Journal of Industrial Engineering International *Islamic Azad University, South Tehran Branch*
January 2008, Vol. 4, No. 6, 7-18

A goal programming model for vehicle routing problem with backhauls and soft time windows

M. Aghdaghi

M.Sc. Graduated Student, Dep. of Industrial Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran

F. Jolai*

Associate Professor, Dep. of Industrial Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran

Abstract

The vehicle routing problem with backhauls (VRPB) as an extension of the classical vehicle routing problem (VRP) attempts to define a set of routes which services both linehaul customers whom product are to be delivered and backhaul customers whom goods need to be collected. A primary objective for the problem usually is minimizing the total distribution cost. Most real-life problems have other objectives addition to this common primary objective. This paper describes a multi-objective model for VRPB with time windows (VRPBTW) and some new assumptions. We present a goal programming approach and a heuristic algorithm to solve the problem. Computational experiments are carried out and performance of developed methods is discussed.

Keywords: Vehicle routing problem; Backhaul; Soft time windows; Goal programming; Heuristic

1. Introduction

In a today's highly competitive world, companies face with the challenge of efficient resource managing because resource management has a direct influence on business performance and profitability. In order to manage the resources involved in supply chains and distribution systems more efficiently, the companies should make effective decisions on their resource management systems. Amongst the resource management problems, Vehicle Routing Problem (VRP) is a well known combinatorial optimization problem arising in transportation logistics that usually involves vehicle routing and scheduling in constrained environments. The capacitated vehicle routing problem (CVRP) as a typical vehicle routing problem is determining a set of routes which minimizes total travel time (cost or distance), originating and terminating at a central depot, for a fleet of vehicles which serve a set of customers with known demands or supplies. Each customer is served exactly once and, furthermore,all customers must be assigned to vehicles such that the restrictions on the capacity of vehicles and the duration of a route are met.

The fact that VRP is both of theoretical and practical interest, because of its real world applications, a considerable amount of research has been done on vehicle routing and scheduling problems by researchers. A good overview of recent researches has been done on VRP and its many variants proposed by Toth and Vigo [20].

In many practical distribution problems each customer is associated with a time interval, known as time window. When the service of customers must be started within the associated time interval, the problem is called Vehicle Routing Problem with Hard Time Windows (VRPHTW). In other cases of vehicle routing problem with time windows (VRPTW), both lower and upper bounds of the time window need not be met, but can be violated at a penalty. These are Vehicle Routing Problems with Soft Time Windows (VRPSTW). To see some surveys about VRPTW refer to [3,7,11,15,16].

Another type of VRP is VRP with backhauls (VRPB) where the customer set is partitioned into two subsets. The first subset contains the linehaul customers, each requiring a given quantity to be delivered. The second subset contains the backhaul cus-

* Corresponding author. E-mail: fjolai@ut.ac.ir

tomers, where a given quantity inbound product must be picked up. A common example in practical situation is the grocery industry, where supermarkets and shops are linehaul customers and grocery suppliers are backhaul customers. Some recent surveys about VRPB can be found in [2,18,19].

The real-life distribution and transportation problems have other objectives addition to minimizing the total travel cost (time or distance). Hence great attention has been paid to multiple objectives VRP in past years [3,7,12,17].

The goal programming (GP) approach is an important technique to model multi-objective problems and helps decision-makers to solve multi-objective decision making problems in finding a set of satisfying solutions. The purpose of GP is minimizing the deviations between the achievement of goals and their aspiration levels. Hong and Park [7] and Calvete et al. [3] have used goal programming approach to model the VRPTW.

In this paper we consider a multi-objective VRP with backhauls and soft time windows (VRPBSTW). A goal programming model is presented for VRPBSTW by using goal programming approach for VRPTW proposed by Calvete et al. [3] and the basis of the mathematical formulation proposed for VRPB by Toth and Vigo [18]. We also consider the limitation for total daily work time of driver as mentioned in [3]. The further distinctions of this paper as compared to [3] are:

- The vehicles are permitted to wait at customer location when they arrive to customer location earlier than lower bound of time windows.
- Primary objective consists of minimizing the total time that vehicles are waiting at customer location.
- Depot departure time of vehicles can be different from each others.
- The vehicles are allowed to remain at the depot.

The rest of the paper is organized as follows. In Section 2, the mathematical formulation of VRPBTW as a goal programming model is described. In Section 3, a heuristic algorithm to solve the model is proposed. In Section 4, first to validate proposed heuristic algorithm, we solve some small size problems exactly and compare the results of the heuristic algorithm with optimal solutions. Then we carry out some experiments using a set of data obtained from Solomon's instances. Finally in Section 5, some overall conclusions are described.

2. Model formulation

The problem is formulated on the basis of the existing mathematical formulation for VRPB, where each customer corresponds to a vertex. Let $G(V, A)$ be a directed graph with vertex set $V = L \bigcup B$. Subsets $L = \{1,2,...,n\}$ and $B = \{n+1,...,n+m\}$ correspond to linehaul and backhaul customer subsets, respectively. The nodes 0 and $n+m+1$ represent the depot, i.e., exiting depot and returning depot respectively. The arc set A denotes all possible connections between the nodes. Let us define $G'(V', A')$ be a directed graph obtained from G by defining $V' = V \bigcup \{0, n+m+1\}$ and $A' = A_1 \bigcup A_2$ where:

$$A_1 = \{(i, j) \in A : i \in L \bigcup \{0\}, j \in L \bigcup B \bigcup \{n+m+1\}\}$$

$$A_2 = \{(i, j) \in A : i \in B, j \in B \bigcup \{n+m+1\}\}. \quad (1)$$

In other words the arc set A' can be partitioned into two disjoint subsets. A_1 contains the arcs from the depot and linehaul vertices to all vertices except node 0. A_2 contains all the arcs from backhaul vertices to backhaul vertices and the depot. So no arc from backhaul to a linehaul vertex will be included in a feasible solution of the model. Given a vertex i, $\alpha(i)$ is defined as a set of vertices that are directly reachable from i i.e., j such that arc $(i, j) \in A'$. Analogously, $\beta(i)$ is defined as a set of vertices which i is directly reachable from them, i.e., j such that arc $(j, i) \in A'$. No arc terminates at node 0 and no arc originates at node $n+m+1$ and each feasible vehicle route corresponds to a path in G' that starts from node 0 and ends at node $n+m+1$. A nonnegative demand d_i, to be delivered or collected depending on its type, is associated with each customer i, and the depot is associated with a fictitious demand $d_0 = 0$. Let c_{ij} denotes the cost and t_{ij} denotes the travel time associated with going from node i to node j through arc (i, j). For virtual arc from node 0 to node $n+m+1$, $c_{ij} = t_{ij}$ =0. Each customer i is associated with a time interval $[a_i, b_i]$ corresponds to soft time windows. A set of k identical vehicles, each with capacity $U(k)$ is available at the depot. Let C_k defined as a fixed cost for using vehicle k. The vehicle must stop at the customer for S_i time instants,

S_i is the service time of customer i. Let us define $T_{\max}$ to denote driver's mandatory working hours per a day. It should be noted $T_{\max}$ does not include overtime. Also $\widehat{T}_{\max}$ is the total time that a driver is lawful to work per a day. In following model, M has been used as a very large positive number and m has been used as a very small negative number.

The purpose of the model is to find a set of separate trips to service all the customers such that following constraints are satisfied:

1. Each trip starts at vertex 0 and ends at vertex $n+m+1$.
2. Each customer vertex is visited by exactly one trip.
3. The total demands of the linehaul (backhaul) customers visited by a vehicle do not exceed the vehicle capacity.
4. The linehaul customers precede the backhaul customer, in other words whenever a routes serves both type of customers, all the linehaul customers must be served before the backhaul customers.

The goals of the model are:

I. Minimize the total cost to service the customers.
II. Minimize total time spent by vehicles for waiting at customers location to begin service.
III. Satisfy soft time window preferences of customers.
IV. Avoid labor works overtime or idle time.
V. Avoid underutilization of vehicles capacity.

Our mathematical programming formulation for VRPBTW consists of the following variables:

- Flow variable x_{ij}^k, $(i, j) \in A$, $k \in K$ that is equal to 1 if arc (i, j) is used by vehicle k and 0 otherwise.
- Time variable w_{ik}, $i \in v$, $k \in K$ that specifies start time of service of node i by vehicle k.
- Positive deviational variables $\xi^+, \mu_{ik}^+, \hat{\mu}_{ik}^+$, γ_k^+, δ_k^+ which ξ^+ corresponds to goal (I), μ_{ik}^+ and μ_{ik}^+ correspond to goal (III), γ_k^+ corresponds to goal (IV) and δ_k^+ corresponds to goal (II).
- Negative deviational variables $\mu_{ik}^-, \hat{\mu}_{ik}^-, \lambda_k^-$, $\hat{\lambda}_k^-, \gamma_k^-$ which $\mu_{ik}^-, \hat{\mu}_{ik}^-$ correspond to goal (III), λ_k^- and $\hat{\lambda}_k^-$ correspond to goal (V) and γ_k^- corresponds to goal (IV).
- Waiting time variable e_{ij}^k measures the time that vehicle k waits at location of customer j when trip through arc (i, j).

Given the above goals, the goal programming model for VRPBTW can be mathematically formulated as shown below:

$$\text{Min } \omega^{(1)}\xi^+ + \sum_{i\in V'}\sum_{k\in K}\omega_{ik}^{(2)}\mu_{ik}^- + \sum_{k\in K}\sum_{i\in V'}\omega_{ik}^{(3)}\mu_{ik}^+ + \sum_{k\in K}\omega_k^{(4)}\lambda_k^- + \sum_{k\in K}\omega_k^{(5)}\hat{\lambda}_k^- + \sum_{k\in K}\omega_k^{(6)}\gamma_k^+ + \sum_{k\in K}\omega_k^{(7)}\gamma_k^+ + \sum_{k\in K}\omega_k^{(8)}\delta_k^+ \quad (0)$$

Subject to:

$$\sum_{(i,j)\in A'} c_{ij}\sum_k x_{ij}^k + \sum_k C_k \sum_{j\in v} x_{0j}^k - \xi^+ = Z_0 \quad (1)$$

$$w_{ik} + \mu_{ik}^- - \mu_{ik}^+ = a_i \sum_{j\in\alpha(j)} x_{ij}^k \quad \forall i \in V', k \in K, \quad (2)$$

$$w_{ik} + \hat{\mu}_{ik}^- - \hat{\mu}_{ik}^+ = b_i \sum_{j\in\alpha(i)} x_{ij}^k \quad \forall i \in V', k \in K, \quad (3)$$

$$\sum_{i\in L} d_i \sum_{j\in\beta(i)} x_{ij}^k + \lambda_k^- = U(k) \quad \forall k \in K, \quad (4)$$

$$\sum_{i\in B} d_i \sum_{j\in\alpha(i)} x_{ij}^k + \hat{\lambda}_k^- = U(k) \quad \forall k \in K, \quad (5)$$

$$w_{n+m+1,k} - T_{\max}\sum_{j\in V} x_{0j}^k + \gamma_k^- - \gamma_k^+ = 0 \quad \forall k \in K, \quad (6)$$

$$\sum_{(i,j)\in A'} e_{ij}^k - \delta_k^+ = 0 \quad \forall k \in K, \quad (7)$$

$$\sum_{k\in K}\sum_{j\in\alpha(i)} x_{ij}^k = 1 \qquad \forall i\in V, \tag{8}$$

$$\sum_{j\in\beta(i)} x_{ji}^k - \sum_{j\in\alpha(i)} x_{ij}^k = 0 \qquad \forall i\in V, \forall k\in K, \tag{9}$$

$$\sum_{j\in\beta(0)} x_{0j}^k \le 1 \qquad \forall k\in K, \tag{10}$$

$$\sum_{i\in\beta(n+m+1)} x_{i,n+m+1}^k \le 1 \qquad \forall k\in K, \tag{11}$$

$$w_{ik} + s_i + t_{ij} - w_{jk} \le (1 - x_{ij}^k)M \qquad \forall (i,j)\in A', k\in K, \tag{12}$$

$$e_{ij}^k - N\ x_{ij}^k \le 0 \qquad \forall (i,j)\in A', k\in K, \tag{13}$$

$$e_{ij}^k - [w_{jk} - (w_{ik} + s_i + t_{ij})] + Nx_{ij}^k \le M \qquad \forall (i,j)\in A', k\in K, \tag{14}$$

$$e_{ij}^k - [w_{jk} - (w_{ik} + s_i + t_{ij})] + nx_{ij}^k \ge m \qquad \forall (i,j)\in A', k\in K, \tag{15}$$

$$w_{n+m+1,k} - \widehat{T}_{\max} \sum_{j\in V} x_{0j}^k \le 0 \qquad \forall k\in K, \tag{16}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall (i,j)\in A', k\in K, \tag{17}$$

$$w_{ik} \ge 0 \qquad \forall i\in V', k\in K, \tag{18}$$

$$e_{ij}^k \ge 0 \qquad \forall (i,j)\in A', k\in K. \tag{19}$$

The set of constraints (1) to (7) are goal programming constraints. Constraint (1) which corresponds to goal (I) allows us to assign a penalty to a deviation from a targeted total delivery cost Z_0. We suppose that Z_0 is a lower bound on the total operational cost. This lower bound can be calculated from typical shape of VRPB which obtained from the model when goal constraint is omitted.

Constraint (1) includes a positive deviational variable ξ^+ which is weighted by $\omega^{(1)}$ in the objective function (0). Constraints (2) and (3) are goal programming constraints on soft time window preferences of customers. Constraints (4) and (5) refer to goal (V). The variables λ_k^- and $\hat{\lambda}_k^-$ are negative deviation variables which indicate underutilization of vehicle capacity in linehaul and backhaul customer subsets, respectively.

Constraint (6) refers to goal (IV). Since $T_{\max}$ denotes the total daily time that a driver should be worked per a day, the deviation variables γ_k^- and γ_k^+ state the idle and overtime labor k, respectively. Constraint (7) allows us to formulate goal (II) where the expression $\sum e_{ij}^k$ measures the total time that each vehicle k waits at customer location to satisfy time window.

Constraints (8) and (9) impose that exactly one arc enters and leaves each vertex associated with a customer, respectively. Analogously, constraints (10) and (11) ensure that all routes leave from the depot (vertex 0) and return to it (vertex $n+m+1$). Constraint (12) guarantees schedule feasibility with respect to time considerations. Constraints (13) to (15) ensure that e_{ij}^k indicates waiting time of vehicle k at customer j location, correctly. Finally constraint (16) refers to upper bound associated with daily working time.

3. Solution approach

On account of large number of variables and constraints, exact solution methods of mathematical programming are not capable to solve large instances of the problem. Thus, we developed a heuristic algorithm which has exploited of two phase algorithms contexts. In two phase algorithms, first the algorithm clusters the vertices into feasible routes then actual route is been constructed in second phase.

Our heuristic attempts to generate feasible routes and find the best solution for the problem. So we must generate feasible routes and select the best one as a best solution. Because of large number of feasible routes, construction of all feasible routes does not take place in reasonable time. Owing to the fact that in real applications the goal (I), i.e. cost of system, is more important than other goals, we exploit an elementary Sweet algorithm which attributed to Gillett and Miller [20] to decompose customers set to some subsets in stage 1. Then we generate feasible routes in these subsets, separately, and select the best one in stage2. Finally in stage 3 we improve the solution of stage 2. A simple implementation of sweet method is as follows:

Assume each vertex i is represented by its polar coordinates (θ_i, ρ_i), where θ_i is the angle and ρ_i is the ray length. Feasible clusters are obtained from assigning the unrouted vertices having the smallest angle to vehicle k. To apply this method to VRPB and to improve its performance, we first rank the members of both linehaul and backhaul sets in increasing order of their ray length, ρ_i, separately, then partition the customers` distribution space into some zones (Figure 1). These zones are obtained by partitioning member of linehaul and backhaul sets to some subsets separately. Finally we sort the vertices of each zone in increasing order of θ_i.

Once customers are ranked, the algorithm starts to assign customers to the routes. Note that, we assumed that if a route serves both linehaul and backhaul customers, the linehaul customers precede the backhaul customers. In order to meet this assumption in our heuristic algorithm, we define parameter TL as a point to transfer assignment of the customers in linehaul set to backhaul set. Let f_k denotes the total time after vehicle k departures depot. Algorithm starts from set of linehaul customers and assign unrouted vertices until f_k becomes greater than *TL*. Once f_k becomes greater than *TL*, the algorithm continues the assigning process on backhauls set. To stop the assigning of vertices to a route, parameter *TB* is used as a limit for route duration. Algorithm assign vertices to route k as long as f_k is less than *TB*. The feasibility test is performed with respect to the constraints for vehicle capacity. To apply soft time window constraint, we define α and β, and impose constraints (2) and (3) to our algorithm as follow:

$$a_i(1-\alpha) \le w_i \le b_i(1+\beta).$$

It is apparent when $\alpha = \beta = 0$, the soft time window transforms to hard time window. The algorithm generates a numerous feasible solutions by changing the values of four parameters α, β, TL and TB to clusters the vertices with a favorable value of objective function.

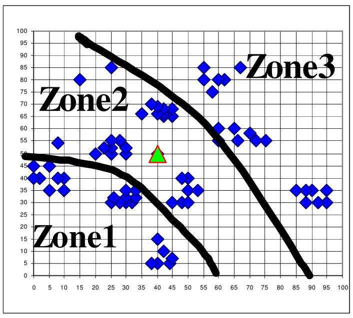

Figure1. Partition the customers' distribution space.

We improve sequence of vertices on route k by solving its corresponding TSP. Our heuristic algorithm can now be specified as follows:

3.1. Heuristic algorithm

3.1.1. Stage 1: Initialization

The purpose of the initialization stage is providing a set of vertices which are grouped and sorted properly and will be used as ''input data'' in the second stage. The data set is determined as below.

1) Create two sets of vertices which set1 contains linehaul customers and set 2 contains backhaul customers.
2) Sort the members of set1 and set 2 in increasing order of their ray length separately.
3) Subtract ray length of each node from last node and select some major differences in ray length as boundary of zones.
4) Sort customers of each zone in ascending order of the angel of polar coordinates (θ_i).

It should be noted that number of zones must be determined by decision maker and algorithm do not suggest an exact way to determine it. So we may face with some forms to place the nodes in the zones. There are two major strategies for zoning the nodes. In first one, there are small numbers of zones, each one has many nodes and in second, there are more zones with fewer nodes. Increasing number of zones improve solution time likely, the quality of solution may reduce, however.

3.1.2. Stage2: Clustering

This stage attempts to create feasible routes and find a set of feasible routes as the best one. Each route is a cluster of nodes, with favorable quality in objective value (objective value contains all goals of model as mentioned in equation (0) in the mode (l). The number of required vehicles is equal to the number of routes obtained in the clustering stage. The procedural steps of the Stage 2 can be described as follows:

Step1. Set $Z^* = +\infty$.

Step2. Set route index $k = 0$ and set the values of TL, TB, α and β in a manner explained below.

Step3. Set $k = k + 1$.

Step4. Select first unrouted customer from linehaul set and assign to route k such that the total customers' demand visited by route k do not exceed the vehicle capacity and w_{ik} that specifies the start time of service of node i, satisfy constraint below:
$a_i(1-\alpha) \le w_i \le b_i(1+\beta)$

Step5. If f_k is not greater than TL and there is any unrouted customer in linehaul set go to Step 4.

Step6. While f_k is less than TB do Step4 on the set of backhaul nodes.

Step7. The procedure is moved back to Step3 and is repeated until all vertices are assaigned to a route.

Step8. Calculate objective function value (Z). If this value is less than Z^*, replace the value of Z^* with Z, ($Z^* = Z$).

Step9. Move back to step2 and continue till stop condition of stage2 occurs (the stop condition is explained below).

Algorithm calculates objective function for variant combinations of parameters TL, TB, α and β values and refers the best value has been found. Accordingly, algorithm starts from a minimum value for TL, TB, α and β, ($TL_1 = TL_{\min}$, $TB_1 = TB_{\min}$, $\alpha_1 = \alpha_{\min}$ and $\beta_1 = \beta_{\min}$) and increases one of the parameters after each iteration r, in Step 2. Stop condition which cited at Step9, occurs when TL, TB, α and β get to the utmost values($TL_{\max}$, $TB_{\max}$, $\alpha_{\max}$ and $\beta_{\max}$). The utmost values of all parameters must be determined by decision maker. The value of TL has effect on number of linehaul and backhaul customers which has placed in a route. In fact TL enables algorithm to generate variant sets of routes with different number of linehaul and backhaul nodes, so the maximum value of TL can be set equal to $TB_{\max}$. The parameter TB refers to constraint (6) so the value of $TB_{\max}$ depends to acceptable deviation from the total daily time that a driver should be working ($T_{\max}$). When $TB_{\max}$ has been set under value of $T_{\max}$, idle time of labor is reasonable and when $TB_{\max}$ is greater than $T_{\max}$, there is likely overtime. In like manner for $\alpha_{\max}$ and $\beta_{\max}$ the values depend to acceptable deviation from soft time window.

Owing to this, algorithm enables decision maker to generate variant solutions through the change of the maximum values of parameters to evaluate impact of weights of the goals on objective value. Since the run time of algorithm is reasonable, it can be helped to decide about the weight of goals in other solving approach of multi-objective VRPs.

3.1.3. Stage3: Sequencing

Once the customers are clustered into groups in the Stage2, a set of feasible routes are constructed. We can improve the solution of Stage 2 easily since the GP model for each group becomes much simpler than the described model for the VRPBTW in Section 2. We determine the best sequence of the vertices in a cluster with applying a GP model to corresponding TSP.

4. Computational results

In this section we present the experimental results were achieved by the two sets of problems. In order to evaluate performance of the presented approximation algorithm, we first attempt to solve some small-size testing problems. Then we used a set of data were obtained from Solomon's instances [16]. In all experiments we suppose following assumptions.

- The distances, travel times and travel costs between nodes are the same ($c_{ij} = t_{ij}$ = distance between node i and node j).
- Fixed cost of vehicle usage is 200.
- The weights of goals were set as follows: $\omega^{(1)} = 0.5$, $\omega^{(2)} = 0.05$, $\omega^{(3)} = 0.05$, $\omega^{(4)} = 0.02$, $\omega^{(5)} = 0.02$, $\omega^{(6)} = 0.05$, $\omega^{(7)} = 0.15$ and $\omega^{(8)} = 0.18$.

The heuristic algorithm discussed in this paper was coded in Matlab Version 6.5. In order to find optimal solution, the testing problems were solved by using Lingo 8.0. All computing processes executed on a PC Pentium iii, 800MHz, 512 MB Ram, under Windows XP.

4.1. Testing and Evaluation

To test and evaluate the performance of the proposed algorithm, we generate three different small-size problems TC, TR and TRC by choosing first 10 nodes of Solomon's C101, R101, and RC101 problems, respectively [16]. In all test problems, we suppose that Nodes 1-6 are linehauls and nodes 7-9 are backhauls. Also the capacity of each vehicle is 45 units.

Table 1 summarizes the results of optimal solution of the test problems. The results of stage2 and stage3 of the heuristic are given in Table 2. It should be noted that the columns %r of the Table 2 indicates the percentage ratio of heuristic solution respect to optimal solution.

The results which have been reported in Table1 and Table 2 show that the heuristic results are close to the optimal results. The Average percentage ratio of the heuristic algorithm solution with respect to the optimal solution value is 92.43%. The average time taken to solve the testing problems is less than 35 seconds.

4.2. Problems based on Solomon's instances

We have constructed problem instances MC1, MR1, and MRC1 based on Solomon's C1, R1, and RC1 data sets, respectively by randomly choosing 53% of the 100 customers to be linehaul customers. Notice that the customer type (linehaul or backhaul) in MC1, MR1, and MRC1 is the same. For example if node 2 is a linehaul customer at MC1, it is a linehaul at MR1 and MRC1 too. In order to analyze the impact of the form of zoning, we compare the results of two different forms of stage1 which have different number of zones. Table 3 and Table 5 show the characteristics of two forms. In Tables 4 and 6 the results obtained from two forms, are reported.

Figure2 shows the comparison of the solutions obtained by applying the proposed algorithm on data sets with two forms.

It is apparent from the results that reduction in number of zones has different impact on MC1, MR1 and MRC1. The zoning applied in form2 performs better than form1 on data sets MC1 and MRC1, impact of changing on MC1 is not considerable however (Figure2-a and c). The results show that on data sets MRC1, on average, the second form improves the solution 6.82%. On set MR1, the second form, worsen the solution. The results show that on data sets MR1 we can see 5.44% raise in objective value when use form2 (Figure2-b).

The large amount of parameters α and β show that satisfying soft time window constraint (goal III) has poor importance for decision maker rather than others goals.

We also solve MR101 for different value of parameters $\alpha_{\max}$ and $\beta_{\max}$ to evaluate the capability of algorithm to generate numerous solutions. The results of this experiment are shown in Table 7 and Figure 3. It should be noted that parameters $\alpha_{\max}$ and $\beta_{\max}$ were increased 0.02 after each iteration. These results help decision maker to analyze behavior of objective value against acceptable deviation from time window. As we can see, by accepting 10% deviation from time window in second row, the value of objective value is reduced by approximately 11%.

5. Conclusion

In this paper, we constructed a linear goal programming model for multi-objective vehicle routing problem with backhaul and soft time window (VRPBTW). The high complexity of the VRPBTW requires heuristic solution strategies for most real-life instances. Hence we developed a simple heuristic to solve the model. The heuristic algorithm consists of the three stages. In the first initialization stage, procedure partitions the set of customers into various groups called zones. We confirmed through the computational experiment that results of stage2 are sensitive to number of zones which determined at stage1. To evaluate the efficiency of our algorithm, we solved some small size instances and compare the results with optimal solution. Also it has been shown that the proposed algorithm could generate a numerous solutions in very short time and enables decision maker to set weight of goals better. As an extra advantage of the algorithm, we discussed it can be used as a base to decide about the weight of goals in other solution approach of multi-objective problems.

Table 1. Optimal solution of the test problems.

Instances	Z^*	CPU time (in second)	# of routes
TC	512.05	3139	2
TR	513.11	784	2
TRC	724.65	134	3

Table 2. The heuristic results of the test problems.

Instances	Stage 2			Stage 3			# of routes
	Z^*	% r	CPU time (in second)	Z^*	% r	CPU time (in second)	
TC	533.15	95.87%	28	525.05	97.46%	2	2
TR	651.01	73.12%	28	586.36	85.72%	4	2
TRC	774.4	94.13%	30	774.4	94.13%	3	3

Table 3. Form 1 of stage 1.

Problem type	# of Linehauls	# of Backhauls	# of Linehaul 's zones	# of Backhaul 's zones
MC1	57	43	4	4
MR1	57	43	4	4
MRC1	57	43	5	5

Table 4. Results of the heuristic on form 1.

Instance	α	β	TL	TB	# of routes	Z^*
MC101	0.9	1.8	800	1100	10	2999.5
MC102	0.9	1.8	800	1050	10	2664.3
MC103	1	1.8	900	1150	9	2353.7
MC104	1	1.4	700	1150	9	1977.8
MC105	1	1.2	800	1050	10	2824.3
MC106	1	1.2	1000	1100	10	2763.7
MC107	1	1.1	950	1050	10	2717.7
MC108	1	0.8	900	1100	10	2573.8
MC109	0.9	0.7	1000	1100	10	2994.4
MR101	0.9	2.4	200	270	9	2178.9
MR102	0.9	2.3	200	300	8	2010.4
MR103	0.9	2.4	200	340	7	1938.2
MR104	0.7	2.6	220	360	6	1792
MR105	0.9	2.2	180	330	8	2098.9
MR106	0.9	1.8	200	300	8	1984.2
MR107	0.8	2.6	160	290	8	1926.3
MR108	0.8	2.4	230	370	6	1826
MR109	0.9	2	160	280	8	1963.8
MR110	0.9	1.9	160	280	8	1904
MR111	0.8	1.7	160	280	8	1901.6
MR112	0.9	0.9	160	280	8	1833.7
MRC101	0.9	2.6	200	340	8	2437.5
MRC102	0.9	2.6	200	380	7	2310.5
MRC103	0.8	2.8	240	380	7	2237.9
MRC104	0.7	2.2	240	380	7	2158.3
MRC105	0.8	2.9	240	380	7	2325.4
MRC106	0.9	3	220	400	7	2338.8
MRC107	0.6	2.2	240	360	7	2206.9
MRC108	0.5	1.8	240	360	7	2155

Table 5. Form 2 of stage 1.

Problem type	# of Linehauls	# of Backhauls	# of Linehaul 's zones	# of Backhaul 's zones
MC1	57	43	3	4
MR1	57	43	3	2
MRC1	57	43	4	4

Table 6. Results of the heuristic on form 2.

Instance	α	β	TL	TB	# of routes	Z^*
MC101	1	1.8	850	1050	10	2991.2
MC102	1	1.2	900	1050	10	2628.5
MC103	1	1.8	900	1150	9	2361.7
MC104	1	1.5	700	1150	9	2026.3
MC105	1	1.5	850	1100	10	2823.3
MC106	1	1.2	850	1100	10	2779.5
MC107	1	1.1	850	1050	10	2703.1
MC108	1	0.7	850	1100	10	2551.9
MC109	1	0.4	850	1050	10	2438.6
MR101	1	2.7	220	280	9	2275.1
MR102	0.9	2.6	220	360	7	2112.7
MR103	0.9	2	220	390	7	2083.2
MR104	0.7	2.6	200	340	7	1937.7
MR105	0.9	2.8	230	350	7	2192.8
MR106	0.9	2.4	220	360	7	2084.9
MR107	0.9	1.3	220	370	7	2020.9
MR108	0.7	2.2	200	330	7	1924.2
MR109	1	2.4	220	380	6	2043.9
MR110	0.8	2.8	230	390	6	2013.2
MR111	0.7	1.4	220	310	8	2001.4
MR112	0.6	1.2	200	330	7	1929.3
MRC101	1	2.9	140	260	9	2241.9
MRC102	1	2.9	140	260	9	2158.4
MRC103	0.7	2.9	140	260	9	2087.6
MRC104	0.7	2.1	140	260	9	2024.8
MRC105	1	2.7	140	260	9	2170.9
MRC106	1	2.4	140	260	9	2153.3
MRC107	1	2.2	140	260	9	2096.5
MRC108	1	0.8	140	260	9	1993.4

Table 7. MR101 soultions for different value of parameters.

$TB_{\max}$	$\alpha_{\max}$	$\beta_{\max}$	TB	α	β	Z^*	CPU time (in second)
200	0	0	180	0	0	5340.6	1
200	0.1	0.1	160	0.1	0.1	4761.6	2
200	0.2	0.2	190	0.2	0.18	4143.1	5
200	0.3	0.3	120	0.18	0.3	3895.5	8
200	0.4	0.4	130	0.22	0.4	3475.2	15
200	0.5	0.5	140	0.48	0.48	3282.4	24
200	0.6	0.6	130	0.6	0.42	3021.2	35
200	0.7	0.7	200	0.62	0.68	2836.5	48
200	0.8	0.8	200	0.72	0.72	2570.2	59
200	0.9	0.9	200	0.72	0.72	2570.2	78
200	1	1	200	0.86	1	2532.7	94

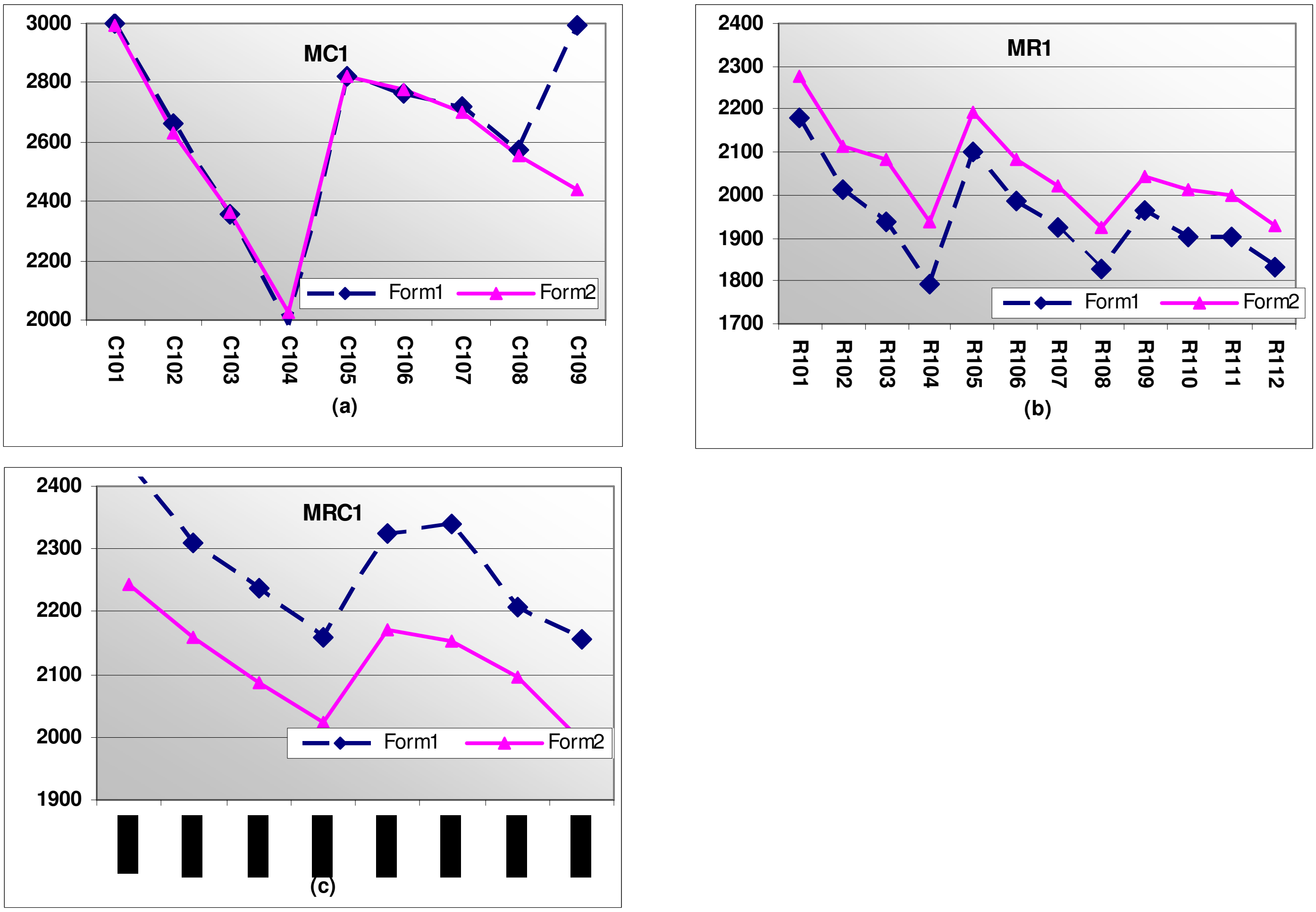

Figure 2. The impact of different zoning methods with MC1, MR1 and MRC1 instances.

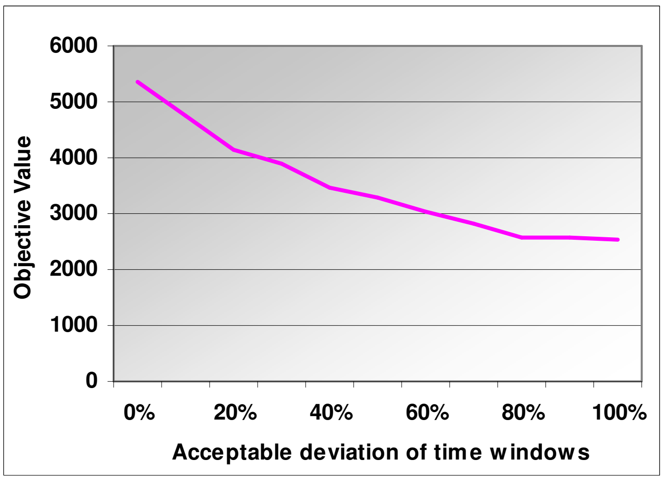

Figure 3. The effect of acceptable deviation from time window on objective function value.

References

[1] Agarwal, Y., Mathur, K. and Salkin, H. M., 1989, A set partitioning based exact algorithm for the vehicle routing problem. *Networks*, 19 731-749.

[2] Branda, J., 2006, A new tabu search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, 173, 540-555.

[3] Calvete, H. I., Gale, C., Oliveros, M. J. and Valverde, B. S., 2007, A goal programming approach to vehicle routing problems with soft time windows. *European Journal of Operational Research*, 177, 1720-1733.

[4] Dentcheval, D., Prekopa, A. and Ruszczynski, A., 2002, Bounds for probabilistic integer programming problems. *Discrete Applied Mathematics*, 124, 55-65.

[5] Glaab, H., 2002, A new variant of a vehicle routing problem: Lower and upper bounds. *European Journal of Operational Research*, 139, 557-577.

[6] Gupta, D. K., 2003, A new algorithm to solve vehicle routing problems (VRPS). *International Journal of Computer Mathematics*, 80, 267-274.

[7] Hong, S. C. and Park, Y. B., 1999, A heuristic for bi-objective vehicle routing with time window constraints. *International Journal of Production Economics*, 62, 249-258.

[8] Irnich, S., Funkeb, B. and Grünert, T., 2006, Sequential search and its application to vehicle-routing problems. *Computers and Operations Research*, 33, 2405-2429.

[9] Junger, M. and Thiene, S., 1998, Introduction to ABACUS-A branch and cut system. *Operations Research Letters*, 22, 83-95.

[10] Katoh, N. and Yano, T., 2006, An approximation algorithm for the pickup and delivery vehicle routing problem on trees. *Discrete Applied Mathematics*, 154, 2335-2349.

[11] Kallehauge, B., Larsen, J. and Madsen, O.B.J., 2006, Lagrangian duality applied to the vehicle routing problem with time windows. *Computers and Operations Research*, 33, 1464-1487.

[12] Lacomme, P., Prins, C. and Sevaux, M., 2006, A genetic algorithm for a bi-objective capacitated arc routing problem. *Computers and Operations Research*, 33, 3473-3493.

[13] Laporte, G., Mercure, H. and Nobert, Y., 1986, An exact algorithm for the asymmetrical capacitated vehicle routing problem. *Networks,* 16, 33-46.

[14] Laporte, G. and Nobert, Y., 1987, Exact algorithms for the vehicle routing problem. *Annals of Discrete Mathematics,* 31, 147-184.

[15] OmbukI, B., Ross, B. J. and Hanshar, F., 2006, Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. *Applied Intelligence,* 24, 17-30.

[16] Solomon, M., 1987, Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research,* 35, 254-265.

[17] Tan, K. C., Chew, Y. H. and Lee, L. H., 2006, A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research*, 172, 855-885.

[18] Toth, P. and Vigo, D., 1997, An exact algorithm for the vehicle routing problem with backhauls. *Transportation science*, 31, 372-385.

[19] Toth, P. and Vigo, D., 1999, A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. *European Journal of Operational Research*, 113, 528–543.

[20] Toth, P. and Vigo, D., 2002, *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, 9(1).