CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Multi-Start Heuristics for the Two-Echelon Vehicle Routing Problem

Teodor Gabriel Crainic
Simona Mancini
Guido Perboli
Roberto Tadei

June 2010

CIRRELT-2010-30

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Multi-Start Heuristics for the Two-Echelon Vehicle Routing Problem

Teodor Gabriel Crainic[1,2,*], Simona Mancini[1,3], Guido Perboli[1,3], Roberto Tadei[3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
[2] Department of Management and Technology, Université du Québec à Montréal, C.P. 8888, succursale Centre-ville, Montréal, Canada H3C 3P8
[3] DAUIN, Politecnico di Torino, C.so Duca degli Abruzzi 24 10129, Torino, Italy

Abstract. In this paper we address the Two-Echelon Vehicle Routing Problem (2E-VRP), an extension of the classical VRP, where the delivery from a single depot to customers is managed by routing and consolidating the freight through intermediate depots that are called satellites. We present a family of Multi-Start heuristics based on separating the depot-to-satellite transfer and the satellite-to-customer delivery by iteratively solving the two resulting routing subproblems, while adjusting the satellite workloads that link them. We present computational results on a wide set of instances up to 50 customers and 5 satellites and compare it with results from literature. Our methods over perform previous existent methods, both in efficiency and in effectiveness.

Keywords. Vehicle routing, heuristics, clustering, path-relinking

Acknowledgements. This project has been partially funded by the Italian Ministero dell'Università e della Ricerca through "Progetto PRIN 2007 - Problemi Integrati di Vehicle Routing e Container Packing: Modelli ed Algoritmi". Funding has also been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grants programs, by the partners of the Chair, CN, Rona, Alimentation Couche-Tard and the Ministry of Transportation of Québec, and by the Fonds de recherche sur la nature et les technologies (FQRNT) through its Team Research Grants program.

---

* Corresponding author: Teodor-Gabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2010

1 Introduction

In this paper, we address the basic, static version of the problem, denoted the *Two-Echelon Vehicle Routing Problem* (*2E-VRP*), which is characterized by a single depot and a given number of satellites. The first level routing problem addresses depot-to-satellites delivery, while the satellite-to-customer delivery routes are built at the second level. The goal is to ensure an efficient and low-cost operation of the system, where the demand is delivered on time and the total cost of the traffic on the overall transportation network is minimized. In Multi-Echelon Vehicle Routing Problems, delivery from one or several depots to customers is managed by routing and consolidating the freight through intermediate depots which are called *satellites*. This approach is closely connected to the design of City Logistics systems for large cities, where it provides the means of efficiently keep large trucks out of city centers, while the last leg of the distribution activities is provided by small and environmental-friendly vehicles. This family of problems differs from the multi-echelon distribution systems that can be found in the literature, which focus on the utilization of facilities and the flow assignment between levels, while, in the case we consider, the key elements concern the management of the fleet and the global routing of vehicles in the system.

This problem is faced frequently in real life applications, both at the strategic level (long term planning) and at the operational one (real-time optimization). Methods which can be applied at both levels must be accurate, and at the same time, very fast. In fact, in long term planning, the *2E-VRP* is part of a simulation framework, which means it must be solved several times during the optimization process and for this reason, computational times should be short. Real-time optimization problems, for which a feasible solution is needed in a quick short time, are often faced at the operative level. On the other hand, accuracy of the solution, is also very important, because, in real applications, even a small gain in the objective function could yield a great saving for the transportation company.

No previously defined methods, either exact or heuristic, are able to solve large problems, which are very common in real applications. Our aim is to develop a tool which could guarantee good accuracy while maintaining good efficiency.

In this paper we introduce and compare heuristics for the 2E-VRP, which are based on separating first and second level routing problems and applying an iterative procedure in which the two resulting subproblems are solved sequentially.

We also report the results of an experimental phase performed on instances of various sizes and layouts. We present first an experimental phase on small instances, which allowed us to compare our different heuristics, then we compare our best methods with heuristics from the literature, and finally, we present computational tests on large size instances, which cannot be solved by the methods obtained from the literature.

We describe the problem statement in Section 2, while in Section 3 we give a literature review. The methods are presented in Section 4 and we report the computational results and analyses in Section 5. Conclusions and perspectives are presented in Section 6.

2 Problem statement

The distribution of freight cannot be managed by direct shipping from the depot to the customers. Instead, freight must be consolidated from the depot to a satellite and then delivered from the satellite to the desired customer. This implicitly defines a two-echelon transportation system: the 1st level connecting the depot to the satellites and the 2nd one the satellites to the customers.

Let us denote the depot with v_0, the set of intermediate depots, called satellites with V_s and the set of customers with V_c. Let n_s be the number of satellites and n_c the number of customers. The depot is the starting point of the freight and the satellites are capacitated. The customers are the destinations of the freight and each customer,i, has an associated demand d_i, i.e. the quantity of freight that has to be delivered to that customer. The demand of each customer cannot be split among different vehicles at the 2nd level. For the first level, we consider that each satellite can be served by more than one 1st-level vehicle, therefore the aggregated freight assigned to each satellite can be split into two or more vehicles. Each 1st level vehicle can deliver the freight of one or several customers, as well as serve more than one satellite in the same route.

Let us define the arc (i, j) as the direct route that connects node i to node j. If both nodes are satellites or one is the depot and the other is a satellite, we can define the arc as belonging to the 1st-level network, while if both nodes are customers, or one is a satellite and the other is a customer, the arc belongs to the 2nd-level network.

We consider only one type of freight, i.e. the volumes of freight belonging to different customers can be stored together and loaded in the same vehicle for both the 1st and the 2nd-level vehicles. Moreover, the vehicles that belong to the same level have the same capacity.

We define a route made up of a 1st-level vehicle which starts from the depot, serves one or more satellites and ends up at the depot, as *1st-level route*. A *2nd-level route* is a route made up of a 2nd-level vehicle which starts from a satellite, serves one or more customers and ends up at the same satellite.

The fleet sizes are fixed and known in advance for both levels. All vehicles belonging to the same level have the same capacity. Satellites are capacitated; their capacity is defined as the maximum number of second level vehicles which can leave from it.

Different satellites may have different capacities.

3 Literature review

Literature on multi-echelon systems is quite huge, but it is mainly focused on flow distribution, while routing costs are usually simplified, or not explicitly considered. The problem we address is similar, but different, to the Multi-Echelon Capacitated Location Distribution Problem, in which location and flow assignment are handled while no routing aspects are considered. For a complete survey of this problem the readers can refer to Salhi and Nagy [11]. For what concern exact methods, different formulations and relaxation have been presented in Gendron and Semet [6], while Albareda-Sambola and Diaz [2] have provided a compact model and tight bounds. For the heuristics approach reference can be made to Barreto et al. [3] who have developed several heuristics based on hierarchical and non hierarchical clustering algorithms and to Wu et al. [12], who have presented a type of heuristics that is based on a simulated annealing embedded in a general framework for the problem solving procedure. Another similar problem is the Inventory Routing Problem, which differs from our problem because it is based on customers usage rather than customers orders, and more attention is given to the choice of the moment in which to serve a customer, with respect to the choice of the way to follow to reach it. For a survey on this subject we can refer the reader to Moin and Salhi [8].

Due to the recent introduction of the problem, the literature on 2E-VRP is somewhat limited. A formulation for the 2E-VRP has been presented by Perboli et al. [9], with which instances of up to 32 customers have been solved to optimality. In the same paper, the authors derived two math-heuristics that are able to address instances of up to 50 customers. Both of them are based on the LP model that is presented in the paper and which works on customer-to-satellite assignment variables. The first math-heuristic, called *Diving*, considers a continuous relaxation of the model and applies a diving procedure to the customer-to-satellite assignment variables which are not integer. A restarting procedure is incorporated to recover possible infeasibilities due to variables fixing. The second one is named *Semi-continuous*; in this method the arc usage variables are considered continuous, while the assignment variables are still considered integer. The method solves this relaxed problem and uses the obtained values of the assignment variables to build a feasible solution for the 2E-VRP. A general time-dependent formulation with fleet synchronization and customer time windows has been introduced by Crainic et al. [5] in the context of two-echelon City Logistics systems. The authors have indicated promising algorithmic directions, but no implementation has been reported.

4 Heuristics for the 2E-VRP

In this paper we apply a separation strategy that splits the problem into two routing subproblems, one at each level. The second level problem can be further decomposed into n vehicle routing problems (VRPs), where n the number of satellites, one for each satellite. In every VRP we consider as depot a satellite and as customers only those which have been assigned to it. The customer-to-satellite assignment problem plays a crucial role in the problem solving. In fact, if we suppose we know the optimal assignment, an optimal solution can easily be obtained, by solving the VRP related to each satellite to optimality, and the resultant VRP at the first level, in which we consider the satellites, with a demand equal to the sum of the customers assigned to it, as customers. The 2E-VRP can be treated as an assignment problem in which the objective function is given by the solution of $n+1$ VRPs, which can be solved using methods form literature. Since the computational time is due, in the greater part, to routing solving, we cannot neglect this information while developing a fast heuristic method. In fact, methods involving large neighborhood exploration, are not adapted to solve this problem, because of the computational time needed to analyze each solution of the assignment problem. In order to develop fast heuristics we need a mechanism which can guide us, in the solutions space, to a promising solution, and allow us to obtain good results without exploring a high number of solutions.

In this section we first present a quick method to find a feasible initial solution, then, a local search heuristic which works on the perturbation of the initial assignments, and finally a family of Multi-Start methods in which this local search is applied on different solutions found applying a randomized perturbation on the customer-to-satellite assignments. Different methods for returning to feasibility if the perturbed solution is unfeasible are also presented.

All the methods work as follows:

1. An initial solution is computed
2. A local search is applied
3. A new perturbed solution is generated
4. If the solution is not feasible a feasibility search algorithm is applied
5. If the solution is feasible, and it is promising (it respects a quality threshold) local search is applied
6. The procedure restarts from 3. until a maximum number of iteration is reached

4.1 Initial solution computation

In order to find an initial solution, we have developed a spatial clustering type of heuristics, from now on called First Clustering (FC). The initial clustering is based on the **direct shipment** criterion, which assigns a customer to the satellite with the smallest Euclidean distance. The assignment must be feasible, with respect to the fleet-size restriction (e.g., in a system with two satellites, a fleet of four vehicles with equal capacity of 6000 units, and a total customer demand of 21000 units, an assignment resulting in a demand of 13000 units for one satellite and 8000 units for the other requires at least 5 vehicles and is therefore not feasible). If the assignment is not feasible, the customer is assigned to the second nearest satellite, and so on until a feasible assignment is found.

The resulting independent VRP can be solved using each exact or heuristics methods for the CVRP. The cost of the second-level solution is computed as the sum of the obtained VRPs solutions. The demand of each satellite is updated according to the assignment and the first level VRP is solved. The combination of the first and second-level VRPs yields a feasible solution for the 2E-VRP that is denoted the *current solution* with a cost that is equal to the sum of the second and the first level routing costs.

4.2 The local search approach: A clustering based heuristic

The heuristic we present, named Clustering Improvement (CI) is a clustering based heuristic which has the aim of improving the assignment given by the initial solution obtained following a local search approach with a first improvement exploration technique, in which the order is given by a distance based rule according to which the neighborhood is explored. The considered neighborhood is defined as the set of assignments in which only one assignment is different from the current solution. Since the neighborhood is small, it can be explored in a quite short time. Nevertheless, computational times can be ulteriorly reduced; in fact, since we explore first the most promising neighbors, when the objective function reaches significant bad values respect to the current best, the probability to obtain an improving solution analyzing the following neighbors become very low. For that reason we decided to define a percentage threshold δ, such that if, while exploring the neighborhood we find a solution which has an objective function value higher than the current best of more than δ, the exploration is terminated. The method works as follows:

1. It starts from an initial solution
2. A neighborhood containing all the neighbors reachable changing one and only one customer assignment is defined
3. The neighborhood is explored following a first improvement strategy
4. If a solution which does not respect a quality threshold or if all the neighborhood

has been explored without finding an improvement the procedure terminates

The pseudocode of the algorithm is the reported in Algorithm 4.2.

Algorithm 1 Clustering Improvement

```
repeat
  sort the customers, in increasing order according to the difference in distances between the customer and the satellite to which it has been assigned in the initial solution and between the customer and the nearest satellite among the ones to which it has not been assigned;
  consider the first customer on the list and assign it to its second-nearest satellite;
  if the new cluster assignment is not feasible with respect to the capacity constraints then
    consider the next customer in the list;
  else
    solve the small independent VRPs for the new clusters;
    update the demand of each satellite according to the new assignment and solve the first-level VRP;
    compute the global cost of the new solution and compare it to the cost of the current solution;
  end if
  if the new solution is better then
    keep it as the initial solution;
    re-start the procedure;
  else
    if the new solution is worse of more than a fixed percentage threshold δ then
      terminate the algorithm
    else
      consider the next customer in the list
    end if
  end if
until the list is empty or a given stopping criterion (maximum number of iterations or computing time) has been reached.
```

4.3 Multi-Start heuristics

Search methods based on local optimization that aspire to find global optima usually require some type of diversification to overcome local optimality. Without a diversification phase, such methods can become localized in a small area of the solution space, with very limited possibility of finding a global optimum. In recent years many techniques have been proposed for the avoidance of local optima. One way to achieve diversification is to re-start the search from a new solution once a region has been extensively explored.

Multi-Start strategies can then be used to guide the construction of new solutions in a long term horizon of the search process. Multi-Start methods are composed by an intensification phase, which is normally a local search approach, (but it could be even a more complex heuristic or metaheuristic), and a diversification phase, in which a new solution, possibly in a different area of the solution space. For a complete overview of Multi-Start methods we refer the reader to Marti [7].

We present a family of Multi-Start heuristics in which the intensification phase is performed applying the Clustering Improvement (CI), while the diversification is actuated by applying a randomized perturbation on the customer-to-satellite assignments, and solving the resulting VRPs. This perturbation method do not imply the feasibility of the obtained solution, because satellites capacity or global fleet size constraints can be violated. If it happens, a feasibility search method (FS) for trying to render feasible the solution is applied. More in details, if the global fleet size constraint has been violated we try to move customers, chosen following a given rule, from the satellite to which belong the less filled vehicle, to another satellite randomly chosen, in order to free that vehicle, and repeat it for a number of iteration equal to the number of extra-vehicles we needed to fulfill the demands. Instead, in case of a violation of the satellites capacity, we remove customers, following an order created according to a given rule, from a satellite whose capacity has been exceeded, and assign him to another satellite randomly chosen, until the capacity constraint is again fulfilled. We repeat it for all the satellites in which the constraint has been violated in the diversification phase. If the new obtained solution is still unfeasible, we do not consider it and reapply the diversification phase in order to find a new solution. The intensification phase is applied only on the most promising solutions, i.e. the ones whose objective value is better of the current best or at least within the percentage threshold δ. We introduce two different rules to generate perturbed solutions and six different strategies for choosing customers to be reassigned in the feasibility research phase. Each perturbation rule can be combined with any feasibility search strategy. The procedure is repeated until a maximum number of iterations has been reached. The pseudocode of the algorithm is given in the following.

Algorithm 2 Multi-Start heuristics

```
find an initial solution s_i
repeat
  generate a perturbed solution s_p
  if the solution is unfeasible then
    apply the feasibility search (FS)
  end if
  if the solution is feasible and it is better then current best or at least within the
  threshold δ then
    apply Clustering Improvement (CI)
  end if
until maximum number of iterations has been reached.
```

4.3.1 Perturbed solution generating rules

We have developed two different rules to generate perturbed solutions. Both are based on the same idea, according to which we define an assignment probability of each customer i to each customer j, called P_{ij}, so that $\sum_i P_{ij} = 1$. Furthermore, we apply a Russian wheel algorithm, based on these probabilities, in order to determine the satellite to which is customer must be assigned in the perturbed solution.

Algorithm 3 Perturbed solution generation

```
for i=1 to n_c do
    for j=1 to n_s do
        calculate P_ij
    end for
    draw an integer number d in the interval [1, 100]
    P_i0 = 0
    j=0
    repeat
        if d ∈ [P_ij, P_ij + P_ij+1] then
            assign i to j
        end if
    until i has not been assigned
end for
```

According to the first rule, named **Linear randomized rule** (RAND1), the probability P_{ij} is computed as:

$$P_{ij} = \frac{1 - \frac{D_{ij}}{\sum_j D_{ij}}}{n - 1} \tag{1}$$

The second rule, named **Majority Prize rule** (PRIZE) works in a different way. Probabilities are computed according to the first rules. They are multiplied by a reduction coefficient $r \in [0, 1]$. A majority prize, MP, is given to the assignment with the highest probability and a smaller prize, SP, is given to the assignment with the second highest one, so that $MP+SP = 1-r$. The probability of the third highest probability assignment remains unvaried, while all the other assignment probabilities are placed equal to zero.

The tendency of the first rule, especially in the case of a high number of satellites, is to give more power to the random component. In fact when the number of satellites n grows, all the assignment probabilities tend to assume a value close to $1/n$. This implies that we could potentially find perturbed solutions very far from the initial one, but would be potentially unfeasible or with a very high objective function. The second rule partially reduces the random component effect, thanks to the prizes we give to the most promising assignments. In this way, we can find perturbed solutions nearer to the initial solution with respect to the first rule, while ensuring solutions distant enough from the initial one are obtained in order to have an high diversification.

4.3.2 Feasibility search strategies

Six different strategies have been developed. In the first one, named **DISTANCE** we move first customers with the highest distance from the satellite, whose reassignment probably has a smaller impact on the cost increment. The second and the third, respectively **MAX_WEIGHT** and **MIN_WEIGHT** are based on the customers demand. According to **MAX_WEIGHT** we move first the customer with the highest demand, which allow us to free a vehicle moving the minimum number of customers, while according to **MIN_WEIGHT**, we move the ones with the lowest demand, which are easier to be assigned to another satellite without violating capacity constraints. The other three strategy apply on a functional which depends both on distance and demand. The first customers to move are those with the highest value of the functional. This functional is computed as $F = \alpha dist_i + \beta d_i$ where α and β indicate the weight we give to the criteria, $dist_i$ indicate the distance between customer i and the satellite to which it has been assigned, while d_i represents the demand of customer i. The three strategies differ for different couple of criteria weights. In **50D_50W** the weights are both equal to 0.5, in **75D_25W** more importance is given to the distance criteria (weight=0.75) with respect to the demand one (weight=0.25), while in **25D_75W** the criteria roles are exchanged (distance weight=0.25, demand weight=0.25).

5 Computational tests

In this section we analyze the behavior of the above proposed heuristics in terms of solution quality and computational efficiency. Computational tests are based on instances with different sizes and layout instances, which are described in Section 5.1. Section 5.2 is devoted to presenting preliminary tests that are useful for comparing the different Multi-Start heuristics among each others in other to determine the best parameters setting. In Section 5.3 we compare some of the best Multi Start heuristics with Clustering Improvement and, then, with the other heuristics obtained from the literature, the math heuristics proposed by Perboli et al. [9] on small and medium sized instances (21-32 customers and 2 satellites, 50 customers and 2-3-4-5 satellites). All the computational times have been obtained by scaling all the computational times to an equivalent CPU time on a 2.5 GHz Intel Centrino Duo of 2.5 GHz by means of the SPECINT benchmarks ([1]). All the VRPs derived by the separation approach have been solved by the Branch and cut method developed by Ralphs [10].

5.1 Instances description

In this section, we introduce two instance sets for 2E-CVRP. The instances cover up to 50 customers and up to 5 satellites. The first set is taken from Set 2 by Perboli et al. [9] and it contains different sized instances (21-32 customers with 2 satellites and 50 customers with 2-4 satellites). For all the instances, the depot has a central position in the customer area. The amount of total demand is in the $90\% - 95\%$ range of the maximum sustainable load ($93,375\%$ and $91,781\%$, respectively) to make sure vehicles are "fully" loaded, while this still makes it relatively easy to find feasible solutions. The cost due to loading/unloading operations is fixed to 0.

In order to broaden the scope of the analysis, we also generated a second set of instances, with 50 customers and 2,3 and 5 satellites. These instances are generated by combining three customer distributions and three satellite location patterns. From now on, we will refer to them as, Set 4. Set 3 from Perboli et al. [9] has been not analyzed in this paper because it contains the same instances as Set 2, with the depot placed in an external zone, but Set 4 recreates the same situation using more realistic customers distributions and satellites locations.

Three different customer distributions have been recreated, representing a regional distribution, downtown and suburb zones in a large city, and a small town, respectively. The three considered satellite distributions are the following: a random distribution, in which satellites are randomly located around the customers area, a sliced distribution, according to which the available area is split into some slices and one satellite is randomly located for each slice, and the third one, which represent the case of city with limited accessibility, (near a river, the sea, etc..) for which only a restricted zone is available for satellites location. For a more accurate description of the instances generation the reader can refer to Crainic et al. [4].

Two instances were generated for each combination of customer distribution, satellite location pattern, and number of customers, for a total of 54 instances.

5.2 Multi-Start Heuristic tuning

In this section, we present the preliminary computational tests conducted on a small subset of Set 4, effectuated applying all the possible combinations of the parameters, *perturbed solution generation rule* and *feasibility search strategy*, in order to determine the best tuning for the Multi-Start heuristics. In Table 1, we report, for each instance, name, number of customers, number of satellites, value of the initial solution (FC), and of the solution obtained by the local search (CI) with respective computational times (expressed in seconds), value of the solution obtained with each couple of parameters and correspondent computational time. For each method we report the sum of the objective functions obtained on all the instances, the averaged computational time and the percentage improvement with respect to CI. Computational results show the good behavior of all the methods with respect to CI, and the limited computational effort requested. The overall best for each instance is underlined (if we have two or more methods which reach the same result we consider as overall best the one reached in the smallest computational time). Since, we cannot find a method which clearly outperform the others, we decided to test on all the set of instances the best four parameters configurations (PRIZE/50D_50W, PRIZE/75D_25W, PRIZE/25D_75W, RAND1/MIN_WEIGHT).

INSTANCE	Cust	Sat	FC	TIME	CI	TIME	PRIZE/DISTANCE	TIME	PRIZE/MAX_WEIGHT	TIME	PRIZE/MIN_WEIGHT	TIME	PRIZE/50D_50W	TIME	PRIZE/75D_25W	TIME	PRIZE/25D_75W	TIME
Instance50-s2-01.dat	50	2	1776.43	0.2	1773.02	29	1648.65	91	1718.59	158	1697.65	269	1699.37	107	1648.65	60	1699.37	106
Instance50-s2-09.dat	50	2	1679.26	0.2	1679.26	11	1679.26	118	1589.79	36	1669.26	74	1589.79	34	1622.98	78	1589.79	34
Instance50-s2-17.dat	50	2	2157.42	0.2	2106.72	30	2106.72	140	2106.72	201	2106.72	210	2106.72	226	2106.72	249	2106.72	180
Instance50-s3-22.dat	50	3	1614.61	0.2	1412.87	11	1412.87	12	1202.10	19	1202.10	19	1202.10	19	1202.10	17	1202.10	18
Instance50-s3-28.dat	50	3	1538.10	0.2	1329.86	12	1074.62	23	1074.62	23	1074.62	23	1074.62	23	1074.62	17	1074.62	21
Instance50-s3-32.dat	50	3	1345.65	0.2	1331.14	16	1120.25	15	1120.25	15	1120.25	15	1120.25	15	1120.25	13	1120.25	15
Instance50-s5-40.dat	50	5	1418.80	1	1387.28	5	1387.28	5	1387.28	5	1387.28	5	1387.28	5	1387.28	5	1387.28	5
Instance50-s5-44.dat	50	5	1243.30	0.2	1227.26	14	1227.26	14	1227.26	14	1227.26	14	1227.26	11	1227.26	11	1227.26	11
Instance50-s5-52.dat	50	5	1194.42	0.2	1145.54	10	1145.54	10	1145.54	10	1145.54	10	1145.54	10	1145.54	10	1145.54	10
SUM/AVG TIME			13967.99	0.29	13392.95	15	12802.45	48	12572.15	53	12630.68	71	12552.93	50	12535.4	51	12552.93	44
IMPROVEMENT							4.41%		6.13%		5.69%		6.27%		6.40%		6.27%	

INSTANCE	Cust	Sat	FC	TIME	CI	TIME	RAND1/DISTANCE	TIME	RAND1/MAX_WEIGHT	TIME	RAND1/MIN_WEIGHT	TIME	RAND1/50D_50W	TIME	RAND1/75D_25W	TIME	RAND1/25D_75W	TIME
Instance50-s2-01.dat	50	2	1776.43	0.2	1773.02	29	1713.03	94	1652.13	60	1686.38	276	1699.20	204	1713.03	215	1699.20	202
Instance50-s2-09.dat	50	2	1679.26	0.2	1679.26	11	1657.32	67	1679.26	91	1609.96	52	1652.15	104	1657.32	78	1652.15	141
Instance50-s2-17.dat	50	2	2157.42	0.2	2106.72	30	2106.72	234	2106.72	174	2106.72	174	2106.72	185	2106.72	305	2106.72	194
Instance50-s3-22.dat	50	3	1614.61	0.2	1412.87	11	1196.79	22	1196.79	22	1196.79	22	1196.79	22	1196.79	22	1196.79	22
Instance50-s3-28.dat	50	3	1538.10	0.2	1329.86	12	1098.59	15	1098.59	15	1098.59	15	1098.59	15	1098.59	15	1098.59	15
Instance50-s3-32.dat	50	3	1345.65	0.2	1331.14	16	1096.00	24	1096.00	24	1096.00	24	1096.00	24	1096.00	24	1096.00	24
Instance50-s5-40.dat	50	5	1418.80	1	1387.28	5	1387.28	5	1387.28	5	1387.28	5	1387.28	5	1387.28	5	1387.28	5
Instance50-s5-44.dat	50	5	1243.30	0.2	1227.26	14	1227.26	14	1227.26	14	1227.26	14	1227.26	14	1227.26	14	1227.26	14
Instance50-s5-52.dat	50	5	1194.42	0.2	1145.54	10	1145.54	10	1145.54	10	1145.54	10	1145.54	10	1145.54	10	1145.54	10
SUM/AVG TIME			13967.99	0.29	13392.95	15	12628.53	54	12589.57	46	12554.52	66	12609.53	65	12628.53	76	12609.53	70
IMPROVEMENT							5.71%		6.00%		6.26%		5.85%		5.71%		5.85%	

Table 1: Multi-Start heuristics tuning

5.3 Comparison with the state of the art

In this section, we compare the heuristic we presented in the previous section among each other and with two math-heuristics from literature: [9], **DIVING** and Semi-relaxed **SEMI**.

The results obtained on the whole Set 2 (21-32-50 customers instances) are reported in Table 2, while in Table 5 (reported in the Annex) we report results obtained on Set 4. Both tables are organized in the same way. More precisely, we report, for each instance, name, number of customers, number of satellites, value of the initial solution (FC), and of the solution obtained by the local search (CI) with respective computational times (expressed in seconds), value of the solution obtained with each couple of parameters and correspondent computational time. Objective function and computational time are reported also for DIVING and SEMI. The last column reports the best lower bound. Values in bold correspond to optimal solution. For each one of our methods we report the sum of the objective functions obtained on all the instances, the averaged computational time and the percentage improvement with respect to CI. The overall best of each instance is underlined. If it has been obtained by two or more methods, we consider as overall best the one obtained within the lower computational time.

INSTANCE	Cust	Sat	FC	TIME	CI	TIME	PRIZE/50D_50W	TIME	PRIZE/75D_25W	TIME	PRIZE/25D_75W	TIME	RAND1/MIN_WEIGHT	TIME	DIVING	TIME	SEMI	TIME	BEST_LIT	TIME	BEST LB
E-n22-k4-s6-17	21	2	424.89	0.13	424.89	1.590	**417.07**	16	**417.07**	16	**417.07**	16	**417.07**	14	417.07	7	417.07	14	**417.07**	21	417.07
E-n22-k4-s8-14	21	2	386.36	0.14	384.96	0.456	**384.96**	9	**384.96**	9	**384.96**	9	**384.96**	9	441.41	9	408.14	7	408.14	16	384.96
E-n22-k4-s9-19	21	2	485.12	0.48	485.12	0.763	472.23	20	472.23	20	472.23	20	472.26	11	472.23	6	470.60	10	**470.60**	16	470.60
E-n22-k4-s10-14	21	2	375.91	0.14	375.91	0.703	375.91	7	375.91	7	375.91	7	**371.50**	14	435.92	8	440.85	0.1	435.92	9	371.50
E-n22-k4-s11-12	21	2	453.77	0.34	453.77	1.180	444.83	15	444.83	15	444.83	15	444.83	15	487.45	6	429.39	10	429.39	16	427.22
E-n22-k4-s12-16	21	2	425.65	0.16	425.65	0.887	403.79	26	403.79	26	403.79	26	403.13	17	425.65	7	439.19	8	425.65	15	392.78
E-n33-k4-s1-9	32	2	774.54	0.11	774.54	3.420	757.56	20	757.56	20	757.56	20	757.56	15	772.57	29	736.92	2	736.92	31	730.16
E-n33-k4-s2-13	32	2	745.39	0.11	745.39	2.730	733.18	25	733.18	25	733.18	25	739.64	12	749.94	28	736.37	6	736.37	34	714.63
E-n33-k4-s3-17	32	2	810.83	0.25	801.21	3.400	754.65	28	754.65	28	754.65	28	787.29	22	801.19	68	739.47	5	739.47	73	707.41
E-n33-k4-s4-5	32	2	796.50	2.10	796.50	8.830	792.89	19	792.89	19	792.89	19	792.89	19	838.31	18	816.59	12	816.59	31	778.73
E-n33-k4-s7-25	32	2	775.85	0.12	756.88	1.880	756.88	15	756.88	15	756.88	15	756.88	15	756.88	18	756.88	42	756.88	59	756.84
E-n33-k4-s14-22	32	2	833.30	0.17	825.06	2.600	824.60	16	824.60	16	824.60	16	824.60	21	779.06	13	779.06	4	779.06	17	779.05
E-n51-k5-s2-17	50	2	614.17	0.24	614.17	0.566	614.17	12	614.17	12	614.17	12	611.88	16	666.83	75	628.53	567	628.53	641	576.97
E-n51-k5-s4-46	50	2	544.70	2.60	533.83	2.790	533.83	46	533.83	46	533.83	46	533.83	45	543.24	72	534.04	257	534.04	329	529.34
E-n51-k5-s6-12	50	2	562.21	0.27	559.00	0.586	546.92	32	546.92	32	546.92	32	546.92	18	560.22	69	554.80	60	554.80	130	541.17
E-n51-k5-s11-19	50	2	612.14	0.27	579.90	0.567	579.90	19	579.90	19	579.90	19	579.90	10	584.09	49	592.06	247	584.09	296	558.27
E-n51-k5-s27-47	50	2	535.77	0.23	535.77	0.352	535.77	17	535.77	17	535.77	17	535.77	15	538.20	85	538.20	224	538.20	310	535.04
E-n51-k5-s32-37	50	2	558.48	0.15	558.48	0.404	555.05	33	555.05	33	555.05	33	555.05	22	584.59	84	587.12	557	584.59	640	552.27
E-n51-k5-s2-4-17-46	50	4	566.60	0.12	565.00	0.138	565.00	5	565.00	5	565.00	5	565.00	5	590.63	280	542.37	1057	542.37	1338	515.75
E-n51-k5-s6-12-32-37	50	4	573.01	0.28	567.00	0.560	567.00	6	567.00	6	567.00	6	567.00	6	571.80	112	584.88	936	571.80	1048	516.02
E-n51-k5-s11-19-27-47	50	4	618.52	0.20	600.00	0.640	600.00	3	600.00	3	600.00	3	600.00	3	724.09	118	724.09	555	724.09	673	511.09
SUM/AVG TIME			12473.71	0.41	12363.03	1.67	12216.19	18.52	12216.19	18.52	12216.19	18.52	12247.96	15.43	12741.37	55.30	12456.62	218.16	12414.57	273.46	11766.86
IMPROVEMENT (CI)							1.19%		1.19%		1.19%		0.93%								
GAP (BEST-LIT)							-1.60%		-1.60%		-1.60%		-1.34%								

Table 2: Computational results for Set 2

As far as the Set 2 analysis is concerned, it can be noticed that all our Multi-Start methods perform sensibly better than DIVING (around 4%) and SEMI (around 2%) in quite smaller computational times. Even CI outperform DIVING and SEMI of respectively, 2.97% and 0.75% within a computational time two order of magnitude smaller. If we compare our results with the best known solution in literature (best between DIVING and SEMI) all the Multi-Start procedures improve of more than 1%. Furthermore we reach the overall best in the 59% of the cases, for an averaged improvement of the literature of 2.63%.

If we analyze Set 4 results we can notice a similar behavior of our methods with respect to Set 2. All our Multi-Start methods perform sensibly better than DIVING (more than 3%) and SEMI (more than 1%) in quite smaller computational times. If compared with the best known solution in literature (best between DIVING and SEMI) Multi-Start procedures obtain very similar results within a computational time one order of magnitude lower. The overall best is reached in the 53% of the cases and yield to an averaged improvement of the literature of 3.44%.

CUSTOMERS	OUR_BEST	LIT_BEST	GAP	WINNING
RANDOM	27333.56	27581.84	-0.90%	44%
URBAN	26059.00	26153.75	-0.36%	50%
TOWN	25401.92	25882.64	-1.86%	50%

Table 3: Aggregated results for customers distribution

SATELLITES	OUR_BEST	LIT_BEST	GAP	WINNING
RANDOM	25555.07	25364.06	0.75%	44%
SLICED	25703.01	25877.38	-0.67%	50%
FORBIDDEN	27187.12	28251.96	-3.77%	50%

Table 4: Aggregated results for satellites distribution

In tables 3 and 4 we report for each kind of distribution, the sum of the best objective functions found by our methods, the sum of the best objective functions in literature, the gap between our performances and the literature (if it is negative it means that we perform better) and the percentage of cases in which we perform better than the literature (winning cases). If we analyze aggregated results for customer distribution, we can notice that we gain in all the cases with respect to the literature, even if the better performances are reached in the case of a small town distribution, in which there is one centroid for each quadrants of the customer location area. This kind of distribution can be also found in large American cities where population is not concentrated in a central zone but is distributed in different high density distribution zones, and in provincial level distribution. For what concern satellites distribution, we perform better than methods from the literature, both in a sliced distribution and in a distribution for cities with limited access, which is the most common distribution we find in real applications, because a lot of cities present geographic constraints (near the sea, near the mountains) which limited the space, around the customer area, available for satellites location, and even if there are not geographic restriction, there are often logistic ones, that avoid the use of some areas. Furthermore, the random satellites distribution, the only one in which we obtain results a little bit worse than literature, is very hard to find in real cases, because the satellites location is always planned following different criteria, and is never done completely random.

6 Conclusions

We have here presented a family of Multi-Start heuristics for the basic Two-Echelon Vehicle Routing Problem, a distribution system where the delivery from a single depot to customers is managed by routing and consolidating the freight through intermediate depots that are called satellites. The heuristics are based on separating the first and second level routing problems and on iteratively solving the two resulting routing subproblems, while adjusting the satellite workloads (customer assignments) that link them.

The experimental results have shown that they all perform well, particularly considering the very limited computational effort necessary, and are more efficient than methods from the literature, which makes this two heuristics an important tool to solve the 2E-VRP. Computational results show also the very good performances of our local search approach, and a good quality of the initial solution computation method.

Future developments could address meta-heuristic frameworks working on neighborhoods based directly on the customer positioning inside the routes, instead of acting on the assignments, allowing to explore neighborhoods without recomputing for each neighbors the whole routing but modifying it locally, which could allow to address larger instances.

Acknowledgments

This project has been partially funded by the Italian Ministero dell'Universit e della Ricerca through "Progetto PRIN 2007 - Problemi Integrati di Vehicle Routing e Container Packing: Modelli ed Algoritmi". Funding has also been provided by the the Natural Sciences and Engineering Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grants programs, by the partners of the Chair, CN, Rona, Alimentation Couche-Tard and the Ministry of Transportation of Québec, and by the Fonds québécois de recherche sur la nature et les technologies (FQRNT Québec) through its Team Research grants program.

While working on this project, T.G. Crainic was NSERC Industrial Research Chair on Logistics Management and Adjunct Professor with the Department of Computer Science and Operations Research, Université de Montréal, and the Department of Economics and Business Administration, Molde University College, Norway.

The authors wish to thank their friend and colleague, Professor Michael Gendreau, for the many enlightening discussions.

References

[1] Standard performance evaluation corporation. 2006. spec cpu2006 benchmarks. http://www.spec.org/cpu2006/results/.

[2] M. Albareda-Sambola and E. Diaz, J.and Fernandez. A compact model and tight bounds for a combined location-routing problem. *Computers & Operations Research*, 32:407–428, 2005.

[3] S. Barreto, C. Ferreira, J. Paixao, and B. Souza Santos. Using clustering analysis in a capacitated location-routing problem. *European Journal of Operation Research*, 179:968–977, 2007.

[4] T. Crainic, G. Perboli, S. Mancini, and R. Tadei. Two-echelon vehicle routing problem: A satellite location analysis. In *City Logistics VI: Proceedings of the 6th International Conference on City Logistics, Puerto Vallarta, Mexique, 30 juin - 2 juillet,*, pages 65–75. Institute for City Logistics, Japan,, 2009.

[5] T.G. Crainic, N. Ricciardi, and G. Storchi. Models for Evaluating and Planning City Logistics Transportation Systems. Publication CIRRELT-2007-65, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Université de Montréal, Montréal, QC, Canada, 2007. submitted to *Transportation Science*.

[6] B. Gendron and F. Semet. Formulations and relaxations for a multi-echelon capacitated location-distribution problem. *Computers and Operations Research*, 36: 1335–1355, 2009.

[7] R. Marti. *Handbook of Metaheuristics*, volume 57, chapter Multi-Start Methods, pages 355–368. Springer New York, 2003.

[8] Moin and Salhi. Inventory routing problems: a logistical overview. *Journal of the Operational Research Society*, 58(9):1185–1194, Settembre 2007.

[9] G. Perboli, R. Tadei, and D. Vigo. The two-echelon capacitated vehicle routing problem. Publication cirrelt-2008-55, CIRRELT Montréal, Canada, 2008. submitted to Transportation Sciences.

[10] Ralphs. Parallel Branch and Cut for Capacitated Vehicle Routing. *Parallel Computing*, 29:607–629, 2003.

[11] S. Salhi and G. Nagy. Location-routing: Issues, models and methods. *European Journal of Operation Research*, 177:649 672, 2007.

[12] T. Wu, C. Low, and J. Bai. Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, 29:1393–.1415, 2002.

7 Annex

INSTANCE	Cust	Sat	FC	TIME	CI	TIME	PRIZE/50D_50W	TIME	PRIZE/75D_25W	TIME	PRIZE/25D_75W	TIME	RAND1/MIN_WEIGHT	TIME	DIVING	TIME	SEMI	TIME	BEST LB
Instance50-s2-01.dat	50	2	1776.43	0.2	1773.02	29	1599.37	87	1599.37	90	1599.37	54	1601.73	334	1590	138	1770	35	1480.57
Instance50-s2-02.dat	50	2	1497.85	0.2	1476.60	37	1446.09	75	1446.09	75	1446.09	75	1446.09	52	1442	27	1445	11	1381.00
Instance50-s2-03.dat	50	2	1772.65	0.2	1772.65	16	1606.46	99	1764.86	88	1606.46	99	1627.87	13	1603	98	1795	73	1478.37
Instance50-s2-04.dat	50	2	1483.48	0.2	1474.60	12	1440.77	30	1440.77	30	1440.77	30	1440.77	27	1789	140	1446	26	1355.30
Instance50-s2-05.dat	50	2	2212.46	0.2	2198.01	32	2188.15	100	2188.15	100	2188.15	100	2188.15	100	2216	73	2207	9	2117.12
Instance50-s2-06.dat	50	2	1313.62	0.2	1313.62	13	1313.62	13	1313.62	13	1313.62	13	1310.80	28	1735	365	1348	24	1227.69
Instance50-s2-07.dat	50	2	1668.44	0.2	1668.44	14	1510.80	85	1600.44	80	1510.80	34	1507.55	88	1486	38	1486	6	1359.23
Instance50-s2-08.dat	50	2	1371.74	0.2	1371.71	22	1369.78	27	1369.78	27	1369.78	27	1369.78	19	1468	240	1408	28	1335.94
Instance50-s2-09.dat	50	2	1679.26	0.2	1679.26	11	1589.79	34	1622.98	118	1589.79	42	1501.73	52	1492	74	1489	6	1363.68
Instance50-s2-10.dat	50	2	1424.80	0.2	1410.42	22	1410.42	62	1410.42	62	1410.42	62	1410.42	29	1568	529	1454	31	1317.99
Instance50-s2-11.dat	50	2	2104.13	0.2	2096.91	38	2082.07	101	2082.07	101	2082.07	101	2082.07	265	2073	72	2070	5	1997.89
Instance50-s2-12.dat	50	2	1284.79	0.2	1284.79	13	1266.21	18	1266.21	18	1266.21	18	1284.79	13	1561	35	1307	15	1127.33
Instance50-s2-13.dat	50	2	1664.40	0.2	1664.40	14	1664.40	14	1664.40	14	1664.40	14	1553.71	145	1555	14	1664	63	1392.19
Instance50-s2-14.dat	50	2	1404.03	0.2	1404.03	22	1404.03	81	1404.03	81	1404.03	81	1404.03	38	1399	9	1404	2	1347.31
Instance50-s2-15.dat	50	2	1670.91	0.2	1670.91	16	1670.91	29	1663.27	54	1670.91	29	1581.57	121	1554	312	1671	99	1404.82
Instance50-s2-16.dat	50	2	1422.71	0.2	1410.92	21	1410.42	30	1410.42	30	1410.42	30	1410.42	21	1418	74	1426	12	1337.45
Instance50-s2-17.dat	50	2	2157.42	0.2	2106.72	30	2106.72	226	2106.72	249	2106.72	180	2106.72	174	2117	110	2116	13	1972.82
Instance50-s2-18.dat	50	2	1289.22	0.2	1289.22	13	1259.56	22	1259.56	22	1259.56	22	1259.56	24	1226	31	1228	10	1126.40
Instance50-s3-19.dat	50	3	1576.82	0.2	1576.82	0.2	1576.82	5	1576.82	5	1576.82	5	1576.82	2	1636	165	1640	11	1405.27
Instance50-s3-20.dat	50	3	1606.34	0.2	1473.23	10	1433.87	26	1433.87	26	1433.87	26	1244.17	23	1296	61	1304	29	1155.63
Instance50-s3-21.dat	50	3	1639.50	0.2	1639.50	0.2	1631.66	7	1639.50	5	1631.66	7	1639.50	3	1591	781	1721	20	1463.66
Instance50-s3-22.dat	50	3	1614.61	0.2	1412.87	11	1215.69	11	1202.10	17	1215.69	11	1196.79	22	1623	38	1535	70	1195.22
Instance50-s3-23.dat	50	3	1906.00	1	1906.00	1	1681.29	46	1768.53	24	1681.29	46	1890.11	23	1866	287	1867	190	1526.87
Instance50-s3-24.dat	50	3	1360.69	0.2	1330.09	10	1175.74	10	1175.74	10	1175.74	10	1171.84	15	1538	107	1367	43	1162.76
Instance50-s3-25.dat	50	3	1617.41	1	1617.41	1	1590.11	11	1590.11	3	1590.11	11	1590.11	4	1765	236	1580	21	1293.66
Instance50-s3-26.dat	50	3	1379.16	0.2	1379.16	10	1161.86	19	1161.86	19	1161.86	19	1200.61	21	1227	74	1211	16	1055.22
Instance50-s3-27.dat	50	3	1505.94	1	1505.94	1	1505.94	2	1505.94	18	1505.94	5	1505.94	6	1664	185	1603	5	1323.44
Instance50-s3-28.dat	50	3	1538.10	0.2	1329.86	12	1101.98	10	1074.62	17	1101.98	10	1101.13	15	1291	169	1284	45	1069.30
Instance50-s3-29.dat	50	3	1855.89	1	1855.89	78	1502.05	47	1444.02	48	1502.05	47	1688.89	5	1897	317	1897	526	1398.13
Instance50-s3-30.dat	50	3	1364.16	0.2	1364.16	11	1364.16	11	1364.16	11	1364.16	11	1136.97	24	1302	100	1252	13	1038.83
Instance50-s3-31.dat	50	3	1533.98	1	1533.98	1	1533.98	4	1533.98	4	1533.98	4	1533.98	7	1670	357	1585	10	1325.67
Instance50-s3-32.dat	50	3	1345.65	0.2	1331.14	16	1095.28	15	1120.25	13	1095.28	15	1096.00	21	1197	78	1225	14	1093.91
Instance50-s3-33.dat	50	3	1574.32	1	1574.32	1	1574.32	3	1574.32	3	1574.32	3	1574.32	6	1624	281	1591	35	1365.61
Instance50-s3-34.dat	50	3	1365.46	1	1363.34	11	1363.34	16	1363.34	16	1363.34	16	1125.43	33	1249	87	1234	15	1123.04
Instance50-s3-35.dat	50	3	1598.66	1	1598.66	1	1429.58	33	1474.14	32	1429.58	33	1598.66	6	1743	305	1728	12	1425.52
Instance50-s3-36.dat	50	3	1366.20	1	1319.34	10	1319.34	10	1319.34	10	1319.34	10	1116.82	16	1264	97	1229	9	1096.37
Instance50-s5-37.dat	50	5	1680.04	0.2	1680.04	0.2	1586.23	2	1586.23	2	1586.23	2	1585.25	2	1712	2706	1556	67	1362.44
Instance50-s5-38.dat	50	5	1340.49	0.2	1340.49	0.2	1340.49	2	1340.49	2	1340.49	2	1340.49	2	1340	2193	1205	68	999.19
Instance50-s5-39.dat	50	5	1604.32	0.2	1604.32	0.2	1604.32	2	1604.32	2	1604.32	2	1604.32	2	1764	2867	1555	66	1355.60
Instance50-s5-40.dat	50	5	1418.80	1	1387.28	5	1387.28	5	1387.28	5	1387.28	5	1387.28	5	1325	623	1288	65	1004.91
Instance50-s5-41.dat	50	5	1950.69	0.2	1950.69	0.2	1762.62	32	1762.62	32	1762.62	32	1762.62	32	1726	2	1735	68	1466.30
Instance50-s5-42.dat	50	5	1559.39	0.2	1559.39	0.2	1559.39	2	1559.39	2	1559.39	2	1559.39	2	1346	2	1324	65	1046.11
Instance50-s5-43.dat	50	5	1687.28	1	1687.28	1	1687.28	1	1687.28	1	1687.28	1	1687.28	1	1652	931	1489	55	1229.97
Instance50-s5-44.dat	50	5	1243.30	0.2	1227.26	14	1227.26	14	1227.26	14	1227.26	14	1227.26	14	1157	1290	1058	42	867.82
Instance50-s5-45.dat	50	5	1756.60	1	1756.60	1	1756.60	1	1756.60	1	1756.60	1	1756.60	1	1469	1	1498	35	1230.52
Instance50-s5-46.dat	50	5	1166.76	0.2	1148.31	7	1148.31	7	1148.31	7	1148.31	7	1148.31	7	1190	2374	1153	22	897.72
Instance50-s5-47.dat	50	5	1903.34	0.2	1903.34	0.2	1700.43	25	1683.13	53	1700.43	25	1628.88	53.0	1628	0.2	1621	26.9	1393.52
Instance50-s5-48.dat	50	5	1319.96	1	1319.96	13	1319.96	13	1319.96	13	1319.96	13	1210.11	13	1210	13	1210	23	953.45
Instance50-s5-49.dat	50	5	1766.33	0.2	1766.33	0.2	1500.39	21	1500.39	21	1500.39	21	1660.76	0.2	1668	2339	1488	51	1297.01
Instance50-s5-50.dat	50	5	1131.65	0.2	1131.65	0.2	1131.65	0.2	1131.65	0.2	1131.65	0.2	1131.65	0.2	1245	931	1183	19	879.18
Instance50-s5-51.dat	50	5	1616.58	0.2	1616.58	0.2	1600.83	0.2	1600.83	0.2	1600.83	23.0	1606.55	0.2	1584	2282	1441	106	1237.38
Instance50-s5-52.dat	50	5	1194.42	0.2	1145.54	12	1145.54	12	1145.54	12	1145.54	12	1145.54	12	1181	436	1190	20	948.45
Instance50-s5-53.dat	50	5	1647.67	0.2	1647.67	0.2	1647.67	0.2	1647.67	0.2	1647.67	0.2	1630.19	43	1608	915	1570	33	1398.01
Instance50-s5-54.dat	50	5	1335.44	1	1304.97	50	1201.90	32	1201.90	32	1201.90	32	1201.90	32	1210	2118	1189	12	996.69
SUM/AVG TIME			84270	0.41	83356	12.32	79905	29.46	80227	32	79905	27.67	79350	37.34	82749	520.88	80336	44.33	69205
IMPROVEMENT (CI)							4.14%		3.75%		4.14%		4.81%						

Table 5: Computational results for Set 4