A VARIABLE NEIGHBORHOOD SEARCH METAHEURISTIC IN SOLVING THE VEHICLE ROUTING PROBLEMS: LOGISTICS CASE EXAMPLES

Dražen Popović, Nenad Bjelić, Milorad Vidović, Gordana Radivojević

a University of Belgrade, Faculty of Transport and Traffic Engineering, Department of Logistics, Serbia

http://doi.org/10.37528/FTTE/9786673954196.010

Abstract: Despite the unquestionable significance of the vehicle routing problem for stakeholders in logistics processes, and numerous research conducted with the goal of improving solution procedures of the problem, its complexity caused that there is no universally applicable procedure for solving different variants of the problem. In this research we present the implementation of the variable neighborhood search algorithm in solving inventory routing, container drayage and travelling repairmen problems, with the goal to introduce an interested reader to the practical implementation of the VNS in solving vehicle routing based logistics problems.

Keywords: variable neighborhood search, vehicle routing problem, inventory routing, container drayage, travelling repairmen.

1. INTRODUCTION

Operation of logistics systems are an area in which efficient solving of the vehicle routing problem (VRP) play very important role in the realization of everyday tasks. Despite the unquestionable significance of the VRP for numerous stakeholders in logistics processes, and hundreds of researches conducted with the goal of improving solution procedures of the VRP, its complexity caused that so far there is no universally applicable procedure for solving different variants of the VRP. As a consequence, accommodation of specially designed procedures is the most efficient way of solving the VRP. Accordingly, implementation of heuristics and metaheuristics algorithms in solving real-life problems has attracted the vast majority of scientific researches, and proved to be very efficient. Usually, some new algorithm, or solving concept, is accepted as a procedure for solving a problem if it improves solution either in terms of solution quality, or computational time, or both.

In this research we focus on the implementation of one metaheuristic concept that proved to be very efficient in solving large number of combinatorial optimization problems. More precisely, we give short overview of the variable neighborhood search (VNS) algorithm’s development and implementation in three research studies conducted at the Logistics department of the Faculty of Transport and Traffic Engineering in the

1 Corresponding author: d.popovic@sf.bg.ac.rs
previous seven years: inventory routing, container drayage and travelling repairmen problems. The goal of this paper is to bring to an interested reader closer look to the implementation of the VNS in solving VRP based logistics problems.

Accordingly, this paper is organized in such a way that in Section 2 we give a short overview of the VRP’s main features. In Section 3 we present main concepts that drive the VNS. Afterwards, in Section 4 we give overview of the implementation of the VNS in solving three before mentioned VRP bases logistics studies. Eventually we give a brief conclusion regarding the experience in using the VNS.

2. THE VEHICLE ROUTING PROBLEM

The VRP is one of the most discussed optimization problems for which a large number of exact and heuristic approaches have been developed. The application areas of the VRP are practically unlimited and refer to any system in which there is a need for transportation on a network of nodes (source locations, user locations, pick up locations, delivery locations etc.). Dantzig and Ramser (1959) presented the original work in which the VRP was first formulated as “The Truck Dispatching Problem”. The authors observed the problem of fuel distribution from a depot (bulk terminal) to a large number of service stations, where solving the problem involves the assignment of stations to capacitated vehicles with routes construction in order to minimize the total distance travelled while meeting the demand of all stations. Since 1959, a large number of papers have been published for the VRP, which discussed various variants of the classical VRP problem. The classical capacitated VRP consists of one depot and a set of users that need to be serviced by a capacitated set of vehicles. Toth and Vigo (2002) define the following basic variants of the VRP (Figure 1):

- Capacitated VRP (CVRP)
- Distance-Constrained CVRP (DCVRP) – vehicles have a maximum allowed route length (time or distance related);
- VRP with Time Windows (VRPTW) – with a service time window in each node;
- VRP with Backhauls (VRPB) – service nodes (or customers) are divided into two sets. One set is for linehaul deliveries, and the other is for backhaul pickups. All linehaul customers must be served before any backhauling in each route;
- VRP with Pickup and Delivery (VRPPD) – each customer can have mixed delivery and pick-up quantities, where both can be serviced simultaneously in one node.

Many extensions of the VRP can be found in the literature that tries to solve problems which came from real-life situations with a different set of specific constraints and/or additional interdependent objectives. One of those approaches is the Inventory Routing Problem (IRP) that extends the classical VRP with simultaneous optimization of routing and inventory related decisions (stock level, time and quantity of delivery, etc.). The Location Routing Problem includes location aspect, while Production Routing Problem includes production aspect. In the case of real-time optimization where important information becomes available during the realization of a solution, we have Dynamic VRP. Also, the Stochastic VRP is a variant where some conditions are uncertain and can be described by a probability distribution. The VRP with synchronization tries to solve routing problems where different vehicles need to be synchronized (e.g. routes of trucks with cargo and mobile cranes should be synchronized for those nodes that do not have
material handling equipment). Travelling Repairman Problem (TRP) is a variant of the VRP in which routing is defined with a simultaneously interest of all customers as a focus, resulting in the increased complexity of the problem. In the case of route consistency requirements (e.g. customers must be visited by certain drivers) we have the Consistent VRP. The split delivery VRP observes the possibility of satisfying a customer’s demand by more than one vehicle (a quantity for a customer can be delivered by multiple vehicle routes). If special vehicles with compartments are used, we have the case of the VRP with compartments (transportation of milk, fuel etc.). When vehicles can be used for multiple routes, we have the case of the Multi trip VRP, while in the case of multiple depots with vehicles we have the Multi depot VRP. In the Clustered VRP customers are partitioned into zones (clusters) that are served by its vehicle. The Arc Routing Problem is a special case of vehicle routing where network links (arcs) between nodes should be visited (instead of the nodes).

![Diagram of basic variants and extensions of the VRP](image)

Figure 1. The basic variants of the VRP and extensions (adapted from Toth and Vigo, 2002 & 2014)

As for the objectives of the VRP models, Toth and Vigo (2002) and Toth and Vigo (2014) define the following typical objectives (standalone or that can be combined in weighted objective function):
minimization of the global transportation costs, based on the travel distance or time and fixed cost per vehicle used;
minimization of the number of vehicles used;
balancing of the routes, regarding total time or vehicle load;
minimization of the penalties regarding the service of the customers;
can be in the scheduling and time-windows context (waiting time, time of finishing the route, etc.);
energy consumption and pollutant emission control in the routing context;
infeasibility penalties as a mean of guiding metaheuristics towards feasible solutions in a hybrid models.

Laporte (2009) published a review paper on the major developments in the field of exact algorithms and heuristics that were developed during the 50 years of research since the publication of the first paper. The author states that the problems with around 100 nodes can be solved by exact models (based on linear and dynamic programming), while for the real-scale problems of larger dimensions various heuristic models are a necessity (due to the NP-hard nature of the VRP). The classical heuristics for solving the VRP can be classified as (Laporte, 2009):

- The Savings Algorithm;
- Set Partitioning Heuristics;
- Cluster-First, Route-Second Heuristics;
- Improvement Heuristics.

Braekers et al. (2015) in their survey paper regarding the VRP literature published between 2009 and June 2015, state that heuristics and metaheuristics are often more suitable for practical applications. The main difference between heuristics and metaheuristics is that the latter includes mechanisms for avoiding the local optima trap. For the observed period in the survey, authors found 233 models of metaheuristics to solving the VRP, while other methods (exact, heuristic, real-time and simulation) had a total of 106 models. Laporte (2009) classified metaheuristics for solving the VRP in three general groups:

- Local Search – with Tabu Search, Simulated Annealing, Deterministic Annealing, VNS, Very Large Neighborhood Search and Adaptive Large Neighbourhood Search;
- Population Search – with Genetic algorithms as best representative;
- Learning Mechanisms – with Neural Networks, Ant colony optimization, Bee Colony Optimization.

The VNS metaheuristics with its variants are one of the most used to solving the VRP and its variants and extensions, as it is shown in Table 1.

Table 1. The analysis of the research related to the VRP and VNS (for the three relevant publishers, until January 2019 using publishers’ search engines)

| PUBLISHER            | NUMBER OF RESEARCH PAPERS |          |
|----------------------|---------------------------|----------|
| Elsevier             | 1957                      | 388      |
| Springer             | 432                       | 231      |
| Taylor & Francis     | 219                       | 112      |
3. THE VARIABLE NEIGHBORHOOD SEARCH

The VNS belongs to the class of heuristic algorithms known as metaheuristic algorithms. Prefix meta refers to the basic characteristic of that class of heuristic algorithms that their strict implementation is not providing a solution to the problem, but it rather gives a framework in which good solution can be obtained by including all specificities of the problem. Metaheuristic algorithms are widely used in solving logistics related combinatorial optimization problems (such as VRP) of real-world scale sizes, mainly because they provide good quality solutions in a relatively reasonable computation time, as well as they are not too hard to comprehend and implement.

The VNS is presented to the research community in 1997 (Mladenović and Hansen, 1997), although Hansen et al. (2019) stated that previous work that initiated VNS formulation had been reported several years earlier. Its practical applicability and permanent researchers’ interest in VNS has resulted that today there are almost no referent literature regarding metaheuristics without a chapter dedicated to the VNS (Glover and Kochenberger, 2003; Labadie et al., 2016, Siarry, 2016). VNS’s principles are used not only for solving combinatorial and general discrete optimization problems but also for continuous global optimization, for automatic programming etc. For more details in versatile use of VNS we refer an interesting reader to the Hansen et al. (2019).

The VNS belongs to the class of metaheuristic algorithms that search for the best solution by perturbing the best known solution (also known as the incumbent solution, or just the incumbent). Perturbation implies that one sort of solution change (referred from now on as a move) is performed in every possible way on the incumbent. For example if the incumbent solution of a route is a sequence of eight nodes then change that implies selection of successive four nodes and change of their position in the solution results in twenty new solutions. Similarly, if a move implies selection of six successive nodes from the incumbent it will result in six new solutions. These two moves are known as Or-opt 4, and Or-opt 6 moves, respectively. All solutions generated by implementation of some move are known as incumbent's neighbourhood for that move. Figure 2. shows complete neighbourhoods of Or-opt 4 and Or-opt 6 moves for the incumbent solution 1-2-3-4-5-6-7-8.

In the case when all solutions from a neighbourhood are compared to the incumbent it is said that the best improvement search strategy is implemented in the neighbourhood exploration. However, since that strategy may be time consuming due to a very large number of solutions in a neighbourhood, alternative strategy is to stop neighborhood exploration as soon as the first improved solution is obtained. That strategy is known as the first improvement strategy.
Regarding the changes on the incumbent, moves can induce change on only one route of the incumbent (intra-route moves), or on multiple routes (inter-route moves). Theoretically there is unlimited number of moves both for intra- and inter-route moves. However, typical moves that proved efficient in solving VRP related problems are Or-opt k (k=1,2,...), 2-opt, 3-opt for the intra-route moves, and 2-opt*, swap, move, cyclic for the inter-route moves.

The VNS procedure implies exploration of more than one neighborhood with systematic selection of neighborhoods to be checked. According to Hansen et al. (2010) “VNS systematically exploits the idea of neighborhood change, both in descent to local minima and in escape from the valleys which contain them”. In that sense, VNS is based the following three observations: (1) a local minimum with respect to one neighborhood most probably is not a local minimum for another neighborhood; (2) a global minimum is a local minimum with respect to all possible neighborhoods; (3) for many problems local minima with respect to one or several neighborhoods are relatively close to each other. Accordingly, the VNS implies implementation of several neighborhoods, which have to be defined prior to the beginning of the procedure, as well as the rules for change of neighborhood to be explored. Therefore, if time limit for algorithm run time is not required the VNS does not require any other parameter except for defined neighborhood structures and rules for their changes. Hansen and Mladenović (2003) present the general VNS (GVNS) which is in a pseudo code form given as Algorithm 1.

Shaking is a procedure whose role in the VNS is escaping the trap of a local optimum. Therefore in the shaking phase a random solution from a neighborhood is selected. That solution is taken as a center of the local search phase in which detailed search of solution space around the center is conducted.

As it can be seen from Algorithm 1, the GVNS implies implementation of the Variable Neighborhood descent (VND) as a local search phase. However, in the basic case of the VNS (BVNS) any other local search algorithm can be applied. The VND is a deterministic local search algorithm which also explores three aforementioned observations upon which the VNS is based. Accordingly it is also based on systematic changes of the neighborhoods according to best or first improvement strategies. Nevertheless, neighborhoods explored in the VND (Nl) do not have to be the same as neighborhoods used in the shaking phase (Nk). Sequence in which neighborhoods are explored is a subject of the algorithm settings. However, because neighborhoods with lower values of
indices $k$ and $l$ will be explored more frequently they are usually related with the neighborhoods of smaller cardinality.

Algorithm 1. General VNS algorithm.

**Initialization.** Select the set of neighborhood structures $\mathcal{N}_k$ for $k=1, \ldots, k_{\text{max}}$ that will be used in the shaking phase, and the set of neighborhood structures $\mathcal{N}_l$ for $l=1, \ldots, l_{\text{max}}$ that will be used in the local search; find an initial solution $x$; choose a stopping condition;

**Repeat** the following sequence until the stopping condition is met:

1. Set $k=1$;
2. Repeat the following until $k=k_{\text{max}}$:
   1. **Shaking.** Generate a point $x'$ at random from the $k$th neighborhood $\mathcal{N}_k(x)$ of $x$;
   2. **Local search** by VND
      1. Set $l=1$;
      2. Repeat the following until $l=l_{\text{max}}$:
         1. **Exploration of neighborhood.** Find the best neighbor $x''$ of $x'$ in $\mathcal{N}(x)$;
         2. **Neighborhood change.** If $f(x'') < f(x')$ set $x=x''$ and $l=1$; otherwise set $l=l+1$;

It should be emphasized that although only two variants of the VNS (GVNS and BVNS) have been mentioned in the previous text, VNS has much more application variants (e.g. reduced VNS, skewed VNS, Variable Neighborhood Decomposition Search, etc.). However, since their detailed description is not in the scope of this research, we advise interesting reader to check some of the following papers Hansen and Mladenovic (2003), Hansen et al. (2010), Hansen et al. (2019) for further details regarding the VNS metaheuristic and its variants.

4. LOGISTICS CASE STUDIES

Since the VNS is a metaheuristic algorithm it requires that all specificities of considered problems must be respected in its implementation in order to obtain quality solutions for acceptable solution time. Bearing in mind that logistic problems are fruitful ground for implementation of the VNS algorithms in this section we give a deeper insight into the implementation of the VNS in solving three vehicle routing based problems with significant practical importance in realization of logistics process. The considered problems are inventory routing problem, container drayage problem and travelling repairmen problem.

4.1 Inventory routing problem

The inventory routing problem (IRP) is an extension to the vehicle routing problem where inventory management segment must also be taken into consideration, usually in the planning horizon of several days. In the IRP, the vehicle routing and the inventory management are simultaneously optimized: for each day of planning horizon delivery quantities incurs decision variable regarding the inventories segment, while vehicle movement for delivering goods incurs decision variables regarding routing segment. The IRP can be applied in systems where the Vendor Managed Inventory (VMI) concept exists which allows a supplier to make decisions regarding day, structure and quantity of deliveries (opposite to traditional distribution systems where customers make
orders). The VMI concept enables the supplier to better utilize the vehicles, while customers have the benefit of shifting the responsibility of inventory management to the supplier. The seminal research paper which observed the IRP was Bell et al. (1983), where authors considered a distribution of liquefied industrial gases and used the linear programming model to obtain the delivery plan for short-term planning horizon. Since then, numerous papers were published on the topic of the IRP in the replenishment of stores and supermarkets, maritime transportation, textile, automotive and petrochemical industries etc. (see Popović, 2015). In general, the IRP for real-life problems cannot be solved to optimality in reasonable computational time due to the problem complexity (routing problem with allocation of deliveries over an observed planning horizon) and therefore a heuristic approach is required. Popović et al. (2012) observed the IRP in fuel delivery using multicomartment vehicles for petrol stations replenishment. Authors presented the MILP and VNS models to solving the observed problem. In the rest of this section, we describe the fuel delivery problem and the VNS approach from Popović et al. (2012).

The authors observed the secondary distribution of different fuel types from one depot location to a set of petrol stations by a designated fleet of multi-compartment vehicles, and for which a single oil company has control over all of the managerial decisions over all of the resources. The problem has the following key assumptions:

- the delivery quantities for several fuel types and set of petrol stations must be determined for each day within the planning horizon;
- fuel is transported by a fleet of homogeneous multi-compartment vehicles of unlimited size (only full compartments are delivered to petrol stations);
- every petrol station has a daily fuel consumption per type;
- petrol stations are equipped with underground tanks of known capacity;
- stations can be served only once a day;
- it is not allowed that inventory levels in petrol stations for any fuel type fall below their defined daily fuel consumption;
- the total inventory costs are assumed to be dependent on the sum of the average stock levels in each day of the planning horizon, whereas transport costs depend on a vehicle’s travel distance;
- one vehicle can visit up to three stations per route (real-life constraint due to the number of compartments in vehicle and fuel types).

The VNS model developed to solve the IRP in fuel delivery is based on three neighborhoods (changes of the delivery plan): relocation of individual compartments; relocation of all compartments for the observed station’s fuel type; and relocation of all compartments for the observed station.

The construction of the initial solution is based on compartment transfer (CT) heuristic approach. The CT heuristic used the idea of solving the inventory model (the MILP used to solve the IRP but without routing decision variables) and iterative change of the delivery plan by transferring the delivery quantities (represented as full compartments) to be realized one or a few days earlier (compartment transfers to days ahead would incur violation of minimal inventory levels). The Sweep method was used for route construction (starting from the station that has the biggest "polar gap" to its nearest preceding station) which showed good results for vehicles with compartments.
The Randomized Variable Neighborhood Descent (RVND) guided search with the first improvement approach was used for the local search procedure. The first occurrence of improvement in neighborhood search is accepted, and the search is repeated until no more improvements can be found. Instead of using the same order of neighborhoods, randomization of the neighborhoods order after each shaking move was applied (for increased diversification of local search). The RVND local search has two levels: the micro level (intra-period search) where routes improvement is carried out in each day of planning horizon; and the macro level (inter-period search) with aim of minimizing the total inventory and routing costs by the reallocation of deliveries over the planning horizon. For the local intra-period search, three neighborhood structures were used: (1) the interchange of a single station between two routes for the same day; (2) the removal of a single station from one route and its insertion into another route for the same day; and (3) 2-opt*, the removal of two arcs (one from each route) and finding the best possible routes reconnections. For the inter-period search, three neighborhoods were used: (1) the transfer of individual compartments; (2) the transfer of all compartments from a fuel type; (3) the transfer of all compartments from a station.

The shaking procedure is based on the deliveries transfer between days of the planning horizon. Shaking procedure has the task of changing only the time instants of each delivery in the current best solution. Four types of shaking were used: (1) shaking of individual compartments; (2) shaking of all compartments of the same fuel type in the observed station and day; (3) shaking of all compartments in the same station in the observed day; (4) combined shaking of all previous three types.

The feasibility of the solution was respected in all search procedures with regards to: minimum and maximum inventory level, vehicle capacity, single route per vehicle, single delivery in each day for a single station, and maximum three stops per route. The algorithm of the VNS heuristic from Popović et al. (2012) is presented in Figure 3. The stopping criterion becomes active in the case when the last neighborhood in the last shaking procedure does not incur an improvement of the current best solution.

To test the proposed models, three set of instances were developed:

- **P1** - small scale instances with 10 petrol stations and planning horizon of 3 days;
- **P2** - medium scale instances with 15 petrol stations and planning horizon of 4 days;
- **P3** - large scale instances with 20 petrol stations and planning horizon of 5 days.

Additionally, three cases of different vehicle types were observed per each instance: vehicles with 3 compartments of 6t; vehicles with 4 compartments of 6t; and vehicles with 5 compartments of 6t. The MILP model was implemented through the CPLEX 12 on a desktop PC with a 2.0 GHz Dual Core processor with 2 GB of RAM memory. All of the input data needed for the model implementation as well as for the heuristics were implemented in Python 2.6.

The MILP model was able to solve only P1 set of instances, while the VNS heuristics obtained good quality solution for P1 compared to optimal solutions (incurs in average 0.271% higher total costs than MILP model, with the lowest error in the case with the 3x6t vehicle type). As for the P2 and P3 instances, the VNS model was able to solve these instances in acceptable computational time with good convergence of the objective function.
4.2 Container drayage problem

The container drayage problem (CDP) involves pickup and delivery operations of full and empty containers usually between an intermodal terminal and set of locations (e.g. shippers) which are generators of transportation requests. The CDP can be classified as the VRPPDTW, which is the case for the majority of research papers in the available literature on container drayage (see Vidović et al. 2016). The main characteristic of the CDP that distinguishes it from most other VRP problems is the size and structure of the cargo. Mainly, transportation requests can be 20ft and 40ft containers (few countries allow 45ft and larger container sizes).
In Vidović et al. (2016) authors developed a generalized multiple matching model formulated as a mixed integer linear program (MILP) and VNS approach to solving homogeneous simultaneous drayage of multi-size containers. The cases with different container vehicle types were observed: standard vehicles (due to the weight limitation vehicle can carry only one 20ft or 40ft container); modular concept vehicles (with capacity of two 20ft or single 40ft container); longer combination vehicles (with capacity of up to four 20ft or two 40ft container on vehicle with attached trailer). Imai et al. (2007) have shown that the container drayage problem is NP-hard and therefore authors developed the VNS model to solve large-scale instances. In the rest of this section, we describe the problem formulation and the VNS approach used in Vidović et al. (2016).

A network of a single terminal and a set of nodes was observed. Nodes can simultaneously have both container demand (-) and supply (+) requests. Each request can be for 20ft or 40ft container, therefore 4 different types of requests exist in the system: 20+, 20-, 40+, 40-. A task node represents one request from a node (each node can have multiple requests and therefore multiple task nodes). A vehicle route can be defined as multiple matching of task nodes to a vehicle, where a solution represents the order of requests to be served by a vehicle. An example of a feasible sequence of task node types for a longer combination vehicle can be (20-,20+,20-,20+,40-,40+). In any arbitrary sequence of task nodes types, these must be ordered in such a way to enable the realization of pickup tasks (free space on the vehicle to accommodate container pickup). The objective function of observed problem tries to minimize total transportation costs of all vehicle routes, where each node has time windows constraints.

The first application of VNS heuristic to solve the CDP was presented by Popović et al. (2014). Vidović et al. (2016) made an extension by introducing additional local search and shaking procedures, as well as new vehicle type. The initial solution procedure is obtained by the sweep method with "polar gap". The construction of routes starts from the task that has the biggest "polar gap" to its nearest preceding task. The "polar gap" between two nodes refers to the angle formed by two nodes viewed from a depot.

For the local search procedure, the Variable Neighborhood Descend (VND) with four neighborhoods was used: (1) reallocating and (2) interchange of tasks between routes, (3) reallocating of nodes that have two tasks in a single route to another route, and (4) interchange of nodes that have up to two tasks between two routes. The two latter neighborhoods make larger changes (increases the diversification of search) in search procedure that cannot be explored by the first two neighborhoods (they are based on single task changes). The local search procedure examines all possible changes in a neighborhood of a current solution for improvement of its objective function. If a change in observed neighborhood incurs improvement, the local search is restarted within the observed neighborhood.

For the shaking procedure, four different methods (Sh_type) were used in the following order: (1) random reallocation of tasks between some pairs of routes, (2) random interchange of tasks between some pairs of routes, (3) "destruction" of randomly chosen route/routes with multiple stops and construction of single stop (direct) routes to each of those stops, and (4) method that combines all previous three methods. During the shaking procedure, each method starts from a small intensity of random change (the number of changes is limited by parameter Sh_size_max) which is and gradually
increased. Also, multiple repetitions of one shaking intensity are made because of the stochastic nature of shaking (to intensify the search, each shaking size is repeated for $Sh_{pass\_max}$). The total number of shaking neighborhoods is defined by the $Sh_{type}$, $Sh_{size\_max}$, and $Sh_{pass\_max}$ parameters. The stopping criterion of the VNS heuristic becomes active when the shaking and local search procedures for the last neighborhood do not incur improvement in current best solution value. The algorithm of the VNS heuristic is presented in Figure 4.

The multiple matching MILP formulation and VNS heuristic were tested on the set of instances generated according to the Solomon VRPTW benchmark problems (http://web.cba.neu.edu/~msolomon/problems.htm). Test instances used to evaluate proposed models can be divided in the following way:

---

Figure 4. The algorithm of the VNS heuristic to solving CDP presented in Vidović et al. (2016)
small scale instances that include two groups: first 10 and first 15 nodes from Solomon instances;

- medium scale instances that include two groups of first 50 nodes from Solomon instances where the first group has fewer tasks than the second;
- large scale instances that include all 100 nodes from Solomon instances.

Mathematical models were implemented by the CPLEX 12.4. on the Intel(R) Core(TM) i3 CPU M380 2.53 GHz with 6 GB RAM, while the VNS model was implemented in C++ and Microsoft Visual Studio 2012.

The research has shown that the proposed MILP model can optimally solve small problem instances and medium scale instances with fewer tasks and only in the case when modular concept vehicles are used. However, the medium scale problem instances with a small number of tasks and longer combination vehicles, as well as the medium scale problem instances with a large number of tasks and large-scale instances, could not be solved to optimality in reasonable computational time.

The VNS heuristic model was able to obtain very accurate results since it solves almost all small-scale problem instances and a certain number of medium scale problems to optimality. In the cases where the MILP model could not find optimal solutions, VNS heuristics obtained results in reasonable computation time. This research has verified that VNS heuristics performances are very good and give an opportunity for solving CDP of realistic size.

The proposed approach is tested in cases when standard, modular concept and longer combination vehicles are used. One important result from the practical point of view is that modular concept vehicles and longer combination vehicles outperform standard vehicles in its drayage characteristics. Of course, modular concept vehicles and longer combination vehicles are more expensive and are not able to reach every network node because they require wider roads with a larger radius, but results show their huge potential for application.

### 4.3 Heterogeneous travelling repairmen problem with time windows

Travelling repairmen problem (TRP) is a special case of the Traveling Salesman Problem (TSP). Salesman in the TSP visits all customers in a planning horizon with the goal to minimize traveled distance. Repairman is less selfish because he does not put his own interest as the focus of the work but rather focuses on better customer satisfaction. Therefore, objective in the TRP is minimization of the sum of times all clients wait until the end of a service. TRP’s objective function has numerous possibilities of application, especially in client oriented processes. Some examples of such processes are: regular and emergency equipment repairs, deliveries and pick-ups of shipments to and from clients, scheduling of special material handling equipment (heavy load hoists, specialized pumps etc.).

A great majority of researches that considered the TRP are based on an assumption that all tasks are already available for a service at the beginning of the planning horizon, meaning the service begins as soon as the repairman reaches task location. However, real situations imply that this does not have to be the case, and if the repairman reaches task’s location before the task is available for a service he will have to wait for the beginning of the service. In Bjelić et al. (2013) and Bjelić (2014) authors considered a
variant of the TRP which respects the possibility that not all tasks are present in the system at the beginning of a planning horizon. Moreover the TRP they considered is further generalized by assuming that service is realized by a fleet of heterogeneous repairmen whose characteristics differ in terms of moving speed between locations and times needed for service realization. Therefore, the analyzed problem is classified as the heterogeneous traveling repairmen problem with time windows (hetTRPTW). Note that the tasks’ time windows in the hetTRPTW are opened on the right hand side. Because of the space limitation, for deeper literature review of the hetTRPTW we refer an interested reader to Bjelić et al. (2013).

As a special case of the TSP, the TRP belongs to the class of NP-hard problems. The cumulative nature of the TRP’s objective function makes this problem harder to solve than its TSP counterpart. Therefore for solving problem instances of practical sizes in Bjelić et al. (2013) authors present a GVNS algorithm presented on Figure 5.

Initial solution for the VNS algorithm is a solution from the widely used insertion algorithm. Precisely authors used the insertion algorithm given in van der Meer (2000) for generation of the initial solution.

As the local search algorithms authors propose the VND. Three moves are used for generating neighborhoods. The first neighborhood structure $\mathcal{N}_1$ is generated by
execution of Or-opt moves on each repairman’s sequence of tasks with the aim of improving the solution by reordering the current service sequence. Reordering implies the removal of 1, 2 and 3 adjacent tasks from the existing sequence of a repairman, and their insertion into all of the possible positions in the remainder of the sequence, excluding their original position. The second neighborhood structure \( N'_2 \) is generated by execution of swap move with one task. Similar to the case of the Or-opt move, swap move keeps the same structure of the solution, regarding a number of tasks allocated to devices. However, the swap move attempts to find an improved solution by exchanging a task served by different repairmen. Relocation move forms the third neighborhood structure \( N'_3 \) by changing the existing structure of the incumbent. Namely, a relocation move attempts to improve a solution by removing one task from a repairman, and inserting it at all possible locations on other repairmen. Order of implemented neighborhoods exploration is selected according to the cardinalities of neighborhoods, meaning that firstly is checked neighborhood generated by the Or-opt 3 move, afterwards Or-opt 2 neighborhood, followed by Or-opt 1, swap and relocate neighborhoods.

For the shaking procedure, two sets of nested neighborhoods are used. ExtSwap move chooses the next region for exploration by a random selection of two tasks for swapping. This procedure is an extension of the swap move because selected tasks can be of the same repairman meaning that it covers domain of the Or-opt move. Nested neighborhood structures \( N_k (k = 1, 2, ..., k_{\text{max}}) \) are formed by successive implementations of the ExtSwap moves on the incumbent. The second set of nested neighborhood structures, \( N_k (k = k_{\text{max}}^1 + 1, k_{\text{max}}^1 + 2, ..., k_{\text{max}}^1 + k_{\text{max}}^2) \), is formed by the successive execution of the Relocation moves used for the generation of \( N'_3 \).

Because of the rapid growth of the solution space, running time of the VNS algorithm prolongs significantly with the increase of a number of tasks, or a number of repairmen in the problem. Therefore, authors tried to reduce neighborhood cardinalities by implementing a reduction strategy which considers only solutions that fulfill predefined conditions. These conditions are defined for each neighborhood structure in the VND and the VNS algorithm. Exploration of reduced neighborhoods in the local search resulted in new version of the VNS algorithm, referred as the RdVNS.

\( N'_1 \) structure is reduced by shortening sequence of tasks on which Or-opt move is performed. Starting from the leftmost task in a task sequence, for each pair of adjacent tasks it is checked if the reverse order of their service provides faster end of service for the firstly served task. If this is not the case it is supposed that there are no tasks in considered sequence that will provide faster end of service for the firstly realized task. Therefore that task is excluded from the sequence on which Or-opt move will be applied. The same procedure is then repeated on a shortened sequence of tasks. However, if the swap of adjacent tasks results in faster realization of the firstly realized tasks, we implement Or-opt move on the whole sequence in order to find a more appropriate position for the leftmost task of the original sequence.

Reduction of the \( N'_2 \) structure is realized by limiting the set of tasks that can be included in the swap procedure of a considered task. Namely, because of the problem’s nature there is a low chance that an improved solution will be found if swap move is executed on the tasks from the beginning and the end of the considered planning period, i.e. on the tasks with considerable time gap between their arrival times. Therefore, a
Coefficient $\alpha$ is introduced. $\alpha$ denotes the ratio of the time period between arrivals of tasks included in the swap move and the time width of the planning horizon. Swap move is allowed only if $\alpha$ is not greater then $\alpha_{crit}$. Level of neighborhood reduction, as well as quality of obtained solutions is influenced by the value of $\alpha_{crit}$. According to the small experimental study the best tradeoff between the reduction level and solution quality is achieved for $\alpha_{crit}=0.3$.

In the case of the $N_3$ structure reduction implies that the inserted task can be put before a task scheduled to another repairman only if arrival of the inserted task happens sooner than the beginning of a service of the previously scheduled task. In this way consideration of solutions in which the repairman waits for the arrival of an inserted task instead of serving an already allotted task is avoided.

Efficiency of presented VNS based algorithms is tested on four benchmark test sets originally used for the TSPTW problem. All considered instances are adjusted to the hetTRPTW by removing right hand sides of the time windows, and by introducing variability of repairmen in the system. In the case of small scale instances (up to 20 tasks) optimal solutions (if possible) are obtained by implementation of branch and bound algorithm found in CPLEX 12.2 with the memory limit of 1GB for the tree structure. In the cases of B&B algorithm, instances are solved once, while in the case of the VNS and RdVNS algorithms the solution procedure is repeated five times.

From 153 instances solved by B&B algorithm, VNS was outperformed, in terms of objective value, only in six with the largest deviation of 1.16%. In all other experiments VNS gave solutions of equal or better quality. The number of solutions with lower value of VNS's objective function increases with the complexity of problem caused by growth in number of tasks and/or number of repairmen in the system.

Relative distances of all RdVNS solutions to the VNS results show that a deviation of RdVNS objective values decrease with the increase of problem complexity. The reason for this pattern is that with larger complexity of the problem the number of good solutions with a relatively small difference in objective values increases as well. This gives better chance to the RdVNS algorithm to find a solution close, in terms of a objective function, to the VNS's solution. Presented results also show that the implementation of reduction strategies resulted in the reduction of running times for more than 50% on average. The largest average time reduction is achieved for the case of two repairmen, and the lowest for the case of ten repairmen in the system.

5. CONCLUSION

VNS is one of the leading metaheuristic approaches in solving numerous combinatorial optimization problems. Probably the most important reason for its wide applicability, beside its simplicity, is its flexibility in implementing different neighborhood exploration strategies, which resulted in several variants of the algorithm. In presented researches the GVNS concept has been used, with implementation of first and best improvement concepts, nested neighborhoods, reduction of neighborhoods, numerous different moves used in local search and shaking phases etc. Quality of solutions obtained in the considered studies, as well as solution times, proved a practical implementation of given VNS algorithms in real-life systems.
Finally it should be emphasized that presented problems are just an example of numerous logistics problems in which the VNS can be successfully applied. Of course, specificity of every problem must be included in algorithms constructions.

ACKNOWLEDGMENT

This work was supported by the Ministry of Education, Science and Technical development of the Government of the Republic of Serbia through the project TR36006, for the period 2011-2019.

REFERENCES

[1] Bell, W.J., Dalberto, L.M., Fisher, M.L., Greenfield, A.J., Jaikumar, R., Kedia, P., et al. (1983). Improving the Distribution of Industrial Gases with an On-line Computerized Routing and Scheduling Optimizer. Interfaces. 13(6), 4-23.

[2] Bjelić, N., (2014). Models for operational planning of a materials handling system with dynamic task arrivals, Doctoral Dissertation, University of Belgrade, Serbia. (In Serbian)

[3] Bjelić, N., Vidović, M., Popović, D., (2013). Variable neighborhood search algorithm for heterogeneous traveling repairmen problem with time windows. Expert Systems with Applications, Volume 40 (15), 5997-6006.

[4] Braekers K., Ramaekers, K., Nieuwenhuyse, I.V., (2015). The Vehicle Routing Problem: State of the Art Classification and Review. Computers & Industrial Engineering (99), 300-313.

[5] Dantzig, G.B., Ramser, J.H, (1959). The Truck Dispatching Problem. Management Science, 6(1), 80-91.

[6] Glover, F., Kochenberger, G. (2003), Handbook of metaheuristics, Kluwer Academic Publishers, Boston, USA.

[7] Hansen, P., Mladenović, N., (2003). A tutorial on variable neighborhood search. Technical Report G-2003-46, Les Cahiers du GERAD, HEC Montreal and GERAD, Canada.

[8] Hansen, P., Mladenović, N., Brimberg, J., Moreno Pérez, J., (2019) Variable Neighborhood Search. In: Gendreau., M., Potvin., J.-Y., Handbook of Metaheuristics 3rd Edition, Springer International Publishing AG, pp. 57-97. https://doi.org/10.1007/978-3-319-91086-4_3.

[9] Hansen, P., Mladenović, N., Moreno Pérez, J., (2010) Variable Neighborhood Search: methods and applications, Annals of Operations Research 175, 367-407. https://doi.org/10.1007/s10479-009-0657-6

[10] Imai, A., Nishimura, E., Current, J, (2007). A Lagrangian relaxation-based heuristic for the vehicle routing with full container load. European Journal of Operational Research 176, 87-105.

[11] Labadie, N., Prins, C., Prodhon, C. (2016) Metaheuristics for vehicle routing problems. Computer engineering series - Metaheuristics set vol. 3, Wiley-ISTE, London, UK.

[12] Laporte, G., (2009). Fifty Years of Vehicle Routing. Transportation Science, 43(4), 408-416.

[13] Mladenović, N., Hansen, P. (1997) Variable neighborhood search. Computers and Operations Research 24, 1097-1100.
[14] Popović, D., Vidović, M., Nikolić, M., (2014). The Variable Neighborhood Search Heuristic for the Containers Drayage Problem with Time Windows. Soft Computing in Industrial Applications, Volume 223 of the series Advances in Intelligent Systems and Computing, 351-364. Online ISBN: 978-3-319-00930-8.

[15] Popović, D., (2015). Inventory Routing Problem: modeling and performance analysis. Doctoral Dissertation, University of Belgrade, Serbia. (In Serbian)

[16] Popović, D., Vidović M., Radivojević G., (2012). Variable Neighborhood Search heuristic for the Inventory Routing Problem in fuel delivery. Expert Systems with Applications, Volume 39 (18), 13390–13398.

[17] Siarry, P. (2016) Metaheuristics, Springer International Publishing Switzerland.

[18] Toth, P., Vigo, D., (2002). The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, ISBN: 0898715792, 9780898715798.

[19] Toth, P., Vigo, D., (2014). Vehicle Routing: Problems, Methods, and Applications, Second Edition. Society for Industrial and Applied Mathematics, ISBN: 1611973597, 9781611973594.

[20] Van der Meer, R. (2000) Operational Control of Internal Transport. PhD Thesis, University of Rotterdam. Retrieved from http://repub.eur.nl/res/pub/859/

[21] Vidović, M., Popović, D., Ratković, B., Radivojević, G., (2016). Generalized mixed integer and VNS heuristic approach to solving the multi-size containers drayage problem. International Transactions in Operational Research, 24(3), 1–32. DOI: 10.1111/itor.12264