

On a cooperative VNS parallelization strategy for the capacitated vehicle routing problem

Panagiotis Kalatzantonakis^[0000-0002-0729-6583], Angelo Sifaleras^[0000-0002-5696-7021], and Nikolaos Samaras^[0000-0001-8201-7081]

Department of Applied Informatics, School of Information Sciences,
University of Macedonia, 156 Egnatia Str., Thessaloniki 54636, Greece,
mai18019@uom.edu.gr, sifalera@uom.gr, samaras@uom.gr

Abstract. It is generally accepted that cooperation-based strategies in parallel metaheuristics exhibit better performances in contrast with non-cooperative approaches. In this paper, we study how the cooperation between processes affects the performance and solution quality of parallel algorithms. The purpose of this study is to provide researchers with a practical starting point for designing better cooperation strategies in parallel metaheuristics. To achieve that, we propose two parallel models based on the general variable neighborhood search (GVNS) to solve the capacitated vehicle routing problem (CVRP). Both models scan the search space by using multiple search processes in parallel. The first model lacks communication, while on the other hand, the second model follows a strategy based on information exchange. The received solutions are utilized to guide the search. We conduct an experimental study using well-known benchmark instances of the CVRP, in which the usefulness of communication throughout the search process is assessed. The findings confirm that careful design of the cooperation strategy in parallel metaheuristics can yield better results.

Keywords: Parallel Metaheuristics · Variable Neighborhood Search · Cooperation strategies · Vehicle routing problem · Intelligent Optimization Methods.

1 Introduction

Dantzig and Ramser [9] introduced the CVRP, which belongs to the class of routing problems and is a variation of VRP, with additional constraints on the capacities of the vehicles. CVRP is an NP-hard problem with notable impact on the fields of transportation, distribution, and logistics. The fact that most NP-hard problems become intractable for exact methods, mainly when dealing with large instances, has motivated researchers in developing a plethora of approximation algorithms, heuristics, and metaheuristics that provide an optimal, or close to the optimal, solution. The Variable Neighborhood Search (VNS) metaheuristic has been successfully applied for solving many discrete and global optimization problems [5,13].

Please cite this paper as:

Kalatzantonakis P., Sifaleras A., Samaras N. (2020) "On a Cooperative VNS Parallelization Strategy for the Capacitated Vehicle Routing Problem". In: Matsatsinis N., Marinakis Y., Pardalos P. (eds) Learning and Intelligent Optimization. LION 2019. Lecture Notes in Computer Science, vol 11968, pp. 231-239, Springer, Cham.

The final publication is available at Springer via https://doi.org/10.1007/978-3-030-38629-0_19

The purpose of this paper is to present two parallel VNS methods using the general VNS variant to tackle the CVRP, and to examine how the level of cooperation between threads can affect the performance and the quality of the solutions.

The remainder of this paper is organized as follows: In section 2, we present related works, in which the impact of communication in parallel algorithms is analyzed. In Section 3, we present two parallel VNS models for the solution of the CVRP. In Section 4, we present the summary of the findings for the models. Finally, conclusions and prospects are summarized in Section 5.

2 Related Work

Recently, parallelization processing methods are increasingly being used in metaheuristics, due to the broadly available multicore processors and distributed computing environments. Contributions focused on the communication strategies are sparse. In [6], Crainic focused on different cooperation-based strategies. In this study, Crainic found that approaches based on asynchronous exchanges of information and the formulation of new knowledge out of exchanged data improve the global guidance of the search and display extraordinary performances. The author noticed that, low level communication schemes are particularly attractive when neighborhoods or populations are large, or the neighbor or individual evaluation is costly. Those low level schemes were classified in Crainic taxonomy with the 1st dimension marked as 1-control. Crainic taxonomy is discussed in Section 3.3.

The articles assessed for the literature review relate to cooperative parallel metaheuristics that, are based on the VNS algorithm and have been applied on several problems. Table 1 sums the related works. The authors of these works focus on the effectiveness of the proposed cooperative models rather than the reasoning for selecting the cooperation strategy.

Table 1. Parallel VNS metaheuristic applied in several problems

Related work	Metaheuristic algorithm	Problem
García-López et. al. [10]	Parallel VNS	P-median
Crainic et al. [7]	Parallel VNS	P-median
Aydin and Sevkli [2]	Parallel VNS	Job shop scheduling
Polacek et al. [11]	Parallel VNS	MDVRPTW
Coelho et al. [4]	Parallel VNS	SVRPDSP
Polat O. [12]	Parallel VNS	VRPDP
Antoniadis et. al. [1]	Parallel VNS	Inventory optimization

3 Information exchange between parallel models

It is generally accepted that, adding cooperation to parallel algorithms provides a critical boost to create solutions of the highest quality. In order to study the effect

of communication between the threads, we created two parallel GVNS models. The Savings Algorithm of Clarke and Wright [3] was used to construct the initial solution for the two models. Both models are using an identical neighborhood structure, consisting of three widely used inter and intra-route operators, i.e., 2-opt (Intra-route), Swap (Inter-route), and Relocate (Inter-route). In order for the two models to have the same resources in the search for a solution, a single thread was used to play the role of the solution warehouse.

3.1 Parallel GVNS - Managed information exchange model

In this model, Clarke and Wright's algorithm provides the initial solution to all the threads, except for one, which will assume the role of the solution warehouse. Communication is asynchronous and dynamically determined by each process. The threads begin their search in the solution space and if a thread finds a better solution then, and only then, communication between the current thread and the solution warehouse is initiated. The thread passes the solution to the warehouse/manager process. At that point, a check is taking place. If a better solution exists in the solution warehouse, then it is adopted by the the thread, and the search continues.

The communication schema used in the managed information exchange model is novel and its purpose is to create a sparse communication graph. The target is to maintain an equilibrium between exploration and exploitation phases. As shown in Figure 1, while three solutions are generated, one gets rejected by the warehouse and the thread that passed that solution adopts a new, better solution from the warehouse.

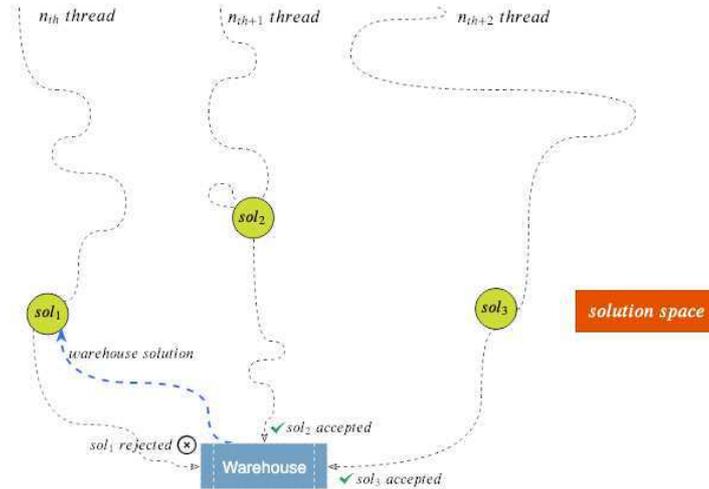


Fig. 1. Three solutions were passed to the warehouse. sol_1 was rejected and the first thread adopted a better solution (sol_3) from the warehouse (blue arrow).

The novelty in the cooperative model resides in the fact that, not only no broadcasting takes place, but also the information exchange between a process and the solution warehouse happens at irregular intervals. Each process dynamically determines those intervals, and even when they occur, the thread might not adopt the available solution from the solution warehouse.

Solution adoption by the warehouse, much like communication initiation is being controlled by each individual process and can be configured to filter-out solutions based on several criteria. The algorithm of this model is shown in Algorithm 1.

Algorithm 1: Pseudo code - Managed information exchange model

```

// fr: Current best solution route
// kmax: Shake k parameter
// timelimit: The time limit stopping criteria
input : fr, kmax, timelimit, initial_solution

while true do
  t ← CpuTime()
  fr' ← Shake(fr, kmax)
  fr'' ← VND(fr', t, timelimit)
  // When a new solution has been found communicate with warehouse
  if thread_solution < thread_current_best then
    thread_current_best ← thread_solution
    if thread_current_best < solution_warehouse then
      // Give solution to warehouse and continue
      solution_warehouse ← thread_current_best
    end
  else
    // Get new solution from solution warehouse and restart.
    thread_current_best, thread_solution ← solution_warehouse
  end
end
if t > timelimit then
  | break
end
if solution_warehouse = optimum then
  | break
end
end

```

3.2 Parallel GVNS - A model with isolated processes

This non-cooperative model, as shown in Figure 2, uses an island-based design where every thread runs the GVNS wholly isolated. All threads utilize identical search procedures. Once the primary solution is produced using the Clarke and Wright algorithm, it is used as a starting point by each thread.

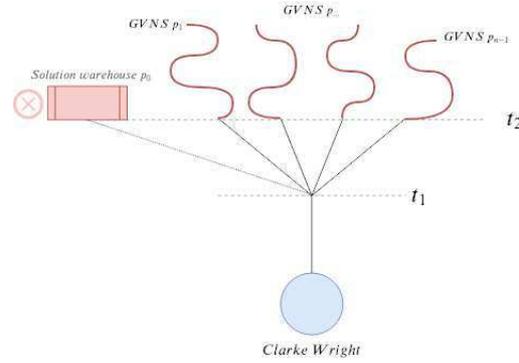


Fig. 2. The Clarke and Wright algorithm generates a solution that is passed to all processes. Best solutions are stored in solution warehouse, but never broadcasted.

Each thread works autonomously, and their paths deviate particularly when the shaking procedure takes place. When the stopping criteria have been met, then all the threads terminate. The best solution is then picked among the list of best solutions. The algorithm of this model is shown in Algorithm 2.

Algorithm 2: Pseudo code - GVNS thread, Isolated model

```

input :  $fr, kmax, timelimit$ 
while true do
   $t \leftarrow CpuTime()$ 
   $fr' \leftarrow Shake(fr, kmax)$ 
   $fr'' \leftarrow VND(fr', t, timelimit)$ 
  if  $t > timelimit$  then
    | break
  end
  // if optimum value exists
  if  $best\_value = optimum$  then
    | break
  end
end

```

3.3 Model Classification

Crainic and Hail [8] suggested three dimensions to classify parallel metaheuristic strategies:

- 1st dimension: Search control cardinality
- 2nd dimension: Search control and communication

- 3rd dimension: Search strategies

According to this taxonomy, the proposed models can be classified as follows:

- The non cooperative model fits into the pC/RS/SPSS classification.
- The cooperative model fits into pC/C/SPSS classification.

The first dimension of this taxonomy defines how the search process for new solutions is controlled; pc stands for poly-Control meaning that, there is more than one process that controls the search operation.

In our case, each single thread has its control for the search operation. The second dimension defines how the information between processes is exchanged. RS stands for rigid synchronization, meaning that little or no information exchange takes place when we use the non-cooperative model. The second dimension in the cooperative model is classified as “Collegial”, thus we extract and adopt only the best solutions when information exchange occurs. The third dimension refers to how new information is created, and the diversity of searches involved. Both models are classified as “SPSS” that stands for “Same initial Point, Same search Strategy”. This makes sense since all the threads use Clarke and Wright as an initial solution and all the threads have an identical neighborhood structure.

4 Computational experiments

This section presents the results of the computational experiments carried out to ascertain the performance of the two parallel GVNS models. The practical relevance of the communication strategy is presented and analyzed.

All the algorithms were implemented in Python 3.7. The experiments were conducted on an Intel Core i9 7940X CPU (3.50 GHz) and 32GB RAM at 3333 MHz. Both models have a single termination criterion; the test is repeated until a certain number of GVNS iterations is met. The two parallel models were tested with the following iterations: 5, 10, 20, 30, 40, 50, 100, 200, and 300. All tests were repeated ten times.

The computational tests were carried out on instances from the X set from the CVRP library [14]. The test set is composed of a subset of the X set containing six instances (X-n110-k13, X-n143-k7, X-n153-k22, X-n256-k16, X-n261-k13 and X-n280-k17). Every instance in the X set was generated by Uchoa with specific characteristics. From the computational effort associated with the instance characteristics and the size of the neighborhood to be explored, the instances can be categorized into the following three groups:

- (a) Easy (X-n110-k13),
- (b) Medium (X-n143-k7, X-n153-k22)
- (c) Hard (X-n261-k13, X-n256-k16, X-n280-k17)

In the results showed in Table 2, we can observe essential differences among the compared methods. The isolated model appears to be much faster but the

cooperative model provides solutions with better quality. “SpI” stands for “seconds per 1 GVNS iteration”. The solution manager never broadcasts the best solution in the cooperative model, in order to minimize the communication overhead. In spite of this fact, the isolated method is 21.108% faster. This can be explained to a large extent by the fact that, communication and solution comparison deprives the search procedure of some CPU cycles. Even though the isolated method is much faster and has a more intense diversification phase, communication between processes appears to yield better results.

Table 2. Comparison of the two GVNS parallel variants

	Isolated scheme	Managed scheme
mean error	10.487%	10.078%
median error	11.460%	11.471%
SpI	4.982	6.315

When we focus on the instances, based on their characteristics and grouped by the computational effort required by the CPU to complete one GVNS iteration, an interesting pattern emerges. As shown in Table 3, information sharing outperforms isolation in hard instances. When the search space is small (easier instances), the non-cooperative method yields better results. Thus, information sharing seems to outperform the independent search method and constitutes a valuable strategy for tackling hard instances when setting small iteration count as a stopping criterion. As shown in Figure 3, after several GVNS iterations, the two methods don’t display essential differences.

Table 3. Model performance based on computational effort

	Isolated scheme	Managed scheme
mean error - (easy)	4.310%	4.955%
mean error - (medium)	11.488%	11.423%
mean error - (hard)	9.131	8.434%
SpI - (easy)	1.234	1.652
SpI - (medium)	5.000	6.153
SpI - (hard)	18.750	20.689

In order to support our findings, we applied a Friedman test to the performance results collected by the two models executed across the same set of instances and obtained a p-value equal to zero showing that, there is enough statistical evidence to consider the two algorithms different. We consider a common significance level $\alpha = 0.05$ as the threshold for rejecting the null hypothesis.

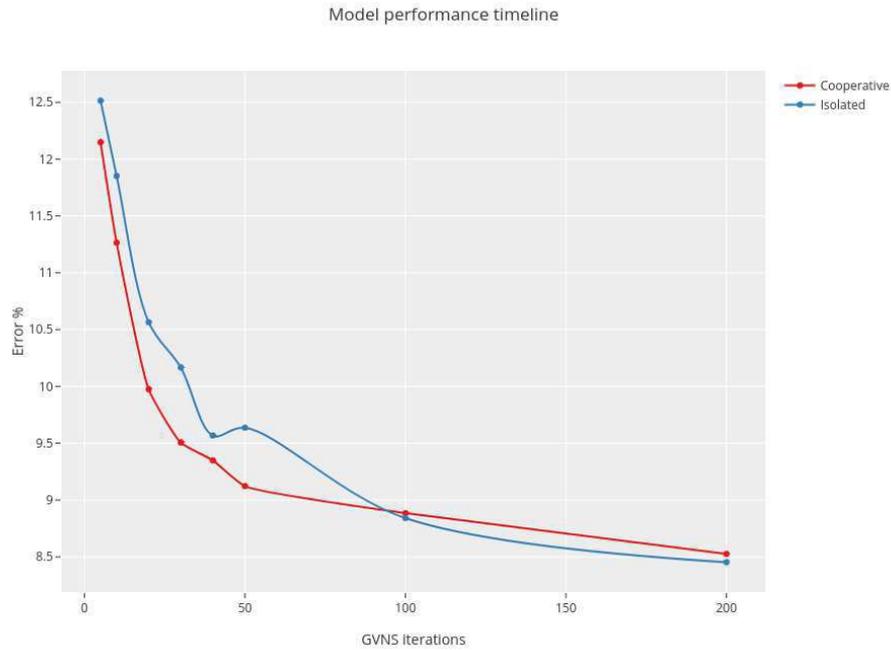


Fig. 3. Performance of the two models at 5,10,20,30,40,50,100 and 200 GVNS iterations

5 Conclusions

In this paper, we proposed two models for the parallelization of the variable neighborhood search for the efficient solution of CVRP. Our goal was to study how the communication between processes affects the performance and the solution quality of parallel algorithms. Well known instances were used in order to compare and analyze the effect of the cooperation strategies between the two parallel metaheuristic models.

Cooperation strategy can have a decisive influence on the quality of the solutions. There is a strong indication that cooperation yields better results over hard instances, whereas in small solution spaces isolation appears to be the best strategy. The timing of communication also appears to play a role since no communication near the end of the search yields better results.

Future studies may include the use of filters to better guide the solution adoption and smarter memory-based strategies to provide better solutions.

Acknowledgements

The second author has been funded by the University of Macedonia Research Committee as part of the “Principal Research 2019” funding scheme (ID 81307).

References

1. Antoniadis, N., Sifaleras, A.: A hybrid CPU-GPU parallelization scheme of variable neighborhood search for inventory optimization problems. *Electronic Notes in Discrete Mathematics* **58**, 47–54 (2017)
2. Aydin, M.E., Sevkli, M.: Sequential and parallel variable neighborhood search algorithms for job shop scheduling. In: *Metaheuristics for Scheduling in Industrial and Manufacturing Applications*, pp. 125–144. Springer Berlin Heidelberg (2008)
3. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* **12**(4), 568–581 (1964)
4. Coelho, I.M., Ochi, L.S., Munhoz, P.L.A., Souza, M.J.F., Farias, R., Bentes, C.: The single vehicle routing problem with deliveries and selective pickups in a cpu-gpu heterogeneous environment. In: *14th IEEE International Conference on High Performance Computing and Communication & 9th IEEE International Conference on Embedded Software and Systems (HPCC-ICISS)*. pp. 1606–1611. IEEE (2012)
5. Coelho, V.N., Santos, H.G., Coelho, I.M., Penna, P.H.V., Oliveira, T.A., Souza, M.J.F., Sifaleras, A.: 5th International Conference on Variable Neighborhood Search (ICVNS'17). *Electronic Notes in Discrete Mathematics* **66**, 1–5 (2018)
6. Crainic, T.: Parallel metaheuristics and cooperative search. In: *Handbook of Metaheuristics*, pp. 419–451. Springer International Publishing (2018)
7. Crainic, T.G., Gendreau, M., Hansen, P., Mladenović, N.: Cooperative parallel variable neighborhood search for the p-median. *Journal of Heuristics* **10**(3), 293–314 (2004)
8. Crainic, T.G., Hail, N.: Parallel metaheuristics applications. *Parallel metaheuristics: A new class of algorithms* **47**, 447–494 (2005)
9. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. *Management Science* **6**(1), 80–91 (1959)
10. García-López, F., Melián-Batista, B., Moreno-Pérez, J.A., Moreno-Vega, J.M.: The parallel variable neighborhood search for the p-median problem. *Journal of Heuristics* **8**(3), 375–388 (2002)
11. Polacek, M., Benkner, S., Doerner, K.F., Hartl, R.F.: A cooperative and adaptive variable neighborhood search for the multi depot vehicle routing problem with time windows. *Business Research* **1**(2), 207–218 (2008)
12. Polat, O.: A parallel variable neighborhood search for the vehicle routing problem with divisible deliveries and pickups. *Computers & Operations Research* **85**, 71–86 (2017)
13. Sifaleras, A., Salhi, S., Brimberg, J. (eds.): *Variable Neighborhood Search: ICVNS 2018, LNCS*, vol. 11328. Springer, Cham (2019)
14. Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., Subramanian, A.: New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research* **257**(3), 845–858 (2017)