

# An Efficient Metaheuristic Approach for the Multi-Period Technician Routing and Scheduling Problem

Abouliakdane Khattara

Electrical Engineering Department,  
Ferhat Abbas Setif 1 University,  
Setif, Algeria  
yekdane87@univ-setif.dz

Wahiba Ramdane Cherif-Khettaf

LORIA, UMR 7503,  
University of Lorraine,  
Nancy, France  
ramdanec@loria.fr

Mohammed Mostefai

Electrical Engineering Department,  
Ferhat Abbas Setif 1 University,  
Setif, Algeria  
mostefai@univ-setif.dz

**Abstract**—In this paper, we address a new variant of the Multi-Period Technician Routing and Scheduling Problem. This problem is motivated by a real-life industrial application in a telecommunication company. It is defined by a set of technicians having distinct skills that could perform a set of geographically scattered tasks over a multi-period horizon. Each task is subject to time constraints and must be done at most once over the horizon by one compatible technician. The objective is to minimize the total working time (composed by routing time, service time, and waiting time), the total cost engendered by the rejected tasks, and the total delay. Two variants of variable neighborhood descent are proposed, and three variants of variable neighborhood search to solve this problem. Computational experiments are conducted on benchmark instances from the literature. An analysis of the performance of the proposed local search procedures is given. The results show that our methods outperform the results of a mimetic method published in the literature.

**Keywords**—technician routing and scheduling problem (TRSP); variable neighborhood search (VNS); variable neighborhood descent (VND)

## I. INTRODUCTION

Technician routing and scheduling problem (TRSP) is a new challenge in logistics for the service sector and especially for utility companies in energy (gas, electricity), telecommunications, and water distribution areas [1]. The TRSP consists of planning tasks assigned to commercial or technical personnel, over a set of periods (days) in order to visit industrial facilities or customers for different types of activities: installation, inspection, repair, and maintenance. Until recently, the TRSPs, both static and dynamic cases, have received a limited attention. Thus, the number of publications and scientific reports is limited, although several variants of the TRSP have been studied in the literature. These variants can be divided into two classes: (i) one period TRSP, and (ii) multi-period TRSP. The one period TRSP has been studied by authors who consider constraints as skills, time windows, tools, spare parts, stochastic service and stochastic travel times, multiple depots, and customer priority [1-6]. For the multi-period TRSP, we can mention [7], which introduces the multi-period technician scheduling problem with experience based on service times and stochastic customers. The aim is to minimize

the expected sum of each day's total service times over a finite horizon. Another multi-period TRSP was proposed in 2007 [8]. This problem consists in computing a schedule for technicians to perform a set of tasks on a five day horizon. The routing aspect is not considered, and tasks have different proficiency skill level constraints, that require a team of technicians. Authors in [10] studied the one-periodic variant of this problem, namely, the service technician routing and scheduling problem by taking on consideration the routing aspect. Authors in [9] presented a multi-period technician routing problem faced by a water distribution and treatment company. In [9], requests were divided into two categories (users requested interventions and company scheduled visits), and the skill constraints were not included.

In this paper, we propose the study of a new multi-period TRSP variant where skill constraints and routing aspects are considered simultaneously, inspired by a realistic application in the telecommunication field. From the above survey, it appears that most papers on TRSP considered several realistic constraints, but to the best of our knowledge, the multi-periodic variant of TRSP with skill constraints and routing aspects has not been considered in the literature. The papers that consider a multi-periodic TRSP with skill constraints, and routing aspects, included other specific constraints as the technician team constraint [10]. Our study is also an extension of the problem studied in [9, 11, 12] in which the skill constraints are ignored. As the considered problem is NP-hard and since it results from the combination of complex constraints, large instances can hardly be solved by exact methods. So, the best way to tackle this problem is by using the metaheuristic approaches. We choose a variable neighborhood search (VNS) to solve our problem, because its effectiveness has been proven on a number of variants of vehicle routing problems (VRP) as the vehicle routing problem with time windows [13], the vehicle routing problem with multiple depots and time windows [14, 15], the periodic vehicle routing problem [16], and the workforce scheduling and routing problem [17].

In this paper, we propose two variants of variable neighborhood descent, as well as three variants of variable neighborhood search to solve the TRSP with skill constraints and routing aspects.

Corresponding author: Abouliakdane Khattara

## II. PROBLEM DESCRIPTION

We consider a multi-period horizon  $H$  of several days (typically one week). For each day  $h \in H$ , a set of technicians  $K$  with different skills are available (a technician has one skill or more). Each technician  $k \in K$  has a known starting and ending location  $d \in D$ , which corresponds to the technician's home or office (the starting location is the same with the ending location). Each technician has a working time limit per day  $Maxtime_{k,h}$ . Requests belong to two categories: non-urgent tasks ( $NT$ ) generated by the company, and urgent requests ( $UT$ ) formulated by customers through a call center, for emergency reasons. Note that  $UT \cup NT = T$ , with  $T$  the set of all tasks known in advance. Let  $si$  be the service time of the task  $i$ . The urgent tasks  $i \in UT$  are planned on a fixed day  $hi$  and are subjected to customer appointments within a given time window ( $bi, ei$ ), where  $bi$  is the beginning of the time window, and  $ei$  the end of the time window. The task  $i$  could be affected to a technician  $k$  if the arriving time at task  $i$  (denoted  $aik$ ) is before the end of the time windows ( $aik < ei$ ). If  $\forall k \in K, aik \geq ei$ , then the task  $i$  is considered as a rejected task. If  $aik \leq bi$ , a waiting time  $Wik$  occurs, with  $Wik = bi - aik$ . If  $aik \leq ei$ , and  $aik + si > ei$ , a delay  $Lik$  occurs, with  $Lik = (aik + si) - ei$ . Non-urgent tasks are characterized by a validity period composed of one or several days  $[hbi, he_i] \in H$ , where  $hbi$  is the early day and  $he_i$  is the deadline for the execution of  $i$ . A request  $i$  requires certain skills (qualifications), and must be executed by only one compatible technician. The goal is to build a set of routes per day and per technician (at most  $|Kh|$  routes per day). Each route  $Rhk$  is a sequence of tasks assigned to only one technician  $k$  and one-day  $h$ . The following constraints must be satisfied: 1) each task must be executed at most once within the validity period or within the time window, 2) the total time of each route  $Rhk$  should not exceed  $Maxtime_{k,h}$ , 3) the competence requirements must be respected, 4) each route must start and end at the same location  $d \in D$ . The objective function, measured in monetary units denoted  $f(x)$ , consists in minimizing three costs: (i) the total working time composed by routing time that depends on the number of kilometers travelled by each technician, the service time and the waiting time, (ii) the total cost engendered by the ejected tasks, and (iii) the total delay.

## III. SOLUTION METHODOLOGY

In this section, we describe the general framework of the variable neighborhood search (VNS), and then we present the basic components of the VNS that we have developed to solve our problem.

### A. Variable Neighborhood Search

VNS is a metaheuristic, or a framework created 1997 [18, 19] for approximately solving optimization problems, including combinatorial and non-linear continuous optimization problems [20, 21]. VNS is based on the systematic changes of neighborhood structures during the search for a (near) optimal solution of a considered problem. These changes occur in both descent phase, to improve the solution, and shaking and perturbation phase that aims to escape local optima traps. The main structure of the VNS (Algorithm 1) is shown in Figure 1.

```

Function VNS_heuristic ( $x, k_{max}, t_{max}$ )
1. repeat
2.   repeat
3.      $x' \leftarrow \text{Shake}(x, k)$ ;
4.      $x'' \leftarrow \text{Improvement\_procedure}(x')$ ;
5.     Neighborhood_Change( $x, x'', k$ );
6.   until  $k = k_{max}$ ;
7.    $t \leftarrow \text{CpuTime}()$ ;
8. until  $t > t_{max}$ ;

```

Fig. 1. Algorithm 1: Variable neighborhood search VNS

The inputs of VNS heuristic are  $x$ ,  $k_{max}$  and  $t_{max}$ , and they present the initial solution, the number of neighborhoods to be explored and the maximum allowed CPU time. The main ingredients of variable neighborhood search include an improvement procedure to improve the current solution and a shaking procedure to perturb the search and escape from the corresponding valley, see lines 3 and 4. The improvement procedure in line 5 could be a single local search or an ordered list of set of neighborhoods.

### B. Initial Solution

We propose to use as an initial solution the best insertion method with sorting list. This method is performed by two steps. In the first step, a list of unserved tasks ( $L$ ,  $L=T$ ), is sorted in increasing order according to validity day ( $VD$ ), that represents the length of the period (number of days) in which the tasks can be done,  $VD_i = he_i - hb_i$ . In the second step, the algorithm select a task  $i$  from the head of  $L$ , and scan all feasible insertions in all routes  $R_{hk}$ . The insertion cost of  $i$  in a route  $R_{hk}$  between two tasks  $x$  and  $y$ , named  $\delta(i, R_{hk}, x, y)$  will be calculated as in (1). The algorithm performs the best insertion.

$$\delta(i, R_{hk}, x, y) = C_{xi} + C_{iy} - C_{xy} + \sum(j \in R_{hk} \cup i) W_{jk} + \sum(j \in R_{hk} \cup i) L_{jk} \quad (1)$$

### C. Local Search Procedures

We propose five local search operators to be used either individually or together to focus the search in the inner loop of VNS. We consider three intra-route and two inter-route local search methods. The best improvement strategy is used for each method. The local search methods are:

- One intra-route relocate: one node (task) from the route is removed and reinserted in other positions in the same route.
- One intra-route exchange: two nodes (tasks) are exchanged in the same route.
- 2 opt: two arcs are removed and reinserted in the same route
- One inter-route relocate: one node (task) from the route is removed and reinserted in one other route in the solution.
- One inter-route exchange: two nodes (tasks) are exchanged between two different routes.

### D. Variable Neighborhood Descent Procedures

The variable neighborhood descent (VND) procedures explore several neighborhood structures either in a sequential or nested (or composite) fashion to possibly improve a given solution [21] because the solution which is a local optimum with respect to several neighborhood structures is more likely

to be a global optimum than the solution generated as a local optimum for just one neighborhood structure. The order of neighborhoods may play an important role in the quality of the final solution [22]. Two variants of VND are discussed in this paper regarding the decision made in neighborhood change procedure. If an improvement has been detected in some neighborhood: (1) Basic VND (B-VND): we return to the first neighborhood on the list, (2) Union VND (U-VND): at each iteration all the neighborhoods in the list are used to explore the search, and the next incumbent solution is the best one found by the best neighborhood. The outline of basic VND is presented in Algorithm 2 (Figure 2). The steps of the sequential neighborhood change which is presented in line 5 (Algorithm 1) and line 7 (Algorithm 2) are given in Algorithm 3 (Figure 3). If an improvement of the incumbent solution in some neighborhood structure occurs, the search is resumed in the first neighborhood structure (according to the defined order) of the new incumbent solution, otherwise the search is continued in the next neighborhood (according to the defined order).

```

Function B-VND ( $x, k_{max}$ )
1. repeat
2.    $stop \leftarrow false;$ 
3.    $k \leftarrow 1;$ 
4.    $x' \leftarrow x;$ 
5.   repeat
6.      $x'' \leftarrow \text{argmin}_{y \in N_k(x)} f(y);$ 
7.     NeighborChange( $x, x'', k$ );
8.     until  $k = k_{max};$ 
9.     If  $f(x') \leq f(x)$  then
10.       $stop \leftarrow true;$ 
11.     end
12. until  $stop = true;$ 
13. return  $x'$ 

```

Fig. 2. Algorithm 2: Variable neighborhood descent VND

```

Function NeighborChange ( $x, x', k$ )
1. If  $f(x') < f(x)$  then
2.    $x \leftarrow x';$ 
3.    $k \leftarrow 1;$ 
4. else
5.    $k \leftarrow k+1;$ 
6. end

```

Fig. 3. Algorithm 3: Neighborhood change procedure

### E. Shaking Procedure

The shaking procedure is used in VNS as mentioned in line 3 of Algorithm 1 in order to hopefully resolve local minima traps. Our shaking procedure consists in selecting a random solution from the  $k^{\text{th}}$  neighborhood structure,  $N_k(x)$ .

## IV. COMPUTATIONAL RESULTS

All developed algorithms were implemented in Matlab and all tests were carried out on a MacBook Pro Intel Core™ i7-3520M with 2.90GHz CPUs, sharing a memory of 8GB (the algorithms use only one CPU). Our problem is an extension of the problem studied in [9, 11, 12], in which the skill constraints are ignored. For this, we use the data instance used in [9, 11, 12], and evaluate the performance of our methods with their mimetic algorithm. We first compare the performance of different local search procedures with the initial solution. Different VND procedures are then tested and compared.

Finally, we compare and evaluate the VNS procedures proposed in this paper.

### A. Description of Experimental Data Sets

In order to evaluate and assess the performance of the proposed approaches, we compare them with the methods proposed by Tricoire in [9, 11, 12]. The instances of [9, 11, 12] are used for tests. For this, the skill constraints in our problem are relaxed, and lunch break constraints are added. They are inspired by a real life case, and they are available with detailed experimental results in [9] as well as on the web site: <http://www.emn.fr/z-auto/routing-pbs/>. All instances have a five-day planning horizon and three technicians available every day. The demands are randomly distributed over a 40km<sup>2</sup> map, and Euclidean distances are used. The technicians drive at a constant average speed of 35km/h. Two sizes of instances are tested C1 with 100 customers and C2 with 180 customers, each one with 5 variants according to the distribution and the percentage of time windows and the percentage of urgent and non-urgent tasks.

### B. Evaluation Of The Performance Of The Local Search Procedures

We study the impact of the locale search procedures. The results are shown in Table I. The first column indicates the name of each instance. Column 2 presents the result found by the initial solution, which is based on the best insertion strategy. The remained columns provide both the Gap and the computing time for each local search operator. The Gap is calculated by (2). Row 13 mentions the average results of all instances. The ranks of the local search operators according to their performances and computing time are provided in the last two rows. The best results are in bold.

$$\text{Gap}\% = (f(x)_{\text{heuristic}} - f(x)_{\text{heuristic+LS}}) / f(x)_{\text{heuristic+LS}} \quad (2)$$

From Table I, we note that the 2 opt operator is the best one in almost all instances but it is the third one in terms of computing time. It is also worth noting that all the operators perform well and they all improve on average at least 7.14% and at most 9.66% the results found by the heuristic.

### C. Variable Neighborhood Descent Procedures

The aim of this section is to evaluate and to compare different variants of VND procedures according to the manner in which the neighborhood is changed after each improvement occurs. Namely, it is obvious that the order of neighborhoods on the list affect the performance of VND procedures [22]. Thus, we take into account two possible orders of LS procedures according to their performance: 1) the value of  $f(x)$ , and 2) the computing time. The used orders are mentioned in Table II. The results of VND procedures on Tricoire instances are summarized in Table III. The settings of the VND variant are provided in column and row headings as described above. For example, in Table III, the values in the two cells, at the intersection of the row C100\_1 and 4<sup>th</sup> column, correspond to value achieved by B-VND that explores neighborhoods using the 1st order, as well as its execution time in second. The next column reports the percentage deviation of the obtained solution compared to best solution of the mimetic algorithm

proposed by Tricoire [9, 11,12]. The deviations are calculated by:

$$Dev\% = (f(x)_{VND} - f(x)_{mimetic}) / f(x)_{mimetic} \quad (3)$$

The next column reports the percentage deviation of the obtained solution compared to the result found by the initial

solution, which is calculated by (2). In Table III, we report the results obtained by B-VND and U-VND using the two proposed orders. The average results are mentioned in the two last rows. In Table III, values in bold followed by a star represent new best solutions obtained by our method.

TABLE I. COMPARISON BETWEEN LOCAL SEARCH PROCEDURES

instances	$f(x)_{Heuristic}$	2 Opt		One intra route relocate		One intra route exchange		One inter route exchange		One inter route relocate	
		Gap	Time (s)	Gap	Time (s)	Gap	Time (s)	Gap	Time (s)	Gap	Time (s)
C1 1	21024.64	14.04%	0.40	12.74%	0.47	12.79%	0.47	0.00%	0.29	3.74%	5.79
C1 2	19347.33	6.15%	0.40	4.63%	0.45	4.65%	0.41	3.11%	5.49	5.30%	4.31
C1 3	19658.79	8.68%	0.39	6.55%	0.43	5.74%	0.36	0.72%	1.53	4.29%	3.45
C1 4	22232.21	9.86%	0.40	7.29%	0.29	7.32%	0.35	7.13%	5.26	6.10%	2.96
C1 5	18219.65	-2.63%	0.19	-2.75%	0.28	-1.84%	0.24	-2.46%	2.58	8.58%	7.05
C2 1	39085.88	10.03%	1.76	7.04%	1.45	7.07%	1.96	15.04%	44.40	8.27%	30.95
C2 2	34873.74	12.22%	1.76	6.69%	1.62	5.89%	1.34	4.29%	22.04	6.47%	20.96
C2 3	36349.52	13.70%	1.70	13.22%	1.75	11.36%	1.52	13.09%	39.18	15.12%	24.57
C2 4	36679.56	8.59%	1.46	7.32%	1.95	6.00%	1.64	7.72%	29.95	7.97%	25.62
C2 5	33700.93	11.00%	1.74	8.42%	1.27	9.58%	1.46	12.25%	28.76	6.23%	19.96
Avg	28117.23	9.66%	1.02	7.47%	0.99	7.14%	0.98	7.17%	17.95	7.58%	14.56
Rank ( $f(x)$ )		1		3		5		4		2	
Rank (Time)		3		2		1		5		4	

TABLE II. ORDERS OF LOCAL SEARCH PROCEDURES

5 Local search operators		1st	2nd
Order of local search	2 Opt	1	3
	One intra route relocate	3	2
	One Intra route exchange	5	1
	One Inter route exchange	4	5
	One inter route relocate	2	4

From the results presented in Table III, we may conclude the following: The VND variants that explore neighborhoods in 1<sup>st</sup> order offer the best results in both objective function and

CPU time compared to the other order type. If we consider the average results over all test instances, it appears that the best average results are obtained by U-VND even when we change the order of neighborhoods. So we can say that the U-VND is more effective than B-VND in terms of the objective function and CPU Time. From the average results over all test instances we show that all our VND procedures implemented and discussed in this paper are competitive and perform better than the mimetic algorithm when solving the same problem. For 6 instances among 10, a new best solution is found by our method.

TABLE III. EVALUATION OF DIFFERENT VARIANTS OF VND

Instances	Orders	B-VND						U-VND						
		Mimetic of Tricoire	1st		2nd		1st		2nd					
		Value	% Dev mimetic	% Dev heuristic	Value	% Dev mimetic	% Dev heuristic	Value	% Dev mimetic	% Dev heuristic	Value	% Dev mimetic	% Dev heuristic	
C100_1	$f(x)$	17893.91	<b>17578.37*</b>	-1.76%	19.61%	17594.18	-1.68%	19.50%	<b>17578.37*</b>	-1.76%	19.61%	17594.03	-1.68%	19.50%
	Time (s)	9.92			11.42			8.97			13.28			
C100_2	$f(x)$	15977.12	17202.92	7.67%	12.47%	17136.03	7.25%	12.90%	17153.67	7.36%	12.79%	17164.61	7.43%	12.72%
	Time (s)	9.52			9.58			8.06			8.73			
C100_3	$f(x)$	16714.03	17493.5	4.66%	12.38%	17529.38	4.88%	12.15%	17491.94	4.65%	12.39%	17538.11	4.93%	12.09%
	Time (s)	7.32			5.93			5.03			6.74			
C100_4	$f(x)$	17489.36	18285.77	4.55%	21.58%	18265.73	4.44%	21.72%	18229.85	4.23%	21.95%	18031.33	3.10%	23.30%
	Time (s)	11.37			12.68			9.5			14.03			
C100_5	$f(x)$	16025.91	16535.47	3.18%	10.19%	16611.1	3.65%	9.68%	16535.47	3.18%	10.19%	16364.41	2.11%	11.34%
	Time (s)	9.83			9.78			9.91			15.47			
C180_1	$f(x)$	28945.36	28607.43	-1.17%	36.63%	29299.56	1.22%	33.40%	<b>28405.93*</b>	-1.86%	37.60%	28579.51	-1.26%	36.76%
	Time (s)	113.28			107.17			84.19			96.87			
C180_2	$f(x)$	31191.12	28156.24	-9.73%	23.86%	27780.88	-10.93%	25.53%	27748.3	-11.04%	25.68%	<b>27729.19*</b>	-11.10%	25.77%
	Time (s)	66.39			72.82			58.22			46.09			
C180_3	$f(x)$	27728.44	26464.43	-4.56%	37.35%	27472.29	-0.92%	32.31%	<b>26034.96*</b>	-6.11%	39.62%	26886.39	-3.04%	35.20%
	Time (s)	85.99			89.47			78.46			77.85			
C180_4	$f(x)$	30245.61	29348.92	-2.96%	24.98%	29522.29	-2.39%	24.24%	30124.57	-0.40%	21.76%	<b>29238.94*</b>	-3.33%	25.45%
	Time (s)	57.23			76.22			52.22			61.71			
C180_5	$f(x)$	28158.57	26880.26	-4.54%	25.37%	26566.25	-5.65%	26.86%	<b>26395.74*</b>	-6.26%	27.68%	26625	-5.45%	26.58%
	Time (s)	78.74			93.93			70.71			60.92			
Average	$f(x)$	23036.94	22655.33	-1.66%	24.11%	22777.77	-1.13%	23.44%	<b>22569.88</b>	<b>-2.03%</b>	24.58%	22575.15	-2.00%	24.55%
	Time (s)	44.96			48.9			38.53			40.17			

It is worth noting that all VND procedures also perform better and they all improve on average at least 23.44% and at most 24.58% the results found by the heuristic. That means that VND procedures improve the solution in average 15% more than the single local search operators (Table I).

#### D. Variable Neighborhood Search Procedures

In this section we evaluate and compare three variants of VNS procedures regarding to the improvement procedures in the inner loop: (1) B-VNS that uses a simple local search in the

inner loop at each iteration and move to the other one as in Algorithms 1 and 3, (2) VNS\_B-VND that uses a B-VND procedure, and (3) VNS\_U-VND that uses a U-VND in the improvement phase. The neighborhoods in the VND procedures are ordered in the 1<sup>st</sup> order. The performances of VNS procedures have been tested on class C1 of the instances described above. We tested our VNS procedures by using 4 different time limits, ranging from 360s to 1140s. For each instance and each VNS we ran the Algorithm 5 times. The results are summarized in Table IV and Figure 4.

TABLE IV. EVALUATION OF DIFFERENT VARIANTS OF VNS

Time limit (s)	Instances	Mimetic of Tricoire	BVNS		VNS_B-VND		VNS_U-VND	
			% Dev Best	% Dev Avg	% Dev Best	% Dev Avg	% Dev Best	% Dev Avg
360	Average C1	16820.07	-0.27%	1.56%	-1.04%	0.18%	-1.25%	0.14%
720			-0.56%	1.06%	-1.59%	-0.37%	-1.87%	-0.27%
1080			-0.56%	0.99%	-1.69%	-0.53%	-2.11%	-0.47%
1440			-0.57%	0.89%	-1.96%	-0.66%	-2.19%	-0.60%
Average			-0.49%	1.12%	-1.57%	-0.34%	-1.85%	-0.30%

For each time limit and each VNS variant, we report the deviation from the best solution found over all variants of instance of class C1 in 5 runs compared to the best solution found by the mimetic algorithm of [9, 11, 12] (named % Dev\_Best in Table IV). We also compute the deviation from the average value of the solutions found in 5 runs for all instances of C1 compared to the best solution found by the mimetic algorithm (named % Dev\_Avg in Table IV). The deviations are calculated by (3).

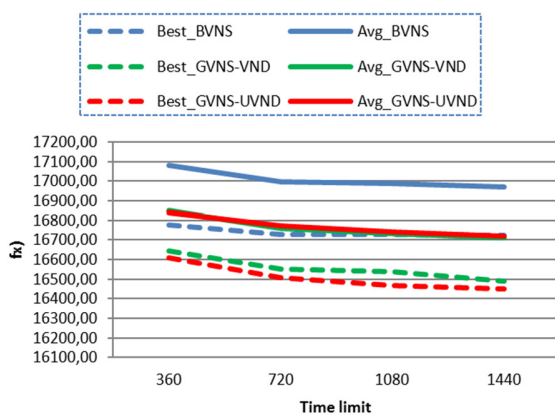


Fig. 4. Comparison between VNS procedures

From the results presented in Table IV and Figure 4 we may draw the following conclusions:

Firstly, we remark that the performance of VNS procedures in this paper depends on the time limit. As we have a long time we achieve the best results. All VNS variants outperform the results of the mimetic algorithm even if our VNS methods are stopped at 360s. The collaboration of all local search procedures is more beneficial than only the use of one local search in the inner loop in the VNS procedure. If we consider the average results over all variants of instance C1, it appears that the best average results are obtained by VNS\_U-VND, and that confirms what we found in the last section.

#### V. CONCLUSION AND PERSPECTIVES

In this paper, we considered a new variant of the Multi-Period Technician Routing and Scheduling Problem motivated by a real-life industrial application in a Telecommunication Company. To solve the problem, two variants of variable neighborhood descent B-VND and U-VND, as well as three variants of variable neighborhood search BVNS, VNS\_B-VND and VNS\_U-VND are proposed. All heuristic methods were tested and compared with the methods proposed by Tricoire [9, 11, 12]. The results confirm the effectiveness of our methods. Regarding future work, we will generate other instances to intensify the experimentations. Also, we will consider the dynamic aspect, where the demands appear dynamically over the planning horizon.

#### REFERENCES

- [1] V. Pillac, C. Gueret, A. L. Medaglia, "A parallel matheuristic for the technician routing and scheduling problem", *Optimization Letters*, Vol. 7, No. 7, pp. 1525-1535, 2013
- [2] J. Xu, S. Y. Chiu, "Effective heuristic procedures for a field technician scheduling problem", *Journal of Heuristics*, Vol. 7, No. 5, pp. 495-509, 2001
- [3] E. Hadjiconstantinou, D. Roberts, "Routing under uncertainty: an application in the scheduling of field service engineers", in: *The vehicle Routing Problem*, pp. 331-352, Society for Industrial and Applied Mathematics, 2001
- [4] E. Delage, *Re-optimization of Technician Tours in Dynamic Environments with Stochastic Service Time*, Ecole des Mines de Nantes, 2010
- [5] C. E. Cortes, F. Ordóñez, S. Souyris, A. Weintraub, "Routing technicians under stochastic service times: a robust optimization approach", *TRISTAN VI: The Sixth Triennial Symposium on Transportation Analysis*, Phuket Island, Thailand, June 10-15, 2007
- [6] V. Pillac, C. Gueret, A. Medaglia, "On the dynamic technician routing and scheduling problem", *5th International Workshop on Freight Transportation and Logistics*, Mykonos, Greece, May 21-25, 2012
- [7] X. Chen, B. W. Thomas, M. Hewitt, "Multi-Period Technician Scheduling with Experience-based Service Times and Stochastic Customers", *Computers and Operations Research*, Vol. 82, No. C, pp. 1-14, 2017
- [8] W. Jaskowski, *ROADEF Challenge 2007: Technicians and Interventions Scheduling for Telecommunications*, Poznan University of Technology, 2007

- [9] F. Tricoire, *Optimisation de Tournees de Vehicules et de Personnels de Maintenance: Application a la Distribution et au Traitement des Eaux*, PhD Thesis, Universite de Nantes, 2006 (in French)
- [10] A. A. Kovacs, S. N. Parragh, K. F. Doerner, R. F. Hartl, "Adaptive large neighborhood search for service technician routing and scheduling problems", *Journal of Scheduling*, Vol. 15, No. 5, pp. 579-600, 2012
- [11] N. Bostel, P. Dejax, P. Guez, F. Tricoire, "Multiperiod planning and routing on a rolling horizon for field force optimization logistics", in: *The Vehicle Routing Problem: Latest Advances and New Challenges*, pp. 503-525, Springer, 2008
- [12] F. Tricoire, N. Bostel, P. Dejax, P. Guez, "Exact and hybrid methods for the multiperiod field service routing problem", *Central European Journal of Operations Research*, Vol. 21, No. 2, pp. 359-377, 2013
- [13] O. Braysy, "A reactive variable neighborhood search for the vehicle routing problem with time windows", *INFORMS Journal of Computing*, Vol. 15, No. 4, pp. 347-368, 2003
- [14] M. Polacek, R. F. Hartl, K. Doerner, M. Reimann, "A variable neighborhood search for the multi depot vehicle routing problem with time windows", *Journal of Heuristics*, Vol. 10, No. 6, pp. 613-627, 2004
- [15] S. Salhi, A. Imran, N. A. Wassan, "The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation", *Computers & Operations Research*, Vol. 52B, pp. 315-325, 2014
- [16] M. Elbek, S. Wohlk, "A variable neighborhood search for the multiperiod collection of recyclable materials", *European Journal of Operational Research*, Vol. 249, No. 2, pp. 540-550, 2016
- [17] R. L. Pinheiro, D. Landa-Silva, J. Atkin, "A variable neighbourhood search for the workforce scheduling and routing problem", in: *Advances in Nature and Biologically Inspired Computing*, pp. 247-259, Springer, 2016
- [18] N. Mladenovic, P. Hansen, "Variable neighborhood search", *Computers & Operations Research*, Vol. 24, No. 11, pp. 1097-1100, 1997
- [19] P. Hansen, N. Mladenovic, "Variable neighborhood search: Principles and applications", *European Journal of Operational Research*, Vol. 130, No. 3, pp. 449-467, 2001
- [20] P. Hansen, N. Mladenovic, J. A. M. Perez, "Variable neighbourhood search: methods and applications", *Annals of Operations Research*, Vol. 175, No. 1, pp. 367-407, 2010
- [21] P. Hansen, N. Mladenovic, R. Todosijevic, S. Hanafi, "Variable neighborhood search: basics and variants", *EURO Journal on Computational Optimization*, Vol. 5, No. 3, pp. 1-32, 2016
- [22] A. Mjirda, R. Todosijevic, S. Hanafi, P. Hansen, N. Mladenovic, "Sequential variable neighborhood descent variants: an empirical study on the traveling salesman problem", *International Transactions in Operational Research*, Vol. 24, No. 3, pp. 615-633, 2016