Simple and effective look-ahead heuristics for the vehicle routing problem

Christian Prins[1], Jorge E. Mendoza[2,3,4]

[1] Institut Charles Delaunay, Université de Technologie de Toyes, France
christian.prins@utt.fr
[2] Institut de Mathématiques Appliquées (IMA) - Université Catholique de l'Ouest, Angers, France
[3] IRCCyN (UMR CNRS 6597), École des Mines de Nantes, France
[4] Departamento de Ingeniería Industrial, Universidad de los Andes, Colombia
jorge.mendoza@uco.fr

Mots-Clés : *vehicle routing problem, heuristics, pilot method*

1 Introduction

The vehicle routing problem (VRP) consists of selecting a set of routes of minimal cost to serve the demands of a set of geographically spread customers using a homogeneous and unlimited fleet of vehicles. In the VRP : (i) the demand of every customer is serviced in a single visit, (ii) the total demand serviced by each vehicle does not exceed its capacity, and (iii) all routes start and end at a central depot. Since the VRP is a hard optimization problem, it is generally solved in practice using heuristics and metaheuristics. Even though metaheuristics regularly outperform heuristics in terms of solution accuracy, the latter are : faster, simpler to understand and implement, and often more flexible to be adapted to different business-specific scenarios. Therefore, heuristics remain the type of approaches more widely used in practice. To expand the heuristics toolbox for the VRP, this paper presents a set of fast, simple, and flexible algorithms that bridge the accuracy gap between classic heuristics and metaheuristics.

2 Look-ahead heuristics

The proposed heuristics are based on the look-ahead approach introduced by Mendoza et al. [1] for the multi-compartment VRP with stochastic demands (MC-VRPSD). As adapted to the VRP, the approach consists of using the *pilot method* [3] to guide a route-first, cluster-second heuristic (RCH) that builds a single giant tour visiting every customer (routing phase) and later splits it into smaller routes that satisfy the vehicle capacity constraints (routing phase). Then, the resulting set of routes undergoes a post-optimization phase. We implemented a set of RCHs combining different routing and clustering procedures. For the routing phase we coded three simple yet effective constructive algorithms for the deterministic traveling salesman problem (TSP), namely, the nearest neighbor (NN), nearest insertion (NI), and farthest insertion (FI) heuristics. On the other hand, for the clustering phase we use two different versions of Prins' optimal tour partitioning procedure : split and split with shifts (split-s). For the post-optimization phase we implemented a first-improvement local search procedure (LS) that couples 4 classical moves, namely, the relocation of one customer or chains of two customers, the swap of two customer, and 2-Opt moves. For a thorough explanation of the clustering and post-optimization procedures the reader is referred to [2].

To illustrate how the pilot method guides our heuristics, consider the RCH that uses the NN algorithm in the routing phase and the split procedure in the clustering phase. Starting with a partial tour visiting only the depot, at every iteration the pilot method makes K copies of the master solution, that is, the partial TSP tour being built, and stores them as partial *pilot solutions*. Then, the K nearest available neighbors are each connected to one of the partial pilot solutions. Using each new pilot solution as the starting point, the pilot method makes a callback to the NN heuristic to complete the generation of the TSP tour and then invokes split to perform the clustering phase, thus building a complete VRP solution. Next, the partial pilot solution leading to the best VRP solution becomes the new master solution and a new iteration begins. Pilot iterations are carried out until every customer is included in the master solution, then split performs the clustering phase and the resulting VRP solution is the one transferred to the post-optimization phase. The completion of the K solutions at each iteration allows the pilot method to look ahead the impact of different local choices on the final output of the RCH rather than only the partial TSP tour being built, thus avoiding myopic choices. The same principle can be applied to any of the six RCHs resulting from combining the 3 routing and the 2 clustering procedures. Henceforth we will refer to each of the pilot-based heuristics using the notation $Pilot(RP, CP, K)$, where $RP \in \{NN, NI, FI\}$ is the routing procedure and $CP \in \{split, split\text{-}s\}$ is the clustering procedure in the guided RCH, and K is the number of pilot solutions to be explored at each iteration.

We tried our heuristics in the 14 classical benchmarks of Christofides, Mingozzi and Toth (CMT) and compared the results against the best known solutions for the set. To assess the relative performance of our pilot-based approaches with respect to classical heuristics, we also implemented an improved version of the Clarke and Wright savings algorithm (ICW) that performs a 3-Opt search on each new trip after the merger and uses the LS procedure in the post-optimization phase. Table 1 summarizes the main results after fine tuning the value of K for each pilot-based heuristic.

Metric	**Pilot(RP,CP,K)**					
	(NN,split,1)	**(NN,split-s,1)**	**(NI,split,1)**	**(NI,split-s,1)**	**(FI,split,1)**	**(FI,split-s,1)**
Avg. Gap	5.31	5.55	5.01	5.65	4.41	4.14
Avg. CPU*	0.00	0.00	0.01	0.01	0.01	0.01
Metric	**(NN,split,3)**	**(NN,split-s,4)**	**(NI,split,5)**	**(NI,split-s,7)**	**(FI,split,8)**	**(FI,split-s,9)**
Avg. Gap	3.88	3.53	3.64	3.22	3.13	2.72
Avg. CPU*	0.02	0.04	0.54	0.81	0.43	0.53
Abs. Imp.	1.43	2.02	1.37	2.43	1.28	1.41
Rel. Imp.	26.86	36.36	27.37	42.99	29.05	34.18

TABLE 1 – **Performance metrics for the pilot-based heuristics.** Avg. Gap(%) : average gap with respect to the best known solution ; Avg. CPU(s) : average execution time ; Abs. Imp.(%) : average improvement with respect to the simple RCH ; Rel. Imp.(%) : relative improvement with respect to the simple RCH. * Coded in Delphi 2007, running on an Intel Core Duo 2.54 GHz

The results show the significant contribution of the pilot method to the accuracy of the heuristics. When the look-ahead mechanism is turned-on (i.e., K > 1), the performance of the RCHs improves above 1.25% in absolute terms and 27% in relative terms with respect to the simple RCHs. These improvements are achieved with a minor increment in the average CPU time, which is in the worst case below 1 second (Pilot(NI,split-s,7)). The results also indicate that our pilot-based heuristics outperform ICW (Avg. Gap of 4.84% and Avg. CPU of 0.03s) in terms of accuracy at a small increment in the execution time. These results tip the balance towards the use of our simple look-ahead heuristics over classical approaches for solving the VRP in practical settings.

Références

[1] J. E. Mendoza, B. Castanier, C. Guéret, A. L. Medaglia, and N. Velasco. Constructive heuristics for the multi-compartment vehicle routing problem with stochastic demands. Technical report, Universidad de los Andes, 2009. `http://hdl.handle.net/1992/1111`.

[2] C. Prins, N. Labadi, and M. Reghioui. Tour splitting algorithms for vehicle routing problems. *International Journal of Production Research*, 47(2) :507–535, 2009.

[3] S. Voß, A. Fink, and C. Duin. Looking ahead with the pilot method. *Annals of Operations Research*, 136(1) :285–302, 2005.