Online Supplement to: A Unified Modeling and Solution Framework for Vehicle Routing and Local Search-based Metaheuristics

Stefan Irnich [a,*]

[a]*Deutsche Post Endowed Chair of Optimization of Distribution Networks, RWTH Aachen University, Templergraben 64, D-52056 Aachen, Germany.*

Key words: local search, vehicle routing, rich VRPs, resource-constrained paths

---

Additional Results for Section 3.3.3 on Preprocessing

The ideas of the 1-level hierarchy can be generalized to hierarchies with more levels. A 2-level hierarchy of seed points divides the giant tour on the first level into sections of size n^α. On the second level, several of these sections are combined to 2-level sections. With $0 < \alpha < \beta < 1$, there are n/n^β second level sections, each of which comprises n^β/n^α sections of level one. Figure 1 depicts the 2-level hierarchy. Any segment ranging from

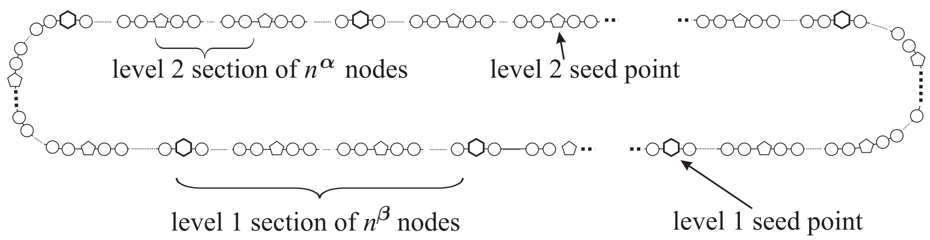

Fig. 1. 2-Level Hierarchy

position i to position j decomposes into a maximum of five smaller sections, i.e., (1) from i to the first level 1 seed point s_1^{first}, (2) from s_1^{first} to the first level 2 seed point s_2^{first}, (3) from s_2^{first} to the last level 2 seed point s_2^{last}, (4) from s_2^{last} to the last level 1 seed point s_1^{last}, (5) from s_1^{last} to j. If i and j fall into the same (first or second) level section, some of the sections are redundant. In order to handle any arbitrarily chosen i and j, the resulting number of segments to consider is

$$\mathcal{O}\left(2\frac{n}{n^\alpha}(n^\alpha)^2 + 4\frac{n}{n^\beta}\frac{n^\beta}{n^\alpha} + 2\left(\frac{n}{n^\beta}\right)^2 + 2\frac{n}{n^\beta}\left(\frac{n^\beta}{n^\alpha}\right)^2\right) = \mathcal{O}\left(n^{\max\{1+\alpha,1-\alpha,2-2\beta,1+\beta-2\alpha\}}\right)$$

* Corresponding author.
Email address: `sirnich@or.rwth-aachen.de` (Stefan Irnich).

since the first term estimates the number of segments (inverted or not) between two consecutive level 1 seed points, the second term for the number of segments between the level 2 and level 1 seed points in the same level 2 section, third between any pair of the level 2 seed points, and finally between any pair of level 1 seed points of the same level 2 section. The term $\max\{1+\alpha, 1-\alpha, 2-2\beta, 1+\beta-2\alpha\}$ is minimal under precondition $0 < \alpha < \beta < 1$ for $\alpha^* = 1/7$ and $\beta^* = 3/7$ yielding an effort of $\mathcal{O}(n^{8/7})$.

Proposition 1 *Segment REFs and inverse segment REFs for a 2-level hierarchy of seed points for a giant tour of length n can be computed in $\mathcal{O}(Rn^{8/7})$ time and space.*

Finally, we see that a giant tour, which is arbitrarily split into k segments, decomposes into a maximum of $3k$ segments of the 1-level hierarchy and $5k$ segments of the 2-level hierarchy. With k fixed, $3k$ and $5k$ is also fixed and with Proposition 1 we get the following result:

Theorem 2 *Any VRP neighborhood of size $\mathcal{O}(n^k)$, in which moves decompose the giant tour into a fixed number of segments, can be searched in $\mathcal{O}(Rn^k)$ time and $\mathcal{O}(Rn^{8/7})$ space.*

Without proof, we remark that a 3-level hierarchy can reduce the effort for preprocessing to $\mathcal{O}(n^{16/15})$ by considering up to $7k$ segments resulting from a k-edge exchange move. In general, an ℓ-level hierarchy leads to the consideration of up to $(2\ell+1)k$ segments and $\mathcal{O}(n^{2^{\ell+1}/(2^{\ell+1}-1)})$ effort.

Additional Results for Section 4 on Modeling Issues

4.4 VRPs with Compatibility Constraints*

Two types of (in)compatibilities have been considered in the literature thus far. First, *site dependencies* (SDVRP, e.g., Cordeau and Laporte (2001)) reflect that some vehicles cannot serve some requests due to the fact that, e.g., special facilities are needed to perform the service or a particular vehicle type is inappropriate for reaching a customer located on a narrow street. To model these types of constraints, we propose considering groups of vehicles and requests that behave identically w.r.t. compatibility. Let $g(i) \in \{1, \dots, G\}$ be the group of a request node $i \in R$, let $h(o), h(d) \in \{1, \dots, H\}$ be the vehicle group of $o \in O$ and $d \in D$ respectively, and let $(\kappa_{gh}) \in \{0,1\}^{G \times H}$ be the compatibility matrix ($\kappa_{gh} = 1$ means that g and h are compatible). If $G \leq H$, we can use G binary resources $\{1, 2, \dots, G\}$ representing which groups of requests are collected along the tour. Except for the reset REFs on arcs (d, o), all other REFs are of the form $f_{ij}^g(T) = \max\{a_j, T\}$, i.e., they are completely determined by the lower bounds of the resource intervals. Route-start nodes have resource intervals $[0,1]^G$; entering a request node j sets the resource $g(j)$ to one, i.e., $a_j = (0, \dots, 0, 1, 0, \dots, 0)^\top$ (with the 1 at position $g(j)$). For each resource $g \in \{1, \dots, G\}$, the compatibility is checked at the route-end nodes $d \in D$ using a resource interval $[0, \kappa_{g,h(d)}]$. Alternatively for $H < G$, one should use H binary resources $\{1, \dots, H\}$ representing the possible vehicle groups. REFs are of the same form as before. At a route-start node $o \in O$, the resource consumption is set to $a_o = (0, \dots, 0, 1, 0, \dots, 0)^\top$ (with the 1 at position $h(o)$). Feasibility is checked at all request nodes i where the resource interval is set to $[0, \kappa_{g(i),h}]$ for each resource $h \in \{1, \dots, H\}$.

Second, *incompatibility among requests* occurs, e.g., in the context of hazardous material transportation or transportation of groceries (frozen and unfrozen goods, different cooling requirements). Again, requests are grouped into classes $\{1,\ldots,G\}$ with class $g(i)$ for request node $i \in R$. Let $(\kappa_{g,g'}) \in \{0,1\}^{G\times G}$ be the compatibility matrix. Entering a request node requires that the resource $g(i)$ is at zero guaranteed by the resource interval $[0,0]$ for this resource (upper bound 1 for other resources). At the same time, when entering i, all incompatible resources, i.e., $g' \in \{1,\ldots,G\}$ with $\kappa_{gg'} = 0$ have to be set to one. Hence, REFs are of the form $f_{ij}^g(T) = \max\{1-\kappa_{g,g(j)}, T_g\}$ for $g \in \{1,\ldots,G\}$.

4.5 Interdependent Resources*

Interdependent resources arise naturally in some real-world applications or they are imposed by modeling issues, especially if one wants to model with REFs that satisfy all of the necessary conditions stated in Section 3. Examples are load-dependent or time-dependent travel costs or several types of non-trivial tariffs, where the cost of a tour depends on the time and distance travelled, the (maximum) load transported, the time spent on traveling, waiting and service etc. Irnich (2006) considers several of these examples and points out the following results: (1) Applications with simultaneous delivery and pickup (VRPSDP) require two dependent resources and REFs with interdependent resource consumptions. These are REFs of the form $f_{ij}(T,T') = (\max\{a, T+t, T'+u\}, \max\{a', T+t', T'+u'\})$ for both, arcs and segments. Their inverses are of the form $f_{ij}^{inv}(S,S') = (\min\{b, S-t, S'+t'\}, \min\{b', S-u', S'-u'\})$. (2) Cost functions with polynomial functions for the load-dependent cost have REFs that can be generalized to segments. (3) Together with the results given by Desaulniers and Villeneuve (2000), limited waiting times, limited working hours (with individual weights for traveling, service, and waiting) can be handled by non-decreasing REFs. REFs have the same form as those of the VRPSDP. In all these cases, the techniques of Section 3 are applicable, so that an acceleration of LS moves is possible. Note further that the modeling of waiting costs can also be done with non-decreasing REFs. Since these cost functions are not separable by arcs, sequential search techniques are not directly applicable but results for the constant time feasibility test of Section 3 remain valid.

Contrary, important real-world constraints and cost functions exist that do not fit into the context of accelerated local search procedures as given in Section 3. The paper (Irnich, 2006) points out that, e.g., soft-time windows, time-dependent travel times, and non-linear tariffs for load-dependent costs do not fit into the unified framework. Finally, applications with multiple time windows can have segment REFs with a high number of cases to distinguish, so that multiple time window constraints do not fit fully into the unified framework. $\mathcal{O}(1)$ feasibility testing is not accomplishable. Nevertheless, the methods of Section 3 remain applicable and result in efficient LS algorithms also for these types of VRPs.

4.6 Heterogeneous Fleet VRPs*

Heterogeneity of the vehicle fleet has been considered by several authors (see, e.g., Toth and Vigo, 2002) and regards the following aspects: different (1) capacities Q^k, (2) fixed costs f^k, (3) travel times t_{ij}^k and maximum route durations T^k, (4) variable costs c_{ij}^k for groups of vehicles of type $k \in \{1,\ldots,K\}$, and (5) site dependencies, see above.

The giant tour representation can directly handle aspects (1) and (2) by defining vehicle type-specific route-start and route-end nodes, i.e., $O = O^1 \cup \cdots \cup O^K$ and $D = D^1 \cup \cdots \cup D^K$. Fixed costs f^k can be put on the connections (o^k, i) for all $o^k \in O^k, i \in R$ while the resource load is only bounded on nodes $d^k \in D^k$ by the resource interval $[0, Q^k]$.

Adding vehicle-specific travel times and route durations (3) requires additional resources to tackle the problem efficiently. We suggest to use $K+2$ resources, one resource $r = k^*$ to record the actual vehicle type, K resources $r = time^k$ to model the travel time according to *each* possible vehicle type k, and one resource $r = time$ for the actual time along the giant tour. Depending on resource k^*, the resource $time$ is updated according to the information gathered in resource $time^{k^*}$. It is important to mention that resources $time^k$ are not bounded, i.e., the corresponding constraints are never violated but the actual time resource $r = time$ is bounded.

Finally, vehicle-dependent costs (4) can be handled similarly to vehicle-dependent travel times, so that $\mathcal{O}(1)$ feasibility checks and cost computations are possible. Hence, the same worst-case results, as derived in Section 3.3, apply here. It is beyond the scope of this paper to give details on the REFs and their generalizations to segments. Note that vehicle-dependent costs forbid the direct use of sequential search techniques because costs are not directly retrievable from the arcs. Nevertheless, the use of lower and upper bounds for arc costs can lead to variants of sequential search procedures with weaker bounding criteria. These criteria allow the acceleration of LS algorithms w.r.t their average case running time.

4.7 Periodic VRPs*

In periodic VRPs (see, e.g., (Cordeau et al., 1997)), customers have to be serviced according to feasible *visiting patterns*, e.g., in a week $\mathcal{T} = \{mo, tu, we, th, fr, sa\}$ two or three visits according to the patterns $mo/we/fr$, $tu/th/sa$, mo/we, tu/fr or we/sa. Periodic problems can be modeled with one request node for each combination of customer and day. Assume that a route on day mo visits three customers i, j and k, customer i is serviced according to visiting pattern $mo/we/fr$, customer j is serviced every day with pattern $mo/tu/we/th/fr/sa$, and customer k is serviced on mo and th only. In our representation the corresponding route on Monday is $p = (o^{mo}, i_{mo}, i_{tu}, j_{mo}, k_{mo}, k_{tu}, k_{we}, d^{mo})$, i.e., the route covers demands of consecutive days for a customer. A feasible route plan, therefore, corresponds with a Hamiltonian cycle in this particularly defined routing graph. By means of specialized non-decreasing REFs it is also possible to ensure the feasibility of routes, i.e., that a route on day $t \in \mathcal{T}$ covers only sequences of consecutive customers/day combinations *starting* with day t. A description of the modeling approach is beyond the scope of this paper but a more detailed report on modeling periodic VRPs with the help of resources is in preparation.

4.8 Inter-Tour Resources and Constraints*

Another strength of the unified framework is that it is able to handle inter-tour resources and constraints by considering the giant route as a single resource-feasible path. Along this path, *global resources* can be updated and limited. First of all, cost is a resource which is accumulated along the entire giant route; it is never reset at route connecting arcs (d, o). Some examples of the usefulness of inter-tour resources and constraints will be sketched in the following paragraphs. Note that in column generation models, the inter-route constraints are those “complicated” constraints which are put into the master program together with the covering constraints, cf. (Desaulniers et al., 2005; Lübbecke and Desrosiers, 2005).

4.8.1* Limiting the Number of Routes of Certain Characteristics

A first example of inter-tour constraints is the requirement that only a limited number of tours with a certain characteristic are allowed. An example is a restricted number of "short" or "long" routes. A first resource measures the (spatial or temporal) length of a route. Whenever a certain limit is exceeded, the route is regarded as "long". A second resource records the number of long routes. This resource is only modified on arcs entering a route-end node $d \in D$, more precisely, incremented by unit if the first resource exceeds the given limit. Such a resource update yields to non-decreasing REFs. Generalizing these REFs to segments of the giant tour is possible, but cumbersome to write down. These REFs do not have proper inverses w.r.t. the second resource for counting long routes. Anyway, such a proper inverse is not really required. The second resource can be propagated in a forward direction along the entire giant route. The resulting resource consumption has to be checked at the very last node only, since it is globally bounded by a fixed upper bound. Using similar modeling tricks, it is possible to enforce so-called *balancing constraints*, e.g., in order to limit the ratio between tours performed by full-time and part-time employees.

Moreover, inter-tour constraints are essential in combined multi-depot and heterogeneous fleet problems. A straightforward approach uses as many route-start and route-end nodes as possible depot/vehicle type pairs exist. In situations where a limited fleet can be assigned to different depots, the number of depot/vehicle pairs exceeds the real size of the fleet (2 depots, 2 types of vehicles with 3 and 4 cars, respectively; the overall number of depot/vehicle pairs is $14 = 2 \cdot (3 + 4)$ but only 7 cars are available). Resources can limit the overall number of vehicles of each type or limit the number of tours departing from a specific depot.

4.8.2* Handling of Sorting Capacities

In postal applications, an important subproblem is the routing of letter mail and parcel collecting tours. These tours bring letter mail to production/sorting centers where mail and parcels are sorted, commissioned, and sent to inbound facilities. Service requirements result in fixed cutoff times at the sorting centers. These cutoffs imply in turn that letters/parcels are brought in constantly over time, always early enough such that the remaining quantity can be processed in the remaining time before cutoff. The effect is that tours have to deliver quantities according to certain input requirements formulated over time. More precisely, we have constraints stating that, for each point T in the planning horizon, the overall quantity delivered by all tours arriving at T or later is bounded from above by an amount $Q(T)$. If the curve $T \mapsto Q(T)$ is discretized over time, corresponding resources and REFs can capture the limited sorting capacities at the processing centers. Similar constraints arise when vehicle routing is considered as a transportation sub-process in supply-chain design where facilities in consecutive stages of the chain have to perform time-consuming processes on homogeneous goods, see (Hempsch and Irnich, 2007).

Additional Results for Section 5 on Computational Results

5.5 Pickup and Delivery Problems*

When solving pickup-and-delivery problems, a request-relocation neighborhood is an interesting option (see, e.g., Toth and Vigo, 1997). Since a request consists of two nodes, a pickup node i^+ and a delivery node i^-, the two nodes are removed from their current positions and inserted into new positions (the case where only one of these nodes is moved is already covered by the relocation move). Clearly, an instance with n requests and maximum route length L implies a neighborhood of size $\mathcal{O}(n^2L)$, which is of the order $\mathcal{O}(n^3)$ when there is no upper bound for the route lengths. The magister degree thesis by Bellscheidt (2005) gives details of how different cases have to be handled in a sequential search procedure (pickup and delivery nodes can be in consecutive or non-consecutive positions before/after the modification, i.e., subcase (h1)-(h4) in Figure 4). It is beyond the scope of this paper to describe these implementations in detail.

For a comparison of search procedures for PDPTW, we have used the benchmark instances of Li and Lim (2001). These instances are constructed similar to the Solomon instances for VRPTW, i.e., there are groups of clustered, random and mixed instances with short as well as long tours. The number n of request nodes varies from about 100 to 1 000 ($n/2$ is the number of pickup-and-delivery requests). For the sake of brevity, we restrict ourselves to reporting results for the request-relocation neighborhood, see Figure 4(h1–h4).

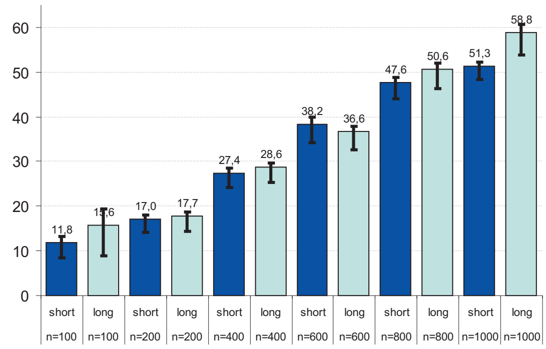

Fig. 2. Speedup of Sequential Search vs. Lexicographic Search for Li&Lim PDPTW Instances, Acceleration Factor $f_{req-reloc}$ for the Request-Relocation Neighborhood

The results for PDPTWs are displayed in Figure 2 and indicate that the speedup of the request-relocation neighborhood is large. As seen for the acb-neighborhood, the empirical evaluation of these $\mathcal{O}(n^3)$ neighborhoods indicates that they benefit even more strongly from the sequential search approach than the classical quadratic neighborhoods do. In spite of the previously presented results, there is no significant difference between instances with short and long routes. A possible explanation for this result could be that pickup-and-delivery routes differ substantially from cost-minimal MTSP tours. Hence, there is already a large fraction of moves that seems to be improving but is, in fact, infeasible. This seems to apply equally to instances with short and long routes.

5.6 Periodic Vehicle Routing Problems*

Benchmarks problems for the PVRP are available from (Cordeau et al., 1997). Since we were computing all REFs in advance, we had to omit the largest instance p13 with $n = 417{\cdot}7 = 2919$ request nodes. The remaining 31 instances range from $n = 20{\cdot}4 = 80$ to

$n = 153 \cdot 6 = 918$. Recall that our implementation uses the giant-tour representation with nodes for customer/day pairs, where a single delivery at customer i at day t covering the demands of days $t, t+1, \ldots, t'$ is encoded as a string $i_t, i_{t+1}, \ldots, i_{t'}$. In order to "relocate a customer" from one route to another, it must be possible to relocate the entire string. Hence, the maximum string length ℓ for Or-opt and string-exchange moves is increased to the maximum length of the time horizon, i.e., to $|\mathcal{T}| = 10$.

The speedups for the PVRP instances are mainly correlated to both parameters, the number of request nodes n and the average number n/r of nodes per route. Since we did not find a meaningful grouping of instances that reflects both parameters, displaying the results in a figure is hardly possible. Thus, we report results for $f_{\mathcal{N}}$ and the six neighborhoods introduced in Section 5.1.1 in textual form: The speedup factor f_{swap} for swaps varies from 2.7 for p14 to 49.4 for p09 with an average of 10.7 over all 31 instances. For the other neighborhoods, the maximum values of $f_{\mathcal{N}}$ were always obtained for the instance p09 with $n = 800$ and $n/r = 100$. The minima resulted from the instances p14 and p17 with $n = 80$ and $n = 160$ request nodes and $n/r = 10$ and $n/r = 13,\bar{3}$ respectively. For 2-opt we computed f_{2-opt} (min/avg/max) 3.0/11.5/28.3, for 2-opt* 2.9/8.3/31.1, for node relocation 3.0/6.9/11.9, Or-opt 2.6/17.6/75.8, and Or-Opt with inversion 2.0/16.5/70.3. As before, the largest speedups were found for the string-exchange neighborhood with 13.6/87.0/473.9 and for the acb neighborhood with 3.5/96.5/1 088.0

References

Bellscheidt, L. 2005. Implementierung und Analyse von sequentiellen Suchverfahren für Pickup- und Delivery Probleme. Magister thesis, Deutsche Post Endowed Chair of Optimization of Distribution Networks, RWTH Aachen University. (in German).

Cordeau, J.-F., Michel Gendreau, G. Laporte. 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* **30** 105–119.

Cordeau, J.-F., G. Laporte. 2001. A tabu search algorithm for the site dependent vehicle routing problem with time windows. *Information Systems and Operations Research* **39** 292–298.

Desaulniers, G., J. Desrosiers, M.M. Solomon, eds. 2005. *Column Generation*. Springer.

Desaulniers, G., D. Villeneuve. 2000. The shortest path problem with time windows and linear waiting costs. *Transportation Science* **34** 312–319.

Hempsch, C., S. Irnich. 2007. Vehicle-routing problems with inter-tour resource constraints. Technical Report 2007-01, Deutsche Post Endowed Chair of Optimization of Distribution Networks, RWTH Aachen University, Aachen, Germany. Available at www.dpor.rwth-aachen.de.

Irnich, S. 2006. Resource extension functions: Properties, inversion, and generalization to segments. Technical Report 2006-01, Deutsche Post Endowed Chair of Optimization of Distribution Networks, RWTH Aachen University, Aachen, Germany. Available at www.dpor.rwth-aachen.de.

Li, H., A. Lim. 2001. A metaheuristic for the pickup and delivery problem with time windows. D. Moldovan, ed., *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01)*. IEEE Press, 160–167.

Lübbecke, M., J. Desrosiers. 2005. Selected topics in column generation. *Operations Research* **53** 1007–1023.

Toth, P., D. Vigo. 1997. Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science* **31** 60–72.

Toth, P., D. Vigo. 2002. An overview of vehicle routing problems. P. Toth, D. Vigo, eds., *The Vehicle Routing Problem*, chap. 1. Siam, Philadelphia, 1–23.