POLITECNICO DI TORINO
Repository ISTITUZIONALE

Multi-start heuristics for the Two-Echelon Vehicle Routing Problem

(Article begins on next page)

10 July 2024

Multi-start heuristics for the Two-Echelon Vehicle Routing Problem

Teodor Gabriel Crainic[23], Simona Mancini[12], and Guido Perboli[12] and
Roberto Tadei[1]

[1] Politecnico di Torino, Turin, ITALY
[2] CIRRELT, Montreal, Canada
[3] School of Management, UQAM, Montreal, Canada

Abstract. In this paper we address the Two-Echelon Vehicle Routing Problem (*2E-VRP*), an extension of the classical Capacitated VRP, where the delivery from a single depot to the customers is managed by routing and consolidating the freight through intermediate depots 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. The common scheme on which all the heuristics are based consists in, after having found an initial solution, applying a local search phase, followed by a diversification; if the new obtained solutions are feasible, then local search is applied again, otherwise a feasibility search procedure is applied, and if it successful, the local search is applied on the newfound solution. Different diversification strategies and feasibility search rules are proposed. We present computational results on a wide set of instances up to 50 customers and 5 satellites and compare them with results from the literature, showing how the new methods outperform previous existent methods, both in efficiency and accuracy.

1 Introduction

The transportation of goods plays a crucial role for most economic and social activities taking place in urban areas. For the city inhabitants, it supplies stores and places of work and leisure, delivers goods at home,and so on. For firms established within city limits, it forms a vital link with suppliers and customers. In the past decade researchers, besides the research for developing *green* vehicles, practitioners and institutions started to be aware that there was the need of developing new methods and technologies for optimizing how we use the resources presently available in order to reduce the impact of the different sources of nuisance (traffic congestion, pollution, reduction of the quality of life), avoiding to slow down the economic, social and cultural development of the urban areas. The implementation of this view is known as City Logistic, which introduces a multidisciplinary approach to urban logistics, as well as all the research projects aiming to build sustainable logistic systems which takes into account the impact

of freight operations on the environment. [6] Under this context, in this paper, we address the *Two-Echelon Vehicle Routing Problem* (*2E-VRP*), [12] which is characterized by a single depot and a set of customers. The delivery of the freight to the customers is not managed by direct shipping from the depot, but by consolidating the freight in intermediate depots, called *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. This problem is frequently faced in real life applications, both at the strategic level (long term planning) and the operational one (real-time optimization). Methods which can be applied at both levels must be accurate and, at the same time, fast. In fact, in long term planning the *2E-VRP* is often part of a larger simulation framework, which means it must be solved several times during the optimization process. Then, the computational times should be short, while maintaining a high accuracy. On the other hand, at the operational level, real-time optimization problems, for which a feasible solution is needed with a limited computational effort, are also often faced.

Our goal is to introduce new methods able to guarantee good accuracy while maintaining high efficiency. In this paper we introduce and compare different 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 sequentially solved. We also report extensive computational tests on instances of various sizes and layouts, comparing the newly defined heuristics with the other heuristics available in the literature.

More in detail, the paper is organized as follows. We define the problem 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 their analysis in Section 5. Conclusions and perspectives are presented in Section 6.

2   Problem definition

The Two-Echelon Vehicle Routing Problem (*2E-VRP*) is the Two-Echelon extension of the Capacitated Vehicle Routing Problem (CVRP), which aims to deliver the freight from the depot to the customers by consolidating the freight through the satellites while minimizing the overall transportation cost [12]. In our model we will not consider the fixed costs of the vehicles, since we suppose they are available in fixed number. Thus, the travel costs are given by the sum of the cost due to the usage by the vehicles of the arcs connecting depot, satellites and customers. These costs are of two types:

- costs of the arcs traveled by 1st-level vehicles, i.e. arcs connecting the depot to the satellites and the satellites between them;
- costs of the arcs traveled by 2nd-level vehicles, i.e. arcs connecting the satellites to the customers and the customers between them.

Let us denote the depot with v_0, the set of 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. Define the arc (i, j) as the direct route connecting node i to node j and c_{ij} its associated traveling cost. If both nodes are satellites or one is the depot and the other is a satellite, we 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 define as *1st-level route* a route made by a 1st-level vehicle which starts from the depot, serves one or more satellites and ends at the depot. A *2nd-level route* is a route made by a 2nd-level vehicle which starts from a satellite, serves one or more customers and ends at the same satellite.

The freight must be delivered from the depot v_0 to the customers set V_c. Let d_i be the demand of the customer i: 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.

The number of 1st-level vehicles available at the depot is m_1. These vehicles have the same given capacity K^1. The total number of 2nd-level vehicles available for the second level is equal to m_2. Moreover, each satellite k has a maximum capacity m_{s_k} expressed in terms of number of vehicles. The 2nd-level vehicles have the same given capacity K^2. No additional limitation on the route size, neither in length nor in number of visited customers is introduced.

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 in all the levels. 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 first-level depot exists and then no first-level routing costs are considered. For a complete survey of this problem the readers can refer to [15]. For what concern exact methods, different formulations and relaxation have been presented in [9], while in [1] a compact model and tight bounds have been provided. A Branch and Cut method has been proposed in [3]. For heuristic methods reference can be made to [2], where the authors developed several heuristics based on hierarchical and non hierarchical clustering algorithms, while, for what concerns metaheuristic methods, we refer to the following papers. In [13], the authors present a two-phases metaheuristic, in which the first phase executes a Greedy Randomized Adaptive Search Procedure (GRASP), based on an extended and randomized version of Clarke and Wright algorithm. This phase is implemented with a learning process on the choice of depots. In a second phase, new solutions are generated by a post-optimization

using a path relinking, while in [17], the authors propose a simulated annealing with a special solution encoding scheme that integrates location and routing decisions in order to enlarge the search space so that better solutions can be found. In [4] an hybrid heuristic based on a column generation scheme where the subproblems are solved using a tabu search algorithm, is presented.

4 Heuristics for the *2E-VRP*

The customer-to-satellite assignment problem plays a crucial role while solving *2E-VRP*, as remarked by the results in [12] and [11]. In fact, if we suppose to know the optimal customer-satellite assignments, *2E-VRP* is partitioned in at most $n_s + 1$ Capacitated VRP (CVRP) instances, one for the 1st-level and one for each satellite with at least a customer assigned. Thus, as in the math-heuristics in [12], we directly focus on the customer-satellite assignments by searching the optimal assignments, delegating state-of-the-art methods for solving the corresponding CVRPs.

Even if the literature on CVRP is quite huge and efficient methods have been developed to solve this problem, the computational time due to CVRP solving could be quite large. Thus, methods involving large neighborhood exploration on the assignments between customers and satellites are not suitable to solve this problem, because of the computational time needed to analyze each customer-satellite assignment change and its impact on the routing. For this reason, the core of our heuristic is a Multi-Start procedure that iteratively perturbs the solution, and a simple local search heuristic able to improve the initial assignment. Moreover, additional rules to prune not-promising assignments, and their corresponding CVRPs instances, are taken over. The main steps of our Multi-start heuristic are the following:

1. *First Clustering.* An initial solution is computed, by assigning each customer to a satellite according to a distance-based greedy rule. Thus, a complete solution is computed by solving the resulting first and second level CVRPs.
2. *Clustering Improvement.* A local search based on a neighborhood which changes one customer-satellite assignment each time is applied to the solution found by the *First Clustering*.
3. While the maximum number of iterations is not reached
   3.1 *Multi-Start.* Given the best solution found so far, the assignments customer-satellite are perturbed according to rules taking into account the cost of the reassignment.
      3.1.1 If the new solution is not feasible, we try to reach again the feasibility by means of the *Feasibility Search* algorithm.
      3.1.2 If the solution is feasible and it is considered promising, i.e. its objective function is within a given percentage threshold of the best solution, the Clustering Improvement phase is applied on it.

In the following, we give a detailed description of the different procedures involved.

4.1 First Clustering

In order to find an initial solution, we develop a clustering-based heuristic, from now on called *First Clustering* (*FC*). FC is based on a cost greedy criterion. More in detail, the procedure, after ordering the customers according to non-increasing order of their demand d_i, assigns each customer to the satellite with the smallest Euclidean distance. If the assignment of the customer to a satellite implies to add an additional vehicle, the procedure checks whether the constraints about the capacity of the satellite or the overall fleet capacity are violated. If so, the assignment is considered as unfeasible and the customer is assigned to the second nearest satellite, and so on until a feasible assignment is found. At the end of this clustering procedure, the customers are assigned to the satellite and the full solution can be computed by solving the first-level CVRP and the second-level CVRPs, one for each satellite with at least one customer assigned to it.

4.2 Clustering Improvement

Clustering Improvement (*CI*) aims to improve the customer-satellite assignments by means of a local search approach. The local search is a first improvement method where the neighborhood solutions are defined by assigning one customer from its original satellite to another one by a cost-based rule. More in details, the rule consists in moving customers from current satellite to nearest available. This trivial idea is very reasonable because it is much more frequent that in the optimal solution a customer is assigned to the nearest satellite or the second nearest one. Furthermore, this consideration holds for each customer distribution and does not depend on the satellite location strategy.

Let define the current solution as the solution given as the initial one to CI if we are at the first iteration or the best solution found at the previous iteration, otherwise. Then, the neighborhood works as follows.

Given the current solution, the customers are sorted by non-decreasing order of the reassignment cost, defined as $RC_i = c_{ij} - c_{ik}$, where i is a customer, j is the satellite to which i is assigned in the current solution, and $k \neq j$ is the satellite such that, moving i from satellite j to satellite k, the capacity constraints on the global second-level vehicle fleet and the satellite k are satisfied and the cost c_{ik} is minimum among the satellites $k \neq j$. This is equivalent to order the customers according to non-decreasing order of the estimation of the change in the solution quality due to the assignment of one customer from the present satellite to its second-best choice. Let be CL the ordered list of the customers.

repeat
  Consider the first customer i in CL;
  if k exists **then**
    remove i from CL;
  else
    terminate the CI algorithm and return the best solution;
  end if

Solve the CVRPs of satellites j and k;
Update the demand of each satellite according to the new assignment and solve the first-level CVRP;
Compute the objective function of the new solution and compare it to the cost of the current solution;
if the new solution is better **then**
   Keep it as new current solution and exit from the neighborhood;
else
   if the new solution has an objective function which is worse than a fixed percentage threshold γ from the objective function of the current solution **then**
      Terminate the CI algorithm and return the best solution;
   else
      Consider the next customer in the list
   end if
end if
until CL is empty

Even if the neighborhood has size $O(n_c)$, the computational time could grow up due to the need of recompute the CVRPs after a change in the customer-satellite assignments. This is the rationale of adding the additional heuristic stopping criterion when the reassignment has an objective function which is significantly worst than the current solution. The worsening of the quality of the solution is measured by the γ parameter. In fact, being the customers ordered by non-decreasing order of RC_i and being RC_i related to the change in the objective function when we assign the customer to another satellite, if the objective function of a neighbor is deteriorating too much, it is unlikely that the following neighbors may bring us an improving solution.

4.3 Multi-Start heuristic

Search methods based on local optimization that aspire to find global optima usually require some type of perturbation to overcome local optimality. Without a perturbation 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 to avoid local optima and a promising way are Multi-Start strategies. They are able to explore different regions of the search space by means of a re-start mechanism. Multi-Start strategies are then used to guide the construction of new solutions in a long term horizon of the search process. The general framework, after generating an initial solution, uses a perturbation mechanism to iteratively build new solutions, which are usually improved by a local search approach (but it could be even a more complex heuristic or metaheuristic). For a complete overview of Multi-Start methods we refer the reader to [10]. In the following, we present our Multi-Start heuristic. The perturbation is done in the *Perturbed Solution Generation* procedure by a cost-driven randomized rule, which changes the customer-to-satellite assignments. This perturbation method does not imply the feasibility of the obtained

solution, because of satellites capacity or global fleet size constraints violation. In this case, a *Feasibility Search (FS)* procedure is applied for bringing back the solution in the feasibility region. Whether the solution is feasible, the *Clustering Improvement* (CI) presented in Section 4.2 is applied to it to improve the solution quality. In order to limit the computational effort, the local search phase is applied only on the most promising solutions, i.e. the ones whose objective value is better of the current best or which objective function is not worse than a fixed percentage threshold δ from the objective function of the best solution. The procedure is repeated until a maximum number of iterations has been reached. In the following, we give more detail about the different rules we tested in Perturbed Solution Generation and Feasibility Search.

Perturbed Solution Generation We present the different rules to generate perturbed solutions. Both are random based, where the probability of a change is proportional to an estimation of the cost due to the reassignment of the customer to another satellite. Generally speaking, for each customer i and satellite j, we define a reassignment probability P_{ij}, $\sum_j P_{ij} = 1$. Then, the perturbation is obtained by considering the customers one after the other and computing the new satellite to which the customer is assigned by a Russian Wheel algorithm, based on the probabilities P_{ij}. The two different definitions of the probabilities P_{ij} are the following:

- *Linear Randomized (LR).* The probability P_{ij} is computed as $P_{ij} = \frac{1 - \frac{c_{ij}}{\sum_l c_{il}}}{n-1}$. The rule assigns the probabilities of each customer in inverse relation to its distance from the satellites. The rationale of this rule, in particular when the number of satellites increases, is to enforce the effect of the random component. In fact when the number of satellites n grows, the probabilities aim to be similar. This implies that we find perturbed solutions very far from the initial one, but potentially unfeasible or with a very high objective function.
- *Majority Prize (MP).* The idea of MP is to give a prize in terms of assignment to the best customer-satellite assignments of each customer, while penalizing the worst ones. For each customer, probabilities $\widetilde{P}_{ij}$ are computed according to LR and the satellite are ordered, for each customer, in non-decreasing order of $\widetilde{P}_{ij}$. Let j_{i1} and j_{i2} the first and the second satellites in the ordered list of customer i. Thus, given two constants $r \in (0,1)$ and $p \in (0.5, 1)$, the assignment probabilities are the following:
  $r\widetilde{P}_{ij}$, if $j \neq j_{i1}, j_{i2}$;
  $r\widetilde{P}_{ij}$ +(1-r)p, if $j = j_{i1}$;
  $r\widetilde{P}_{ij}$ +(1-r)(1-p), if $j = j_{i2}$.

Feasibility Search Let suppose that after the Perturbed Solution Generation phase we obtain a solution which is infeasible. Aim of the Feasibility Search procedure is to guide the solution towards the feasibility space. Thus, if the global fleet size constraint has been violated we try to move customers from

the satellite to which belong the less filled vehicle, to another satellite randomly chosen, in order to free that vehicle. In case of a violation of the satellite capacity, we remove customers from a satellite whose capacity has been exceeded, and assign them 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. If the new obtained solution is still unfeasible, the solution is discarded. In the following, we present the six different strategies we developed to choose the customers to be moved in order to achieve the feasibility:

1. *COST*. We move first customers with the highest cost from the satellite;
2. *MAX_DEM*. We move first the customer with the highest demand. This allows us to free a vehicle moving the minimum number of customers;
3. *MIN_DEM*. We move the customers with the lowest demand. The rationale is that the lower is the demand of the customer we are moving, the easier it is assigned to another satellite without violating capacity constraints;
4. The following three strategies uses both the cost and demand-based rules, by maximizing the expression $\alpha scost_i + \beta d_i$, where α and β are the weights we give to the criteria, $scost_i$ indicate the cost between customer i and the satellite to which it has been assigned, while d_i represents the demand of customer i. According to our tests, the best rules are the following
   (a) **25C_75D**. The parameters are set $\alpha = 0.25$ and $\beta = 0.75$.
   (b) **50C_50D**. The parameters are set $\alpha = 0.5$ and $\beta = 0.5$;
   (c) **75C_25D**. The parameters are set $\alpha = 0.75$ and $\beta = 0.25$;

This strategies are not applied sequentially. Tests for determining the most performing one among them are presented in Section 5.

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. We compare our heuristics in their best setting with the other heuristics obtained from the literature, the math-heuristics proposed in [12], as well as the best lower bounds from the literature [11]. We do not report explicitly a comparison with MIP solver, because they solve exactly only small instances (up to 32 customers and 2 satellites), while the quality of their solutions becomes very poor when instances grow up to 50 customers, making them not any more competitive. More details can be found in [12]. All the methods presented in this paper are implemented in C++ and tested on a 2.5 GHz Intel Centrino Duo, while the CVRP instances built by the different procedures are heuristically solved by the Branch and Cut method developed by [14], an exact method based on an implicit solutions enumeration with additional constraints, with a time limit of 5 seconds.

The instances we used cover up to 50 customers and 5 satellites and can be grouped into two sets:

  – *S1*. It contains all the instances of Set 2 in [12]. The set contains 21 instances obtained as extensions of data sets E-n22-k4, E-n33-k4 and E-n51-k5 for the

CVRP problem introduced in [5]. The cost matrix of each instance is given by the corresponding CVRP instance. The capacity of the 1st-level vehicles is 2.5 times the capacity of the 2nd-level vehicles, to represent cases in which the 1st-level is made by trucks and the 2nd-level is made by smaller vehicles (e.g., vehicles with a maximum weight less than 3.5 t). The capacity and the number of the 2nd-level vehicles is equal to the capacity of the vehicles of the corresponding CVRP instance. The satellites are located at the same position of some randomly chosen customers. The instances range between 21 and 50 customers and consider 2 or 4 satellites.

– *S2*. Instances taken from [7]. We consider the instances with 50 customers, combining three customer distributions and three satellites location patterns, with 2, 3, and 5 satellites.

Preliminary computational tests on a small subset of S2 have been effectuated in order to determine the behavior of the different rules used in the Perturbed Solution Generation and Feasibility Search procedures. (For the detailed results, see [8]). From the point of view of the perturbation, a better behavior of the Majority Prize rule while from the point of view of the Feasibility Search, the best results are given by the rules which linearly combine cost and demand($25C\_75D$, $50C\_50D$, $75C\_25D$). We also performed a tuning of the parameters involved in the different procedures. We do not report the detailed results, but, according to our tests, the best values are the following: $\delta = 0.1$, $\gamma = 0.1$, $r = 0.5$, $p = 0.8$, $ITER = 100$.

5.1 Comparison with State-Of-The-Art Algorithms

In this section, we compare the results of First Clustering, Clustering Improvement and Multi-Start heuristics with the two math-heuristics by [12], the Diving and the Semi-Relaxed heuristics, as well as with the best lower bounds taken from the literature ([12], [11]). Due to the different workstation used, in order to make the computational times comparable we scale the results for the math-heuristics, as well and the lower bounds to a 2.5 GHz Intel Centrino Duo by means of the SPECINT benchmarks [16].

The results obtained on sets S1 are reported in Table 1, which is organized as follows:

– Columns 1-3 and 10-12. Instance name (E-nx-ky-sa-b-c-d , where x indicates the number of customers, y the maximum number of vehicles and letters from a to d, the customers at which the satellites is located), number of customers, and number of satellites.
– Columns 4 and 5. Objective function and computational time in seconds obtained by the First Clustering.
– Columns 6 and 7. Objective function and computational time in seconds obtained by the Clustering Improvement.
– Columns 8-9. For the best version of the Multi-Start ($50C_50D$), we give the objective function and computational time.

- Columns 13-16. We report the results of the state-of-the-art algorithms. More precisely, DIVING and SEMI columns refer to the Diving and the Semi-Relaxed heuristics.[12].
- Column 17-18. Objective function and computational time of the composite heuristic obtained by taking the best between Diving and Semi-Relaxed heuristics.
- Column 19. Column BEST LB gives the best lower bound computed for each instance ([12], [11]).

Values in bold correspond to optimal solutions. For each method we report the single values for each instance. The last three rows give a summary of the results of each method, providing the sum of the objective functions, the average computational time, the percentage improvement with respect to CI and the percentage gap with the results from the literature. The overall best of each instance is emphasized. If it has been obtained by two or more methods, we consider as overall best the one obtained within the lower computational time. As far as set S1 analysis is concerned, it can be noticed that the different versions of the Multi-Start heuristic perform sensibly better than DIVING (around 4%) and SEMI (around 2%) with a smaller computational effort. Even CI outperforms the math-heuristics of 2.97% and 0.75%, respectively, with a reduction of the computational effort of two order of magnitude. If we compare our results with the composite heuristic which consider the best of the two math-heuristic, the Multi-Start heuristics still improve of more than 1%. Furthermore, if we consider the results instance by instance, we notice how our heuristics reach the overall best in the 59% of the cases, with an average improvement of the literature of 2.63%.

Tables reporting results obtained on S2 can be found in [8]. All our Multi-Start methods perform sensibly better than Diving (more than 3%) and Semi-Relaxed (more than 1%) in quite smaller computational times. If compared with the best solution from literature, Multi-Start procedures obtain very similar results within a computational time one order of magnitude lower. The overall best is reached in 53% of the cases and yield to an averaged improvement of the literature of 3.44%.

6   Conclusions

In this paper, we presented a family of Multi-Start heuristics for the Two-Echelon Vehicle Routing Problem, a newly defined Multi-Echelon variant of the classical CVRP. The experimental results have shown that they all perform well, particularly considering the very limited computational effort needed by our algorithms, and are more efficient than the other heuristic methods from the literature. 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 will address larger instances and meta-heuristic frameworks working on neighborhoods directly based on the customer positioning

Table 1. Computational results for set S1

INSTANCE	Cust	Sat	FC		CI		MP/50C_50D	
			OF	TIME	OF	TIME	OF	TIME
E-n22-k4-s6-17	21	2	424.89	0.13	424.89	1.590	**417.07**	16
E-n22-k4-s8-14	21	2	386.36	0.14	384.96	0.456	**384.96**	9
E-n22-k4-s9-19	21	2	485.12	0.48	485.12	0.763	472.23	20
E-n22-k4-s10-14	21	2	375.91	0.14	375.91	0.703	375.91	7
E-n22-k4-s11-12	21	2	453.77	0.34	453.77	1.180	444.83	15
E-n22-k4-s12-16	21	2	425.65	0.16	425.65	0.887	403.79	26
E-n33-k4-s1-9	32	2	774.54	0.11	774.54	3.420	757.56	20
E-n33-k4-s2-13	32	2	745.39	0.11	745.39	2.730	733.18	25
E-n33-k4-s3-17	32	2	810.83	0.25	801.21	3.400	754.65	28
E-n33-k4-s4-5	32	2	796.50	2.10	796.50	8.830	792.89	19
E-n33-k4-s7-25	32	2	775.85	0.12	756.88	1.880	756.88	15
E-n33-k4-s14-22	32	2	833.30	0.17	825.06	2.600	824.60	16
E-n51-k5-s2-17	50	2	614.17	0.24	614.17	0.566	614.17	12
E-n51-k5-s4-46	50	2	544.70	2.60	533.83	2.790	533.83	46
E-n51-k5-s6-12	50	2	562.21	0.27	559.00	0.586	564.92	32
E-n51-k5-s11-19	50	2	612.14	0.27	597.90	0.567	597.90	19
E-n51-k5-s27-47	50	2	553.77	0.23	553.77	0.352	553.77	17
E-n51-k5-s32-37	50	2	558.48	0.15	558.48	0.404	555.05	33
E-n51-k5-s2-4-17-46	50	4	566.60	0.12	565.00	0.138	565.00	5
E-n51-k5-s6-12-32-37	50	4	573.01	0.28	567.00	0.560	567.00	6
E-n51-k5-s11-19-27-47	50	4	618.52	0.20	600.00	0.640	600.00	3
SUM/AVG TIME			12491.71	0.41	12399.03	1.67	12270.19	18.52
IMPROVEMENT (CI)							1.04%	
GAP (LIT)			0.62%		-0.13%		-1.16%	

INSTANCE	Cust	Sat	DIVING		SEMI		LIT		BEST LB
			OF	TIME	OF	TIME	OF	TIME	
E-n22-k4-s6-17	21	2	417.07	7	417.07	14	**417.07**	21	417.07
E-n22-k4-s8-14	21	2	441.41	9	408.14	7	408.14	16	384.96
E-n22-k4-s9-19	21	2	472.23	6	470.60	10	**470.60**	16	470.60
E-n22-k4-s10-14	21	2	435.92	8	440.85	0.1	435.92	9	371.50
E-n22-k4-s11-12	21	2	487.45	6	429.39	10	429.39	16	427.22
E-n22-k4-s12-16	21	2	425.65	7	439.19	8	425.65	15	392.78
E-n33-k4-s1-9	32	2	772.57	29	736.92	2	736.92	31	730.16
E-n33-k4-s2-13	32	2	749.94	28	736.37	6	736.37	34	714.63
E-n33-k4-s3-17	32	2	801.19	68	739.47	5	739.47	73	707.41
E-n33-k4-s4-5	32	2	838.31	18	816.59	12	816.59	31	778.73
E-n33-k4-s7-25	32	2	756.88	18	756.88	42	756.88	59	756.84
E-n33-k4-s14-22	32	2	779.06	13	779.06	4	779.06	17	779.05
E-n51-k5-s2-17	50	2	666.83	75	628.53	567	628.53	641	576.97
E-n51-k5-s4-46	50	2	543.24	72	534.04	257	534.04	329	529.34
E-n51-k5-s6-12	50	2	560.22	69	554.80	60	554.80	130	541.17
E-n51-k5-s11-19	50	2	584.09	49	592.06	247	584.09	296	558.27
E-n51-k5-s27-47	50	2	538.20	85	538.20	224	538.20	310	535.04
E-n51-k5-s32-37	50	2	584.59	84	587.12	557	584.59	640	552.27
E-n51-k5-s2-4-17-46	50	4	590.63	280	542.37	1057	542.37	1338	515.75
E-n51-k5-s6-12-32-37	50	4	571.80	112	584.88	936	571.80	1048	516.02
E-n51-k5-s11-19-27-47	50	4	724.09	118	724.09	555	724.09	673	511.09
SUM/AVG TIME			12741.37	55.30	12456.62	218.16	12414.57	273.46	11766.86
IMPROVEMENT (CI)									
GAP (LIT)									

inside the routes, instead of acting on the assignments. For a more detailed discussion of results we refer to [8].

References

1. Albareda-Sambola, M., Diaz, J., Fernandez, E.: A compact model and tight bounds for a combined location-routing problem. Computers & Operations Research 32, 407–428 (2005)
2. Barreto, S., Ferreira, C., Paixao, J., Souza Santos, B.: Using clustering analysis in a capacitated location-routing problem. European Journal of Operation Research 179, 968–977 (2007)
3. Belenguer, J., Benavent, E., Prins, C., Prodhon, C.: A branch-and-cut method for the capacitated location-routing problem. Computers & Operation Research 38-6, 931–941 (2010)
4. Boulanger, C., Semet, F.: A column generation based heuristic for the capacitated location-routing problem. In: EU/MEeting (2008)
5. Christofides, N., Mingozzi A., Toth, P.: The Vehicle Routing Problem. In: Christofides, N., A., M., Toth, P., Sandi, C. (eds.) Combinatorial Optimization, pp. 315–338. John Wiley, New York (1979)
6. Crainic, T.G., Ricciardi,N. and Storchi, G.: Models for evaluating and planning city logistics systems. *Transportation Science*, 43:432–454, 2009.
7. Crainic, T.G., Mancini, S., Perboli, G., Tadei, R.: Two-echelon vehicle routing problem: A satellite location analysis. PROCEDIA Social and Behavioral Sciences 2, 5944–5955 (2010)
8. Crainic, T.G., Mancini, S., Perboli, G., Tadei, R.: Multi-start heuristics for the two-echelon vehicle routing problems. Tech. Rep. CIRRELT-2010-30, CIRRELT (2010)
9. Gendron, B., Semet, F.: Formulations and relaxations for a multi-echelon capacitated location-distribution problem. Computers & Operations Research 36, 1335–1355 (2009)
10. Marti, R.: Handbook of Metaheuristics, vol. 57, chap. Multi-Start Methods, pp. 355–368. Springer New York (2003)
11. Perboli, G., Tadei, R., Masoero, F.: New families of valid inequalities for the two-echelon vehicle routing problem. Electronic Notes on Discrete Mathematics (2010), forthcoming, 10.1016/J.ENDM.2010.05.081
12. Perboli, G., Tadei, R., Vigo, D.: The two-echelon capacitated vehicle routing problem. Publication cirrelt-2008-55, CIRRELT Montréal, Canada (2008), and Transportation Science, forthcoming
13. Prins, C., Prodhon, C., Wolfer-Calvo, R.: Solving the capacitated location-routing problem by a grasp complemented by a learning process and a path relinking. 4OR 4, 221–238 (2006)
14. Ralphs, T.K.: Parallel Branch and Cut for Capacitated Vehicle Routing. Parallel Computing 29, 607–629 (2003)
15. Salhi, S., Nagy, G.: Location-routing: Issues, models and methods. European Journal of Operation Research 177, 649–672 (2007)
16. Standard Performance Evaluation Corporation: SPEC CPU2006 benchmarks (2006), http://www.spec.org/cpu2006/results/
17. Yu, V., Lin, S., Lee, W., Ting, C.: A simulated annealing heuristic for the capacitated location routing problem. Computers & Industrial Engineering 58, 288–299 (2010)