Heuristics for the routing of trucks with double container loads*

Michela Lai[1], Massimo Di Francesco[2], and Paola Zuddas[3]

1 Department of Mathematics and Computer Science, University of Cagliari
Via Ospedale 72, 09124, Cagliari, Italy
mlai@unica.it
2 Department of Land Engineering, University of Cagliari
Via Marengo 2, 09123, Cagliari, Italy
mdifrance@unica.it
3 Department of Mathematics and Computer Science, University of Cagliari
Via Ospedale 72, 09124, Cagliari, Italy
zuddas@unica.it

Abstract

This research addresses a problem motivated by a real case study. A carrier must plan the routes of trucks in order to serve importers and exporters. What is original in this vehicle routing problem is the impossibility to separate trucks and containers during customer service and the opportunity to carry up to two containers per truck. Customers may demand more than one container and may be visited more than once. Moreover, according to the carrier's policy, importers must be served before exporters. In order to address this Vehicle Routing Problem with backhaul and splits, a linear integer programming model is proposed. This research aims to show to what extent an exact algorithm of a state of the art solver can be used to solve this model. Moreover, since some instances are too difficult to solve for the exact algorithm, a number of heuristics is proposed and compared to this algorithm. Finally, the heuristics are compared to the real decisions of the carrier who has motivated this problem.

1998 ACM Subject Classification H.4.2 Types of Systems Logistics

Keywords and phrases Split Vehicle Routing Problem, Backhaul, Drayage, Container transportation, Heuristics

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.84

1 Introduction

This paper addresses a vehicle routing problem, which is motivated by a real case study. A carrier is in charge of planning the distribution of container loads by trucks and containers based at a port. The carrier has a homogeneous fleet of trucks carrying up to two containers and the planning of routes must be performed within 10 minutes. Two classes of customers must be served: importers and exporters. The importers need to receive full container loads from the port and the exporters need to ship container loads to the port. Typically customers need to be served by more than one container and must be visited by more than one truck.

According to the carrier's policy, trucks and containers cannot be uncoupled during customer service, because truck drivers are required to check the right execution of operations. As a result, in this problem there are no pickups or deliveries of loaded and empty containers:

* This work was partially supported by Grendi Trasporti Marittimi.

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 84–93
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

during customer service containers are filled or emptied and moved away by the same trucks used for bringing containers to customers.

Moreover, since the container loads of exporters are typically not ready before the afternoon, the carrier policy is to serve importers before exporters. As a result, empty containers leaving from importers can be moved to exporters, where they are filled and shipped to the port.

This problem belongs to the class of Vehicle Routing Problem (VRP) with backhauls, because deliveries must be performed before pickups [1]. However, in classical VRP with backhaul each customer must be visited only once, whereas in our problem multiple visits at each customer are allowed. Therefore, although several solution methods exist in VRP with backhauls [2], [3], [4], [5], they may be suboptimal because splits are not considered.

There are also some similarities to drayage problems, which consists of picking up and delivering of full containers. Typically they are separated from trucks during customer service [6], whereas this is not possible in this problem. The closest problem setting was probably faced by Imai et al. [7], who studied the optimal assignment of own and chartered trucks to a set of delivery and pickup pairs. As in our setting, tractors and containers cannot be uncoupled, but the capacity of trucks is limited to one container only. Homogeneous fleets with one container trucks are also considered in [8] and [9].

The objective of this paper is to propose an optimization model accounting for the original characteristics of this problem. The model minimizes distribution costs, such that all customers are served as requested, truck capacity constraints hold and importers are served before exporters.

Since we are required to determine efficient solutions rapidly, this paper also aims to propose a number of heuristics for this problem. The common idea between these heuristics is to build an initial set of routes in which all customers are either importers or exporters. Next, these routes are merged according to different criteria, one for each heuristic.

The contributions of this paper are:

- to present a problem with some original characteristics, which have not been investigated in the rich VRP literature;
- to model the problem by a linear integer programming formulation;
- to propose and evaluate a number of heuristics with respect to an exact algorithm of a state-of-art solver.

The paper is organized as follows. In Section 2 the problem description is presented. The problem is modeled in Section 3. Solution methods are described in Section 4. The heuristics are tested in Section 5. Section 6 presents a summary of conclusions and describes future research perspectives in the field.

2 Problem description

Consider a fleet of trucks and containers based at a port. Trucks carry up to two containers and serve two types of customer requests: the delivery of container loads from the port to importers and the shipment of container loads from exporters to the same port. Typically customers need to ship or receive more than one container load. Therefore, usually each customer must be served by multiple containers and must be visited more than once.

A relevant characteristic of this problem is the impossibility to separate trucks and containers during customer service. As a result, when importers receive container loads by trucks, containers are emptied and moved away by the same trucks used for providing container loads. Similarly, when exporters are served by empty containers, containers are filled and moved away by the same trucks used for providing empty containers.

According to the carrier's policy, importers must be served before exporters. As a result, routes may consist in the shipment of container loads from the port to importers, the direct allocation of empty containers from importers to exporters and the final shipment of container loads from exporters to the port. Therefore, trucks can serve in a route up to four customers (two importers and two exporters). Every pair of containers can be shipped in a truck. All containers leaving from importers can be used to serve exporters, no incompability occurs between customers and trucks, which can serve almost any customer, and there are no priorities among importers and among exporters.

It is worth noting that the number of container loads to be picked up and delivered is generally different. When the number of container loads delivered to importers is larger than the number of container loads shipped by exporters, several empty containers must be moved back to the port. When the number of container loads delivered to importers is lower than the number of container loads shipped by exporters, several empty containers must be put on trucks leaving from the port, in order to serve all customers.

The movement of trucks generate routing costs. In this problem, all trucks lead to the same routing costs per unitary distance. Moreover, handling costs are paid to put containers on trucks at the port. The objective is to determine the routes of trucks in order to minimize routing and handling costs, such that customers are served as requested, truck capacity constraints hold and importers are served before exporters.

3 Modeling

We consider a port p, a set I of importers, a set E of exporters, and a set K of different trucks, whose transportation capacity is 2 containers. An integer demand of $d_i \geq 0$ containers is associated with each customer $i \in I \cup E$. It represents the number of loaded containers requested to serve import customer $i \in I$ and it is also equal to the number of empty containers returned by this customer. When $i \in E$, d_i represents the number of empty containers requested to serve export customer $i \in E$ and it is also equal to the number of loaded containers shipped by this customer.

Consider a direct graph $G = (N, A)$, where $N = \{p \cup I \cup E\}$ and the set of arcs A includes all allowed ways to move trucks:

- from the port to any importer and any exporter;
- from an importer to the port, any other importer and any exporter;
- from an exporter to the port and any other exporter.

More formally, the set A is defined as $A = A_1 \cup A_2$, where

$$A_1 = \{(i, j) | i \in I \cup p, j \in N, i \neq j\}$$
$$A_2 = \{(i, j) | i \in E, j \in E \cup p, i \neq j\}$$

The operation cost c_{ij} for any truck traversing arc $(i, j) \in A$ is supposed to be nonnegative. Let h_{pj} be the nonnegative handling cost of a container put on and picked from any truck at the port p to serve node $j \in N$.

The following decision variables are defined:

- x_{ij}^k: Routing selection variable equal to 1 if arc $(i, j) \in A$ is traversed by truck $k \in K$, and 0 otherwise;
- y_{ij}^k: Integer variable representing the number of loaded containers moved along arc $(i, j) \in A$ by truck $k \in K$;

- z_{ij}^k: Integer variable representing the number of empty containers moved along arc $(i,j) \in A$ by truck $k \in K$.

The problem can be formulated as follows:

$$min \sum_{k\in K} \left[\sum_{(i,j)\in A} c_{ij} x_{ij}^k + \sum_{j\in N} h_{pj}(y_{pj}^k + z_{pj}^k) \right] \tag{1}$$

$s.t.$

$$\sum_{k\in K}\sum_{l\in N} y_{il}^k = \sum_{k\in K}\sum_{j\in p\cup I} y_{ji}^k - d_i \qquad \forall i \in I \tag{2}$$

$$\sum_{k\in K}\sum_{l\in N} z_{il}^k = \sum_{k\in K}\sum_{j\in p\cup I} z_{ji}^k + d_i \qquad \forall i \in I \tag{3}$$

$$\sum_{l\in N} y_{il}^k \le \sum_{j\in p\cup I} y_{ji}^k \qquad \forall i \in I, \forall k \in K \tag{4}$$

$$\sum_{l\in N} z_{il}^k \ge \sum_{j\in p\cup I} z_{ji}^k \qquad \forall i \in I, \forall k \in K \tag{5}$$

$$\sum_{k\in K}\sum_{l\in p\cup E} y_{il}^k = \sum_{k\in K}\sum_{j\in N} y_{ji}^k + d_i \qquad \forall i \in E \tag{6}$$

$$\sum_{k\in K}\sum_{l\in p\cup E} z_{il}^k = \sum_{k\in K}\sum_{j\in N} z_{ji}^k - d_i \qquad \forall i \in E \tag{7}$$

$$\sum_{l\in p\cup E} y_{il}^k \ge \sum_{j\in N} y_{ji}^k \qquad \forall i \in E, \forall k \in K \tag{8}$$

$$\sum_{l\in p\cup E} z_{il}^k \le \sum_{j\in N} z_{ji}^k \qquad \forall i \in E, \forall k \in K \tag{9}$$

$$y_{ij}^k + z_{ij}^k \le 2x_{ij}^k \qquad \forall (i,j) \in A, \forall k \in K \tag{10}$$

$$\sum_{j\in N} x_{ji}^k - \sum_{l\in N} x_{il}^k = 0 \qquad \forall i \in N, \forall k \in K \tag{11}$$

$$\sum_{j\in N} x_{pj}^k \le 1 \qquad \forall k \in K \tag{12}$$

$$\sum_{k\in K}\sum_{i\in I\cup E} z_{ip}^k - \sum_{k\in K}\sum_{i\in I\cup E} z_{pi}^k = \sum_{i\in I} d_i - \sum_{i\in E} d_i \tag{13}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall (i,j) \in A, \forall k \in K \tag{14}$$

$$y_{ij}^k \in \{0,1,2\} \qquad \forall (i,j) \in A, \forall k \in K \tag{15}$$

$$z_{ij}^k \in \{0,1,2\} \qquad \forall (i,j) \in A, \forall k \in K \tag{16}$$

Container handling and truck routing costs are minimized in the objective function (1).

Constraints from (2) to (5) concern the service to importers. Constraints (2) and (3) are the flow conservation constraints of loaded and empty containers respectively at each importer node. Constraints (4) and (5) check the number of loaded and empty containers in each truck entering and leaving from importers: when a truck leaves from each importer, the number of loaded containers cannot increase and the number of empty containers cannot be reduced.

Constraints from (6) to (9) concern the service to exporters. Constraints (6) and (7) are the flow conservation constraints of loaded and empty containers, respectively, for each exporter node. Constraints (8) and (9) control the number of loaded and empty containers in each truck entering and leaving from exporters: when a truck leaves from each exporter, the number of loaded containers cannot be reduced and the number of empty containers cannot be increased.

Constraint (10) imposes that the number of containers moved by each truck is not larger than the transportation capacity. Constraints (11) are the flow conservation constraints for trucks at each node. Constraint (12) guarantees that trucks are not used more than once. Constraint (13) represents the flow conservation of empty containers at port p.

Finally, constraints (14), (15) and (16) define the domain of decision variables.

4 Solution methods

Several solution methods can be adopted to solve the previous problem. Generally speaking, they can be divided into exact and heuristic methods. In this paper we illustrate to what extent a well-known exact algorithm can be used to face this problem. This analysis is performed by the solver Cplex, which solves integer programming models by a branch-and-cut algorithm.

However, most Vehicle Routing Problems are *NP-hard* and, also in our problem, there is little hope of finding exact solution procedures for large problem instances [11]. Therefore, in this section we propose a number of heuristics, which can be used to tackle the problem at hand.

All proposed heuristics are composed of two phases. The first phase, which is the same for all heuristics, determines an initial solution, in which all routes serve either importers or exporters. In the second phase, each heuristic implements a different rule to merge the routes determined in the first phase. Finally, the best heuristic in terms of objective function is selected.

In the first phase, we face two vehicle routing problems with splits: the first has importers only, whilst the second has exporters only. However, Split Vehicle Routing Problems are also known to be difficult. Therefore, since an efficient metaheuristic for this class of problems has been proposed by Archetti et al. [10], their algorithm has been chosen to determine routes in the first phase.

The routes determined in the first phase are merged in the second phase according to different saving-based heuristics. Savings represent the routing costs achieved by merging two routes instead of leaving them separately. Given a route i with importers only and a route j with exporters only, the saving generated by their merging is computed as $s_{ij} = c(i) + c(j) - c(ij)$, in which $c(i)$ and $c(j)$ are the respective costs of routes i and j, and $c(ij)$ is the cost of the merged route. Savings are saved in a matrix, in which the number of rows is equal to the number of routes serving importers in the first phase and the number of columns is equal to the number of routes serving exporters in the first phase.

In this paper, the order of visits between pairs of importers and pairs of exporters is not changed after the merging. To clarify, let us consider for instance two importers i_1 and i_2 and two exporters e_1, e_2. Assume that the routes determined in the first phase are p, i_1, i_2, p and p, e_1, e_2, p. If these routes are merged, the final route is p, i_1, i_2, e_1, e_2,p. Therefore, the possibility of visiting importer i_2 before importer i_1 and exporter e_2 before e_1 is not taken into account.

Some definitions are provided for the sake of clarity in the presentation of the heuristics:

Row i represents the $i-th$ route of importers, as determined in the first phase;

Column j represents the $j-th$ route of exporters, as determined in the first phase;

Entry s_{ij} $>= 0$ is the saving generated by the merging of routes i and j. Only nonnegative savings are considered. When an entry s_{ij} takes value 0, the merging is not allowed. Whenever two routes i and j are merged by a heuristic, the related saving s_{ij} is set to 0;
$\boldsymbol{m_i}$ Number of columns (routes) that can be merged with the route represented by row i;
$\boldsymbol{m_j}$ Number of rows (routes) that can be merged with the route represented by column j;
$\boldsymbol{avrg_i}$ is the average of all savings in row i;
$\boldsymbol{avrg_j}$ is the average of all savings in column j.

We propose eight heuristics, whose solution is denoted by $s_{0,\dots,7}$:

Heuristic 0 (H0) This heuristic does nothing and returns routes as determined in the first phase:
Step$_0$ $s_0 = \emptyset$.
Step$_1$ For each row i, insert i into s_0.
Step$_2$ For each column j, insert j into s_0.

Heuristic 1 (H1) This heuristic determines the maximum saving for each route of importers and selects the best routes serving exporters:
Step$_0$ $s_1 = \emptyset$.
Step$_1$ For each row i, select the largest s_{ij}. Merge routes i and j, if any, and insert the new route into s_1.
Step$_2$ For each row i not involved in any merging, insert i into s_1.
Step$_3$ For each column j not involved in any merging, insert j into s_1.

Heuristic 2 (H2) This heuristic determines the maximum saving for each route of exporters and selects the best routes serving importers:
Step$_0$ $s_2 = \emptyset$.
Step$_1$ For each column j, select the largest s_{ij}. Merge routes i and j, if any, and insert the new route into s_2.
Step$_2$ For each row i not involved in any merging, insert i into s_2.
Step$_3$ For each column j not involved in any merging, insert j into s_2.

Heuristic 3 (H3) This heuristic gives priority to routes of importers that can be merged with a low number of other routes.
Step$_0$ $s_3 = \emptyset$.
Step$_1$ Search for row i with the lowest value of m_i. If any, go to *Step*$_2$, otherwise go to *Step*$_3$.
Step$_2$ Select the largest s_{ij} for i, merge routes i and j and insert the new route into s_3. Go to *Step*$_1$.
Step$_3$ For each row i not involved in any merging, insert i into s_3.
Step$_4$ For each column j not involved in any merging, insert j into s_3.

Heuristic 4 (H4) This heuristic gives priority to routes of exporters that can be merged with a low number of other routes.
Step$_0$ $s_4 = \emptyset$.
Step$_1$ Search for column j with the lowest value of m_j. If any, go to *Step*$_2$, otherwise go to *Step*$_3$.
Step$_2$ Select the largest s_{ij} for j, merge routes i and j and insert the new route into s_4. Go to *Step*$_1$.
Step$_3$ For each row i not involved in any merging, insert i into s_4.
Step$_4$ For each column j not involved in any merging, insert j into s_4.

Heuristic 5 (H5) This heuristic gives priority to routes of both importers and exporters, that can be merged with a low number of other routes:

$Step_0$ $s_5 = \emptyset$.

$Step_1$ Search for row i with the lowest value of m_i and the column j with the lowest value of m_j. If $m_i <= m_j$, go to $Step_2$, otherwise go to $Step_3$. If no routes can be merged, go to $Step_4$.

$Step_2$ Select the largest s_{ij} for i, merge routes i and j and insert into s_5. Go to $Step_1$.

$Step_3$ Select the largest s_{ij} for j, merge routes i and j and insert into s_5. Go to $Step_1$.

$Step_4$ For each row i not involved in any merging, insert i into s_5.

$Step_5$ For each column j not involved in any merging, insert j into s_5.

Heuristic 6 (H6) This heuristic differs from the previous one in the selection of savings: we choose the closest saving to the average of all available savings, instead of the largest one:

$Step_0$ $s_6 = \emptyset$.

$Step_1$ Search for row i with the lowest value of m_i and the column j with the lowest value of m_j. If $m_i <= m_j$, go to $Step_2$, otherwise go to $Step_3$. If no routes can be merged, go to $Step_4$.

$Step_2$ Select the closest s_{ij} to $avrg_i$, merge routes i and j and insert into s_6. Go to $Step_1$.

$Step_3$ Select the closest s_{ij} to $avrg_j$, merge routes i and j and insert into s_6. Go to $Step_1$.

$Step_4$ For each row i not involved in any merging, insert i into s_6.

$Step_5$ For each column j not involved in any merging, insert j into s_6.

Heuristic 7 (H7) This heuristic merges the routes with the largest saving in the matrix:

$Step_0$ $s_7 = \emptyset$.

$Step_1$ Select for the largest s_{ij} in the saving matrix. If any, go to $Step_2$, otherwise go to $Step_3$.

$Step_2$ Merge routes i and j and insert the new route into s_7. Go to $Step_1$.

$Step_3$ For each row i not involved in any merging, insert i into s_7.

$Step_4$ For each column j not involved in any merging, insert j into s_7.

After the execution of all heuristics, we select the best one in terms of objective function.

5 Experimentation

In this section we test the previous heuristics on artificial and real instances. The real instances are provided by a shipping company operating in the port of Genoa (Italy).

Tests are performed on both artificial and real instances. Five classes of artificial instances have been generated:

- 10 customers;
- 20 customers;
- 30 customers;
- 40 customers;
- 50 customers.

In each class the coordinates of nodes are fixed. The instances of a class differ in the number of importers and exporters. The heuristics are implemented in the programming language C++. Tests are performed by Cplex 12.2 running on a four-CPU server 2.67 GHz

64 GB RAM. Since a major requirement of this problem is to determine solutions in a few minutes, Cplex is set to stop after 10 minutes. Computational results are indicated in Table 2, in which the following notation is used:

- $|I|$: Number of importers;
- $|E|$: Number of exporters;
- $H0, \ldots, H7$: Objective function returned by Heuristic 0,. . . , Heuristic 7;
- *t(s)*: The total execution time (in seconds) to solve the related instance, i.e. it represents how long it takes to run the first phase plus the time spent to run all heuristics $H0, \ldots,$ $H7$;
- *% Gap from CPLEX* : gap between the best heuristic and the best upper bound provided by CPLEX within 10 minutes;
- *Optimality gap*: Optimality gap between lower and upper bounds in Cplex after 10 minutes.

The string *n.s.* means that Cplex cannot provide a feasible solution within 10 minutes.

Table 2 shows that only one instance with 10 customers can be optimally solved. Cplex does not provide feasible solutions within 10 minutes for all instances with 40 and 50 customers, whereas all heuristics can solve these instances within 10 minutes. Generally speaking, the heuristic $H7$ is the most promising in terms of the objective function.

Real instances, which have about 40 customers, have no feasible solutions by Cplex within 10 minutes. In this case we compare the best heuristic to the carrier's decisions in terms of total travelled distances. Results on the real instances are shown in Table 1, in which the following notation is used:

- *Instances* The instance considered;
- $|I|$ Number of importers;
- $|E|$ Number of exporters;
- $|K|$ Number of trucks;
- *Carrier's decisions* The total travelled distance according to the carrier's decisions (km);
- *Decisions* The total travelled distance according to the best heuristic (km);
- *% Improvement*: gap between the best heuristic and the carrier's decisions;
- *Criterion* The heuristic(s) providing the best solution;

Table 1 The solutions of real instances.

Instances	$\lvert I\rvert$	$\lvert E\rvert$	$\lvert K\rvert$	Carrier's decisions	Best Heuristic		
					Decisions	% Improvement	Criterion
Instance 1	7	34	41	16503	**16196**	-1.90	7
Instance 2	10	28	31	13369	**11701**	-14.26	7
Instance 3	3	31	39	13702	**13602**	-0.74	1, 3, 7
Instance 4	6	34	36	13263	**12328**	-7.58	7
Instance 5	3	28	41	13180	**12869**	-2.42	1, 3, 7

Table 1 shows that in each instance the best heuristic always improves the carrier's decisions. The improvement seems to be particularly relevant when $|I|$ increases and becomes closer to $|E|$, due to the larger search space of feasible routes.

6 Conclusion

This paper has investigated a vehicle routing problem with some original characteristics, such as the opportunity to carry two containers per truck and the impossibility to separate trucks and containers during customer service. We have formulated an integer linear programming model for this problem. An exact algorithm was used to solve several artificial instances, but it was able to solve only instances with few customers. Several heuristics are proposed and tested on both artificial and real instances. According to our tests, the most promising heuristic in terms of objective function is $H7$, because high-quality routes are built from the beginning by the maximum saving.

The comparison with the carrier's decisions shows that the heuristics represent a promising instrument to improve its current decision-making process, because they yield significant savings in distances travelled by trucks.

Research is in progress to face problems with heterogeneous fleets of trucks, time windows and larger transportation capacities. New heuristics will be developed accounting for the specific characteristics of these problems.

Acknowledgements The authors want to thank professor Claudia Archetti, who made available her Tabu Search code for the Split Vehicle Routing Problem. The authors are also grateful to Antonio Musso, managing director of Grendi Trasporti Marittimi, who introduced to them the application that motivated this research.

References

1 P. Toth, D. Vigo. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, USA, 2002.
2 S. Anily. *The Vehicle Routing Problem with Delivery and Back-haul Options*. Naval Research Logistic Quarterly, 43, 415-434, 1996.
3 A. Mingozzi, S. Giorgi and R. Baldacci. *An Exact Method for the Vehicle Routing Problem with Backhauls*. Transportation Science, 33, 315-329, 1999.
4 P. Toth, D. Vigo. *An Exact Algorithm for the Vehicle Routing Problem with Backhauls*. Transportation Science, 31, 372-385, 1997.
5 P. Toth, D. Vigo. *A Heuristic Algorithm for the Symmetric and Asymmetric Vehicle Routing Problem with Backhauls*. European Journal of Operational Research, 113, 528-543, 1999.
6 R. Zhang, W.Y. Yun, I.K. Moon. *Modeling and optimization of a container drayage problem with resource constraints*. International Journal of Production Economics, 133, 351-359, 2011.
7 A. Imai, E. Nishimura, J. Current. *A Lagrangian relaxation-based heuristic for the vehicle routing with full container load*. European Journal of Operational Research, 176, 87-105, 2007.
8 A. Caris, G.K. Janssens. *A Local search heuristic for the pre- and end-haulage of intermodal container terminals*. Computers and Operations Research, 36, 2763-2772, 2009.
9 R. Namboothiri and A.L. Erera, *Planning local container drayage operations given a port access appointment system*. Transportation Research Part E: Logistics and Transportation Review, 44, 185-202, 2008.
10 C. Archetti, M. G. Speranza, A. Hertz. *A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem*. Transportation science, 40, 1, 64-73, 2006.
11 C. H. Papadimitriou. *Computational Complexity*. Addison Wesley Longman, 1993.

Table 2 The solutions of artificial instances.

		HEURISTICS										CPLEX
$\lvert I\rvert$	$\lvert E\rvert$	$H0$	$H1$	$H2$	$H3$	$H4$	$H5$	$H6$	$H7$	t(s)	% Gap from CPLEX	% Optimality Gap
						10 CLIENTS						
2	8	24170.26	20872.07	21392.98	**20672.07**	21392.98	22985.55	22985.55	**20672.07**	6	0.00	4.06
5	5	23674.27	21544.40	20916.21	22411.33	20066.19	20729.28	21131.93	**19960.83**	0	0.00	0.00
8	2	24067.78	23528.75	**21109.77**	23175.59	**21109.77**	21289.41	22863.04	**21109.77**	0	1.95	4.04
						20 CLIENTS						
2	18	46436.64	41566.04	43639.35	**40766.04**	43639.35	43319.77	43315.09	**40766.04**	0	-2.93	7.51
5	15	46204.46	36873.01	42743.03	**36233.01**	41845.95	40636.38	40378.63	**36233.01**	0	-4.48	7.91
10	10	46326.75	34751.04	36256.89	35277.56	36608.49	35976.40	38127.17	**33092.29**	0	4.20	4.40
15	5	46204.46	40021.52	37336.19	40222.93	38196.39	40480.64	42463.88	**37261.67**	0	-1.08	6.29
18	2	46204.46	44019.68	43050.79	44877.00	43050.79	44459.09	44753.25	**42975.11**	42	-2.41	7.02
						30 CLIENTS						
2	28	69121.47	63708.18	66750.74	**62728.18**	66730.74	66871.79	66682.33	**62728.18**	3	-6.38	9.52
5	25	69906.97	60019.63	66365.54	59423.33	65468.45	64809.86	66863.83	**59359.63**	2	-1.76	6.66
10	20	70704.03	54774.37	60574.18	54740.43	59587.76	58581.67	64047.55	**54129.24**	1	-5.63	14.27
15	15	70905.38	52092.61	50934.97	52687.38	53340.51	51172.63	60816.68	**49218.46**	1	-4.44	10.37
20	10	70905.38	59170.97	55004.74	60576.35	55574.39	58998.61	63808.75	**54757.12**	1	n.s	n.s.
25	5	69481.27	66695.13	61075.33	66554.36	61075.33	64538.63	65977.58	**61074.67**	2	-13.11	15.51
28	2	68982.30	69906.35	**67217.27**	68563.65	**67217.27**	68361.82	68361.82	**67217.27**	3	-5.17	9.93
						40 CLIENTS						
2	38	101302.11	95645.86	99695.34	**93985.86**	100062.51	96291.21	96952.05	**93985.86**	17	n.s	n.s.
5	35	101852.55	88764.09	91654.05	**88244.09**	92053.54	95498.95	99401.32	**88224.09**	11	n.s	n.s.
10	30	101582.68	76194.52	82999.48	75968.82	85442.09	86621.65	90626.66	**75117.98**	49	n.s	n.s.
15	25	101776.10	70920.95	76016.82	70347.44	75518.30	74980.22	89368.12	**68484.46**	3	n.s	n.s.
20	20	101895.27	76525.84	71514.12	76941.13	71709.14	75641.02	89044.40	**70795.53**	2	n.s	n.s.
25	15	102349.92	83733.24	76749.41	83267.22	76805.40	87505.43	95387.34	**76261.72**	5	n.s	n.s.
30	10	101286.43	91838.75	85772.27	92560.11	85888.37	91851.87	95243.18	**85506.14**	6	n.s	n.s.
35	5	101735.75	99057.56	**95072.84**	97828.28	**95072.84**	98095.70	100364.94	**95072.84**	11	n.s	n.s.
38	2	100722.47	100265.05	**97736.69**	99516.60	**97736.69**	100109.32	100109.32	**97736.69**	11	n.s	n.s.
						50 CLIENTS						
2	48	132095.20	125233.97	130617.35	**124513.97**	130761.45	130857.73	130959.03	**124513.97**	20	n.s	n.s.
5	45	131432.46	120617.96	126258.08	**120197.96**	125786.41	125447.20	128393.50	**120197.96**	23	n.s	n.s.
10	40	132098.19	113127.96	116088.33	**112687.96**	116348.43	120467.73	124113.76	**112687.96**	15	n.s	n.s.
15	35	133459.72	102304.67	113048.46	**100770.48**	113085.01	111324.61	121595.94	100837.89	8	n.s	n.s.
20	30	132673.52	94279.48	98966.22	92089.72	103620.29	99575.14	113407.89	**91542.11**	5	n.s	n.s.
25	25	131859.53	84807.36	91318.81	83254.21	92807.30	86028.26	112155.51	**82373.34**	5	n.s	n.s.
30	20	131907.73	89854.06	86239.26	88105.14	86865.49	90237.42	111915.47	**83923.10**	6	n.s	n.s.
35	15	132079.70	103480.43	96405.91	104061.17	96672.63	106294.63	117055.94	**95771.71**	8	n.s	n.s.
40	10	131450.60	114221.90	106093.40	116702.34	106920.76	115479.38	124521.98	**105976.92**	10	n.s	n.s.
45	5	132793.70	123151.73	113013.98	122472.64	114208.57	123625.22	124787.33	**112704.72**	19	n.s	n.s.
48	2	132892.88	127984.28	**122462.37**	126229.39	**122462.37**	128170.93	130264.79	**122462.37**	35	n.s	n.s.