A Variable Neighborhood Search Approach for the Two-Echelon Location-Routing Problem

Martin Schwengerer[1], Sandro Pirkwieser[2], and Günther R. Raidl[1]

[1] Institute of Computer Graphics and Algorithms
Vienna University of Technology, Vienna, Austria
martin.schwengerer@chello.at raidl@ads.tuwien.ac.at
[2] Destion – IT Consulting OG, Vienna, Austria
pirkwieser@destion.at

Abstract. We consider the two-echelon location-routing problem (2E-LRP), a well-known problem in freight distribution arising when establishing a two-level transport system with limited capacities. The problem is a generalization of the $\mathcal{NP}$-hard location routing problem (LRP), involving strategic (location), tactical (allocation) and operational (routing) decisions at the same time. We present a variable neighborhood search (VNS) based on a previous successful VNS for the LRP, accordingly adapted as well as extended. The proposed algorithm provides solutions of high quality in short time, making use of seven different basic neighborhood structures parameterized with different perturbation size leading to a total of 21 specific neighborhood structures. For intensification, two consecutive local search methods are applied, optimizing the transport costs efficiently by considering only recently changed solution parts. Experimental results clearly show that our method is at least competitive regarding runtime and solution quality to other leading approaches, also improving upon several best known solutions.

1 Introduction

We focus on a problem in the field of freight transportation, the Two-Echelon Location Routing Problem (2E-LRP)[1]. This problem is a generalization of the classical Location Routing Problem (LRP), dealing with a two-level distribution system to deliver goods from suppliers to customers as this is beneficial (and sometimes even mandatory) for many real world applications. Basically, the 2E-LRP includes components of classical $\mathcal{NP}$-hard problems like the Facility Location Problem (FLP) and the Vehicle Routing Problem (VRP), in combination with the requirements of an additional echelon. The 2E-LRP arises when selecting locations for two types of facilities (platforms and satellites) through which goods may be transported to final customers. This includes placing two types of facilities at given locations, assigning customers to satellites, satellites to platforms, and to serve these bipartite supply chain by two fleets of vehicles. Additionally, the vehicles as well as the facilities may impose capacity constraints representing e.g. load limits. Possible extensions of the problem may deal with direct routes from the platform to customers, either by a first-level vehicle or by locating vehicles of the second-level fleet at the platforms. In contrast to the FLP, where simple out-and-back routes are used, the 2E-LRP generally allows multiple stops per route. Similar to the LRP, efficiently solving the 2E-LRP usually requires strategic (platform and satellite location), tactical (allocation) as well as operational (routing) planning decisions at the same time. Considering these aspects in a subsequent way might seem beneficial in terms of complexity but unfortunately tends to result in suboptimal solutions compared to all-embracing, nested approaches [2].

From now on, we will refer to the echelon platform/satellite part of the problem as first level and to the satellite/customer part as second level, respectively.

The 2E-LRP can be defined on a complete, undirected, weighted graph $G = (V, E)$, with $V = V_C \cup V_S \cup V_P$ being the set of nodes consisting of n customers $V_C = \{0, \ldots, n-1\}$, m potential satellites $V_S = \{n, \ldots, n+m-1\}$ and o potential platforms $V_P = \{n+m, \ldots, n+m+o-1\}$, and $E = \{\{i,j\} \mid i,j \in V_C,\ i \neq j\} \cup \{\{i,j\} \mid i \in V_C,\ j \in V_S\} \cup \{\{i,j\} \mid i \in V_S,\ j \in V_P\}$ being the set of edges. For any pair of nodes $i, j \in V$ $c_{ij} \geq 0$ represents the given travel cost. Each facility $i \in V_D \cup V_P$ has an associated maximal capacity W_i and opening costs O_i. Furthermore, two homogeneous fleets F_1 and F_2 of K_1 (K_2) vehicles, each having capacity $Q_1 > 0$ ($Q_2 > 0$), are available at the platforms (satellites). The fixed cost of using a single vehicle of F_1 (F_2) is given by $f_1 > 0$ ($f_2 > 0$), and each vehicle is limited to perform one single route. Further, each customer $j \in V_C$ has defined a demand $d_j > 0$ which has to be satisfied. Additionally, we assume that the total capacity of the satellites as well as of the platforms can satisfy the whole demand.

The 2E-LRP then deals with finding a setting of platforms and satellites to be opened as well as determining efficient vehicle routes from the platform to the satellites and from satellites to customers on G such that:

- Each route starts and ends at the same opened facility,
- each customer j is visited exactly once by a vehicle of F_2, satisfying its demand d_j,
- each satellite is visited exactly once by a vehicle of F_1, satisfying its demand $d_{satellite}$ which is the sum of the demands of the assigned customers,
- the total number of vehicle routes of the first (second) level N_1 (N_2) is less than or equal to K_1 (K_2),
- the total load of each route transported by a vehicle of the first (second) fleet does not exceed the vehicle capacity limit Q_1 (Q_2),
- for each opened platform or satellite i the total load of each route assigned to it does not exceed the facility capacity limit W_i,
- and the total costs of opening platforms and satellites, fixed costs for used vehicles, and corresponding travel costs are minimized.

In this work we adapt and extend a a successful Variable Neighborhood Search (VNS) for the LRP [3] in order to account for the additional echelon in the 2E-LRP by modifying existing neighborhood structures, introducing new ones, adapting the local search procedures, and applying some additional improvements.

The remainder is organized as follows. Related work is presented in the next section, the VNS is the topic of Section 3 and experimental results are given in Section 4. Section 5 finishes the work with concluding remarks.

2 Related Work

The 2E-LRP is a special case of the Multi-Echelon Location-Routing Problem (MELRP) having only one level of satellites between platforms and customers. In [4] Gonzalez-Feliu defines a unified notation and a general model for the MELRP and gives a literature review on diverse variants, including the 2E-LRP. A literature research on the 2E-LRP itself only reveals a rather sparse number of publications, most of them appeared in recent years although the problem itself was already described in the 80s by Jacobsen and Madsen [1] in the context of newspaper delivery. Boccia et al. [5] introduce three mixed integer programming models for the 2E-LRP and present corresponding results. The same authors further proposed a tabu search using an *iterative-nested approach*, splitting the problem in two LRPs and solving them in a bottom-up manner [6]. A different heuristic has been proposed by Nguyen et al. [7], considering a slightly modified version of the problem using only a single platform. Their approach is based on a Greedy Randomized Adaptive Search (GRASP) using four different construction heuristics, a learning process and path relinking. In [8] the same authors suggest a multi-start iterated local search with a tabu list and path relinking, which, according to their evaluations, clearly outperforms the former approach. Among these algorithms, one of the currently most effective method has been proposed by Contardo et al. in [9]. It is an Adaptive Large Neighborhood Search (ALNS) applying eight destroy and four repair operations and an embedded local search. An interesting aspect is that none of these present-day heuristics is a population-based approach. To the best of our knowledge, only the contribution by Jin et al. [10] is based on evolutionary principles using a hybrid genetic algorithm with a tabu search, which, however, tackles a two-layer LRP having infinite platform capacities. Unfortunately the authors did not provide results on available or common instances making a meaningful comparison difficult.

As the 2E-LRP consists of locating and routing components, it is closely related to various other problems in freight distribution. For example, the Two-Echelon Vehicle Routing Problem (2E-VRP), which deals with a two-level routing of vehicles, can be seen as a special case of the 2E-LRP, containing no opening costs, always opened satellites and no facility capacities. For the 2E-VRP, Crainic et al. [11] proposed a multi-start heuristic by separately solving the platform-to-satellite and satellite-to-customer subproblems. A different approach by Hemmelmayr et al. uses an ALNS [12] with destroy and repair components and embedded local search methods. Another related problem is the Two-Echelon Single-sourced Capacitated Facility Location Problem, considering two-level facility location and allocation similar to the 2E-LRP, for which Tragantalerngsak et al. proposed six heuristics based on Lagrangian relaxation [13]. A further related problem is the Two-Echelon Uncapacitated Facility Location Problem, in which satellites and customers can be supplied by multiple sources. Gao and Robinson [14] introduce a dual-based branch and bound algorithm for it. For more details about two-echelon freight transport optimization we refer to [15].

Finally, we remark that the classical LRP can be considered a special case of the 2E-LRP in which the first level is removed, e.g. by using only one platform and zero transport costs to the satellites; two recents works are reported in [3, 16]. In fact, the current work is methodically based on the former approach [3], also a VNS, sharing some features with it. Therefore we omit to give details here but refer to the next section.

3 Variable Neighborhood Search for the 2E-LRP

Variable Neighborhood Search [17] is a well-known metaheuristic exploiting a set of (not necessary disjunct) neighborhood structures. Essentially, it applies random steps in neighborhoods of growing size as diversification mechanism, called shaking, each followed by an application of an embedded local search for intensification.

As many other problems in the field of freight distribution, the 2E-LRP is particularly challenging due to some strict constraints resulting in a rugged search space and limiting the amount of possible (legal) solutions for many neighborhoods. Hence to smooth the search space, our VNS relaxes the vehicle load as well as the facility load restrictions by allowing infeasible solutions in combination with penalties for excesses of these constraints. As the same (constant) weightings might be inefficient for the whole search process, we use an *adaptive* penalty function which gradually adjusts both values. Depending on whether the search is in a feasible or infeasible region regarding a constraint, it decreases or increases the corresponding penalty, respectively. We observed that useful weights and initial values often depend on the problem instance. After some experiments, we decided to use the average traveling costs c_{avg} of the second fleet as initial value with $c_{\text{avg}}/5$ ($c_{\text{avg}} \cdot 5$) as lower (upper) bound for the penalty.

Initially, our VNS creates a set of candidate solutions by the following construction heuristic, finally picking the best one for further improvement.

1. Sum up the customer demands to get a lower bound for the accumulated facility capacities of each echelon $W_{\text{LB}} = \sum_{j \in V_C} d_j$.
2. Select satellites to be opened at random one by one until their combined capacity is greater than or equal to W_{LB}.
3. Select platforms to be opened at random one by one until their combined capacity is greater than or equal to W_{LB} (as the satellite demand never exceeds the customer demand).
4. Assign each customer to be visited by a vehicle of its nearest opened satellite, thereby considering the penalties for satellite load violations.
5. Apply the well-known Clarke and Wright savings algorithm [18] until no more routes can be feasibly merged due to the vehicle load restriction.

6. If we end up with more than K_2 second-level routes, least customer routes are selected and customers contained therein are re-added in a greedy way, allowing (penalized) excess of vehicle load.
7. Assign each opened satellite with at least one route to be visited by a first-level vehicle of its nearest opened platform, considering penalties for platform load violations.
8. Apply step 5 in an analogous way on the first-level routes.
9. Similar to the second-level fleet, in case of ending up with more than K_1 first-level routes, least satellite routes are selected and satellites contained therein are re-added in a greedy way.

The reader may notice that our initialization method is rather simple in contrast to others like those suggested in [8, 7]. However, a better initial solution does not necessarily lead to a better final result, in fact it might even be counterproductive for the subsequent heuristic search. For example also [9], one of the so far leading approaches, uses a comparable initialization procedure, and similar observations hold for problems like the LRP [3].

For shaking we define seven different types of neighborhood structures via possible moves on a current solution. These are used with several perturbation sizes, denoted by δ, providing a total of 21 shaking neighborhoods (i.e. $k_{\max} = 21$). Note however, that not all of them are always useful. Some of our benchmark instances deal with a single, fixed platform, which renders some neighborhoods rather useless. Our implementation detects such cases automatically and skips the corresponding neighborhoods. In the following these seven basic neighborhood structures are described.

Exchange-segments: Exchange two random segments of variable length between two routes at the same facility (platform or satellite). This includes also a customer (or satellite) relocation, as one of the segments might be empty. The facility is selected at random with a probability directly (i.e. in a one to one fashion) according to the number of supplied customers or satellites. Since satellites usually serve more customers than platforms serve satellites, the former are selected more often, leading to the desired focus on second-level routes.

Exchange-segments-two-facilities: Exchange two random segments of variable length where both segments are located at two distinct facilities in the same echelon. The selection criterion for the first facility, and hence the type of echelon, is as for *exchange-segments*, whereas the second facility of the same type is then determined using equal probabilities. In case of only one opened platform, always two satellites are selected (if possible) and if only a single satellite is opened, *exchange-segments* is applied. This neighborhood structure facilitates the (re-)assignment of customers (satellites) to the opened satellites (platforms).

Change-two-satellites: Opens a currently closed satellite and closes an opened satellite, ensuring that the actual lower bound regarding the satellite capacity W_{LB} is still satisfied. After opening the selected new satellite a relocation procedure is applied which tries to relocate routes in a cost saving manner by removing the old satellite of a route, connecting the first and last customer and placing the new satellite at minimum cost between two consecutive customers. In case no route is relocated, a more investigative method is used, trying to reduce costs by relocating single customers. Afterwards the routes of the satellite to be closed are relocated to an opened satellite one by one in the least expensive way, taking potential penalties into account. After such a neighborhood move the number of opened/closed satellites stays the same. In case all available satellites are already opened, the following *change-satellite* is applied instead.

Change-satellite: This neighborhood deals with opening/closing a single satellite. A satellite is selected at random for changing its status (opened to closed or vice versa), thereby taking care of maintaining the required lower bound regarding the satellite capacity W_{LB}. As for *change-two-satellites*, it is tried to relocate customers to a newly opened satellite and shift routes from a closed satellite. This neighborhood move allows to alter the number of opened/closed satellites.

Change-two-platforms: Similar to *change-two-satellites*, in this move a closed platform is opened and subsequently a different opened platform is closed, ensuring that the actual lower bound regarding the platform capacity W_{LB} is still satisfied. Again, after opening a platform it is tried to minimize total costs by rerouting complete routes to it. In a subsequent step, the routes of the closed platform are relocated to open platforms one by one in a greedy way, allowing but penalizing excesses of platform capacities. After such a move, the number of opened/closed platforms stays the same. In case all available platforms are already opened, the following *change-platform* is applied instead.

Change-platform: The platform-specific counterpart of *change-satellite* selects a platform at random and changes its status from opened to closed or vice versa. In case of opening, existing first-level routes are relocated to the newly opened platform in a cost-saving manner while in case of closing, the assigned routes are relocated to other opened platforms. Also here it is ensured that the sum of platform capacities is equal to or greater than W_{LB} by restricting the search space to valid moves, which also may result in infeasible neighboring solutions.

In this work we consider a fixed shaking neighborhood order, detailed in Table 1. A greater focus is laid on exchanging segments and selecting optimal satellites, as these two parts play a major role for obtaining good solutions, while rather drastic changes involving whole platforms occur less often.

For intensification, each newly derived solution is subject to the well-known 3-opt intra-route exchange method, considering only recently changed routes. As a second method, we propose a 2-opt* [19] inter-route exchange local search, which tests for each pair of routes if exchanging their end segments would lead to a better solution. Both methods are applied in a best improvement manner until a local optimum is encountered. Since 2-opt* is considerably more demanding than 3-opt it is primarily applied on new incumbent solutions, but additionally with a probability of 0.2 on newly derived solutions lying within 3% to the current incumbent.

Although this basic VNS already performs relatively well, it seems that besides always accepting better solutions as usual, also considering solutions having

Table 1. Order and parametrization of the shaking neighborhoods as used by the VNS.

k	$\mathcal{N}_k$
1–5	*Exchange-segments* of maximal length $\delta = k$
6	*Exchange-segments* of maximal lengths bounded by corresponding route size
7–11	*Exchange-segments-two-facilities* of maximal length $\delta = k - 6$
12	*Exchange-segments-two-facilities* of maximal lengths bounded by corresponding route size
13–15	*Change-two-satellites* is applied up to $\delta = k - 12$ times
16–18	*Change-satellite* is applied up to $\delta = k - 15$ times
19–20	*Change-two-platforms* is applied up to $\delta = k - 18$ times
21	*Change-platform* is applied once

worse costs sometimes can boost efficiency. This is done in a systematic way using the Metropolis criterion as was previously done in [20, 3], originating from simulated annealing [21]. Similar to the penalty weights we use an instance specific value for the initial temperature $T_0 = c_{\text{avg}}/10$ and apply a linear cooling scheme, decreasing the value after every 100 iterations by $T_0 \cdot 100/i$, with i iterations in total as a limit. In addition, we further enhance this method with a *reheating* procedure which resets the temperature to its initial value in case no improvements occurred over a longer period, taking $i/8$ iterations for it.

4 Experimental Results

Our algorithm has been implemented in C++, compiled with GCC 4.6 and performed on a single core of an Intel Xenon E5540 with 2.53 GHz. For the evaluation we took three benchmark data sets already used in previous work, comprising 147 instances in total. Two sets were proposed by Nguyen [8] and are provided by Prodhon [22]: the first set *Prodhon* is an extension of 30 instances originally created for the LRP by adding a single platform at coordinates $(0, 0)$. The instances are named n-m-#clusters[b][BIS], containing 20 to 200 customers n, either 5 or 10 satellites m, up to three customer clusters, and having different vehicle capacities ('b' denotes high) as well as separated clusters (denoted by 'BIS'). The second set *Nguyen*, with a naming schema n-m-$\{N, NM\}$), consists of 24 instances having also only one platform, between 25 and 200 customers and 5 or 10 satellites; 'N' denotes a normal distribution while 'NM' a multi-normal distribution. Finally, the third set *Sterle* was generated according to [6] and provided by Sterle [23]. It consists of three (sub-)sets of instances, called I1, I2 and I3, containing different distributions of the satellites. These instances have multiple platforms with restricted capacities and contain 8 to 200 customers, 3 to 20 satellites and 2 to 5 platforms; they are denoted as *I[1|2|3]-o-m-n*.

Our VNS is run for $i = 5 \cdot 10^6$ iterations for sets Prodhon and Nguyen, and for $i = 7.5 \cdot 10^6$ iterations for set Sterle; the difference is due to obtaining comparable runtimes to previous methods to perform a fairer comparison. Initially we create 10 candidate solutions and pick the best one as start solution of the VNS. Penalty weights are adapted by multiplying or dividing them by 1.000005 depending on whether or not the solution at hand is feasible with respect to the corresponding constraint, respectively. This factor might seem quite small, but we opted for a smooth adaptation, which was also confirmed to work well by preliminary test results.

The results for the different instance sets are shown in Tables 2–4. To obtain meaningful results we performed the VNS 20 times per instance. Column BKS states so far *best known solution* values from [9, 8], For our VNS we list minimal costs (best of 20 runs), and following average values over 20 runs: costs, corresponding coefficients of variation (i.e. standard deviations divided by average values) in percent (CV [%]), average CPU-times in seconds (t [s]) and average times for obtaining the best solutions in the corresponding runs ($t_{\min}$ [s]). Due to limited space we, on the one hand, give in these tables also the costs of the overall best solutions we obtained, i.e. also including further, differently parameterized test runs, and, on the other hand, state for Sterle's instances in Table 4 only those having ≥ 50 customers or where a new BKS could be obtained; full details are provided via an online supplementary [3]. Results printed bold improve upon the BKS. Furthermore, best, minimum and average costs are also expressed as percentage gaps w.r.t. BKS.

The results indicate that the VNS performs very well in general, reaching on small instances almost every time the optimal or best known solution (for a list of proven optimal solutions, we refer to [9]). In detail, our approach was able to reach the BKS in all 71 instances with less than 50 customers at least once and could even improve two of them. On the 76 larger instances with $n \geq 50$ customers, the VNS was able to find 35 of the previous BKS and even 30 new, improved solutions. Concerning the average results of our 20 runs per instance, the solution quality is only 0.45% worse than the previously known BKS with an average coefficient of variation of 0.40%. Since larger instances are harder to solve and contain more local optima in general, it is not surprising that solution values vary in these cases to a larger extent. However, it can be observed that the runtime of the VNS increases only moderately with increasing instance size, making the approach attractive for solving larger instances in relatively short time.

Table 5 displays a comparison of the VNS with other state-of-the-art approaches from literature: GRASP+PR [7], MS-ILS+PR [8], and ALNS [9]. As common reference, we now use the currently best known solutions, i.e. including those found by our method. Comparisons among runtimes must be done with care, especially Nguyen et al. used a notably slower machine (Intel Pentium 4 with 3.4 GHz), whereas those utilized by Contardo et al. is slightly faster than ours (Intel Xeon E5472 with 3.0 GHz). Note that for GRASP+PR and MS-ILS+PR, the authors present only the best solutions, and results on Sterle's set are limited to instances having at least 50 customers (marked by an asterisk in the table). It is clearly shown that the VNS outperforms both, GRASP+PR and MS-ILS+PR, although the comparisons concerning only best solutions should

[3] see at http://www.ads.tuwien.ac.at/w/Research/2E-LRP

Table 2. Results of the VNS on Prodhon's instances.

Instance	BKS	best of 20 runs		average over 20 runs					overall best	
		costs	%-gap	costs	CV [%]	%-gap	t [s]	$t_{\min}$ [s]	costs	%-gap
20-5-1	89075	89075	0.00	89075.00	0.00	0.00	63.46	1.54	89075	0.00
20-5-1b	61863	61863	0.00	61863.00	0.00	0.00	82.84	0.27	61863	0.00
20-5-2	84478	84478	0.00	84489.50	0.06	0.01	62.22	11.33	84478	0.00
20-5-2b	60838	60838	0.00	61033.80	0.68	0.32	125.43	0.00	60838	0.00
50-5-1	130843	130843	0.00	130859.30	0.04	0.01	80.10	15.85	130843	0.00
50-5-1b	101530	101530	0.00	101548.80	0.06	0.02	127.87	34.87	101530	0.00
50-5-2	131825	131825	0.00	131825.00	0.00	0.00	96.71	11.28	131825	0.00
50-5-2b	110332	110332	0.00	110332.00	0.00	0.00	198.21	11.95	110332	0.00
50-5-2BIS	122599	122599	0.00	122599.00	0.00	0.00	111.58	90.66	122599	0.00
50-5-2bBIS	105696	105696	0.00	105935.50	0.14	0.23	197.73	155.29	105696	0.00
50-5-3	128379	128379	0.00	128436.00	0.10	0.04	79.75	9.03	128379	0.00
50-5-3b	104006	104006	0.00	104006.00	0.00	0.00	131.25	6.34	104006	0.00
100-5-1	319137	**318225**	-0.29	**318667.10**	0.10	-0.15	225.73	153.13	**318134**	-0.31
100-5-1b	257349	**256991**	-0.14	257436.35	0.11	0.03	301.11	219.65	**256878**	-0.18
100-5-2	231305	231305	0.00	231340.00	0.02	0.02	203.70	131.04	231305	0.00
100-5-2b	194729	194763	0.02	194812.70	0.02	0.04	240.39	202.27	194729	0.00
100-5-3	244194	244470	0.11	245334.90	0.12	0.47	174.10	123.70	**244071**	-0.05
100-5-3b	194110	195381	0.65	195586.20	0.11	0.76	180.41	111.35	195381	0.65
100-10-1	358068	**352694**	-1.50	**357381.40**	1.02	-0.19	233.37	166.94	**351243**	-1.91
100-10-1b	297167	298186	0.34	300239.15	0.43	1.03	299.03	194.47	297907	0.25
100-10-2	305402	**304507**	-0.29	**304931.20**	0.11	-0.15	247.66	193.78	**304438**	-0.32
100-10-2b	264389	**264092**	-0.11	264592.00	0.12	0.08	307.05	207.54	**263873**	-0.20
100-10-3	313249	**311447**	-0.58	**312701.25**	0.21	-0.17	226.64	141.31	**310312**	-0.94
100-10-3b	264096	**260516**	-1.36	**261577.90**	0.22	-0.95	302.85	217.89	**260328**	-1.43
200-10-1	552816	**548730**	-0.74	**552488.90**	0.45	-0.06	1009.49	748.05	**548703**	-0.74
200-10-1b	448236	**445791**	-0.55	**448095.45**	0.43	-0.03	634.59	575.87	**445301**	-0.65
200-10-2	498199	**497451**	-0.15	513673.40	3.34	3.11	1158.23	832.45	**497451**	-0.15
200-10-2b	423048	**422668**	-0.09	432487.00	2.12	2.23	730.12	695.92	**422668**	-0.09
200-10-3	533732	**527162**	-1.23	**529578.00**	0.30	-0.78	970.42	903.46	**527162**	-1.23
200-10-3b	404284	**402117**	-0.54	404431.25	0.30	0.04	591.90	556.91	**401672**	-0.65
Average			-0.21		0.35	0.20	313.13	224.14		-0.26

be done with care. Compared to ALNS our VNS seems to be at least competitive with only marginal differences between the best and average values. Especially on Prodhon's instances the VNS provides better results on average while on the other instance sets ALNS achieves slightly better results. Also the last line of Table 5, stating how often the (new) BKS could be obtained by the corresponding method, emphasizes the good performance of the VNS.

5 Conclusions

In this work, we extended and adapted a previous variable neighborhood search (VNS) initially proposed for the location-routing problem to tackle the two-echelon location-routing problem (2E-LRP) with capacitated vehicles and facilities. Two new types of neighborhood structures were introduced dealing with the additional echelon by opening and closing platforms including vehicle route relocations. Existing neighborhood structures were modified by also considering first-level routes from platforms to satellites. For smoothing the search space, violations of capacity constraints are allowed but penalized in the objective function; corresponding penalty weights are automatically adapted. The VNS

Table 3. Results of the VNS on Nguyen's instances.

Instance	BKS	best of 20 runs		average over 20 runs					overall best	
		costs	%-gap	costs	CV [%]	%-gap	t [s]	$t_{\min}$ [s]	costs	%-gap
25-5N	80370	80370	0.00	80370.00	0.00	0.00	75.60	2.61	80370	0.00
25-5Nb	64562	64562	0.00	64562.00	0.00	0.00	90.51	0.10	64562	0.00
25-5MN	78947	78947	0.00	78947.00	0.00	0.00	61.15	0.96	78947	0.00
25-5MNb	64438	64438	0.00	64438.00	0.00	0.00	88.52	0.04	64438	0.00
50-5N	137815	137815	0.00	137815.00	0.00	0.00	116.30	35.07	137815	0.00
50-5Nb	110094	110094	0.00	110204.40	0.21	0.10	131.91	51.24	110094	0.00
50-5MN	123484	123484	0.00	123484.00	0.00	0.00	124.94	40.64	123484	0.00
50-5MNb	105401	105401	0.00	105687.00	0.38	0.27	201.51	47.57	105401	0.00
50-10N	115725	115725	0.00	115725.00	0.00	0.00	143.31	23.65	115725	0.00
50-10Nb	87315	87315	0.00	87345.40	0.10	0.03	175.86	66.35	87315	0.00
50-10MN	135519	135519	0.00	135519.00	0.00	0.00	144.06	10.33	135519	0.00
50-10MNb	110613	110613	0.00	110613.00	0.00	0.00	217.73	13.24	110613	0.00
100-5N	193228	193228	0.00	200685.05	3.18	3.86	168.12	101.32	193228	0.00
100-5Nb	158927	164156	3.29	164508.10	0.17	3.51	257.59	144.08	159429	0.32
100-5MN	204682	204682	0.00	206567.40	1.09	0.92	183.99	158.99	204682	0.00
100-5MNb	165744	165744	0.00	166357.35	0.25	0.37	314.87	246.88	165744	0.00
100-10N	212847	**212729**	-0.06	214585.60	0.62	0.82	222.99	167.17	**209952**	-1.36
100-10Nb	155489	155489	0.00	155790.60	0.17	0.19	352.13	251.36	155489	0.00
100-10MN	201275	201275	0.00	203798.05	0.88	1.25	229.30	162.90	201275	0.00
100-10MNb	170625	170625	0.00	170791.25	0.16	0.10	347.26	282.88	170625	0.00
200-10N	347395	**346181**	-0.35	349584.15	0.74	0.63	640.67	525.01	**345267**	-0.61
200-10Nb	256171	256759	0.23	264228.90	1.66	3.15	906.96	790.98	256759	0.23
200-10MN	326454	**325747**	-0.22	332207.50	1.04	1.76	453.32	440.96	**323801**	-0.81
200-10MNb	289742	**289239**	-0.17	292036.65	0.89	0.79	944.14	842.69	**287802**	-0.67
Average			0.11		0.48	0.74	274.70	183.63		-0.12

sometimes also accepts worse solutions in a simulated annealing fashion, applying reheating if no improvement occurs over some time. Thorough experimental results on available benchmark sets reveal the very promising performance of our method, achieving results that are at least competitive to leading approaches for this problem, both in terms of solution quality and required runtime. Moreover, we were able to improve upon 32 of 147 (21.8%) of the best known solutions.

It might be promising to augment the presented method with further local search heuristics or even suitably combine it with exact, possibly mixed integer programming based approaches in order to achieve further improvements.

References

1. Jacobsen, S.K., Madsen, O.B.G.: A comparative study of heuristics for a two-level routing-location problem. European Journal of Operational Research **5**(6) (1980) 378–387
2. Salhi, S., Rand, G.K.: The effect of ignoring routes when locating depots. European Journal of Operational Research **39**(2) (1989) 150–156
3. Pirkwieser, S., Raidl, G.R.: Variable neighborhood search coupled with ILP-based very large neighborhood searches for the (periodic) location-routing problem. In Blesa, M.J., et al., eds.: Hybrid Metaheuristics: 7th International Workshop, HM 2010. Volume 6373 of LNCS., Springer (2010) 174–189
4. Gonzalez-Feliu, J.: The n-echelon location routing problem: concepts and methods for tactical and operational planning. Working Papers halshs-00422492, HAL (2009)

Table 4. Results of the VNS on Sterles's instances; note that here we can only present results for a subset of all instances, but the averaged values are stated for the whole subsets each (lines "Average" and "Total Average").

Instance	BKS	best of 20 runs		average over 20 runs					overall best	
		costs	%-gap	costs	CV [%]	%-gap	t [s]	$t_{\min}$ [s]	costs	%-gap
I1-4-10-25	1607.94	**1559.36**	-3.02	**1565.92**	0.31	-2.61	139.52	12.43	**1559.36**	-3.02
I1-5-8-50	1162.44	1162.44	0.00	1175.67	0.95	1.14	168.54	47.88	1162.44	0.00
I1-5-10-50	1132.63	1132.63	0.00	1139.37	0.54	0.60	189.48	23.10	1132.63	0.00
I1-5-10-75	1540.23	1540.23	0.00	1554.63	0.41	0.94	237.73	135.45	1540.23	0.00
I1-5-15-75	1686.21	1686.21	0.00	1709.52	1.28	1.38	265.80	151.73	1686.21	0.00
I1-5-10-100	2124.09	**2123.44**	-0.03	2152.04	1.34	1.32	353.37	222.23	**2123.44**	-0.03
I1-5-20-100	1973.08	**1971.09**	-0.10	1981.38	0.59	0.42	491.68	287.53	**1971.09**	-0.10
I1-5-10-150	1883.44	1886.9	0.18	1908.02	0.76	1.31	1241.05	1020.53	**1882.6**	-0.04
I1-5-20-150	1869.53	**1841.51**	-1.50	1870.66	1.11	0.06	1358.06	930.26	**1835.30**	-1.83
I1-5-10-200	2443.8	2461.55	0.73	2488.20	0.68	1.82	1994.27	1706.85	2461.55	0.73
I1-5-20-200	2219.54	**2180.01**	-1.78	**2217.33**	0.96	-0.10	1932.42	1586.05	**2180.01**	-1.78
Average			-0.18		0.48	0.36	334.56	212.67		-0.20
I2-5-8-50	1121.13	1121.13	0.00	1122.13	0.27	0.09	175.83	83.10	1121.13	0.00
I2-5-10-50	1256.44	1256.44	0.00	1257.02	0.15	0.05	184.16	53.00	1256.44	0.00
I2-5-10-75	1691.15	1691.15	0.00	1702.61	0.45	0.68	240.27	83.72	1691.15	0.00
I2-5-15-75	1644.79	1743.46	6.00	1760.23	0.51	7.02	281.80	154.31	1743.46	6.00
I2-5-10-100	2231.21	2242.67	0.51	2281.94	0.57	2.27	332.43	232.81	2242.67	0.51
I2-5-20-100	1996.34	1999.61	0.16	2016.66	0.52	1.02	465.89	332.72	1999.61	0.16
I2-5-10-150	1728.05	1728.23	0.01	1746.83	0.90	1.09	951.50	774.49	1728.23	0.01
I2-5-20-150	1630.29	**1615.26**	-0.92	1636.21	0.95	0.36	1106.31	740.71	**1615.26**	-0.92
I2-5-10-200	2147.51	2155.78	0.39	2184.76	1.37	1.73	1522.63	1224.11	2155.78	0.39
I2-5-20-200	2049.01	**2039.67**	-0.46	2086.87	1.26	1.85	1807.87	1537.49	**2035.6**	-0.65
Average			0.18		0.40	0.66	304.07	179.94		-0.06
I3-2-8-25	951.59	**951.56**	< 0.01	**951.56**	0.00	< 0.01	120.65	23.55	**951.56**	< 0.01
I3-5-8-50	1162.44	1162.44	0.00	1174.18	1.03	1.01	163.75	47.81	1162.44	0.00
I3-5-10-50	1207.31	1207.31	0.00	1210.20	0.29	0.24	190.60	55.88	1207.31	0.00
I3-5-10-75	1721.47	1721.47	0.00	1727.94	0.31	0.38	245.00	117.26	1721.47	0.00
I3-5-15-75	1483.14	**1478.92**	-0.28	1486.39	0.62	0.22	256.56	152.54	**1478.92**	-0.28
I3-5-10-100	2178.35	**2177.86**	-0.02	2203.79	1.07	1.17	345.54	250.82	**2177.86**	-0.02
I3-5-20-100	2035.37	**2022.55**	-0.63	2037.09	0.67	0.08	480.41	336.27	**2022.55**	-0.63
I3-5-10-150	1274.44	1284	0.75	1293.43	0.64	1.49	904.36	611.32	1276.2	0.14
I3-5-20-150	1235.86	1240.77	0.40	1253.21	0.92	1.40	984.91	779.43	1235.86	0.00
I3-5-10-200	1766.46	1768.35	0.11	1788.41	0.79	1.24	1441.66	1213.78	**1765.33**	-0.06
I3-5-20-200	2531.21	**2515.8**	-0.61	2554.57	0.76	0.92	1944.09	1589.40	**2514.28**	-0.67
Average			-0.01		0.27	0.29	287.63	171.57		-0.05
Total Average			0.0		0.39	0.44	308.75	188.06		-0.11

5. Boccia, M., Crainic, T.G., Sforza, A., Sterle, C.: Location-routing models for designing a two-echelon freight distribution system. Technical Report CIRRELT-2011-06, University of Montreal (2011)
6. Boccia, M., Crainic, T., Sforza, A., Sterle, C.: A metaheuristic for a two echelon location-routing problem. In Festa, P., ed.: Experimental Algorithms. Volume 6049 of LNCS. Springer (2010) 288–301
7. Nguyen, V.P., Prins, C., Prodhon, C.: Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. European Journal Of Operational Research **216**(1) (2012) 113–126
8. Nguyen, V.P., Prins, C., Prodhon, C.: A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. Engineering Applications of Artificial Intelligence **25**(1) (2011) 56–71
9. Contardo, C., Hemmelmayr, V.C., Crainic, T.G.: Lower and upper bounds for the two-echelon capacitated location routing problem. Technical Report CIRRELT-

Table 5. Comparison of leading approaches for the 2E-LRP; an asterisk marks the limited sets of Sterle (and corresponding results) considered by Nguyen et al.

Set	GRASP+PR		MS-ILS+PR		ALNS			VNS		
	%-gap$_{\min}$	t_{avg}	%-gap$_{\min}$	t_{avg}	%-gap$_{\min}$	%-gap$_{\text{avg}}$	t_{avg}	%-gap$_{\min}$	%-gap$_{\text{avg}}$	t_{avg}
Prodhon	1.79	14.20	0.94	178.30	0.32	0.83	465.82	0.08	0.50	313.13
Nguyen	1.52	19.70	0.71	112.20	0.15	0.48	191.98	0.26	0.89	274.70
I1	-	-	-	-	0.24	0.41	306.76	0.05	0.60	334.56
I2	-	-	-	-	0.25	0.45	331.01	0.25	0.72	304.07
I3	-	-	-	-	0.05	0.22	329.86	0.04	0.35	287.63
I1*	6.24	-	3.39	917.10	0.42	0.96	839.60	0.16	1.31	835.43
I2*	5.84	-	3.06	928.00	0.79	1.39	913.70	0.76	1.81	745.16
I3*	6.42	-	1.76	935.10	0.17	0.66	909.85	0.14	0.98	698.53
Average	-	-	-	-	0.20	0.48	330.03	0.14	0.61	302.82
Average*	3.36	-	1.60	426.73	0.37	0.86	538.26	0.28	1.10	461.64
No. BKS	8*		14*		111			136		

2011-63, University of Montreal (2011)

10. Jin, L., Zhu, Y., Shen, H., Ku, T.: A hybrid genetic algorithm for two-layer location-routing problem. In: New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on. (2010) 642–645
11. Crainic, T.G., Mancini, S., Perboli, G., Tadei, R.: Multi-start heuristics for the two-echelon vehicle routing problem. Technical Report CIRRELT-2010-30, University of Montreal (2010)
12. Hemmelmayr, V.C., Cordeau, J.F., Crainic, T.G.: An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Technical Report CIRRELT-2011-42, University of Montreal (2011)
13. Tragantalerngsak, S., Holt, J., Rönnqvist, M.: Lagrangian heuristics for the two-echelon, single-source, capacitated facility location problem. European Journal of Operational Research **102**(3) (1997) 611–625
14. Gao, L.L., Robinson Jr., E.P.: A dual-based optimization procedure for the two-echelon uncapacitated facility location problem. Naval Research Logistics **39**(2) (1992) 191–212
15. Gonzalez-Feliu, J.: Two-echelon freight transport optimisation: unifying concepts via a systematic review. Post-Print halshs-00569980, HAL (2011)
16. Contardo, C., Cordeau, J.F., Gendron, B.: A grasp + ilp-based metaheuristic for the capacitated location-routing problem. Technical Report CIRRELT-2011-52, University of Montreal (2011)
17. Hansen, P., Mladenović, N., Brimberg, J., Moreno Pérez, J.A.: Variable neighborhood search. In Gendreau, M., Potvin, J.Y., eds.: Handbook of Metaheuristics, 2nd edition. Springer (2010) 61–86
18. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Operations Research **12**(4) (1964) 568–581
19. Potvin, J.Y., Rousseau, J.M.: An exchange heuristic for routeing problems with time windows. Journal of the Operational Research Society **46** (1995) 1433–1446
20. Hemmelmayr, V.C., Doerner, K.F., Hartl, R.F.: A variable neighborhood search heuristic for periodic routing problems. European Journal of Operational Research **195**(3) (2009) 791–802
21. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598) (1983) 671–680
22. Prodhon, C.: `http://prodhonc.free.fr/` (December 2011)
23. Sterle, C.: Private communication (2011)