Distance-based path relinking for the vehicle routing problem

Kenneth Sørensen
Faculty of Applied Economics
University of Antwerp
Prinsstraat 13, 2000, Antwerp, Belgium
Email: kenneth.sorensen@ua.ac.be

Marc Sevaux
Lab-STICC
Université Bretagne-Sud
F-56321 Lorient, France
Email: marc.sevaux@univ-ubs.fr

1 Path relinking for the vehicle routing problem

Path relinking is a relatively new metaheuristic technique for combinatorial optimization, proposed by Glover (see e.g. [3]). Path relinking attempts to find new good solutions by examining the solutions that are on a path from an initial (incumbent) to a final (guiding) solution. By definition, each move on the path makes the solution more different from the initiating solution and more similar to the guiding solution. Moving on the path is done by a neighbourhood operator, just like in any local search algorithm. The technical difference with ordinary local search is that the neighbourhood search strategy that decides which move to execute is not based on the quality of the resulting solution, but on the distance in the solution space between the resulting solution and the guiding solution. A move that takes the solution closer to the guiding solution will be preferred over one that takes it further away, regardless of the quality of the resulting solution.

Constructing a path relinking procedure therefore requires us to select a move operator to use and a distance measure in the solution space between two solutions. The distance measure than can be used to determine whether a move brings a solution closer to the guiding solution and whether the resulting solution can be considered to be “on a path from incumbent to guiding solution”.

Usually, path relinking is not used as a standalone solution method, but combined with other metaheuristics, such as tabu search or GRASP, see e.g. [1, 7, 9, 10]. Although the capacitated vehicle routing problem (CVRP) is one of the best-known combinatorial optimization problems, path relinking approaches for this problem are few and far between. This is partially due to the fact that a VRP solution is most naturally represented as a set of permutations of customers, each member of the set representing a tour. Contrary to problems that have a natural binary or vector representation, it is not immediately obvious what is meant by...
2 Distance-based path relinking

Distance-based path relinking (DBPR) is a secondary local search meta-heuristic that exploits the search history of a local search to improve solutions. It is an improvement heuristic that can enhance the performance of local search algorithms. The method involves generating new solutions by altering the current solution in a way that respects the structure of the search history. This is particularly useful in combinatorial optimization problems where the solution space is vast and the search for good solutions is computationally expensive. By leveraging the history of the search, DBPR can avoid revisiting previously explored regions of the solution space, thus improving the efficiency of the search process.
2. Given the list of solutions that transforms the incumbent into the guiding solution, in which order do we execute this list of moves?

3. Which solution from the path do we return upon termination?

4. Can a path-relinking approach be used as a stand-alone optimizer or do we need to integrate it in a (local search) procedure?

3 Previous work and experiments

A proof-of-concept procedure was developed and published in [12]. In this contribution, we developed a path relinking procedure based on the “add-remove” edit distance, which corresponds to the minimal number of “add-remove” moves that have to be made to transform a solution into another one. One of the goals of this research is to generalize this procedure and perform the same analysis for more move types and their corresponding distance measures, resulting in a procedure like the one depicted in Algorithm 1.

Algorithm 1: Path relinking for the VRP

\[
\begin{aligned}
\text{Input: } & \text{Two solutions } s \text{ and } t \\
\text{Output: } & \text{A solution } u \text{ on a path from } s \text{ to } t \\
\text{Calculate } & n = d(s, t) \text{ and determine the list } M = \{m_1, \ldots, m_{n-1}\} \text{ of moves that transforms } s \text{ into } t \\
\text{Set } & u_0 = s; \\
\text{for } i = 1 \text{ to } n-1 \text{ do} \\
& \text{Perform move } m_i \in M (u_{i-1} \rightarrow u_i); \\
& \text{Choose solution } u \text{ to return from the set of generated solutions } \{u_0, \ldots, u_{n-1}\}; \\
\end{aligned}
\]

Although some promising results have already been obtained, this research is ongoing. Detailed results will be presented at the conference.

References


