

Sequential order vs random order in operators of variable neighborhood descent method

Darmawan Satyananda*¹, Sapti Wahyuningsih²

Mathematics Department, Universitas Negeri Malang, Semarang St. No. 5 Malang, Indonesia,
telp: (0341) 552182

*Corresponding author, email: darmawan.satyananda.fmipa@um.ac.id¹,
sapti.wahyuningsih.fmipa@um.ac.id²

Abstract

Many optimization problems require heuristic methods to solve the problem. Variable Neighborhood Search (VNS) is a metaheuristic form that systematically changes its "neighborhood" in search of solutions. One method in VNS is Variable Neighborhood Descent (VND), which performs a deterministic neighborhood change. The change of the neighborhood in VND can be done in a random and sequential order. This paper compares sequential and random neighborhood selection methods in solving Capacitated Vehicle Routing Problem (CVRP) problems. There are 6 intra-route neighborhood structures and 4 inter-route structures used. CVRP problems are taken from several dataset providers. The initial solution is formed by Sequential Insertion method. The experimental results show that the random selection of neighborhood operators can provide a more optimal route length (in 10 of 13 datasets used) than that of sequential selection (only better in 3 dataset). However, the random selection takes more iterations to reach convergent state than the sequential one. For sequential selection, determination of the neighborhood structure's order affects the speed to the convergent state. Hence, a random selection in VND method is more preferable than sequential selection.

Keywords: CVRP, local search, neighborhood selection methods, VND, VNS

Copyright © 2019 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Vehicle Routing Problem (VRP) have long been known as "difficult" problems where no algorithm in polynomial time proportional to the size of the instance can solve all the problems (belongs to *NP-hard* family). The exact method may solve a "big" problem, however, it takes a very long execution time. By using the exhaustive search (brute force), it would have obtained a solution (it's optimal), but this would require a very large computation time (it isn't efficient). Algorithms that are significantly faster than exhaustive search is still possible to be designed for some problems, however, still not in polynomial time [1]. Hence, a heuristic algorithm is introduced that can immediately provide an approximate solution which is sometimes considered as the optimal solution. Researchers focused on approximate algorithm that provides near-optimal solution in quick time [2].

The problem found in heuristics is local optimum (local minima), i.e. a solution that never reaches a feasible state (global optimum value). In this instance, the algorithm is "trapped" in certain circumstances where the operation only generates values around the local optimum area (optimum at within a neighborhood is not necessarily optimum in another one). The global optimum is a proven optimum solution for all locals. To overcome the problems that arise in the heuristics, metaheuristic is raised; it is a framework to build heuristics for combinatorial problems and global optimization [3].

One of the frameworks offered to generate solutions using metaheuristic is Variable Neighborhood Search (VNS) metaheuristic which exploits systematically the idea of neighborhood change, both in descent to local minima and in escape from the valleys which contain them [3]. A variant of VNS is Variable Neighborhood Descent (VND) which explore neighborhood in a deterministic way. They both belong to a local search metaheuristic group, applying iteratively to various neighborhood structures, modifying an initial solution [4]. The use of various neighborhood structures in the local search will result in a better solution than the original solution.

Types and number of neighborhood operator in VND affect performance of VND [4], but in other hand it will increase operation time as iteration will increase as well. Sequence of operator is important since it influences how fast convergence will be reached. Operators can be examined in sequential or random [5, 6], moreover, in arbitrary way. It is necessary to study about method of the selection, to find better configuration of operators needed in finding best optimal solution of VRP.

This paper discusses the comparison of a selecting VND operator in sequential and random order, in CVRP. The paper is organized as follows. In Introduction Section, the literature related to VNS, VND, and neighborhood operators are reviewed. Next section is about methodology. Finally, test results for some instances and some computational results are discussed in Results and Discussion Section.

Variable Neighborhood Search (VNS)

Metaheuristic is a common framework for building heuristics for combinatorial issues and global optimization. One of the most widely used metaheuristic is Variable Neighborhood Search (VNS) developed by Mladenović since 1995 [3]. VNS systematically changes the neighborhood, either in the gradual form leaving local optimum or "jumping" from the valley that leads the local optimum by using perturbation. Simply put, "neighborhood" is a set of solution results from the application of a particular operation. From an initial solution, an operation can generate various alternative solutions; those are "neighborhoods". Figure 1 shows the local optima and perturbation conditions for jumping to other expected circumstances.

Basically, VNS consists of stages:

- Shaking (perturbation): to determine a solution x' to be fixed. Shaking is done with the initial solution x using operations taken from a particular neighborhood.
- Repairing x' with local search using a particular neighborhood operator, generating x''
- Change Neighborhood by comparing local search results x'' with initial solution x . If x'' is better than x , then x'' is considered as a new x and the iteration is reset back to the first neighborhood operator. Otherwise next operator is applied to x' .

The problem that arises is to determine what operators can be included in the neighborhood structure, and how the sequence, so as to produce global or near-optimal. Variable Neighborhood Descent (VND). VND is obtained when the neighborhood change is determined deterministically, by eliminating the random shaking stage. VND explores the solution space using various neighborhood structures to ensure that there are optimal solutions in all different neighborhoods thus it can be considered as global solutions. Algorithm of VND can be seen in Figure 2 [3].

N_ℓ is a neighborhood structure as a set of operators applied to a neighborhood, x is an initial solution, and $N_\ell(x)$ is set of solutions using ℓ^{th} neighborhood of x . At the exploration of neighborhood stage, a local search for a neighborhood that gives the best x' of x is performed, started from the first neighborhood. At the next stage ("move or not" stage), a neighborhood index is set to the first neighborhood when the solution x' is better than x , and to the next neighborhood when x' equal or even worse than x . The iteration stops after all neighborhoods are examined and no improvement can be found. This is to ensure that x' are optimum for all neighborhoods, starting from the first neighborhood.

VND (even VNS) is based on principles: a local optimum in a certain neighborhood structure does not correspond to local optimum in other neighborhood structure (but in many problems, they are close each other), and global optimum corresponds to all local optimum in all neighborhood structures [6]. Choice of the neighborhoods at each iteration is main factor form the performance of the VND. The choice of structure (operator to find a "best" solution in certain neighborhood) can be done sequentially, nested, or a combination of the two [6, 7]. Moreover, random neighborhood selection gives better results than deterministic selection [8], and does not require a neighborhood setting [9]. The repetition to the first neighborhood by [7] is referred as Basic VND. There are other variants, such as Pipe VND (continue searching for the same neighborhood), Cyclic VND (proceed to the next neighborhood), and Union VND (a neighborhood that is a composite of several neighborhoods, so the search is continued in the combined neighborhood).

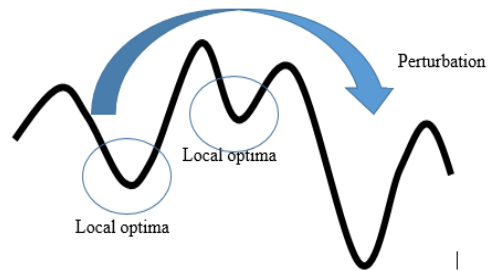


Figure 1. Local optima condition and perturbation to jump from lofrical optima

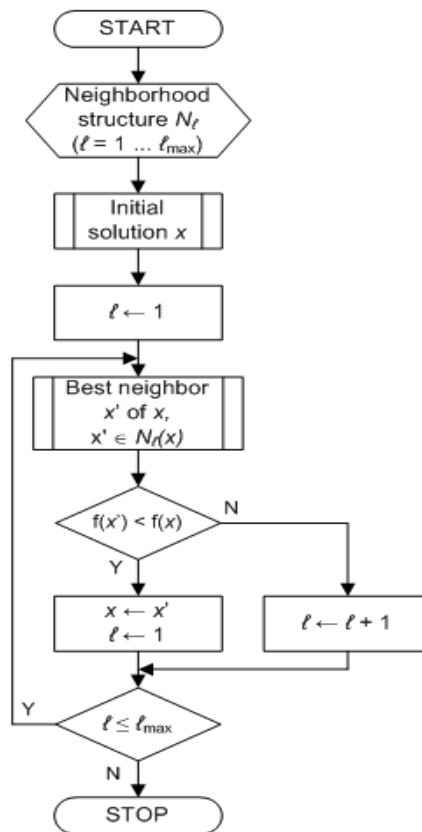


Figure 2. Stages of VND

Some things noticed in choosing neighborhood are: (1) Number of neighborhood operators used, (2) The type of neighborhood operators used, (3) The order of the neighborhood, and (4) Whether the use of the neighborhood ensure the most optimal results. Mistakes in selection can result in enormous computing time, remain trapped in a certain neighborhood, and less optimal results.

To the best of author's knowledge, the basic structure of VND (and VNS as well), is still relatively remain unchanged. Variations is in addition of other method (as a hybrid VND), variations in number and type neighborhood operators, and area of implementation. Other interesting feature is implementation parallel programming due to a great number of iterations in local search process and to take advantage of multi-core computer processors [8].

VNS and VND are reported have been successfully implemented in many cases such as Traveling Salesman Problem [7], and Vehicle Routing Problem [2, 5], Location Routing Problem [10], Flowshop Scheduling Problem [11], Redundancy Allocation Problem [12], Order Batching Problem [13], Dial-a-ride problem (special case of VRP for patient transportation or

disabled people) [14], and in minimum labelling Steiner tree problem [15], and Course Time Table Problem [16]. Those references reported that VNS/VND could improve the objective function compared to other method.

VND in many cases is combined (hybridized) with other method. Genetic Algorithm is hybridized with VND for flexible job shop scheduling problems [2], [17] used Iterated Local Search (ILS) VND in VRPTW, [11] used ILS VND in flowshop scheduling problem, [18] used ILS VND in VRP with Multiple Trip, [6] used Intensified ILS Method with Random VND in VRPTW, and [19] that used Iterated VND for CVRP problems. Those articles proved that the combination could give better result, compared to other method and other best results known. A careful consideration must take into account, since ILS uses VND in two steps: to refine initial result, and in another iteration to meet acceptance criterion, as in [2]. Another interesting hybridization are in [20, 21] that used GRASP (*Greedy Randomized Adaptive Search Procedure*) in VNS and VND to control randomness in greedy heuristic.

Neighborhood Operators

Neighborhood operator can be classified into two classes: intra-neighborhood and inter-neighborhood. Intra-neighborhood operations are performed on a single route with the goal of minimizing the overall distance, and inter-neighborhood operations are performed on two or more different routes with the goal of minimizing the overall distance and even minimizing the number of vehicles [5].

Examples of inter-neighborhood are [9]:

- Shift(1,0) or 1-0 Exchange Neighborhood: One customer i is moved from route r_1 to r_2 .
- Shift(2,0) or 2-0 Relocate Neighborhood: Two adjacent customers i and j (or an arc (i, j)) are moved from route r_1 to route r_2 . The transfer of the opposite arc (j, i) is also considered.
- Swap(1,1) or 1-1 Exchange Neighborhood: Interchange between customer i from route r_1 and customer j from route r_2 .
- Swap(2,1) or 1-2 Exchange Neighborhood: Interchange between adjacent customers i and j from route r_1 and customer k from route r_2 . The opposite arc (j, i) is also considered as well.
- Swap(2,2) or 2-2 Exchange Neighborhood: Interchange between adjacent customers i and j from route r_1 and two other adjacent customers k and l , from route r_2 . The opposite arcs (j, i) and (l, k) are also considered, yielding 4 possible combinations.
- Cross Exchange: exchange of two arcs (two strings of consecutive nodes) from two routes, preserving the order of customers in each string [7], in swapping two paths belonging to two different routes [5].

Examples of intra-neighborhood are [8]:

- Or-opt: one to three consecutive nodes are moved into other position in the route.
- Two-opt: two nonadjacent edges are removed, and two new edges are formed then to make a new route.
- Exchange: exchanging two nodes position. This is another version of swap(1,1), but it applied to the same route.
- Reinsertion: a node is removed, and inserted into other position in the route.

2. Research Method

This paper compares selecting neighborhood structure, in sequential and random order, during process of repairing the initial CVRP route using VND Metaheuristic. The initial route was generated based on the Sequential Insertion algorithm based on [22]. There were 10 neighborhood operators used for local search: 1-0 Exchange neighborhood (N1), 1-1 Exchange neighborhood (N2), 1-2 Exchange neighborhood (N3), 2-2 Exchange neighborhood (N4), 2-0 Relocate neighborhood (N5), Cross-Exchange neighborhood (N6); and (b) intra route operator: Or-opt neighborhood (N7), Two-opt neighborhood (N8), Exchange (N9), Reinsertion (N10). The initial route establishment and repair process was implemented in a computer program created with Borland Delphi, executed in computer with Intel i-7 microprocessors, with 32 GB memory and operating system Microsoft Windows 7.

To test the comparison, 13 data tests were used. Those were taken from <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/> of various publisher: Augerat, Christofides and Eilon, Christofides-Mingozzi-Toth, Fisher, Taillard, and Uchoa. Number of customers ranged from 16 to 200 customers. Properties of each data set are presented in Table 1.

Table 1. Properties of Each Dataset

Dataset	Publisher	Number of depot & customer	Capacity of vehicle	Best Record
E-n13-k4	Christofides and Eilon	13	6000	247
P-n16-k8	Augerat set P	16	35	450
B-n31-k5	Augerat set B	31	100	672
A-n32-k5	Augerat set A	32	100	784
B-n45-k5	Augerat set B	45	100	751
F-n45-k4	Fisher	45	2010	724
A-n60-k9	Augerat set A	60	100	1354
P-n60-k15	Augerat set P	60	80	968
B-n63-k10	Augerat set B	63	100	1496
tai75a	Taillard	76	1445	1618,36
M-n101-k10	Christofides-Mingozzi-Toth	101	200	820
X-n101-k25	Uchoa [23]	101	206	27591
M-n200-k17	Christofides-Mingozzi-Toth	200	200	1373
X-n200-k36	Uchoa [23]	200	402	58578

For random neighborhood selection, 5 executions were performed and its average result was taken then. The selection of neighborhood in sequential order was only done in 1 run time because it did not involve random number generation, hence the algorithm always gave the same result at every running for each data set. Objective for each running is to find the route with shortest length, that is route that started from depot, passes every customer, and back to the depot as final destination. There is no requirement on restrictions of route length for all datasets.

3. Results and Analysis

The computer program made was then executed using test data prepared. Table 2 summarizes the results of program execution. The best results from this experiment are then compared to best record for each dataset. Table 2 presents result of each dataset. For each case was recorded best value, worst value, average value, number of iterations, and ratio of best case to best record known, both for sequential and random selection. The table presents fact that selecting operators in random order can provide a more optimal route length than the sequential order on 10 of 13 datasets. This is in accordance with the [12] which states that random selection can provide better results. In general, number of iterations in random selection is less than iteration in sequential selection. Number of iterations get higher as the number of customers increases, since more local search are performed (in "exploration of neighborhood" stage), and more comparisons to check optimality of a new modified route (in "move or not" stage).

Table 2. The Result of Experiments in Worst and Best Cases

Dataset	Random					Sequential				
	Avg.	Worst	Best	Ratio to best record (%)	Average Number of iterations	Avg	Worst	Best	Ratio to best record (%)	Number of iterations
E-n13-k4	348,58	1898	242	-2,07	800	367,52	3063	237	-4,22	717
P-n16-k8	490,03	1970,95	451,34	0,30	2106	490,03	1696,31	462,69	2,74	1919
B-n31-k5	1209,14	33741,84	687,55	2,26	33123,6	1209,14	40236,89	717,13	6,29	28573
A-n32-k5	1395,24	61423,91	832,52	5,83	42600,8	1395,24	50608,88	828,7	5,39	41406
B-n45-k5	1105,97	45139,9	736,7	-1,94	97104	1237,62	75617,72	770,31	2,51	46873
F-n45-k4	1696,36	213882,3	744,44	2,75	212434,6	1696,36	140940,7	732,77	1,20	254398
A-n60-k9	1752,61	70392,34	1367,53	0,99	292676	1752,61	67325,4	1393,46	2,83	318498
P-n60-k15	1113,58	17515,32	985,45	1,77	127335,6	1113,58	15734,86	995,61	2,77	203174
B-n63-k10	2005,23	96245,7	1561,74	4,21	270407	2005,23	70257,4	1578,3	5,21	495880
tai75a	2128,22	191476,2	1661,36	2,59	917429,6	2128,22	139625,7	1764,73	8,29	535307
M-n101-k10	1278,02	137612,2	822,81	0,34	1410667,2	1278,02	96846,79	834,71	1,76	1941136
X-n101-k25	30732,39	625645,4	28263,16	2,38	499014,2	30732,39	509152,3	28514,68	3,24	765420
M-n200-k17	1079,61	305078,3	1470,1	6,60	19990537,4	1079,61	276206,5	1491,27	7,93	39873211
X-n200-k36	66401,62	1837675	60853,73	3,74	3730677,2	66401,62	1643422	60939,3	3,87	7294713

Figure 3 is graphs showing the results obtained per iteration for E-n13-k4 as an example of dataset. X-axis states number of iteration, and Y-axis states length of route found. Data of these graphics are taken from the best result on each iteration. For each picture, leftside

graphic is for random selection, and rightside graphic is for sequential selection. Selection of operators in sequential order provides a more stable output graph than a random selection at an equivalent level, but there are operations that produce "jumping" results, making it the worst route as shown in Figure 3 (b). For example, in Figure 3 (b), the spikes are about in length of 3000 unit, and mostly other results are about below 500 unit. Meanwhile, in Figure 3 (a), the highest results are about in level of 625 unit, and in average, results in the end of iteration are about 300 unit. This pattern is also found in other datasets. In Table 2, column of "Worst" stated the worst case, with value much higher than the average length.

The graphs on the figures also showed that an operator could generate a completely "new" route, which is not just an improvement over the previous route. Such a route was noted as a "spike", which had much worse length than previous route, and surely it would not be chosen because a route must be minimal on all structure (N1-N10) to be said as a global optimum. From observation, this maximum-length route occurs in two-opt (N8) operation. The existence of this "leap" can be used as a consideration for not using of this operator on local search.

From all datasets, unfortunately VND method still could not give better result than best result known in the reference (except for E-n13-k4), but still in acceptable level. Ratio best value to best record for all dataset ranged from -2.07% to 6.6% for random selection, and ranged from -4.22% to 8,29% for sequential selection). In average, ratio of best results in random order is 2.13% which is better than average for selection order, that is 3.56%.

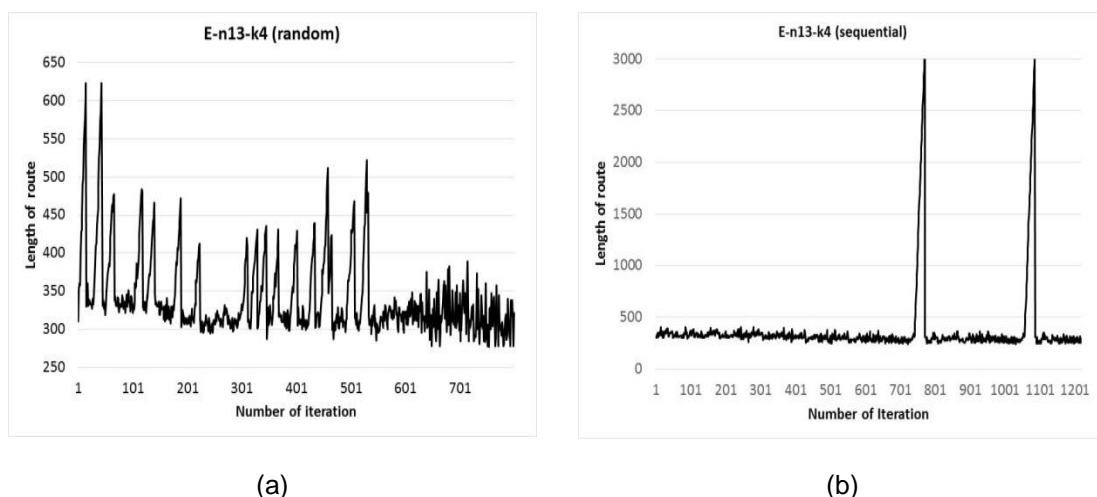


Figure 3. The Result of execution for E-n13-k4

Route length relates to customer's demand and capacity of vehicle (CVRP assumes that number of vehicles is unlimited [24]). If the total demand of all customers is far beyond the capacity of the vehicle, the more routes are generated. The number of routes will affect the number of iterations for local search and for checking the optimization of a route. This number of iterations is also directly proportional to the number of customers on a route. For example, the Shift (1,0) operation on a route that contains n customers will be performed n times to find the best improvement. If there exists m route, there will be $m \times (m-1) \times n$ displacement. If there a route x' which is better than x , the neighborhood will be re-examined to ensure that the result is the global optimum with respect to all neighborhoods. The more iterations occur, the longer the execution time would be. The dataset that has the longest execution time is M-n200-k17, which had 19 million iterations that took 6 minutes to complete.

Maximizing number of customers in a route depends on demand of each customer. If the demand is relatively large compared to capacity of vehicle, then there will be relatively large unoccupied space in a vehicle. Demand for each customer in CVRP cannot be splitted to maximizing vehicle capacity, contrary to VRP with Split Demand (VRPSD). In [3] stated that some considerations that influence the result: whether to use first or best improvement,

reducing the neighborhood, and distribution of neighborhoods. Number of operators used needs to be considered in reducing the neighborhood. There are several operators that are similar in exchange and relocate type, hence it can be considered to choose only one of them. Usage of similar operator does not much influence optimum result [25].

Reduced operators will have an impact on processing time and the number of iterations required, as fewer "move or not" steps are performed. Specifically, for sequential selection, operator sequences should be considered by combining intra-route and inter-route neighborhoods or choosing the most advantageous operator. Other ways of improvement can be considered also, as in implementation of parallel programming [8], and global vector (global memory) to store all generated neighbor solutions to avoid any duplication during the execution [26]

4. Conclusion

The use of the VND framework may be considered in the improvement process, particularly on VRP issues. The selection of operators in random order have proven provide better results on the data test used than that of the sequential selection. The types and sequence of operators can be considered further in other experiments in order to provide better results, as well as implementation of parallel programming to shorten execution time in cases of great number of customers.

Acknowledgements

This article is part of research entitled "Utilization of Google Map Services for the Application of VRP Problem Solving" (year 2017, first year of two years planned). Special thanks addressed to Directorate of Research and Community Services (DRPM), Indonesian Ministry of Research, Technology, and Higher Education (Kemenristekdikti) as funding this research.

References

- [1] Woeginger GJ. Exact Algorithms for NP-Hard Problems: A Survey. In: Jünger M, Reinelt G, Rinaldi G (eds) Combinatorial Optimization—Eureka. *You Shrink! New York: Springer-Verlag*. 2003; 185–207.
- [2] Harzi M, Krichen S. Variable neighborhood descent for solving the vehicle routing problem with time windows. *Electron Notes Discret Math* 2017; 58: 175–182.
- [3] Hansen P, Mladenović N. Variable neighborhood search. In: Burke EK, Kendall G (eds) Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. *New York: Springer Science+Business Media*. 2014; 313–337.
- [4] Chen B, Qu R, Bai R, et al. An investigation on compound neighborhoods for VRPTW. In: *Communications in Computer and Information Science*. 2017, 3–19.
- [5] Pop PC, Fuksz L, Marc AH. A Variable Neighborhood Search Approach for Solving the Generalized Vehicle Routing Problem. *Hybrid Artif Intell Syst*. 2014; 8480: 13–24.
- [6] Ribas S, Subramanian A, Machado I, et al. A hybrid algorithm for the Vehicle Routing Problem with Time Windows. *Proc Int Conf Ind Eng Syst Mana*. 2011; 1243--1252.
- [7] Mjirda A, Todosijević R, Hanafi S, et al. Sequential variable neighborhood descent variants: an empirical study on the traveling salesman problem. *Int Trans Oper Res*. 2017; 24(3): 615–633.
- [8] Subramanian A, Drummond LMA, Bentes C, et al. A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Comput Oper Res*. 2010; 37(11): 1899–1911.
- [9] Silva MM, Subramanian A, Ochi LS. An iterated local search heuristic for the split delivery vehicle routing problem. *Comput Oper Res*. 2015; 53: 234–249.
- [10] Jarboui B, Derbel H, Hanafi S, et al. Variable neighborhood search for location routing. *Comput Oper Res*. 2013; 40(1): 47–57.
- [11] Yahyaoui H, Krichen S, Derbel B, et al. A hybrid ILS-VND based hyper-heuristic for permutation flowshop scheduling problem. In: *Procedia Computer Science*. 2015, 60: 632–641.
- [12] Liang YC, Lo MH, Chen YC. Variable neighbourhood search for redundancy allocation problems. In: *IMA Journal of Management Mathematics*. 2007; 18(2): 135–155.
- [13] Menéndez B, Pardo EG, Alonso-Ayuso A, et al. Variable Neighborhood Search strategies for the Order Batching Problem. *Comput Oper Res*. 2017; 78: 500–512.
- [14] Parragh SN, Doerner KF, Hartl RF. Variable neighborhood search for the dial-a-ride problem. *Comput Oper Res*. Epub ahead of print 2010. DOI: 10.1016/j.cor.2009.10.003.

- [15] Consoli S, Darby-Dowman K, Mladenović N, et al. Variable neighbourhood search for the minimum labelling Steiner tree problem. *Ann Oper Res*. 2009; 172: 71–96.
- [16] Soria-Alcaraz JA, Ochoa G, Sotelo-Figueroa MA, et al. Iterated VND versus hyper-heuristics: Effective and general approaches to course timetabling. In: Melin P, Castillo O, Kacprzyk J (eds) *Studies in Computational Intelligence*. Springer, Cham. 2017; 687–700.
- [17] Gao J, Sun L, Gen M. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Comput Oper Res* 2008; 35: 2892–2907.
- [18] Belver M, Gomez A, Lopez J, et al. Solving Vehicle Routing Problem with Multiple Trips using Iterative Local Search with Variable Neighborhood Search. In: *Int'l Conf. Artificial Intelligence | ICAI'17*. 2017; 286–290.
- [19] Chen P, Huang H kuan, Dong XY. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Syst Appl*. 2010; 37(2): 1620–1627.
- [20] Zeng ZY, Xu WS, Xu ZY, et al. A Hybrid GRASP+VND heuristic for the two-echelon vehicle routing problem arising in city logistics. *Math Probl Eng*; 2014. Epub ahead of print 2014. DOI: 10.1155/2014/517467.
- [21] Salehipour A, Sörensen K, Goos P, et al. Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem. *4OR quarterly journal of the Belgian, French and Italian Operations Research Societies*. 2011; 9(2): 189–209.
- [22] Joubert J, Claassen S. A sequential insertion heuristic for the initial solution to a constrained vehicle routing problem. *ORiON J ORSSA* 2006; 22(1): 105–116.
- [23] Uchoa E, Pecin D, Pessoa A, et al. New benchmark instances for the Capacitated Vehicle Routing Problem. *Eur J Oper Res* 2017; 257(3): 845–858.
- [24] Baldacci R, Mingozzi A. A unified exact method for solving different classes of vehicle routing problems. *Math Program*. 2009; 120(2): 347–380.
- [25] Bögl M, Zäpfel G, Affenzeller M. *Computational study of neighborhood operator performance on the Traveling Salesman Problem with Time Windows in neighborhood search based frameworks (RTS, VNS)*. In: *LINDI 2011-3rd IEEE International Symposium on Logistics and Industrial Informatics, Proceedings*. 2011; 39–44.
- [26] Ren L, Duhamel C, Quilliot A. A hybrid ILS / VND heuristic for the One-Commodity Pickup-and-Delivery Traveling Salesman Problem. In: *International Workshop on Green Supply Chain*. 2012. Epub ahead of print 2012. DOI: 10.13140/RG.2.1.4192.5286.