GENVNS-TS-CL-PR: A heuristic approach for solving the vehicle routing problem with simultaneous pickup and delivery [1]

Cruz R. C. [a,2] Silva T. C. B. [a,2] Souza M. J. F. [a,2]
Coelho V. N. [a,2] Mine M. T. [b,2] Martins A. X. [a,2]

[a] *Federal University of Ouro Preto, Ouro Preto, MG, 35400-000, Brazil*
[b] *GAPSO, Rio de Janeiro, RJ, 22290-160, Brazil*

Abstract

This work addresses the Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD). Due to its complexity, we propose a heuristic algorithm for solving it, so-called GENVNS-TS-CL-PR. This algorithm combines the heuristic procedures Cheapest Insertion, Cheapest Insertion with multiple routes, GENIUS, Variable Neighborhood Search (VNS), Variable Neighborhood Descent (VND), Tabu Search (TS) and Path Relinking (PR). The first three procedures aim to obtain an good initial solution, and the VND and TS are used as local search methods for VNS. TS is called after some iterations without any improvement through of the VND. The PR procedure is called after each VNS iteration and it connects a local optimum with an elite solution generated during the search. The algorithm uses an strategy based on Candidate List to reduce the number of solutions evaluated in the solution space. The algorithm was tested on benchmark instances taken from the literature and it was able to generate high quality solutions.

Keywords: Vehicle Routing Problem with Simultaneous Pickup and Delivery, Variable Neighborhood Search, Variable Neighborhood Descent, Tabu Search

1   Introduction

The Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRP-SPD) is a derivation of the Vehicle Routing Problems (VRP), in which clients require pickup and delivery services. This problem was proposed by [7] and it is an NP-hard class problem, once it can be reduced to VRP when all demands are null. Given the difficulty to reach optimal solutions in practical cases, VRPSPD is usually solved by heuristic procedures.

Zachariadis et al. [15] use a hybrid technique that combines the metaheuristics Tabu Search [5] and Guided Local Search [14]. In [16], an evolutionary algorithm is proposed. This one uses an adaptive memory to save characteristics of high quality solutions obtained during the search. These characteristics are used to generate new solutions in regions that probably will reach better results, and then they will be improved by Tabu Search method.

Souza et al. [11] introduced an algorithm called GENILS, based on the Iterated Local Search – ILS [3] metaheuristic. GENILS uses the best solution generated by the methods Route-by-Route Cheapest Insertion [2], Cheapest Insertion with Multiple Routes [13] and an adaptation of the GENIUS heuristic [4] as initial solution. It uses Variable Neighborhood Descent – VND [3] as local search method of ILS. The authors show that GENILS outperforms the results of [15].

Subramanian et al. [12] presented a parallel algorithm for solving the VRPSPD, so-called PILS-RVND. It is a multi-start heuristic, where in each iteration an initial solution is generated by the procedure Cheapest Insertion with Multiple Routes. This solution is refined by ILS method using the local search procedure VND with random order of neighborhood (RVND). The RVND explores the solution space through six movements, and the neighborhood order is chosen at random in each iteration. The experiments were performed on two clusters of computers, one with an architecture composed of 128 cores and another with 256. The results obtained on benchmark instances proved that it is the best performance algorithm known so far.

In this work, it is expanded and improved the GENILS algorithm [11]. Instead of the ILS metaheuristic, it uses the Variable Neighborhood Search – VNS metaheuristic [6]. The proposed algorithm embodies three search strategies to GENILS: the Tabu Search as an alternative local search procedure of

VNS after a number of iterations of VND without any improvements; a list of candidates to avoid the analysis of non-promising solutions and the Path Relinking applied to every generated local optimum.

2 Problem Description

The VRPSPD can be defined as follows. Let $G = (V, E)$ be a complete graph with a set of vertices $V = \{0, \cdots, n\}$, where the vertex 0 represents the depot and the remaining ones the clients. Each edge $(i, j) \in E$ has a non-negative cost c_{ij} and each client $i \in V' = V - \{0\}$ has non-negative demands d_i for delivery and p_i for pickup. Let $C = \{1, \cdots, m\}$ be a set of homogeneous vehicles with capacity Q. The VRPSPD consists in constructing a set up to m routes in such a way that: (i) every route starts and ends at the depot; (ii) all the pickup and delivery demands are accomplished; (iii) the vehicle's capacity is not exceeded in any part of a route; (iv) a client is visited by only a single vehicle; (v) the sum of costs is minimized.

Figure 1 shows a solution sample for VRPSPD involving 19 clients and vehicles with load capacity $Q = 150$ unities.
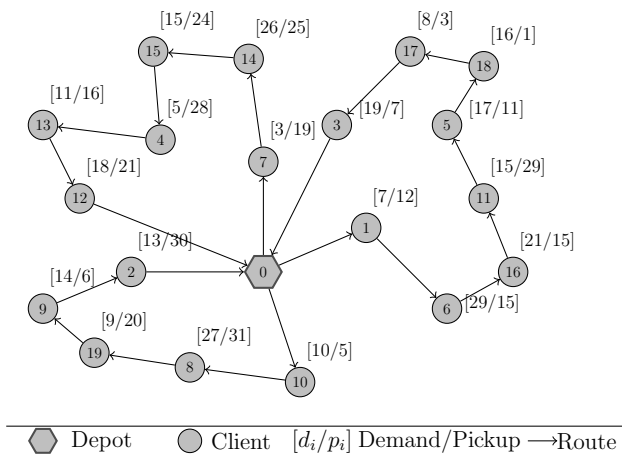

Fig. 1. Example of VRPSPD.

On the superior left corner on Figure 1, the vehicle route starts at the depot, visits the client 7 to deliver 3 product units and to pickup 19 units. Next, it visits the clients 14, 15, 4, 13 and 12, in this order, delivering and collecting products, and returns to the depot.

3 Proposed Algorithm

The proposed algorithm, so-called `GENVNS-TS-CL-PR`, is an improvement of algorithm GENILS [11]. It combines the heuristic procedures Route-by-Route Cheapest Insertion [2], Cheapest Insertion with Multiple Routes [13], an adaptation of GENIUS heuristic – named VRGENIUS – proposed by [11], VNS, VND, Tabu Search (TS) and Path Relinking (PR). The first three procedures are used to generate an initial solution.

Unlike GENILS, that uses ILS to refine this initial solution and VND as local search method, `GENVNS-TS-CL-PR` uses VNS to refine the initial solution, and VND and TS as local search methods. TS is applied after *iterMaxVND* iterations of VND without any improvements on the current solution. In addition, a List of Candidates to avoid non-promising moves is another strategy proposed in this work. Furthermore, the proposed algorithm applies, periodically, the PR method between a local optimum generated by VND or TS and a solution from an elite list formed during the search. Its pseudo-code is outlined on Algorithm 1.

Algorithm 1 `GENVNS-TS-CL-PR`

1: $\gamma \leftarrow$ random real number in the interval $[0.0, 0.7]$;
2: $s^A \leftarrow$ *Route-by-RouteCheapestInsertion*(γ);
3: $s^A \leftarrow VND(s^A)$;
4: $nRotas \leftarrow$ number of routes in solution s^A;
5: $s^B \leftarrow CheapestInsertionMultipleRoutes(\gamma, nRoutes)$; $s^B \leftarrow VND(s^B)$;
6: $s^C \leftarrow VRGENIUS(nRoutes)$; $s^C \leftarrow VND(s^C)$; ;
7: $s \leftarrow argmin\{f(s^A), f(s^B), f(s^C)\}$;
8: $iter \leftarrow 1$; $p \leftarrow 1$
9: **while** ($iter \leq maxIter$) **do**
10:    **for** $i \leftarrow 0$ **until** p **do**
11:       $s' \leftarrow Shake(s)$;
12:    **end for**
13:    **if** ($iter \leq iterMaxVND$) **then**
14:       $s'' \leftarrow VND(s')$;
15:    **else**
16:       $s'' \leftarrow TabuSearch(s', SizeTabuSearch, iterMaxTS)$;
17:    **end if**
18:    *PathRelinking(solElite, s'')*;
19:    **if** ($f(s'') < f(s)$) **then**
20:       $s \leftarrow s''$; $iter \leftarrow 1$; $p \leftarrow 1$
21:    **else**
22:       $iter \leftarrow iter + 1$;
23:    **end if**
24:    **if** ($iter \leq iterMaxShake$) **then**

25:      $p \leftarrow p + 1$
26:    **end if**
27:    *RefreshEliteSet(s");*
28: **end while**
29: Return s;

In line 2 of Algorithm 1 a solution s is built using the Route-by-Route Cheapest Insertion method. In this method a route is progressively built. It consists of a greedy approach that takes into account the least additional cost regarding the insertion of the client k between the clients i and j of the route. The cost of insertion e_{ij}^k is evaluated by the Equation (1).

$$e_{ij}^k = (c_{ik} + c_{kj} - c_{ij}) - \gamma(c_{0k} + c_{k0}) \tag{1}$$

The second part of Equation (1) corresponds to a incentive used to avoid late insertions of clients remotely located. The cost from the depot and back is weighted by the factor γ chosen in line 1. When the capacity of the vehicle is exceeded by inserting a client, another route is initiated. The constructive procedure ends when all the clients have been added to the solution s.

In lines 5 and 6, the methods Cheapest Insertion with Multiple Routes [13] and VRGENIUS [11] are applied. Each solution generated by these methods are refined through the VND procedure. The best solution found is used as initial solution for VNS.

The proposed algorithm uses VND and Tabu Search (TS) as local search procedure of the VNS method. First, the VND procedure is enabled. After *iterMaxVND* iterations without improvements on the current solution, the TS procedure is triggered (line 16 of Algorithm 1). In order to avoid that algorithm stops in a local optimum, the shaking level p is increased after *iterMaxShake* iterations without improvement (line 25). The objective of this strategy is to leave the current region and look for other more promising regions of the solution space.

When the algorithm finds a solution better than the best solution (line 19), the variable *iter* returns to 1 and the shaking level p returns to the initial level (line 20). Thus, the VND procedure is again triggered and TS is disabled until *iterMaxVND* iterations without improvements.

In order to explore the solution space, the movements and the perturbations are the same of [11]. As a way to explore trajectories that connect high-quality solutions, in each VNS iteration the Path Relinking technique is applied between one local optimum produced by the local search method (VND or TS) and one solution of the elite set chosen at random (line 18).

To reduce the number of movements that are not promising, a movement

involving customers i and j is only evaluated if the cost c_{ij} is less than the average of the costs between all pairs of clients. Computational experiments showed that this value does not eliminate the best solutions of the solution space.

4   Computational Experiments

GENVNS-TS-CL-PR was coded in C++, using the Visual Studio C++ 2005 Compiler and executed on a computer *Quad core*, 1,66 GHz with 4 GB of RAM running Windows Vista operational system. In order to test it, we used the instances of [2], [9] and [8].

The results achieved by GENVNS-TS-CL-PR can be seen in Table 1, where the proposed algorithm was compared with three other important algorithms of the literature: the evolutionary algorithm of [16], the parallel algorithm of [13] (256 cores), and the GENILS algorithm of [11]. In this Table, the lines A, B and C, represent the sets of instances of [2], [9] and [8], respectively. The column "Number of instances" shows the number of instances of each set. The columns "Average Gap^{Best}" and "# Best value" mean the average of Gap^{Best} of each instance set analyzed, and the number of solutions for each algorithm that are equal to the best known value in the literature, respectively. The gap of an instance i is calculated by the equation (2), in which $Best_i^{Lit.}$ is the best known value in the literature for the instance i and $Best_i^{Alg}$ represents the best value obtained by the respective algorithm.

$$Gap_i^{Best} = 100 \times (Best_i^{Alg} - Best_i^{Lit.})/Best_i^{Lit.} \qquad (2)$$

Table 1

Comparison between GENVNS-TS-CL-PR and literature results

Set of Inst.	Number of Inst.	Zachariades [16]		Subramanian [13]		Mine [11]		GENVNS-TS-CL-PR	
		Average Gap^{Best}	# Best Value	*Average* Gap^{Best}	# Best Value	*Average* Gap^{Best}	# Best Value	*Average* Gap^{Best}	# Best Value
A	40	0	40	0	40	0	40	0	40
B	14	0.76	4	0.65	8	0.94	4	0.69	6
C	18	0.31	8	0.01	15	0.19	12	0.18	12

We also made empirical probability analysis according to [1] in order to verify how quickly the compared algorithms can find a target value. The experiments showed that the algorithm GENVNS-TS-CL-PR always reached the target value faster than the algorithm GENILS. Further, the separate contribution of each component (TS, CL and PR) was also tested, and the results

confirmed their individual importance in the proposed algorithm. All these experiments are reported in [10].

5 Conclusions

The objective of this research was to propose a competitive algorithm with respect to computational time and solution quality when compared with well known heuristics available in the literature for VRPSPD. The proposed method, named `GENVNS-TS-CL-PR`, uses three construction procedures: Route-by-Route Cheapest Insertion, Cheapest Insertion with Multiple Routes and an adaptation of GENIUS heuristic. The best solution found is used as initial solution. The local search phase is performed by an algorithm based on VNS, which uses VND and Tabu Search algorithms to refine the solutions. The TS algorithm is used only after a number of iterations without improvements of VND. To reduce the number of movements that are not promising, a movement involving clients i and j is only evaluated if the cost c_{ij} is less than the average of the distances between all pairs of customers. Moreover, a Path Relinking scheme was developed.

`GENVNS-TS-CL-PR` has shown to be quite competitive when compared with three important algorithms from the literature. The algorithm proposed by [13] presented the best results; however, it uses 256 cores while the others use only one. When comparing only the sequential algorithms, `GENVNS-TS-CL-PR` shows the best results.

References

[1] Aiex, R. M., M. G. C., Resende, and C. C. Ribeiro, *Probability Distribution of Solution Time in GRASP: An Experimental Investigation*, Journal of Heuristics **8** (2002), 343–373.

[2] Dethloff, J., *Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up*, OR Spektrum **23** (2001), 79–96.

[3] Gaspar-Cunha, A., C. H. Antunes, and R. Takahashi, *Manual of Metaheuristics and Evolutionary Computation* (in portuguese), Coimbra University Press, (2012), ISBN 978-989-26-0150-2.

[4] Gendreau, M., A. Hertz, and G. Laporte, *New insertion and post optimization procedures or the traveling salesman problem*, Operations Research **40** (1992), 1086–1094.

[5] Glover, F. and M. Laguna, "Tabu Search", Kluwer Academic Publisher, Boston, (1997).

[6] Hansen, P., N. Mladenovic, and J. A. M. Perez, *Variable neighborhood search: methods and applications*, 4OR: Quart. J. of the Belgian, French and Italian operations research societies **6** (2008), 319–360.

[7] Min, H., *The multiple vehicle routing problem with simultaneous delivery and pick-up points*, Transportation Research A **23** (1989), 377–386.

[8] Montané, F. A. T., and R. D., Galvão, *A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service.* Computers and Operations Research **33** (2006), 595–619.

[9] Salhi, S., and G. Nagy, *A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling*, Journal of the Operational Research Society **50** (1999), 1034–1042.

[10] Silva, T. C. B., *GENILS-TS-CL-PR: A heuristic algorithm for solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery* (in portuguese), M.Sc. Thesis. Dept. of Computer Science, Federal University of Ouro Preto, Brazil (2012). Available at: `http://www.decom.ufop.br/prof/marcone/projects/vrpspd/ThaisCotta-dissertation.pdf`.

[11] Souza, M. J. F., M. T. Mine, M. S. A. Silva, and L. S. Ochi, and A. Subramanian, *A hybrid heuristic, based on Iterated Local Search and GENIUS, for the Vehicle Routing Problem with Simultaneous Pickup and Delivery*, International Journal of Logistics Systems and Management **10** (2011), 142–157.

[12] Subramanian, A., L. M. A. Drummond, C. Bentes, L. S. Ochi, and R. Farias, *A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery*, Computers and Operations Research **37** (2010), 1899–1911.

[13] Subramanian, A., E. Uchoa, A. Pessoa, and L. S. Ochi, *Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery*, Operations Research Letters **39** (2011), 338–341.

[14] Voudouris, C., and E. Tsang, *Partial constraint satisfaction problems and guided local search*, in Proceedings of the Second International Conference on the Practical Application of Constraint Technology (PACT'96) (1996), 337–356.

[15] Zachariadis, E. E., C. D. Tarantilis, and C. T. Kiranoudis, *A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service*, Expert Systems with Applications **36** (2009), 1070–1081.

[16] Zachariadis, E. E., C. D. Tarantilis, and C. T. Kiranoudis, An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries, European Journal of Operational Research **202** (2010), 401–411.