210, 220, 230, 240\ 250\}$, we have 60 instances. At most 12 small vehicles and 6 large vehicles can be used. Figure 4 illustrates the objective value and the time at which the solution is found.

Figure 6 illustrates the effect of service time limit for each vehicle type on the following measures.

- Number of vehicles in use;
- Capacity Utilization: total demand delivered by the vehicles divided by the vehicle capacity;

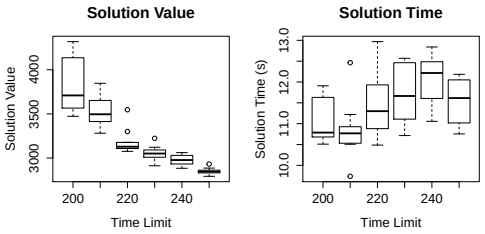

Figure 4: Performance of TS with different time limits.

- Time Utilization: total travel time divided by the time limit;
- More Costly Arcs: total number of more costly arcs used by a vehicle type divided by the total number of arcs used by the vehicle type;
- Remote Customers: total number of remote customers visited by a vehicle type divide by the total number of remote customers.

As shown in Figure 6, small vehicles should be used to visit remote customers since the travel cost is smaller. Furthermore, large vehicles are useful only when the duration constraint is restrictive.

6.3.2 Capacity Constraint

Similarity, we perform sensitivity on the capacity constraint. Vehicle capacities (for both the small vehicles and large vehicles) are reduced by a ratio among values of $\{0.6, 0.7, 0.8, 0.9, 1\}$. For example, 0.6 means that the capacities are reduced by 40% whereas 1 means that the capacities are unchanged. The time limit is fixed to 250. Figure 5 illustrates the objective value and the time at which the solution is found. Figure 7 illustrates the effect of vehicle capacity on the measures.

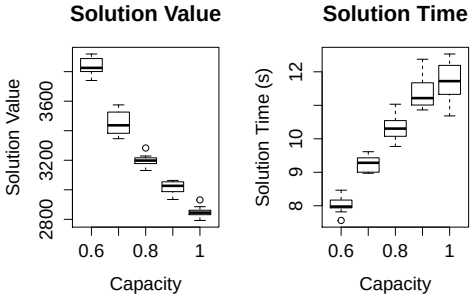

Figure 5: Performance of TS with different vehicle capacities.

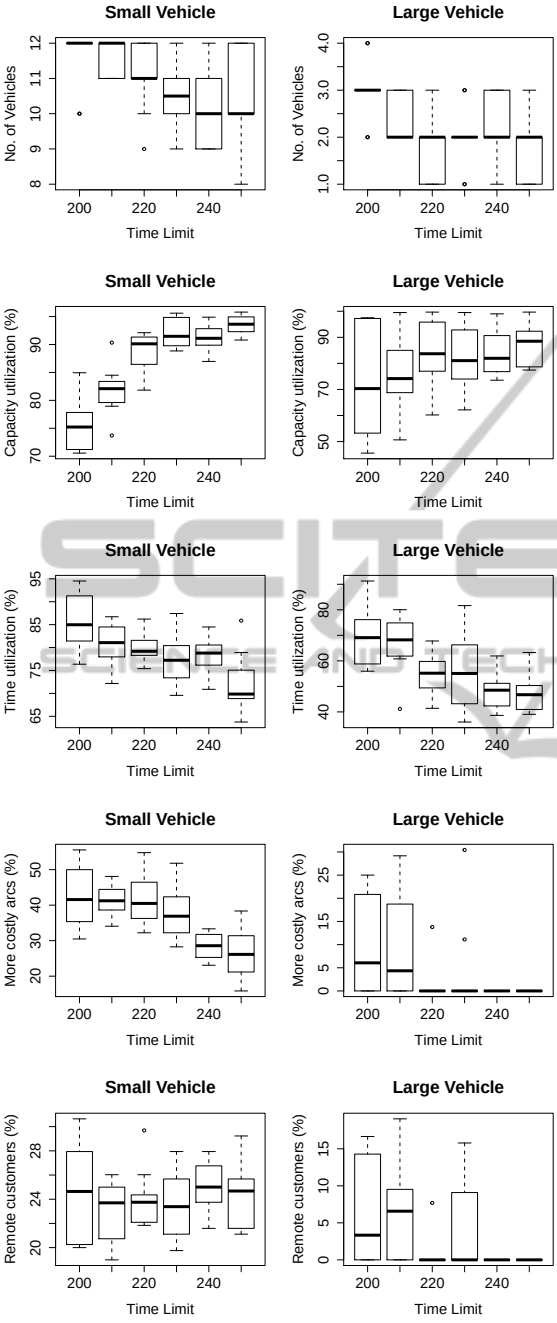

Figure 6: The effect of time limit.

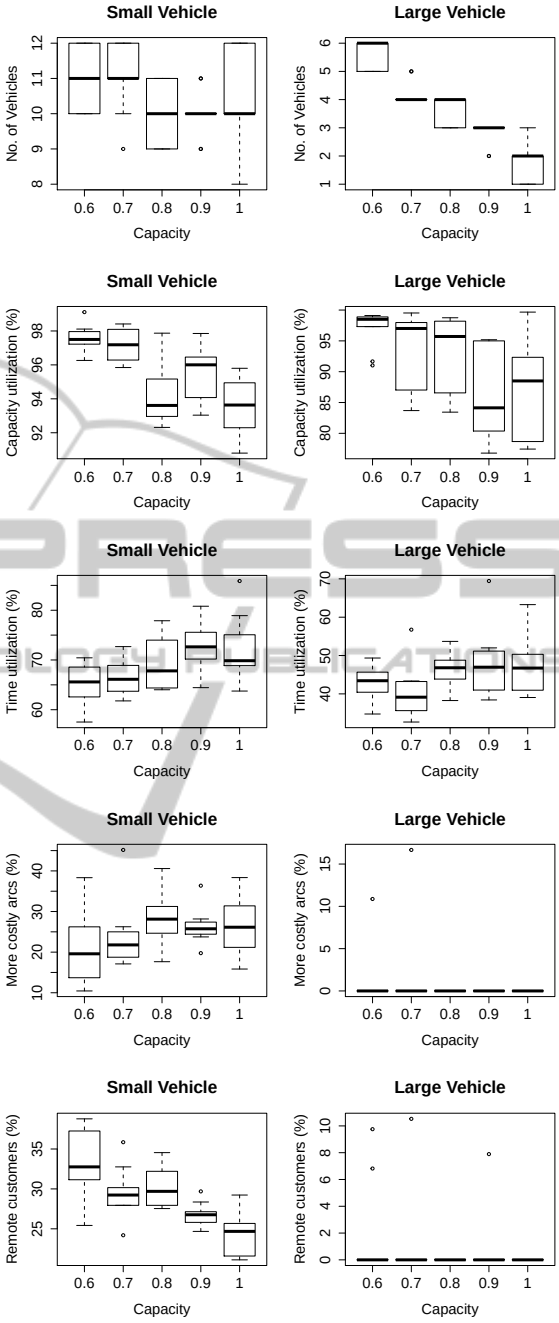

Figure 7: The effect of capacitiy limit.

7 EXPECTED OUTCOME

The proposed insertion heuristic is able to produce feasible solutions quickly. When extra time is provided, near-optimal solutions could be obtained using the tabu search heuristic.

We have performed extensive computational tests on HVRP instances that reflects different practical considerations. For sensitivity analysis, a number of measures are introduced to characterise the solutions. The results are summarized below.

- large vehicles are more useful when the duration constraint or the capacity constraint is restrictive;
- the fleet composition depends on the vehicle capacities;

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

- small vehicles tend to use more costly arcs than large vehicles;
- small vehicles should be used to visit remote customers since the travel cost is lower.

REFERENCES

Baldacci, R., Battarra, M., and Vigo, D. (2008). Routing a heterogeneous fleet of vehicles. In Golden, B., Raghavan, S., and Wasil, E., editors, *The vehicle routing problem: latest advances and new challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 3–27. Springer US.

Baldacci, R. and Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. *Math. Programming*, 120:347–380.

Bräysy, O. and Gendreau, M. (2005). Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Sci.*, 39:104–118.

Cordeau, J., Laporte, G., Savelsbergh, M., and Vigo, D. (2007). Vehicle routing. In Barnhart, C. and Laporte, G., editors, *Transportation, Handbooks in Operations Research and Management Science*, volume 14, pages 367–428. Elsevier, Amsterdam.

Dai, J. and Zhou, C. (2008). *Beer Distribution in China*. PhD thesis, The School of Industrial and Systems Engineering, Georgia Institute of Technology.

Garaix, T., Artigues, C., Feillet, D., and Josselin, D. (2010). Vehicle routing problems with alternative paths: An application to on-demand transportation. *Eur. J. Oper. Res.*, 204:62–75.

Gendreau, M., Laporte, G., Musaraganyi, C., and Taillard, E. (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Resesarch*, 26:1153–1173.

Golden, B., Assad, A., Levy, L., and Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Resesarch*, 11:49–66.

Ho, S. and Gendreau, M. (2006). Path relinking for the vehicle routing problem. *J. Heuristics*, 12:55–72.

Renaud, J. and Boctor, F. (2002). A sweep-based algorithm for the fleet size and mix vehicle routing problem. *Eur. J. Oper. Res.*, 140:618–628.

Rochat, Y. and Taillard, E. (1995). Probabilistic diversification and intensification in local search for vehicle routing. *J. Heuristics*, 1:147–167.

Taillard, E. (1999). A heuristic column generation method for the heterogeneous fleet vrp. *RAIRO - Operations Research*, 33:1–14.

Yaman, H. (2006). Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Math. Programming*, 106:365–390.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS