2013 46th Hawaii International Conference on System Sciences

A Well-Arranged Simulated Annealing Approach for the Location-Routing Problem with Time Windows

Halil Ibrahim Guenduez
RWTH Aachen University
guenduez@or.rwth-aachen.de

Hueseyin Memet Kadir
RWTH Aachen University
hueseyin.kadir@rwth-aachen.de

Abstract

Location and route planning are implemented independently in most distribution networks. Low-quality solutions are obtained if sequential methods, e.g. locate depots first and plan routes second, are used. In this paper we include aspects of route planning in location planning and consider it as a location-routing problem (LRP). Here, we present a well-arranged simulated annealing approach for a single-stage LRP with time windows and compare its performance with a sequential and a tabu search heuristic. Although the presented approach improves the costs of the tabu search approach slightly the computational time is reduced significantly.

1. Introduction

The European logistics market is one of the most important economic factors and indicators with a market volume of € 930 billion in 2010 [10]. Transportation and warehousing represent the largest part of the market volume with a share of 42% and 26%, respectively. Thus, in distribution logistics the bulk of costs are comprised of location, transportation, and handling costs. To reduce them, facility location and route planning strategies are crucial choices. Both the capacitated facility location problem (CFLP) and the vehicle routing problem (VRP), have been studied and solved intensively over the last decades as individual models. Salhi and Rand [19] revealed that the independent tackling of both problems leads to suboptimal solutions. The set of location-routing problems within location theory combines location and route plan decision levels. In recent years, the attention has increased and many of the published works deal with real problems. For example, military [13], evacuation [4], and the paper industry [7] to name just a few. Especially in postal logistics, a huge variety of LRP applications exists (e.g. [3], [21]). An overview of the current state of scientific research is provided by Nagy and Salhi [15].

In this paper we consider a single-stage location-routing problem with time windows (SSLRPTW) and capacity restrictions at both customers and depots, as presented by Guenduez [6]. Time restrictions in LRP have hardly been considered up to now. Jacobsen and Madsen [9] consider a latest delivery time at customers in a two stage newspaper distribution network. Wasner and Zaepfel [21] mainly restrict the routes to maximum time duration in a parcel distribution network. In some real problems, external truck companies perform the routing, and desire to implement route plans for a long period of time whereby, of course, time windows of customers are respected. Executing routes for a long period of time improves the ability of external truck companies to schedule their staff and vehicle fleet more efficiently and reduces extreme planning fluctuations. Furthermore, this leads to cost reductions, from which clients also can benefit during contract negotiations. Thus, the routing in these cases is tactical rather than operational. The considered problem is derived from a real application case of a high quality mail pickup and delivery service for special customers in an urban area of Germany. In Germany a next-day delivery is possible if letter and bulk mail are posted before approximately 5 p.m. An early reception of mail enables law chambers, offices, and agencies to deal with their customers' business in the morning and afternoon, and to post the return mail before 5 p.m. Thus, they are able to give quick response without having to use additional express services if the delivery is early in the morning and the pickup up is before 5 p.m. In fact, daily time windows exist between approximately 6-8 a.m. for delivery and between approximately 3-5 p.m. for pickup. In Fig. 1 the executed network is depicted. The mail is delivered from sorting centers to small depots, and vice versa. In the case of delivery, they are used as breakup sites and in the case of pickup, they are used as consolidation sites. The depicted network includes a transportation stage between sorting centers and depots, where also other major freight is shipped to or from other

DOI 10.1109/HICSS.2013.69

locations. Therefore, this transportation stage can be omitted for our case. Finally, the question of 'where to locate the depots' leads to the SSLRPTW.

Guenduez [6] proposed a heuristic based on tabu search and showed that a simultaneous decision on depot locating and route planning leads to much better solutions than sequential decisions: locating depots first and building route plans after. Guenduez used a systematic complete neighborhood search of a solution to investigate the solution space. Although the used neighborhood is restricted to small fraction of the whole solution space, the main flaw is a high computation time caused by the systematic complete neighborhood search. Therefore, we propose a well-arranged simulated annealing approach based on ideas by Osman [16] for the vehicle routing problem and Osman and Christofides [17] for the capacitated clustering problem and compare it with Guenduez' tabu search algorithm. They observed that search strategies checking neighbors only randomly by doing random moves tend to observe parts of the solution space intensively and leave other parts of the solution space unobserved. To overcome this flaw the main idea is to combine a systematic neighborhood search with a randomized component and to apply the first improving or accepted move. In order to save computing time, the remaining neighborhood search is carried out to the applied move. The contribution of this work is to present an approach which leads to the same or better solution as Guenduez' tabu search in significantly less computation time.

This paper is organized as follows. Section 2 introduces the required notation, defines the problem, and proposes an integer linear optimization model. The subproblems are presented in section 3, followed by the explanation of the proposed hybrid heuristic in section 4. Computational results are presented in section 5. We close with some concluding remarks.

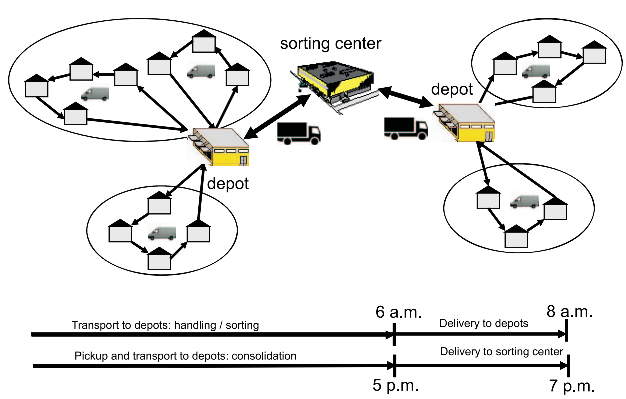


Figure 1: An example network structure

2. SSLRPTW: Definition and Model

The following definition of the single-stage location-routing problem with time windows (SSLRPTW) is borrowed from Guenduez [6]. It is defined on a weighted (not necessarily complete) directed graph $G = (V, A, C, T)$. The node set V consists of a subset D of m potential depot sites and a subset $I = V \setminus D$ of n customers. C and T are weights, corresponding to the traveling costs c_{ij} and the traveling time t_{ij} (includes service time s_i at node i), associated with the set of arcs A linking any two nodes i and j. Each depot site has a capacity Q_d, opening hours $[open_d, close_d]$, and opening costs F_d. Further, each customer has a demand q_i and has to be served during the time window $[a_i, b_i]$. For service purposes, a homogenous fleet K of vehicle with capacity C is available and any subset can be placed at any depot site. The task is to determine the location of open depots, the assignment of the customers to open depots, and the vehicle routes serving the customers with minimum overall costs such that the following constraints hold:

- Each customer is assigned exactly to one open depot and served by exactly one vehicle during the depot's time window (a waiting time w_i at customer i is allowed).
- Each vehicle is used once at the most.
- Each vehicle route begins and ends at the same open depot during the opening hours.
- The vehicle load does not exceed the vehicle capacity.
- The total demand of the customers assigned to an open depot does not exceed the depot capacity.

The following binary variables are necessary:

$$y_d = \begin{cases} 1, & \text{if depot } d \text{ is open} \\ 0, & \text{otherwise} \end{cases}$$

$$z_{di} = \begin{cases} 1, & \text{if customer } i \text{ is assigned} \\ & \text{to depot } d \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ij}^k = \begin{cases} 1, & \text{if node } j \text{ is directly} \\ & \text{visited after node } i \\ & \text{by vehicle } k \\ 0, & \text{otherwise} \end{cases}$$

Further, we need the following time variables:

T_i : arrival time at customer i
w_i : waiting time at customer i
$start_d^k$: departure time of vehicle k at depot d
end_d^k : return time of vehicle k at depot d

The linear program of the SSLRPTW can now be stated as follows:

$$\min \sum_{d \in D} F_d \cdot y_d + \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} \cdot x_{ij}^k \tag{1}$$

subject to

$$\sum_{i \in I} q_i \cdot z_{di} \leq Q_d \cdot y_d \qquad \forall d \in D \tag{2}$$

$$\sum_{k \in K} \sum_{i \in V} x_{ij}^k = 1 \qquad \forall j \in I \tag{3}$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{ji}^k = 0 \qquad \forall k \in K,\ i \in V \tag{4}$$

$$\sum_{d \in D} \sum_{j \in V} x_{dj}^k \leq 1 \qquad \forall k \in K \tag{5}$$

$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in V \setminus S} x_{ij}^k \geq 1 \qquad \forall S \subseteq I,\ 2 \leq |S| \tag{6}$$

$$\sum_{i \in V} \sum_{j \in I} q_j \cdot x_{ij}^k \leq C \qquad \forall k \in K \tag{7}$$

$$\sum_{s \in V} (x_{ds}^k + x_{si}^k) - z_{di} \leq 1 \qquad \forall d \in D,\ i \in I,\ k \in K \tag{8}$$

$$T_i + w_i \geq a_i \qquad \forall i \in I \tag{9}$$

$$T_i \leq b_i \qquad \forall i \in I \tag{10}$$

$$start_d^k - open_d \geq 0 \qquad \forall k \in K,\ d \in D \tag{11}$$

$$close_d - end_d^k \geq 0 \qquad \forall k \in K,\ d \in D \tag{12}$$

$$-M(1 - x_{ij}^k) - (T_j - T_i - w_i - t_{ij}) \leq 0 \qquad \forall i \in I,\ j \in I,\ k \in K \tag{13}$$

$$M(1 - x_{ij}^k) - (T_j - T_i - w_i - t_{ij}) \geq 0 \qquad \forall i \in I,\ j \in I,\ k \in K \tag{14}$$

$$-M(1 - x_{dj}^k) - (T_j - start_d^k - t_{dj}) \leq 0 \qquad \forall d \in D,\ j \in I,\ k \in K \tag{15}$$

$$M(1 - x_{dj}^k) - (T_j - start_d^k - t_{dj}) \geq 0 \qquad \forall d \in D,\ j \in I,\ k \in K \tag{16}$$

$$-M(1 - x_{id}^k) - (end_d^k - T_i - w_i - t_{id}) \leq 0 \qquad \forall i \in I,\ d \in D,\ k \in K \tag{17}$$

$$M(1 - x_{id}^k) - (end_d^k - T_i - w_i - t_{id}) \geq 0 \qquad \forall i \in I,\ d \in D,\ k \in K \tag{18}$$

$$x_{ij}^k \in \{0,1\}\ \forall\ i,j \in V,\ k \in K \tag{19}$$

$$y_d \in \{0,1\} \qquad \forall\ d \in D \tag{20}$$

$$z_{di} \in \{0,1\} \qquad \forall\ d \in D,\ i \in I \tag{21}$$

$$T_i,\ w_i \in \mathbb{Z}^+ \qquad \forall\ i \in I \tag{22}$$

$$start_d^k,\ end_d^k \in \mathbb{Z}^+ \qquad \forall\ d \in D \tag{23}$$

The objective function minimizes the sum of depot and transportation costs. Capacity constraints of the open depots and the used vehicles are satisfied through inequalities (2) and (7). Constraints (3) and (4), known as 'degree constraints', guarantee the uniqueness and continuity of a route performed by a vehicle. Each vehicle is used once at the most through constraints (5). Subtours consisting of only customers are eliminated by constraints (6). Constraints (8) ensure that a customer is only served by a vehicle assigned to the same open depot. While constraints (9) and (10) imply that the arrival time (with additional waiting time) at a customer is within its time window, constraints (11) and (12) guarantee that each route starts and ends at a depot during its open hours. The arrival time, the departure time, and the return time on a route performed by a vehicle are determined by inequalities (13)-(18). If $x_{ij}^k = 1$ holds, then inequalities (13) and (14) reduce to the equation $T_j = T_i + w_i + t_{ij}$, otherwise to the relaxed inequality $-\infty \leq T_j - T_i - w_i - t_{ij} \leq \infty$. The same holds for (15)-(16) and (17)-(18) for $start_d^k$ and end_d^k instead of T_i and T_j, respectively. Finally, the integrality constraints (19)-(23) state the binary or integer nature of the decision variables. For the purpose of time variables description, the time horizon is discretized into time points and coded as integer values.
This formulation includes $O(|K| \cdot |V|^2)$ binary variables, $O(\max\{|K \cdot |D|, |I|\})$ integer variables, and $O(2^{|V|} - 2^{|D|})$ constraints. Therefore, only very small-scale instances can be solved with commercial solvers. Thus, the use of heuristics is essential for large-scale SSLRPTW instances.

3. Subproblems: Location, Allocation, and Vehicle Routing

The location of depots, the allocation of customers, and the construction of routes are essential parts of the SSLRPTW. We decompose the overall problem into the mentioned subproblems, describe them briefly in this section, and propose a solution method, which is a modular part of our approach.

3.1. Capacitated Facility Location Problem with Time Windows

If we neglect routes, the SSLRPTW reduces itsself to the capacitated facility location problem (CFLP), first introduced by Balinski [1], with additional time windows (CFLPTW) and single-sourcing constraints. The task is to determine the number and location of open depots and the assignment of the customers to open depots with a minimum sum of fixed costs and assignment costs, such that the following constraints hold:

- Each customer is assigned exactly to one open depot (single-sourcing) and is reachable before the customer's of its time window.
- The earliest departure time at a depot is after its opening time.
- The latest arrival time at a depot from each assigned customer is before its closing time.
- The total demand of the customers assigned to an open depot does not exceed the depot capacity.

The CFLPTW model is consistent with the CFLP if time feasibility checks are executed during a preprocessing phase. Therefore, we need to check whether

- $open_d + t_{dj} \leq b_j$ and
- $\max\{open_d + t_{dj}, a_j\} + t_{jd} \leq close_d$

hold, otherwise the assignment variable z_{dj} can be fixed to zero or rather arcs (d, j) and (j, d) can be removed from G. The CFLPTW with single-sourcing constraints is stated as follows:

$$\min \sum_{d \in D} F_d \cdot y_d + \sum_{(d,j),(j,d) \in A} (c_{dj} + c_{jd}) \cdot z_{dj} \tag{24}$$

subject to

$$\sum_{i \in I} q_i \cdot z_{di} \leq Q_d \cdot y_d \qquad \forall\, d \in D \tag{25}$$

$$\sum_{d \in D} z_{di} = 1 \qquad \forall\, I \in I \tag{26}$$

$$z_{di} \in \{0,1\} \quad \forall\, d \in D,\, i \in I \tag{27}$$

$$y_d \in \{0,1\} \qquad \forall\, d \in D \tag{28}$$

The objective function (24) minimizes the sum of all fixed and assignment costs. Assignment costs are defined by travel costs from depot to customer and back. Inequalities (25) guarantee that a customer can only be assigned to an open depot and that capacity constraints of the depots are respected. Constraints (26) are single-sourcing restrictions. Finally, constraints (27) and (28) define the binary assignment variables and location variables, respectively. An allocation problem with time windows (APTW) occurs if open depots have already been determined. The time feasibility check can be outsourced to a preprocessing phase. Hence, only constraints (25) and (26) remain for the open depots. Large-scale instances with 1000 customers and 362 potential depots have been solved exactly with the commercial solver IBM ILOG CPLEX Optimizer 12.2.

3.2. Vehicle Routing Problem with Time Windows

The (multi-depot) vehicle routing problem with time windows (MDVRPTW) occurs as a subproblem if depot locations and a feasible assignment of customers are known. Then, the task is to construct routes for each depot and its assigned customers with the following constraints:

- Each customer is served by exactly one vehicle during the customer's time window (waiting time is allowed).
- Each route begins and ends at the depot during the opening hours.
- The vehicle load does not exceed the vehicle capacity.

The goal is to minimize the overall transportation costs. We omit a description of a full model and refer the reader to Toth and Vigo [20]. At the beginning, we construct routes with the savings heuristic of Clarke and Wright [5] and obtain the initial solution. Each customer is served individually by a separate route. Combining two routes, serving customers i and j, results in a cost savings $S_{ij} = c_{id} + c_{dj} - c_{ij}$ with d as the serving depot. We link customers i and j with maximum positive savings such that the combined route is time and capacitative feasible. Further, we restrict the linking of customers to those who retain the traversing order of the previous two routes, because reversing a route order leads in most cases to time infeasibility. The savings procedure is applied iteratively.
Local search methods are used to improve feasible solutions. Arc-exchange operators are applied to find neighboring solutions. The following arc-exchange moves are applied: 2-opt, 2-opt*, Or-opt, relocate, exchange, and cross-exchange. The 2-opt tries to improve a route by replacing two node disjoint arcs by two other arcs. 2-opt* is similar to 2-opt, but it combines two different routes by deleting one arc of each route and replacing them by two new arcs which combine the first part of the first route with the second part of the second route and the first part of the second route with the second part of the first route. Thus, the orientation of the routes is preserved.
The Or-opt is another arc-exchange operator and a special

case of 3-opt preserving the orientation of the route. Three disjoint arcs are replaced by new three arcs, such that all three route parts preserve their orientation after concatenation with the new arcs. By using a relocate operator, a customer is moved from one route to another and by using an exchange operator, customers of different routes are simply swapped. Thus, we are able to change the assignments of customer if the relocate or exchange operation is applied to routes assigned to different depots. The used operators are described in Braeysy and Gendreau [2]. Resource extension function generalized to segments introduced by Irnich [8] are used to check the feasibility of the operators in constant time. Algorithm 1 gives an overview of how the multi-depot vehicle routing problem is solved in our work.

Algorithm 1 Multi-Depot VRPTW

```
 1: for each depot and its assigned customers do
 2:     Construct routes with the savings method.
 3: end for
 4: for each depot and its routes do
 5:     Improve routes by arc-exchange operators in the following sequence: 2-opt, 2-opt*, Or-opt, relocate, exchange, cross-exchange.
 6:     if an operator improves a solution then
 7:         Stop and update the route(s). Go to 4.
 8:     end if
 9: end for
10: for each pair of depots and their routes do
11:     Improve routes by arc-exchange operators in the following sequence: relocate, exchange .
12:     if an operator improves a solution then
13:         Stop and update the route(s). Go to 10.
14:     end if
15: end for
16: if an operator in 11 improved the solution at least once then
17:     Go to 4.
18: end if
```

4. Well-Arranged Simulated Annealing

'Locate first and route second'-type heuristics are sequential methods for LRP. In our work, we refer to the more suitable approach of nested methods, presented by Nagy and Salhi [14], and we use the sequential method as initial solution and for comparison purposes. An alternative approach could be that of iterative methods (e.g. Daskin [18]), where both problems are treated equally. These methods iterate between the location and routing phases until a stop criterion is met. The drawback of iterative methods is that the location solution space cannot be searched intensively as in the nested methods, due to an intensive routing phase for the whole distribution system after each location and allocation decision. First, we define the neighborhood structure of the SSLRPTW by the moves add, drop, and shift, introduced by Kuehn and Hamburger [11]. Opening a depot corresponds to an 'add' move and closing a depot corresponds to a 'drop' move. A 'shift' move is a simultaneous add and drop move. A depot location change mostly influences a connected area of *closely* located depots with their associated customers and the changed depot and its customers. Therefore, all three moves are restricted to a *region* and a *catchment area* of an open depot, where after each move a MDVRPTW is solved. In addition, we define a neighborhood relation between two depots to specify the term *closely* in this context. The following definitions are adopted from Nagy and Salhi [14].

Definition 1 *Two depots d_1 and d_2 are neighbors if and only if at least one customer i exists, such that d_1 and d_2 are the nearest two depots to customer i.*

Definition 2 *The region R(d) of an open depot d consists of the depot d itself, its customers, neighbor depots, and their associated customers.*

Definition 3 *The catchment area CA(d) of an open depot d is the smallest rectangle comprising depot d and its customers.*

The neighborhood relation, the region of a depot d, and its catchment can be created easily. Moreover, we restrict add and shift moves to the catchment area of an investigated depot d. Closed depots too far from depot d are not considered because they have only a slight direct influence on each other. We introduce now a well-arranged approach based on simulated annealing which accepts the first improving neighbor in a simulated annealing manner. The first neighbor with total costs less than the actual solution or with worse total costs but accepting the simulating annealing criterion becomes the new actual solution. We use Lundy and Mees' [12] flexible cooling procedure enriched with Osman's idea of resets [16]. The cooling procedure allows a parameterized temperature lowering starting with temperature T_{Start} and reducing it to T_{Stop} by α reduction steps. A temperature reduction step occurs always after comparing the actual solution costs with a neighbor's costs. At each step there is a chance of accepting a worse neighbor with the probability $e^{-\Delta}/T$: Δ is the cost difference of the actual solution's and neighbor's costs and T is the actual temperature. The search strategy is well-arranged or systematic because the generation of the neighbors to compare the actual solution with is generated by add, drop, and shift moves in a strict order. If a neighbor's costs is less than the actual solution's costs or the simulated annealing criterion accepts the neighbor's costs, the search procedure does not start over completely. Instead, the remaining add, drop, and shift moves are carried out to the the new actual solution to generate next neighbors. This search strategy was successfully applied by Osman [16] for the vehicle routing problem and by Osman and Christofides [17] for the capacitated clustering problem. Search strategies checking neighbors randomly by doing random moves tend to observe parts of the solution space intensively and leave other parts of the solution space unobserved. The systematic generation of neighbors avoids this. Using the simulated annealing criterion to decide whether to accept a neighbor or not gives us the opportunity to overcome local optima. Osman proposes another method to overcome local optima with so called 'resets'. A reset is a massive temperature increase to overcome local optima and heat the search strategy to go on checking its neighbors more intensely.

While tabu search tries to evade local optima once found by introducing a memory in the form of tabu lists, Osman's search strategy relies on the memoryless simulated annealing criterion and additional temperature resets. Temperature resets lead to non-monotonic cooling schedules.

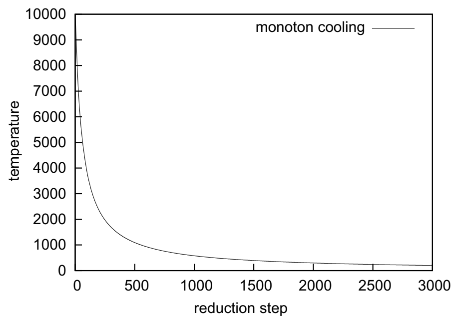
Figure 2 shows two cooling schedules: a) Monotonic cooling schedule as proposed by Lundy and Mees, b) Nonmonotonic cooling, as proposed by Lundy and Mees, enriched with three temperature resets by Osman. Both schedules cool the starting temperature $T_{Start} = 10000$ down to $T_{Stop} = 200$ in $\alpha = 3000$ reduction steps.

Algorithm 2 shows how the search strategy works. The initial solution is calculated by solving a CFLPTW first and a MDVRPTW for the depot customer assignments second. The starting point for creating and checking all neighbors is line 5. For every open depot d of the solutions S, a drop move is done in line 7, an add move is performed in line 13, and finally a shift move is checked in line 18 for every closed depot in the catchment area of the current investigated open depot. A depot is closed, a formerly closed depot becomes open, a depot is swapped with a closed depot, and the region's customers are reallocated solving an APTW. While the CFLPTW's solution leads to first depot customer assignment, where every customer is assigned to a depot, the APTW solution rearranges the assignment of customers to open depots only for small regions, as in Definition 2. Solving MDVRPTWs afterwards, leads to a new solution fragment which needs to be embedded back into the actual solution. In Algorithm 2 this is performed only if the simulated annealing acceptance criterion in the lines 8, 14, or 19 returns the boolean value 'true'. In those cases the systematic search continues for the new created actual solution with the next open depot in line 5. If the accepted solution has costs lower than the best cost value encountered, the solution will be recorded. The 'for' loop in line 5 will be triggered again if the reset counter is less than three. Resets are performed if none of the solution neighbors is accepted or the temperature falls under the minimum temperature Δ_{min}. In the first case the actual solution is a local optimum. In the latter case the search has reached its maximum neighbor comparison count. The temperature is raised to the maximum of $\Delta_{max}/2^{resetCount}$ and *temperature_best*. This ensures that the temperature is raised high enough to allow random neighbor changes to escape from local optimum.

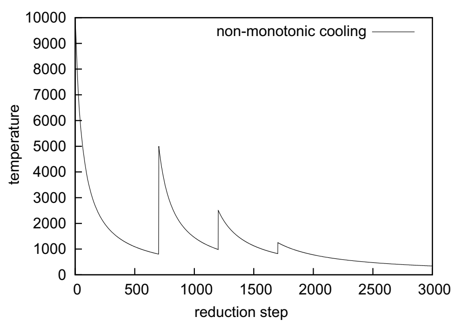
In contrast to the tabu search (TS) of Guenduez [6], we do not search the whole neighborhood for the best improving neighbor. We accept the first improving neighbor and continue the search in the regarded neighborhood with the next open depot in line 5. The algorithm presented tries to overcome local optima with memoryless methods. First, the acceptance criterion has always the chance to accept worse solutions. Second, the reset, the massive temperature increase, will mostly ensure that worse solutions will be accepted and the search procedure continues in different regions of the solution space.

Algorithm 3 shows how the simulated annealing acceptance criterion is exactly checked. The cost difference Δ between the actual solution and the regarded neighbor is calculated. If Δ is positive, the neighbor's costs value is lower than the actual solution's cost value and the acceptance criterion returns 'true'. If Δ is negative, the neighbor's costs are higher and the neighbor becomes the new actual solution only with the probability 'eValue'. 'eValue' equals to $e^{(\Delta/temperature)}$. Since Δ is negative, the expression is lower than 1. Independent of the return value, the temperature is decreased in line 2 with the help of the decrement ratio β. The starting temperature needs to be related to cost values. The adjustment of the starting temperature and other simulated annealing-specific parameters is performed once before Algorithm 2 starts. Algorithm 2 runs once without any acceptance checking just to get an impression of the cost values, their deviations for the first neighborhood, and the count of feasible neighbors. The variable 'neighborCosts' saves encountered neighbor cost values. Δ_{min} holds the minimum difference between any neighbor costs and Δ_{max} saves the maximum difference. The starting temperature then is initialized to Δ_{max}. α denotes the count of reduction steps after which the minimum temperature Δ_{min} is reached. We limit the reduction steps by either the product of the feasible solutions count in the first neighborhood with the depot count or by 20000. For 1000 customer instances, this product gets too high, so finally the reduction steps are limited to 20000. The decrement ratio β then is initialized using Δ_{min}, Δ_{max}, and α.

We call the introduced algorithm 'F1S' and create a variant named 'F1F', changing α, the number of reduction steps until Δ_{min} is reached. We also reduce the maximum number of resets to 1. α is set to $100 \cdot |D|$. For 1000 customer instances, this will lead to a maximum of 36200 neighbor comparisons, while F1S is bounded to max. 20000 comparisons. F1F cools slower than F1S but uses only 1 reset instead of 3.



(a) monotone



(b) non-monotonic

Figure 2: Cooling schedules

5. Computational Study

The proposed heuristic was coded in C++ and executed on an Intel Core2Duo 2.15 Ghz computer with 6 GB RAM. IBM ILOG CPLEX Optimizer 12.2 was applied to solve the CFLPTW and APTW.

We used SSLRPTW instances proposed by Guenduez [6], which are extensions of the VRPTW Solomon instances. The class RC1 with 1000 customers is used, in order to have large-scale instances with clustered and randomly distributed customers. As fixed costs we used 3000 monetary units (MU) combined with depot capacity of 3000 quantity units (QU). In the following, we denote the 10 instances as 'Test Class' (TC). The demand of all customers is 17822 QU and the minimum required number of open depots is 6. Overall, the generated instances consist of 362 potential depots. The vehicle capacity of 200 QU is adopted from the Solomon instances. Further, the travel costs c_{ij} and the travel time match the Euclidean distance multiplied by factor 10.

Algorithm 2 Hybrid simulated annealing

Initial solution S and total costs $C(S)$.
$S_{best} = S$, $resetCount = 0$.
while $resetCount \leq 3$ **do**
  $success = false$
  for each depot d in S **do**
    Calculate the Region $R(d)$.
    Drop open d from $R(d)$. Solve APTW and then MDVRPTW for $R(d) \setminus \{d\}$. Calculate costs $C(R(d) \setminus \{d\})$.
    if check($C(S|_{R(d)\setminus\{d\}})$, $C(S_{best})$) **then**
      $S = S|_{R(d)\setminus\{d\}}$, $t\_best = temperature$, $success = true$
      break
    end if
    for each closed depot $\widetilde{d}$ in $CA(d)$ **do**
      Add $\widetilde{d}$ to $R(d)$. Solve APTW first and then the MDVRPTW for $R(d) \cup \{\widetilde{d}\}$. Calculate the costs $C(R(d) \cup \{\widetilde{d}\})$.
      if check($C(S|_{R(d)\cup\{\widetilde{d}\}})$, $C(S_{best})$) **then**
        $S = S|_{R(d)\cup\{\widetilde{d}\}}$, $t\_best = temperature$, $success = true$
        break
      end if
      Swap $\widetilde{d}$ with d in $R(d)$. Solve APTW and then the MDVRPTW for $(R(d) \cup \{\widetilde{d}\}) \setminus \{d\}$. Calculate the costs $C((R(d) \cup \{\widetilde{d}\}) \setminus \{d\})$.
      if check($C(S|_{(R(d)\cup\{\widetilde{d}\})\setminus\{d\}})$, $C(S_{best})$) **then**
        $S = S|_{(R(d)\cup\{\widetilde{d}\}\setminus\{d\}\}}$, $t\_best = temperature$, $success = true$
        break
      end if
    end for
  end for
  if *not success* **or** $temperature \leq \Delta_{min}$ **then**
    $resetCount = resetCount + 1$
    $temperature = max(\Delta_{max}/2^{resetCount}, t\_best)$
  end if
end while

Algorithm 3 check(C,C_B)

$\Delta = C_B - C$, $eValue = e^{(\Delta/temperature)}$
$temperature = temperature/(1 + \beta \cdot temperature)$
if $\Delta > 0$ **then**
  S_{best} = Solution of C
  return true
end if
if ($\Delta < 0$ **and** $eValue > randomValue(0,1)$) **then**
  return true
else
  return false
end if

Tables 1 - 3 list the results of TC. Columns '♯ dep.' correspond to the number of open depots. Column Δ provides the deviation of the initial solution costs from the minima

Algorithm 4 Initialize SA

1: Execute Algorithm 2 without the if statements containing the execution of checkAcceptance just to gather all *neighborCosts* and the count *feasibilityCount* of feasible neighbors.
2: sort(*neighborCosts*)
3: $\Delta_{min} = neighborCosts(2) - neighborCosts(1)$
4: $\Delta_{max} = neighborCosts(last) - neighborCosts(1)$
5: $temperature = \Delta_{max}$
6: $\alpha = min(feasibilityCount \cdot |D|, 20000)$
7: $\beta = (\Delta_{max} - \Delta_{min})/(\alpha \cdot \Delta_{max} \cdot \Delta_{min})$

of the solution costs computed with one of the heuristics TS, F1S, and F1F. The three columns 'TS', 'F1S', and 'F1F' list either the percental deviation of the cost's or the computation time's minima, respectively. Since those minima are computed from the cost or time values obtained for TS, F1S, and F1F, each row contains a zero value for deviation costs and deviation times. The column with the zero entry indicates the algorithm variant with the best costs or time value. Table 3 lists the number of open depots of the best solution calculated by TS, F1S, and F1F. The computational results proof that location-routing approach leads to much better solutions than the sequential approach. The initial solution opens between 64 and 67 depots (see Table 2). The open depot count for the improved solution is reduced down to 17-21. The CFLPTW model considers round trip costs to customers and overestimates the route costs, which are drastically lowered solving MDVRPTWs. Since distances to customers are weighted quite highly in the CFLPTW model, the number of open depots is high compared to the best found solution. Therefore, time windows are less restrictive factors for the initial solution. Time windows become more restricting when depots are reduced and customers are located to them considering tour costs calculated by solving MDVRPTWs. We observed a cost reduction between 32% and 55%. This is mainly caused by the high fixed costs and the reduction of open depots. The highest cost reductions have been achieved for the instances I03 and I04.

If we compare the heuristics, we see that the maximal percental cost deviation is 1.47%. The best results are mainly found by the hybrid heuristics 'F1F'. On the other hand, the solutions of the three heuristics do not differ so much, so that we cannot say that one of the used heuristics is outperforming the others according to the best objective value. The average percental deviation from the calculated best solution for TC of TS is 0.44%, F1S is 0.43%, and F1F is 0.30%. Thus, our propose approach provides solutions of slightly better mean quality. Although the objective values calculated by F1S, F1F, and TS are tight, the solution structure can differ quite a lot, especially in the number of open depots (see Table 3). For example, in instance I04, the maximal percental cost deviation is 0.16%, but all three heuristics vary in the number of open depots. Hence, even the location of depots and the assignment of customers to tours differ considerably in local areas of the three solutions (see Figure 3). Moreover, the TS best solutions have in almost all instances at least one or more open depots than the best solutions of F1S and F1F, respectively.

Further, the comparison of the computation time shows that the presented heuristics F1S and F1F have a better performance than TS. The computation times differ a lot for the regarded instances: between 24095s and 55571s (see Table 1) for the fastest computations. As expected, F1F computes the fastest solution for the most instances. The mean computation time of TS is 66470 s, F1S is 56304 s, and F1F is 53061 s. The mean percental deviation of the best computation times of TS is 91.2%, F1S is 55.3%, and F1F is 33.1%. Overall, F1S and F1F compute on average better solutions in less computation time compared to TS.

Table 1: Calculation time results of TC

inst.	min time	deviation time		
id	seconds	TS	F1S	F1F
I01	55571.13	2.6	0	108.9
I02	41527.62	0	126.1	201.7
I03	37336.90	56.6	0	15.1
I04	45693.53	43.0	23.9	0
I05	30372.50	176.8	183.4	0
I06	24095.11	253.3	56.7	0
I07	25539.83	170.6	12.0	0
I08	39744.80	37.6	7.3	0
I09	46614.55	108.1	0	4.8
I10	32024.64	63.6	143.4	0

Table 3: Number of open depots

inst.	♯ dep.		
id	TS	F1S	F1F
I01	25	25	22
I02	21	19	21
I03	20	17	17
I04	18	15	16
I05	21	22	19
I06	21	19	19
I07	21	21	19
I08	21	18	17
I09	21	20	17
I10	20	17	17

6. Conclusion

In this paper, both location and routing problems are tackled together, leading to the single-stage location-routing

Table 2: Computational results of TC

inst.	initial solution		min costs			deviation costs		
id	♯ dep.	costs	♯ dep.	costs	Δ	TS	F1S	F1F
I01	67	352204	22	266782	32.02	0.89	0.55	0
I02	66	338148	21	246819	37.00	0.35	0.51	0
I03	67	331772	17	226089	46.74	0	0.75	0.76
I04	67	322968	16	207711	55.49	0	0.07	0.16
I05	67	347828	19	258189	34.72	0.64	1.02	0
I06	67	347118	19	256364	35.40	0	0.51	1.59
I07	64	337319	21	252069	33.82	0.65	0	0.17
I08	67	342972	17	245452	39.73	0.13	0.32	0
I09	67	343380	17	244739	40.30	0.27	0.56	0
I10	67	340189	17	238533	42.62	1.49	0	0.35

problem with time windows (SSLRPTW). This consists of locating depots, assigning customers, and planning routes with consideration of time windows and capacity restrictions at depots and customers. An initial solution is obtained by a sequential method. First, an appropriate capacitated facility location problem with time windows is formulated as an integer linear program and solved with CPLEX. Afterwards, a multi-depot vehicle routing with time windows is solved. The proposed heuristic based on simulated annealing to solve large-scale instances is well-arranged or systematic because the moves are executed in a strict order. Instead of starting over the neighborhood search after a new solution is accepted, the remaining moves are executed for the new solution. We use add, drop, and shift moves to define and to explore the neighborhood of existing solution. These moves are restricted to a defined region and catchment area of an investigated depot. A new solution, not necessarily better, will be accepted if the simulating annealing criterion is met. Further, we use a flexible cooling procedure enriched with resets, that means a massive temperature increase is used to overcome local optima by checking intensively the neighbors of a solution. The basic ideas are derived from Osman [16].

The proposed heuristics have been tested on 10 instances with varying customer time windows. The obtained solutions are compared with the tabu search algorithm presented by Guenduez [6]. For most of the instances, the proposed heuristics led to better solutions in a significantly shorter computation time. In cases where the tabu search algorithm calculated better solutions, the proposed heuristics were able to find solutions of similar overall costs. Therefore, the average percental deviation from best calculated solutions is lower for the proposed heuristics than for the compared tabu search. Hence, the proposed heuristics provide solutions of better mean quality in less average computation time. Although better solutions have been calculated, we do not know how good these solutions are. Future research should include the calculation of 'good' lower bounds for the SSLRPTW model in order to measure the solution quality of the presented heuristics. Moreover, different initial solution methods could lead to improved solutions and should be involved in future research. The proposed approach could be adopted to a multi-stage LRP with time windows. Then, instead of time windows a generalization to earliest pickup and latest delivery aspects could be worth further investigations.

References

[1] M.L. Balinski, "Integer programming methods, uses, computation", Management Science 12, INFORMS, 1965, pp. 253-313

[2] O. Braeysy and M. Gendreau, "Vehicle routing with time windows, Part I: Route construction and local search algorithms", Transportation Science 39(1), INFORMS, 2005, pp-119-139

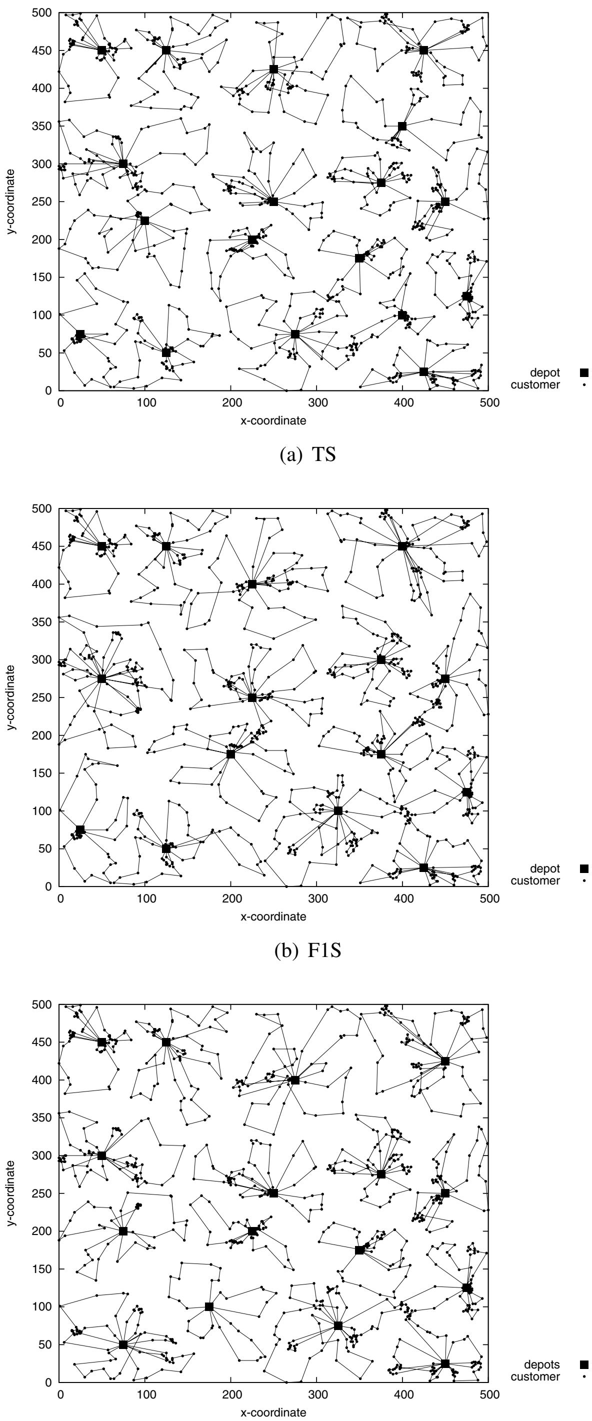
[3] A. Bruns, A. Klose, and P. Staehly, "Restructuring of Swiss parcel delivery services", OR Spectrum 22, Springer, 2000, pp. 285–302

[4] Y. Chan, W.B. Carter, and M.D. Burns, "A multi-depot, multiple-vehicle, location-routing problem with stochastically processed demands", Computers & Operations Research 28, Elsevier, 2001, pp. 803–826

[5] G. Clarke and J.W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points", Operations Research 12, INFORMS, 1964, pp. 568-581

[6] H.I. Guenduez, "The Single-Stage Location Routing Problem with Time Windows", Lecture Notes in Computer Science 6971, Springer, 2011, pp. 407-428

[7] H. Gunnarsson, M. Roennqvist, and D. Carlsson, "A combined terminal location and ship routing problem", Journal of the Operational Research Society 57(8), Palgrave Macmillan, 2006, pp. 928–938



(a) TS

(b) F1S

(c) F1F

Figure 3: Results of instance TC-I04

[8] I. Irnich, "Resource extension functions: properties, inversion, and generalization to segments", OR Spectrum 30, Springer, 2008, pp. 113-148

[9] S.K. Jacobsen, O.B.G. Madsen, "A comparative study of heuristics for a two-level routing-location problem", European Journal of Operational Research 5, Elsevier, 1980, pp. 378–387

[10] P. Klaus, C. Kille, and M. Schwemmer, "TOP 100 in European Transport and Logistics Services, issue 2011/2012", Deutscher Verkehrsverlag, Hamburg, 2011

[11] M.J. Kuehn and A.A. Hamburger, "A heuristic program for locating warehouses", Management Science 9, INFORMS, 1963, pp. 643-666

[12] M. Lundy and A. Mees, "Covergence of an Annealing Algorithm", Mathematical Programming 34, Springer, 1986, pp. 111-124

[13] K.G. Murty, P.A. Djang, "The U.S. army national guards mobile training simulators location", Operations Research 47(2), INFORMS, 1999, pp. 175–183

[14] G. Nagy and S. Salhi, "Nested heuristic methods for the location-routeing problem", Journal of the Operational Research Society 47(9), Palgrave Macmillan, 1996, pp. 1166-1174

[15] G. Nagy and S. Salhi, "Location-routing: Issues, models and methods", European Journal of Operational Research 177, Elsevier, 2007, pp. 649-672

[16] I.H. Osman, "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem", Annals of Operations Research 41, Springer, 1993, pp. 421-451

[17] I.H. Osman and N. Christofides, "Capacitated clustering problems by hybrid simulated annealing and tabu search", International Transactions in Operations Research I, Wiley, 1994, pp. 317-336

[18] J. Perl and M.S. Daskin, "A unified warehouse location-routing", Transport Research Part B 19B(5), Elsevier, 1985, pp. 381-396

[19] S. Salhi and G.K. Rand, "The effect of ignoring routes when locating depots", European Journal of Operational Research 39, Elsevier, 1989, pp. 150-156

[20] P. Toth and D. Vigo, "The vehicle routing problem", SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics, Philadelphia, 2002

[21] M. Wasner, G. Zaepfner, "An integrated multi-depot hub-location vehicle routing model for network planning of parcel service", Operations Research Letters 20, Elsevier, 2004, pp. 403–419