Diversification of Neighbourhood via Constraint-based Local Search and Its Application to VRPTW

Hoong Chuin Lau[(1)], Yun Fong Lim[(2)]
[(1)]School of Computing, National University of Singapore,
3 Science Drive 2, Singapore 117543.
E-mail: lauhc@comp.nus.edu.sg
[(2)] School of Industrial and Systems Engineering, Georgia Institute of Technology
21, Atlanta, GA 30332-0205.
E-mail: yflim@isye.gatech.edu

Abstract. In this paper, we present a simple constraint-based diversification technique to enhance local search in solving optimization problems. This technique differs from existing diversification methods in that it is generic so long as the given optimization problem can be expressed as a constraint satisfaction problem. In particular, we will focus our discussion on the Vehicle Routing Problem with Time Windows, an important combinatorial optimization problem in logistics management. Experimental results show that this technique is capable of assisting tabu search to escape from local optimality. Thus our approach provides opportunities for improving the quality of the solutions.

Keywords: Local Search, Diversification, Constraint Satisfaction Problem, Vehicle Routing Problem with Time Windows.

1. Introduction

Local search techniques have been shown to be quite efficient in solving various large-scale combinatorial optimization problems. Some well-known approaches in local search are simulated annealing (Kirkpatrick et al., 1983), genetic algorithms (Holland, 1975), tabu search (Glover, 1997), etc. In contrast with exact approaches, local searches are efficient in the sense that satisfactory results can usually be obtained with significantly less computational effort. However, a common weakness of these approaches is that they may sometimes be trapped in a local optimum with poor quality. This barrier may not be easy to overcome even with fastidious tuning of various parameters in respective methods.

One of the approaches to overcome this bottleneck is to diversify the search, where diversification will drive the search to explore new regions in the search space such that the objective value can be further optimized. An example is the work of Rochat and Taillard (Rochat and Taillard, 1995) in which of diversification is applied in a tabu search setting to solve the Vehicle Routing Problem with Time Windows (VRPTW).

Instead of manipulating upon the various move operations in local search to create a diversified search neighbourhood, we propose a very different technique. We model the given optimization problem as a constraint satisfaction problem which is then solved by a generic local search algorithm, also proposed in this paper. In doing so, diversification becomes a straight-forward process of modelling, rather than a process of inventing algorithms for specific problems.

Without loss of generality, we will assume that the given optimization problem is a cost minimization problem, and that this problem can be modelled as a constraint satisfaction problem P. Given a locally optimal solution (derived from an arbitrary problem-specific local search algorithm), we feed this solution as the initial solution for our constraint-based local search algorithm which operates in two stages. In the first stage, some of the constraints of P are relaxed hence allowing the search to *further* minimize the cost by exploring into infeasible regions. Restoring feasibility is the second stage of the constraint-based local search. In this stage, all relaxed constraints are resumed. The search continues until a new feasible solution is obtained. The quality of this new solution (if found) is normally worse than the one initially returned by the two-phase approach. However, the good news is that this provides opportunities to enter a possibly more promising region, to which the standard local search is applied again. The overall process will be repeated up to a pre-set maximum number of times.

In this paper, we will discuss our generic constraint-based diversification technique and apply it to solve VRPTW. We believe this approach can be readily applied to other optimization problems. In VRPTW, a set of routes must be constructed so that a set of geographically-distributed customers can be serviced by a fleet of vehicles, subject to vehicle capacity and time window constraints. The objective is to first minimize the number of vehicles used and then the total travel distance.

In terms of related research, Rochat and Taillard (Rochat and Taillard, 1995) introduced a probabilistic diversification and intensification technique that overcomes the problem of local optimality. They applied this technique on tabu search approaches tailored for VRPTW. They showed that the result of the first level tabu search may be significantly improved with this technique. Several works have been carried out advocating the hybrid use of constraint programming and local search. For example, Pesant and Gendreau (Pesant and Gendreau, 1999) applied constraint programming to evaluate the local neighborhood to find the best local moves. There is also constrained-directed local search proposed in (Shaw, 1998; Backer et al., 2000; Kilby et al., 2000). In (Shaw, 1998), for example, the author presented a method called Large Neighbourhood Search (LNS) for VRP in which a part of a given solution is extracted and then reinserted into the partial solution using a quasi-complete search process. If the reinsertion procedure generated a better solution, then the solution is kept. This process is repeated until certain stopping criterion is met. The result produced with this technique is competitive with other meta-heuristic approaches.

In this paper, we illustrate that our constraint-based diversification technique can assist tabu search for solving VRPTW to escape from local optimality. A local optimal solution is first generated by tabu search. Using a strategy similar to LNS, a part of this solution is extracted and passed to the constraint-based local search. In the first stage, the time window and capacity constraints are relaxed hence allowing the search to further minimize the cost by exploring into infeasible regions. In the second stage all the relaxed constraints are resumed. The search continues until a new feasible solution is obtained. This feasible solution (if found) is then passed back to the tabu search and the whole process is repeated up to a pre-set maximum number of times.

This paper is organized as follows. Section 2 illustrates the problem definition of VRPTW and the well-known two-phase approach involving a construction technique to build an initial solution and a tabu search to improve the solution. In section 3, we present our diversification technique where VRPTW is modelled as a linear constraint satisfaction problem and then solved by a local search method. We demonstrate how this technique can cooperate with a two-phase approach, resulting in an improved algorithm. Finally we present the experimental results, over the well-known Solomon's benchmark problems (, 1983), together with some concluding remarks.

2. Problem Definition and Well-Known Approach

2.1. Vehicle Routing Problem with Time Windows

We consider the standard VRPTW, in which a fleet of identical vehicles makes deliveries of a single commodity to customers from a central depot. Each customer has its own demand and must be served within a specified time window by one and only one of the vehicles. A *route* consists of one or more than one customers served by the same vehicle that starts and ends its deliveries at the depot. The load of each vehicle cannot exceed its loading capacity and every vehicle can at most serve one route.

The problem has the following inputs:

K = number of vehicles
N = number of customers
C = capacity of each vehicle
W_i = demand of customer i
S_i = service duration for customer i
a_i = earliest possible start time of delivery to customer i
b_i = latest possible start time of delivery to customer i
D_{ij} = travel distance from customer i to customer j
T_{ij} = travel time from customer i to customer j

Note that S_i represents the duration needed to serve customer i. Hence, if customer i is served starting from t and assume that the following customer to be served is j, then the earliest time that j will be served is $t + S_i + T_{ij}$.

The first objective is to find a feasible solution that minimizes the number of vehicles needed and the second objective is to minimize the total distance travelled by the vehicles.

2.2. Two-Phase Approach

It has been shown that a successful methodology for solving VRPTW is to construct an initial set of feasible routes that serve all the customers (construction phase) and subsequently improve the existing solution (improvement phase), namely a two-phase approach. Various techniques have been investigated for both phases (Kilby et al., 2000). In this paper, we use a greedy insertion algorithm to build the initial solution and apply tabu search associated with a set of traditional vehicle routing move operators to improve the solution. Details are given as follows.

2.2.1. *Construction Phase*

In the first phase, a least cost greedy insertion algorithm is used for generating an initial solution. This is an iterative algorithm with the following steps:

1. let $R := \{\mathrm{r}_k \mid k = 1, ..., K\}$ be the current set of routes and $U := \{u_i \mid i = 1, ..., N\}$ be the set of unassigned customers

2. for each customer u_i in U, compute the cost (the resultant increment in travel distance) of inserting it into every feasible insertion position p_j in every route r_k in R;

3. perform an insertion with the minimum cost (let this customer be $u_{i'}$);

4. remove $u_{i'}$ from U;

5. repeat steps 2 to 4 until $U = \phi$.

The initial feasible solution is then improved in the improvement phase of the method.

2.2.2. *Improvement Phase*
We perform tabu search to improve the solution given by the insertion algorithm described above. In this paper, the steepest decent search is applied. A *move* in this approach corresponds to one of the traditional vehicle routing move operations. Here, we only focus on two types of move operations, namely *exchange* (Taillard, 1993) and *relocate* (Savelsberg, 1988).

An exchange operation swaps two customers from two different existing routes, whereas in a relocate operation, a customer is removed from the original route and is reinserted into an existing route (it could be the same route but at different position). A move is considered feasible if the corresponding operation does not violate any requirement (for instance, time window and vehicle capacity constraints, etc) in VRPTW. Hence the neighbourhood of the current solution is defined by all the feasible moves. In each iteration of the steepest decent approach, the feasible move that gives the best improvement (or least deterioration) of the cost is selected.

To avoid the search from revisiting the same solution in near future, the tabu search mechanism is introduced. A *tabu list* that records the n previous moves performed is maintained in memory. The number n is usually called the *tabu length*. A move is considered tabu if it is in the tabu list. Moreover, a move is aspired if the resultant cost is lower than the cost of the best solution encountered.

If the best move selected by the steepest decent approach is tabu and not aspired, then the next best move in the neighbourhood of the current solution is considered; otherwise, make the selected move. The improvement process in this phase continues until a pre-set maximum number of iterations $maxIter$ is reached or a number of non-improving iterations $maxStag$ have been performed. The key steps of the improvement phase are given as follows:

1. let the current solution x be the feasible solution generated in the construction phase, and set $z^* = \infty$;

2. choose the best move *bestMove* from the neighbourhood of the current solution;

3. if *bestMove* is tabu and not aspired, let *bestMove* be the next best move in the neighbourhood and go to step 3, otherwise accept *bestMove* and update the solution x and cost $z(x)$;

4. if $z(x) < z^*$, then $x^* = x$ and $z^* = z(x)$;

5. repeat steps 2 to 4 until the number of iterations equals $maxIter$ or $maxStag$ non-improving iterations have been performed;

6. output x^* and z^*.

3. Proposed Approach

In this section, we propose a constraint-based local search method, and demonstrate how this method may serve as a diversifier for the two-phase approach discussed in the previous section. We then propose a co-operative framework for the two-phase approach to work in tandem with the diversifier.

3.1. Constraint-based Local Search Method

We model VRPTW as a linear constraint satisfaction problem and introduce an efficient local search method for solving this class of constraint satisfaction problems.

3.1.1. *Modelling VRPTW as Constraint Satisfaction Problem*

In our model, the constraints of VRPTW are represented as a set of *hard* linear constraints, whereas the optimization objective is represented by a set of *soft* constraints. A feasible solution in this formulation is an assignment to all variables in the model that satisfies all constraints in the hard category, whereas an optimal solution is a feasible solution with the minimum total degree of violation contributed by the soft constraints.

The degree of violation of a constraint $X \geq Y$ is defined as $max(0, (Y - X))$. Our goal is to obtain an assignment to all variablesthat minimizes the total degree of violation of the soft constraints while satisfying all the hard constraints.

The model is based on the "previous" and "next" nodes representation, where every node references the nodes that come before and after it. There are N nodes representing N customers. In addition, there is an additional node for each vehicle representing the starting and ending point (depot) of its route. Thus we have $N + K$ nodes (where the first N nodes correspond to the N customers) and K vehicles. The variables in this model are:

$$\begin{aligned} x_{ij} &= 1, \text{ if node } j \text{ is visited immediately after node } i; \\ &= 0, \text{ otherwise}; \end{aligned} \tag{1}$$

where $i,\ j = 1,\ ...,\ N + K$. Note that $x_{ii} = 0$, for $i = 1,\ ...,\ N$.

$$\begin{aligned} v_{ik} &= 1, \text{ if node } i \text{ is visited by vehicle } k; \\ &= 0, \text{ otherwise}; \end{aligned} \tag{2}$$

where $i = 1,\ ...,\ N + K$ and $k = 1,\ ...,\ K$. Note that $\forall i > N$: $v_{ik} = 1$, if $i - N = k$; otherwise $v_{ik} = 0$. This means each additional node is mapped onto one vehicle.

$$t_i \in [a_i,\ b_i]; \tag{3}$$

where t_i is the start time of delivery to node i for all $i = 1,\ ...,\ N$, and $[a_i,\ b_i]$ represents the delivery time window of customer i.

The hard constraints are given as follows. Let $min(E)$ and $max(E)$ denote the minimum (resp. maximum) possible value of the expression E over all possible values of variables in the expression. For example, $min(vik - v_{jk})$ means the minimum value of $v_{ik} - v_{jk}$ over all possible indices i, j and k. These min-max functions serve as the 'big M' which converts an implicational constraint into a linear constraint.

1. *One-successor constraint* - each node must have exactly one successor:

$$\sum_{j=1}^{N+K} x_{ij} = 1; \tag{4}$$

   $\forall i = 1,\ ...,\ N + K$ [1].

2. *One-predecessor constraint* - each node must have exactly one predecessor:

$$\sum_{i=1}^{N+K} x_{ij} = 1; \tag{5}$$

   $\forall j = 1,\ ...,\ N + K$.

[1] Note that each equality can be represented by two ("$\geq$" and "$\leq$") inequalities.

3. *One-vehicle constraint* - each node can only be visited by one vehicle:

$$\sum_{k=1}^{K} v_{ik} \;=\; 1; \tag{6}$$

$\forall i = 1,\ ...,\ N + K.$

4. *Vehicle consistency constraint* - if $x_{ij} = 1$, then $v_{ik} = v_{jk}$:

$$\begin{aligned} v_{ik} - v_{jk} &\geq M_1 \times (1 - x_{ij}); \\ v_{ik} - v_{jk} &\leq M_2 \times (1 - x_{ij}); \end{aligned} \tag{7}$$

$\forall i,\ j = 1,\ ...,\ N + K,\ i \neq j$, and $\forall k = 1,\ ...,\ K$ (where $M_1 = min(v_{ik} - v_{jk})$ and $M_2 = max(v_{ik} - v_{jk})$).

5. *Vehicle capacity constraint* - the load of each vehicle k cannot exceed the capacity C of the vehicle:

$$\sum_{i=1}^{N} v_{ik} W_i \;\leq\; C; \tag{8}$$

$\forall k = 1,\ ...,\ K.$

6. *Service time constraint* - if $x_{ij} = 1$, then $t_i + S_i + T_{ij} \leq t_j$:

$$t_i + S_i + T_{ij} - t_j \;\leq\; M_3 \times (1 - x_{ij}); \tag{9}$$

$\forall i,\ j = 1,\ ...,\ N$, and $i \neq j$ (where $M_3 = max(t_i + S_i + T_{ij} - t_j)$).

The objective of VRPTW, namely minimizing the total number of vehicles and the travel distance are modelled as the following set of soft constraints:

Number of vehicles constraints -
$\forall k = 1,\ ...,\ K$ s.t. $\sum_{i=1}^{N} v_{ik} < \theta$:

$$\sum_{i=1}^{N} v_{ik} \;=\; 0; \tag{10}$$

$\forall k = 1,\ ...,\ K$ s.t. $\sum_{i=1}^{N} v_{ik} \geq \theta$:

$$\sum_{i=1}^{N} v_{ik} \;\geq\; \theta; \tag{11}$$

Travel distance constraint -

$$\sum_{j=1}^{N+K} x_{ij} D_{ij} \ \leq \ 0; \tag{12}$$

$\forall i = 1, \ ..., \ N + K$.

Constraints (1) to (12) form a linear constraint satisfaction problem for VRPTW which can be solved by **LS(CP)**.

Constraints (10) and (11) require some explanation. Intuitively, it says that for routes which are "short" (i.e. whose length is shorter than some $\theta > 0$), we want to make them null, whereas for the rest of them, we want to prevent them from becoming shorter. In doing so, we are in fact attempting to remove short routes by relocating the customers on the routes. This formulation came about through the observation that locally optimal solutions are the ones that often contain "short" routes (containing 1 or 2 customers). Hence, in our experiments, we set $\theta = 3$. Note that if we had directly modelled modelled minimization of number of vehicles directly via the constraint (like the Travel distance constraint):

$$\sum_{k=1}^{K} (\sum_{i=1}^{N} v_{ik} \geq 1) \ \leq \ 0; \tag{13}$$

it would not have produced the desired result, since this constraint is too weak in effecting customer relocation.

3.1.2. *Solving CSP with Local Search*

To solve a constraint satisfaction problem via local search, we propose a scheme (called **LS(CP)** (Henz et al., 2000)) that generalizes the WSAT(OIP)(, 1983) and the WSAT (Selman et. al, 1994) algorithms.

LS(CP) is a greedy local search method which incorporates a tabu mechanism as well as a random walk component to explore the search space and overcome the effects of local minima. The algorithm always works on a full assignment to all the variables in the given CSP instance. Beginning with an initial assignment to all the variables (not necessarily a feasible solution), for example a random assignment, the algorithm inclines to reduce the degree of violation of the constraints in every iteration by performing a local flip. It maintains a saved copy of assignment whenever it encounters a better solution according to certain comparison criteria. The search continues until the number of iterations reaches a preset value maxMoves or some solution criteria are met. The above process is repeated up to maxTries times.

To deal with hard and soft constraints, constraints are further partitioned into levels, hence forming a constraint hierarchy. This constraint hierarchy induces a partial ordering on the solution space, and LS(CP) compares solutions in terms of degree of violation over this partial ordering. For details of LS(CP), the reader is referred to (Henz et al., 2000).

3.2. Co-operative Framework

We now proceed to describe a co-operative framework between the the two-phase approach discussed in Section 2.2 and our constraint-based diversification technique. By doing so, the overall algorithmic framework is capable of moving from one local optimum to another in the objective landscape, thus enchancing the likelihood of finding a better solution.

We use a strategy similar to LNS (Shaw, 1998) in which a part of a solution returned from tabu search is extracted and passed to LS(CP) to diversify the search neighbourhood of the current solution.

The co-operative framework is given as follows.

1. Set $count = 0$.
2. Generate a feasible solution x with the two-phase approach. Set $x^* = x$, and $z^* = z(x)$ ($z(x)$ is the cost of solution x)
3. Extract $P\%$ of the customers to form a sub-VRPTW instance.
4. Pass the sub-VRPTW instance and x to LS(CP).
   a) Relax all time windows and vehicle capacity constraints and allow LS(CP) to reduce the cost by exploring into infeasible regions.
   b) Resume all the constraints relaxed in step 5 and continue search until a new feasible solution is obtained.
5. Feed the partial solution back to tabu search.
6. Perform tabu search with the modified solution and this will generate a new feasible solution x.
7. If $z(x) < z^*$, then $x^* = x$ and $z^* = z(x)$.
8. $count = count + 1$. If $count < maxIter2$, go to step 3; otherwise, output x^* and z^*.

In step 3, we first measure the *tightness* of each route (a route is tight when the arrival times are very close to the start times of delivery to customers in that route). Routes are then sorted in descending order according to their tightness. We choose the last few routes that consist of roughly $P\%$ of the customers from the list. With this selection strategy, those routes with tight schedule will be preserved. Step 4(a) is where the diversification takes place, since tabu search in general does not handle infeasibility. In addition, the quality of the new partial solution returned from LS(CP) in step 4(b) is usually worse than the one initially returned by the two-phase approach. However, the good news is that this provides opportunities to enter a possibly more promising region, to which tabu search is applied again in step 6.

4. Experimental Results

In this section, we compare the experimental results of the two-phase approach discussed in Section 2.2 as well as our proposed co-operative framework presented in Section 3.2. Numerical experiments are performed over the well-known Solomon's benchmark problems.

In these experiments, we set tabu length $n = 50$, $maxIter = 100$, and $maxStag = 20$. For the co-operative framework, we set $P = 90$ and $maxIter2 = 3$. In step 4a of co-operative framework, we relax the constraints defined in Eqs. (6)-(9). We performed our experiments on a SUN Sparc II workstation, and impose a time-out of 10 CPU minutes on each test case.

Table 4 shows the results (in terms of the number of vehicles needed and the travel distance) obtained by the best published results taken from the literature and our results.

Contributions to the best-known results from heuristics are taken from:

- Cordeau *et al.* (, 1983)- *R107, R108, RC104, RC106, R204, RC201, RC207*
- Chiang and Russell (Chiang and Russel, 1996) - R207
- Homberger and Gehring (Homberger and Gehring, 1999) - *R103, R109, R112, R201, R203, R208, R210, R211, C202, RC203, RC204, RC205, R110*
- Rousseau *et al.*(Rousseu et al.,) - *R111, RC105, R202, R205, R209, RC206, RC208*
- Rochat and Taillard (Rochat and Taillard, 1995) - *R101, R104, R105, R106, C101, C102, C103, C104, C105, C106, C107, C108,*

C109, RC103, R206, C201, C202, C203, C204, C205, C206, C207, C208, R102

– Taillard *et al.* (Taillard et al., 1997) - *RC101, RC102, RC107, RC108*

It can be observed that our results are inferior compared to the best published results. Yet, when compared against a specific algorithm, our approach performs well. For instance, when matched against the results of Rochat and Taillard (Rochat and Taillard, 1995) (as summarized in (Schulze and Fahle, 1999) Row 1 of Tables 3 and 4 for compatible CPU times), we observe the following results, as shown in Table 4.

5. Conclusion

In this paper, we discussed a simple constraint-based technique for diversifying local search. This method is simple and applicable to any optimization problem so long as the given problem can be cast as a constraint satisfaction problem. A future work that follows logically would be to apply this technique on different optimization problems. Clearly, applying our technique directly may not yield the best possible results. The challenge therefore is in designing problem-specific heuristics to strengthen the various steps in our proposed co-operative framework. For instance, in the case of VRPTW, we need to design algorithms to find a good subset of routes to be extracted, and to determine which constraints should be relaxed (since not all constraints need to be relaxed).

Acknowledgments

This work is supported by Kent Ridge Digital Laboratories and the National University of Singapore.

References

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, *Science*, 220, 671-680, 1983.

J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

F. Glover and M. Laguna, *Tabu Search*, Reading, Kluwer Academic Publishers, 1997.

Y. Rochat and É. D. Taillard, Probabilistic diversification and intensification in local search for vehicle routing, *Journal of Heuristrics*, 1, 147-167, 1995.

Table I. Our Solutions against the Best Published Results for Solomon's Test Cases

	Best		Ours			Best		Ours	
	Vehicle	Distance	Vehicle	Distance		Vehicle	Distance	Vehicle	Distance
C101	10	828.94	10	828.94	R201	4	1252.37	5	1287.36
C102	10	828.94	10	834.64	R202	3	1191.70	5	1192.58
C103	10	828.06	10	834.56	R203	3	942.64	4	1017.97
C104	10	824.78	10	846.32	R204	2	849.62	3	920.90
C105	10	828.94	10	828.94	R205	3	994.42	3	1098.97
C106	10	828.94	10	828.94	R206	3	912.97	3	1066.28
C107	10	828.94	10	828.94	R207	2	914.39	3	890.62
C108	10	828.94	10	828.94	R208	2	731.23	3	791.01
C109	10	828.94	10	828.94	R209	3	909.86	3	1032.98
C201	3	591.56	3	591.56	R210	3	955.39	4	968.89
C202	3	591.56	3	591.56	R211	2	910.09	3	853.01
C203	3	591.17	3	617.62	RC101	14	1696.94	17	1714.24
C204	3	590.6	4	735.70	RC102	12	1554.75	15	1540.48
C205	3	588.88	3	588.88	RC103	11	1262.02	13	1353.03
C206	3	588.49	3	588.49	RC104	10	1135.48	11	1201.05
C207	3	588.29	3	595.93	RC105	13	1633.72	16	1557.60
C208	3	588.32	3	588.32	RC106	11	1427.13	14	1442.76
R101	19	1650.8	20	1652.00	RC107	11	1230.54	12	1276.67
R102	17	1486.12	19	1461.70	RC108	10	1139.82	11	1156.67
R103	13	1292.85	13	1175.67	RC201	4	1406.94	5	1526.63
R104	10	982.01	12	1024.52	RC202	3	1389.57	4	1343.26
R105	14	1377.11	15	1422.28	RC203	3	1060.45	4	1180.08
R106	12	1252.03	14	1264.69	RC204	3	799.12	4	976.88
R107	10	1113.69	12	1136.18	RC205	4	1302.42	5	1371.61
R108	9	964.38	11	1001.26	RC206	3	1153.93	5	1342.05
R109	11	1194.73	14	1239.62	RC207	3	1062.05	4	1145.76
R110	10	1124.4	12	1151.49	RC208	3	829.69	3	938.56
R111	10	1096.72	13	1131.53					
R112	9	1003.73	10	1027.77					

Table II. Summary of Comparison

	RT	Ours
C1 Total Number of Vehicles	10.00	10.00
C1 Total Distance	832.59	832.13
C2 Total Number of Vehicles	3.00	3.00
C2 Total Distance	595.38	612.25
R1 Total Number of Vehicles	12.83	14.00
R1 Total Distance	1208.43	1211.54
R2 Total Number of Vehicles	3.18	3.55
R2 Total Distance	999.63	960.43
RC1 Total Number of Vehicles	12.75	13.63
RC1 Total Distance	1381.33	1385.05
RC2 Total Number of Vehicles	3.62	4.25
RC2 Total Distance	1207.37	1232.65

G. Pesant and M. Gendreau, A Constraint Programming Framework for Local Search Methods, *Journal of Heuristics*, 5(3), 255-279, 1999.

P. Shaw, Using constraint programming and local search methods to solve vehicle routing problems, in *Proceedings of CP'98*, 417-431, Springer-Verlag, 1998.

É. D. Taillard, Parallel iterative search methods for vehicle routing problems, *Networks*, 23, 661-676, 1993.

Y. Rochat and F. Semet, A tabu search approach for delivering pet food and flour in Switzerland, *Journal of the Operational Research Society*, 45, 1233-1246, 1994.

M. M. Solomon, Algorithms for the vehicle routing and scheduling problem with time window constraints, Operations Research, 35, 254-265 (1987).

P. Kilby, P. Prosser, and P. Shaw, A comparison of traditional and constraint-based heuristic methods on vehicle routing problems with side constraints, *Journal of Constraints*, 5(4), 389-414, 2000.

M. W. P. Savelsbergh, Computer aided routing, Centrum voor Wiskunde en Informatica, Amsterdam, 1988.

J. P. Walser, Domain-independent local search for linear integer optimization, *Pd. D. Thesis*, Universität des Saarlandes, Saarbrücken, Germany (1998).

B. Selman, H. Kautz, and B. Cohen, Noise strategies for improving local search, in *Proceedings of AAAI-94*, 337-343, 1994.

M. Henz, Y. F. Lim, S. C. Lua, X. P. Shi, J. P. Walser and R. Yap, Solving Hierarchical Constraints over Finite Domains, In Proc. *6th Int'l Symp. on Artificial Intelligence and Mathematics*, Fort Lauderdale, Florida 2000.

W. C. Chiang and R. A. Russell, "Simulated Annealing Metaheuristics for the Vehicle Routing Problem with Time Windows," *Annals of Operations Research*, 63, 3-27, 1996.

J.F. Cordeau, G. Laporte, and A. Mercier, "A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows," Working Paper CRT-00-03, *Centre for Research on Transportation*, Montreal, Canada, 2000.

J. Homberger and H. Gehring. Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows, *INFOR*, 37, 297-318, 1999.

L.M. Rousseau, M. Gendreau and G. Pesant, Using Constraint-Based Operators to Solve the Vehicle Routing Problem with Time Windows, *Journal of Heuristics*, forthcoming.

E. Taillard, P. Badeau, M. Gendreau, F. Geurtin, and J.Y. Potvin, 1997.A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows,*Transportation Science*, 31, 170-186, 1997.

J. Schulze and T. Fahle, A parallel algorithm for the vehicle routing problem with time window constraints, *Annals Operations Research*, 86, 585-607, 1999.

B. D. Backer, P. Kilby, P. Prosser and P. Shaw, "Solving Vehicle Routing Problems using Constraint Programming and Metaheuristics", *Journal of Heuristics*, 6(4), 501-525, 2000.

P. Kilby, P. Prosser and P. Shaw, "A Comparison of Traditional and Constraint-based Heuristic Methods on Vehicle Routing Problems with Side Constraints", *Journal of Constraints*, 5(4), 389-414, 2000.