K-medoid Algorithm with Adaptive Large Neighborhood Search for the VRPTW

Mehdi Nasri, Imad Hafidi and Abdelmoutalib Metrane

February 6, 2020

A K-medoid Algorithm with Adaptive Large Neighborhood Search for the VRPTW

Mehdi Nasri[1], Imad Hafidi[1], Abdelmoutalib Metrane[1]

Sultan Moulay Slimane University, National School of Applied Sciences, Khouribga, Morocco
mehdi.nassrii@gmail.com
imad.hafidi@gmail.com
ab.metrane@gmail.com

Mots-clés : *VRPTW, K-medoid, ALNS.*

1 Introduction

In this paper, we propose a new approach combining the k-medoid and the Adaptive Large Neighborhood Search (ALNS). The strategy fits into the class of algorithms cluster first - route second. Indeed, changing the way of clustering may change the efficiency of the solution of the Vehicle Routing Problem with Time Windows (VRP-TW). This fact, can be demonstrated by inserting a pre-processing step based on the K-medoid algorithm. In this stage, we subdivise the group of nodes of the general problem into small sets of customers which represent subproblems. Notice that the use of K-medoid is not restrictive and we can adopt of course other techniques such that K-means or density based spatial (DBS) clustering. Each subproblem is solved implicitly by applying the ALNS.

2 Description of our approach

Our methodology can be described on two steps. From one hand, the manner of clustering uses the K-medoid as a paradigm to tackle the pre-treatement. On the other hand, the second step is devoted to select the adequate routes. This is the wiespread ideas behind this approach. In the following, we provide more precise statements related to each step in more details :

Phase 1 : It consists in identifying a set of clusters through a $K-$medoids algorithm. The main idea of this iterative clustering algorithm is to divide the input data set into K distinct clusters $C_1,\ldots,C_K$. We first begin by selecting K of the N input data points as the initial medoids. Then, we associate each data point x_i to the nearest centroid C_j by computing a specific spatio-temporal measure between specified instance x_i and cluster center c_j and then we pick the cluster which have a minimum measure. We assign each data to the closest cluster j. The next step is to recompute the position of the centroids from individuals attached to the groups by taking the average of the all data points that belong to each cluster. The new centroid C_j is the mean of all points x_i assigned to cluster j in previous step. We repeat the previous steps until none of the cluster assignments change. We obtain then a partition of the instances in K groups characterized by their centroids.

Phase 2 : This is the most crucial phase which aims at selecting the different neighborhoods according to some strategy for the effectiveness of the search process. The strategy adopted in this work is to let ALNS solve each subproblem related to each cluster separately. The used ALNS is a metaheuristic proposed by Ropke and Pisinger in 2006 [3]. It is a common technique used to enhance a locally optimal solution. Given an initial solution obtained by a construction method, it is based on the idea of improving the initial solution by applying various destroy and repair operators to generate large neighborhoods through which the search space is explored. Finally, we collect the solutions related to each subproblem and we gather them to obtain a complete solution when the subsolutions will be the routes of the final solution.

3 Computational results

In this paper we present a comparison between the ALNS algorithm and an approach combining the K-medoid and the ALNS algorithm applied to the VRP-TW. The performance of the proposed algorithms have been evaluated by considering benchmark instances adapted from the literature (Solomon benchmark (1987)[1]). The comparison with a set of classic benchmark instances is presented for these problems, comparing the solution quality and execution time :

Table : Comparison between ALNS and K-medoids + ALNS in terms of objective function

Instance	Best known results	K-medoids + ALNS	ALNS
c103	826.3	869.3	884.1
c104	822.9	858.8	872.3
c105	827.3	856	873.6
r105	1355.3	1397.1	1429.5
r106	1234.6	1282.2	1305.1
r107	1064.6	1115.4	1130
rc205	1154	1197.8	1213.7
rc206	1051.1	1099.3	1112.5
rc208	777.3	819.1	841.6

The results show the improvement of the final objective function in the approach which combines the K-medoid with the ALNS compared to the ALNS.

Table : Comparison between ALNS and K-medoids + ALNS in terms of execution time

Instance	K-medoids preprocessing step(ms)	K-medoids + ALNS for 1000 iterations (ms)	ALNS for 1000 iterations (ms)
c103	111	23541	23410
c104	98	23328	23212
c105	116	23576	23446
r105	147	27245	27083
r106	139	27150	26993
r107	126	27123	26979
rc205	261	34290	34015
rc206	252	34256	33992
rc208	247	33999	33737

From the table, we can observe that the K-medoid preprocessing step doesn't take much execution time. Then it doesn't compromise on the execution time of the whole approach.

Références

[1] *M. Solomon Algorithms for the Vehicle Routing and Scheduling Problem with time Window Constraints. Operations Research*, 254–265, (1987).

[2] *S. E. Cömert, H. R. Yazgan, I. Sertvuran, H. Şengül. A new approach for solution of vehicle routing problem with hard time*, 2017

[3] *S. Ropke, D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science*, 455–472, 2006.