Regensburger
DISKUSSIONSBEITRÄGE
zur Wirtschaftswissenschaft

A Cluster Based Scatter Search Heuristic for the Vehicle Routing Problem

University of Regensburg Discussion Papers in Economics
No. 415, November 2006

Rolf Wendolsky* Stephan Scheuerer†

*University of Regensburg
93040 Regenburg, Germany
rolf.wendolsky@wiwi.uni-regensburg.de

†Fraunhofer-Center for Applied Research on Technologies
for the Logistics Service Industries (ATL)
90411 Nuremberg, Germany
stephan.scheuerer@atl.fraunhofer.de

UNIVERSITÄT REGENSBURG

Wirtschaftswissenschaftliche Fakultät

A Cluster Based Scatter Search Heuristic for the Vehicle Routing Problem

Rolf Wendolsky* Stephan Scheuerer†

*University of Regensburg
93040 Regenburg, Germany
rolf.wendolsky@wiwi.uni-regensburg.de

†Fraunhofer-Center for Applied Research on Technologies for the Logistics Service Industries
90411 Nuremberg, Germany
stephan.scheuerer@atl.fraunhofer.de

1 Introduction

The *Vehicle Routing Problem* (VRP) is one of the most studied problems in the field of Operations Research. It consists of finding least cost routes for a set of homogeneous vehicles located at a depot to geographically scattered customers. Each customer has a known demand and service duration. The routes have to be designed such that each customer is visited only once by exactly one vehicle, each vehicle route starts and ends at the depot and the total capacity and total service time of a vehicle may not be exceeded.

Closely related to the VRP is the *Capacitated Clustering Problem* (CCP). The CCP considers partitioning a set of weighted points, the customers, into p distinctive clusters, so that a capacity limit on the clusters may not be exceeded. For a given cluster, a cluster center is that customer of the cluster from which the sum of the distances to all other customers in the cluster is minimized. The objective of the CCP is to find a partition that minimizes the sum of the distances from all cluster centers to all other customers in their clusters.

The VRP can be considered as an 'extension' of the CCP in the way that for each cluster in the CCP solution, additionally a route through all cluster customers and the depot has to be constructed to generate the routing information.

Both, the VRP and the CCP, are computationally hard to solve and are usually tackled by heuristic approaches. For literature, for example, on the VRP see Toth and Vigo [9], Laporte et al. [6] and Cordeau et al. [2], and for CCP Scheuerer and Wendolsky [8], and Ahmadi [1].

In a previous article [8], we described an application of the Scatter Search methodology to solve the CCP. This algorithm had an excellent performance compared to other ones based on benchmark problems. We determined that our CCP Scatter Search algorithm can - with some modifications - successfully be adopted to the VRP, too. This approach will now be described.

2 The Scatter Search Heuristic

2.1 Overview

The Scatter Search methodology has successfully been applied to a widespread variety of combinatorial optimization problems. Closely related to Scatter Search is the Path Relinking principle. For details on the methodologies the reader is referred to Glover, Laguna and Martí [3] and Laguna and Martí [5], for a general overview on metaheuristics to Glover and Kochenberger [4].

Table 1 shows the general outline of our Scatter Search heuristic. The outline is similar to the one of the CCP heuristic described in Scheuerer and Wendolsky [8]. In our approach, the reference set follows a two-tier design and consists of a set of high quality solutions B_1 and a set of diverse solutions B_2 with size b_1 and b_2 respectively. No duplicates are allowed in the reference set.

In the following, the details on the specific methods will be described to solve the VRP. Thereby let *cluster* denote a set of customers and let *route* denote a set of customers with an additional fixed routing-order for vehicle routing. Note that using this relationship, the routes in the VRP can also be viewed as clusters (by neglecting the routing order) and that a route can therefore either be identified by its cluster center or by its route index.

Table 1: General outline of the Scatter Search algorithm

```
Procedure Scatter Search
begin
    CreateInitialSolution
    ImproveSolutions
    GenerateDiversifiedSolutions
    ImproveSolutions
    UpdateReferenceSet

    Converged := FALSE
    NewSolutions := TRUE
    while (Stopping criterion not met and Converged = FALSE) do
        if (NewSolutions = TRUE) then
            GenerateSubsets
            CombineSolutions
        else
            Converged := GenerateDiversifiedSolutions
        endif
        ImproveSolutions
        NewSolutions := UpdateReferenceSet
    end
end
```

2.2 Initial Solution Creation

The aim of this method is to construct an initial solution. Thereby the method first selects p geographically scattered center candidates among all the customers and then assigns the remaining customers to their nearest center candidate to create an initial CCP solution. In a second step, for every cluster in the CCP solution a route is initially constructed and improved with a TSP heuristic. Note that at this step of the algorithm, the VRP as well as the CCP solution may not be feasible regarding capacity and service time. The resulting VRP solution is improved by the improvement method (see section 2.3) and then forwarded to the diversification generation method (see section 2.4).

2.3 Solution Improvement

To locally improve solutions, the algorithm tries to shift one or exchange two different customers between neighbouring routes and performes the best possible move as long as any improvement is found. Thereby infeasible solutions, that are solutions having overcapacity or overtime, are strictly penalized to favour the selection of feasible solutions. The evaluation of a move is done by cheapest insertion: When shifting a customer u into another tour $T = (i_0, \ldots, i_k)$, its best position in T is determined by evaluating

$$e(u,T) = \min_{0 \le l \le k-1} \left(c(i_l, u) + c(u, i_{l+1}) - c(i_l, i_{l+1}) \right) \tag{1}$$

A 2-opt and an Or-opt TSP-heuristic is used to post-optimize the two modified routes.

To speed up the search process, moves are only considered between neighbouring routes. Our definition for neighbouring routes is based on Ahmadi's definition [1] who introduced it in the context of clustering: We call a route R_i an m-neighbour of route R_j if there exists a customer in R_i that is among the m nearest customers to a customer in R_j.

To further speed up the search, we use a special data structure that holds information about the change in overcapacity, overtime, total distance and total time for the best move between every pair of neighboring routes. The basic form of our data structure is adopted from Osman [7]. Having performed a move, only the route-entries in the data structure belonging to the two modified routes have to be reevaluated and updated; the others remain unchanged. This helps to significantly reduce computing time.

2.4 Diversification Generation

The method creates a set S ($|S| \le \gamma$) of diversified solutions to a given solution s. Using the cluster view of the VRP, the method tries to maximize the number of different clusters and thereby different routes per generated solution.

Starting from a set M containing the p cluster centers in the given solution s and a set N with the remaining $n-p$ customers, we define an array $\bar{C}$ of n customers as center candidates for new solutions. The array $\bar{C}$ is ordered in a way that every successive list of p entries ensures a sufficient diverse set of center candidates to the other solutions. It is constructed by iteratively choosing the customer $c \in N$ with the maximum distance to the $p-1$ previously selected centers, initially $p-1$ centers from M. This customer c is then deleted from N. If all remaining customers in N have been assigned to the array $\bar{C}$, the p customers from M are selected the same way, so that the array has a final length of n center candidates. Given the array, iteratively p successive entries of the array are selected as cluster centers for new solutions, beginning with the first position in the array and by iteratively moving forth by one position. For each set of new cluster centers, every remaining customer is assigned to its nearest center candidate to create a (not necessarily feasible) CCP solution. To ensure a sufficient number of new solutions for the candidate set S, the sets M and N may be redefined with different initial customers and the process may be re-run.

This way, a set S of CCP solutions will be created. To transfer every new CCP solution into a (not necessarily feasible) VRP solution, for every cluster a route through all cluster customers and the depot is created and improved via a TSP heuristic. The corresponding VRP solution finally is locally improved as described in section 2.3.

This method also checks for premature convergence by setting the Boolean variable *'Converged'* equal to true if no further seed solution in the reference set can be found. If this is the case, no further diversified solutions can be created and the algorithm ends prematurely. Note that every reference set solution can only be used once as a seed solution. Also, whenever this method is called, we force the search to include new solutions into the reference set by deleting all solutions in B_2 and by keeping only the best $b_1/2$ solutions in B_1.

2.5 Reference Set Update

The diversification and the combination method fill up a candidate pool S ($|S| \leq \gamma$) of solutions from whom every solution will be considered for admission into the reference set, either because of its quality (reference set B_1 with size b_1) or its diversity (reference set B_2 with size b_2):

The first b_1 best solutions in S may replace solutions in B_1 if they have a lower evaluation value than those. After eliminating the solutions that were admitted to B_1 from S, up to b_2 solutions in B_2 may be replaced with solutions in S. To do so, a diversification criterion is needed. For this purpose it is assumed that if two VRP solutions have the same cluster centers there is a high probability that their routes contain many similar parts. Therefore we define $div(s_1, s_2)$ to be the *absolute* diversification value between two solutions s_1 and s_2 and calculate the value of $div(s_1, s_2)$ as the number of customers with equal center assignment in both solutions plus p times the number of equal centers. The absolute diversification value for two solutions lies between 0 and $n + p^2$ and the lower the value, the more diverse are the two solutions. The *relative* diversification value $div(s, D)$ of a solution s to a set of solutions D is then computed as the sum over the absolute diversification values from s to all solutions in D.

We calculate the relative diversification value of every solution in S and of the solution at the first position in B_2 to all solutions in B_1. If a solution in S has a lower diversification value than the solution currently at the first position in B_2, it is replaced by this solution. Then, having fixed the assignment for the first position in B_2, we calculate the relative diversification value for each solution in S for the second position in B_2 relative to all solutions in B_1 and to the solution on the first position in B_2. Position three in B_2 is calculated relative to all solutions in B_1 and the first and second solution in B_2 and so on. No duplicates are allowed in the reference set.

Whenever a new solution is included in B_1, we update a longterm memory used for tie-breaking in the diversification method that stores information on the number of times a customer is chosen as cluster-center in a B_1 solution. For all customers that represent centers in this solution, the counters for the center frequency are incremented by one.

2.6 Subset Generation

Instead of generating all possible subsets of solutions in the reference set, this method creates subsets up to a size of three solutions: first, all possible 2-element subsets are created; second, every 2-element subset is used as basis for a new 3-element subset by including the best solution not in this subset. Subsets that do not contain at least one new reference set solution, that is a solution that has not yet been in the reference set in previous iterations, will not lead to new solutions in the combination method and are therefore eliminated by this method.

2.7 Solution Combination

We use Path Relinking as a combination method to generate new solutions. For every solution $s \in P$ in a given set of input solutions P, a path is constructed guided by other solutions, called target solutions. A duplicate free pool with the best γ generated solutions is forwarded after the process to serve as candidate solutions for the update of the reference set.

To guide the search, the method first maps every route of the input solutions in P with the "closest" route in every other input solution. The routes are mapped using the cluster view of the routes and a regret function: Let $reg(k, s_j)$ be the regret value for a cluster center k in solution s_i towards another solution s_j $(i \neq j \ ; \ s_i, s_j \in P)$, computed as the difference between the distance from k to its second nearest center and the distance to its nearest center in s_j. Furthermore, let $reg(k, P)$ be the sum of the regret values from k in solution $s_i \in P$ to every other solution in the subset of solutions P to be combined. Then, the method starts by calculating the value $reg(k, P)$ for every cluster (respectively route) and every solution in P. The cluster with the highest regret value $reg(k, P)$ is then mapped with its nearest clusters in every other solution. For the remaining unassigned clusters, the regret values are re-calculated and the assignment continues until all p clusters are mapped. Finally, the clusters of every solution are assigned a temporary index from 1 to p such that the mapping of the clusters is expressed through a similar number in every solution.

After mapping, the solutions in P are combined by Path Relinking via $|P|$ different paths, using each solution exactly once as a starting point for a new path. The paths are built by moving customers from a starting solution s_i into routes (respectively clusters) they are assigned to in the target solutions $s_j \in P$ $(i \neq j)$ according to the temporary index. During the search, moves are restricted to allow for simple shift moves only. The shift moves between two routes are evaluated and finally computed as in the improvement method, see section 2.3. The best possible move is performed - regardless of solution feasibility (known as tunneling [5]) - until all customers are included in one of their target clusters. As long as there is an improvement in the evaluation function value, moves from one target cluster to another target cluster are allowed, too.

Each μ-th solution as well as the best new solution found on a path is improved by the improvement method and is considered as a candidate for insertion into the candidate pool for the update of the reference set.

3 Results

Currently, the algorithm has only been experimentally implemented and further optimization in programming and tuning of the parameters has to be done. Therefore, we cannot say for sure how the VRP heuristic performs in the end. However, we expect the VRP approach to having a competitive performance to other leading VRP heuristics in the literature. First results on the classical VRP-benchmark problems seem to confirm our expectations.

References

[1] Ahmadi, Samad (1998): Metaheuristics for the Capacitated Clustering Problem. Ph.D. thesis. University of Kent at Canterbury, UK.

[2] Cordeau, Jean-François, Gendreau, Michel, Laporte, Gilbert, Potvin, Jean-Yves, and Semet, Frédéric (2002): "A Guide to Vehicle Routing Heuristics". In: *Journal of the Operational Research Society* **53**, 512–22.

[3] Glover, Fred, Laguna, Manuel, and Martí, Rafael (2003): Scatter Search and Path Relinking: Advances and Applications, In: Glover, Fred, and Kochenberger, Gary A. (eds.): *Handbook of Metaheuristics*. (International Series in Operations Research & Management Science, vol. 57). Kluwer Academic Publishers, Boston, USA, 1–35.

[4] Glover, Fred, and Kochenberger, Gary A. (eds.) (2003): *Handbook of Metaheuristics*. (International Series in Operations Research & Management Science, vol. 57). Kluwer Academic Publishers, Boston, USA.

[5] Laguna, Manuel, and Martí, Rafael (2003): *Scatter Search - Methodology and Implementations in C*. Kluwer Academic Publishers, Boston Dordrecht London.

[6] Laporte, Gilbert, Gendreau, Michel, Potvin, Jean-Yves, and Semet, Frédéric (2000): "Classical and Modern Heuristics for the Vehicle Routing Problem". In: *International Transactions in Operational Research* **7**, 285–300.

[7] Osman, Ibrahim H. (1993): "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem". In: *Annals of Operations Research* **41**, 421-451.

[8] Scheuerer, Stephan, and Wendolsky, Rolf (2004): "A Scatter Search Heuristic for the Capacitated Clustering Problem". In: *European Journal of Operational Research*, In Press, Corrected proof available online 12 October 2004.

[9] Toth, Paolo, and Vigo, Daniele (2002): *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA.