UDC: 658.818.3:519.87

DOI: http://dx.doi.org/10.7708/ijtte.2015.5(1).08

TWO-PHASE ALGORITHM FOR SOLVING HETEROGENEOUS TRAVELLING REPAIRMEN PROBLEM WITH TIME WINDOWS

Nenad Bjelić[1], **Dražen Popović**[2]

[1, 2] *University of Belgrade, Faculty of Transport and Traffic Engineering, Vojvode Stepe 305, 11000, Belgrade, Serbia*

Received 22 November 2014; accepted 19 January 2015

Abstract: Heterogeneous travelling repairmen problem with time windows (hetTRPTW) is customer oriented problem with large possibilities for practical applications in logistics area. Models and algorithms developed for solving one problem with a cumulative objective function may be, with a little effort, transformed for solving similar problem with a cumulative function. In that sense, aim of this paper is to present results obtained by implementing an algorithm developed for solving cumulative capacitated vehicle routing problem in solving hetTRPTW.

Keywords: heterogeneous travelling repairmen problem with time windows, two phase local search, variable neighborhood descent.

1. Introduction

Travelling repairman problem (TRP) is combinatorial optimization problem very similar to the well known travelling salesman problem (TSP). Namely, just like in the case of a salesman in the TSP, the TRP implies that a repairman has to visit all nodes in a network, starting from the depot node, where nodes represent breakdowns in a system at the beginning of the planning horizon. The main difference between salesman and repairman is in their objective. Precisely, while salesman tries to finish his job as soon as possible, repairman's objective is to finish the job in such a way that the sum of all clients' waiting time till the breakdown repair is minimized. Cumulative nature of repairman's objective function is the reason why solving TRP has been proven to be computationally harder then solving TSP. On the other hand, the cumulative objective function, which minimizes the average time clients wait till the end of service (Lysgaard and Wøhlk, 2014), is useful for practical implementation in numerous systems whose primary objective is clients' satisfaction. Example of logistics systems suitable for this kind of objective function are: material handling systems for loading and unloading ships in a port, delivery tasks of express mail companies, internal transportation systems with automated guided vehicles, etc. Beside logistics systems the cumulative objective function has been used for modeling different systems for providing emergency help after natural disasters. Moreover, wide applicability of the cumulative objective function, as well as the importance it has on different research areas, resulted that, besides TRP, it can be found in the literature as the minimum latency problem (MLP), the delivery man problem (DMP) and the cumulative capacitated vehicle routing problem (CCVRP).

[1] Corresponding author: n.bjelic@sf.bg.ac.rs

In this research we focus on solving generalized TRP called heterogeneous travelling repairmen problem with time windows (hetTRPTW), meaning that there is a fleet of heterogeneous repairmen for executing tasks, as well as that each task is characterized with a time window. Therefore, the rest of the paper is organized in such way that in the following section we present, formulate and give brief literature overview of the hetTRPTW. In section three we present two heuristic algorithms for solving hetTRPTW. In section four we tested efficiency of presented algorithms by implementing them for solving test instances. Finally, we give a conclusion about obtained results.

2. Heterogeneous Traveling Repairmen Problem with Time Windows

Although different approaches for solving TRP has been developed, such as polynomial algorithms for solving special cases of the problem (Garcia et al., 2002), exact (Wu et al., 2004; Silva et al., 2012), approximation (Chaudhuri et al., 2003) and heuristic (Salehipour et al., 2011; Mladenović et al., 2013) algorithms for solving general cases of the problem, and different problem formulations (Fischetti et al., 1993), that is not the case with the more general variant of the problem which includes existence of tasks' time windows (TRPTW). This is because more general variants of the problem attracted researchers' attention only in recent years. Nevertheless, in the first paper regarding TRPTW (Tsitsikils, 1992) author showed that TRPTW is strongly NP-complete problem.

hetTRPTW generalizes the TRP in two ways. The first one implies that there is a fleet of repairmen that fix breakdowns, i.e. execute tasks. All repairmen in the fleet are heterogeneous regarding time needed for task execution and velocity while moving between task locations. The second way implies that every task is related with the time window, meaning that task occurrence times are respected. Precisely, one of the main assumptions of the TRP is that all tasks are already in the system at the beginning of the planning horizon, meaning that as soon as repairman reaches task location it can start serving it. However, in real systems very frequently this is not possible until all serving conditions are fulfilled. For example, courier in an express mail company cannot deliver a shipment until client's office is opened; automated guided vehicle cannot start transfer of an item between locations until the item is available for the pick up at the origin location; services on a ship cannot start until ship has arrived in a port; etc. Generally, there are a lot of systems in which it is possible to know in advance occurrence times of tasks (breakdowns) within planning horizon. By including those tasks in the planning process it is possible to achieve better system performances regarding the cumulative objective function compared with the strategy that considers only tasks available for the service from the beginning of the planning horizon. Therefore, in this research we consider a problem in which all tasks to be served are not present in the system at the beginning of the planning horizon, but their occurrence times are known. It should be emphasized that there is no right hand side of task's time window, meaning that task can be served at any time after the occurrence in the system.

There are several papers in the literature related to the hetTRPTW. In Bjelić et al. (2010) authors presented two mathematical formulations of the problem. Metaheuristic algorithms for solving hetTRPTW are given in papers Bjelić and Vidović (2011) and Bjelić et al. (2013a, 2013b). The average best results in terms of the objective function values are achieved by the variable neighborhood search (VNS) algorithm presented in the paper Bjelić et al. (2013b).

Beside the papers that consider hetTRPTW, there are papers that consider homogenous fleet of repairmen (van der Meer, 2000), as well as papers that consider problems with one repairman and time windows (Heilporn et al., 2010), and the existence of more than one repairman but without time windows (Ngueveu et al., 2010; Ribeiro and Laporte, 2012; Ke and Feng, 2013).

Mathematical formulation of the hetTRPTW is based on the network flow through a complete, directed, asymmetrical and weighted graph G = (*N,E,t*). If *V* denotes set of available repairmen, where $m = |V|$ is their number, and *P* denotes set of tasks, whose number is *n* ($n = |P|$) then a solution of the hetTRPTW is represented as a set of *m* Hamiltonian paths trough *G* with the objective that the sum of tasks' waiting times till the end of service is minimized.

Graph *G* is defined with a set of nodes, *N*, and a set of edges, *E*, where every edge from *E* is related with a weight *t*. Set of nodes is defined as $N = P \cup \{0, n+1\}$, where nodes 0 i *n*+1 are dummy nodes representing a depot, i.e. there are not tasks that requires service. Node 0 denotes the depot as the starting point of a repairman's route, while node *n*+1 denotes the depot as the last point in a route. The set of edges, *E*, is defined as $E = \{(v,i,j) : \forall v \in V, \forall i,j \in N\}$, meaning that each pair of nodes $(i,j) \forall i,j \in N$ is connected with *m* parallel edges, where each edge is dedicated to only one repairman from *V*. Every edge from *E* has its weight t_{ij}^{v} which represent time required by repairman *v* to visit node *j* after node *i*, including the service in the node *j*. Accordingly, it is clear that t_{ij}^{v} is calculated as $t_{ij}^{v} = \frac{d_{ij}}{v^{v}} - s_{j}^{v}$, where d_{ij} is distance between nodes *i* and *j*; v^{v} is velocity of a repairman *v*; and s_{j}^{v} is serving time of task *j* by a repairman *v*.

This formulation provides the opportunity to model the problem as MILP (Bjelić et al., 2013b) which, because of the problem's NP-completeness, is not suitable for the practical implementations. Therefore, solving problems of real sizes implies implementation of some suboptimal algorithms. Heuristic algorithms have proven to be very efficient in solving combinatorial optimization problems. For solving TRP, and all of its generalized variants, perturbation based algorithms, such as VNS, GRASP, and ALNS, gave very good results.

Based on similarities between CCVRP and hetTRPTW, in terms of existence of the cumulative objective function and a presence of more than one repairman in the system, but at the same time respecting their differences (absence of time windows in the CCVRP and homogeneous fleet of servers), the main objective of this research is to investigate the efficiency of using two phase heuristic algorithm, presented in Ke and Feng (2013), for solving hetTRPTW.

3. Handling Devices Allocation Problem – HDAP

Gravel distribution by inland waterway transportation includes three main phases:

loading of gravel by a suction dredger into barges, transport of gravel to the ports or unloading locations, and unloading of gravel by a handling facility that usually consists of pontoon mounted crane and belt conveyor. Because of high costs, a number of handling facilities is usually relatively small, and requires successive relocation of handling equipment between different unloading locations. Accordingly, providing efficient and cost effective service of loaded river barges needs appropriate scheduling plan for handling equipment, which means defining sequence of unloading locations that should be served by each handling device.

The problem may be introduced in the following way. For a given collection of barges unloading tasks find a set of assignments to minimize the sum of unloading service times that include both waiting for the service and handling devices transfer times. This problem will be referenced in the rest of the paper as Handling Devices Allocation Problem - HDAP.

Obviously, HDAP may be seen as a variant of the hetTRPTW in which fleet of heterogeneous unloading devices is represented as a fleet of heterogeneous repairmen, and set of unloading tasks is represented as the set of repair tasks. Therefore, in the rest of the paper we firstly present a heuristic algorithm for solving hetTRPTW, and after that we test the efficiency of presented algorithm by implementing it for solving HDAP.

4. Two Phase Local Search Algorithm

Two phase algorithm presented in Ke and Feng (2013) is a perturbation based algorithm, meaning that it tries to improve the incumbent solution, x_b, by performing systematically organized changes in the solution structure. This algorithm is referenced in the rest of the paper as orig2Phase algorithm. Search for an improved solution in the org2Phase algorithm implies that in every pass through it two phases are executed. In each phase starting solution is firstly deteriorated by applying predefined set of perturbation moves, after which it is improved as much as it is possible by a second set of perturbation moves.

In that sense, org2Phase algorithm starts by executing randomly selected Exchange or Cross perturbation move H times. In cases where $m > 4$, H is randomly selected element from the set $[2, \min(m/2,10)]$, otherwise $H = 2$. After execution of the deterioration moves, final solution of the first phase, $x_{ib}$is obtained by implementation of the local search algorithm (explained in detail in section 4.2).

Second phase tries to improve problem solution by improving order of service at each repairman in the fleet. Therefore, for every repairman with more than seven tasks order of service is firstly deteriorated by implementing randomly selected 3-OPT or 4-OPT move, after which it is improved by implementing 2-OPT moves. This procedure is executed maximally T times, where T is algorithms parameter. It should be noted that second phase is not realized for the cases of repairmen with seven, or less, tasks because, according to the experience of org2Phase algorithm authors, local search algorithm from the first phase provides very good order of service for such repairmen. The search for an improved solution by implementing 2-OPT move is realized according to the best improvement search strategy. Finally, if the solution after 2-OPT search is better that the current best solution, local search algorithm from phase one is executed with the improved solution as the initial solution.

Alg. 1.

Pseudo code of the org2Phase algorithm

```
Initialize $x_b$ as in chapter 3.1
Until the ending conditions are met do the following:
  /*Phase one*/
  $x \leftarrow x_b$
  from the set {Exchange move, Cross move} select a perturbation move in a random manner
  in a random manner select a positive integer H from the interval $[2, \min(m/2,10)]$
  execute the selected perturbation move H times successively on x and obtained solution denote as y
  obtain a solution $x_{ib}$ by applying a local search algorithm with y as a starting solution
  if $f(x_{ib}) < f(x_b) : x_b \leftarrow x_{ib}$
  /*Phase two*/
  improvement $\leftarrow$ False
  for i=1 to m do:
    $x \leftarrow x_{ib}$
    If route contains more than seven tasks:
      for j=1 to T do:
        from the set {3-OPT move, 4-OPT move} select a perturbation move in a random manner
        obtain solution y by implementing selected move on solution $x$
        the best solution from 2-OPT neighborhood of the solution y denote by z
        if $f(z) < f(x_{ib})$ :
          $x_b \leftarrow z$
          improvement $\leftarrow$ True
          break
  if improvement = True:
    improve $x_{ib}$ by implementing a local search algorithm
  if $f(x_{ib}) < f(x_b) : x_b \leftarrow x_{ib}$
```

Source: Ke and Feng (2013)

There are numerous stopping criteria for a perturbation algorithm, such as: maximum number of iteration, maximum number of iteration without solution improvement, running time of an algorithm etc. As a stopping criterion for org2Phase algorithm authors proposed maximal number of successive realizations (Max_{iter}) of first and second phases.

4.1. Initial Solution

Because perturbation algorithms search for a better solution by exploring neighborhoods of currently the best solution they require an initial solution. In the case of Org2Phase algorithm an initial solution is obtained by implementing regret insertion algorithm. General idea behind the regret insertion algorithm is to get insight into costs of following insertions so that task with the highest cost in following insertions is inserted in current step. Precisely, regret algorithm applied in org2Phase algorithm inserts task *i* to the repairman *j* according to the values of Δ_{ij} which represents objective value increase due to adding task *i* for the service to the repairman *j* ($j \in \{1,\ldots,m\}$). If by v_{ik} , where $v_{ik} \in \{1,\ldots,m\}$, we denote a repairman for which insertion of task *i* results in *k*th lowest increase of the objective function, i.e. if stands that $\Delta_{iv_{ik}} \le \Delta_{iv_{ik'}}$ for all $k \le k'$, then insertion decision is based on a regret value calculated as $\Delta_{iv_{i3}} \le \Delta_{iv_{i1}}$. In other words, decision of inserting task *i* is based on difference between the values of inserting task on a repairman with the lowest increase of the objective values, and the repairman with the 3rd lowest increase of the objective function. In case there are less than three repairmen in the fleet decision is based on a difference between objective function increases of two repairmen. In every step of the regret algorithm the task with the highest regret value is inserted. In the case of two tasks with the same regret value, task for insertion is selected in a random manner.

4.2. Local Search Algorithm

Local search algorithm is of great importance for heuristic algorithm performances because it searches for an improved solution in a closed proximity of the best found solution. Implementation of a local search algorithm implies defining of the following algorithm's features: initial solution; a set of perturbation moves and the order in which they will be executed in the search; and finally the improvement strategy. Concerning initial solutions of the local search algorithm implementations in org2Phase algorithm, from the pseudo code given in the Algorithm 1 it is clear that in first phase initial solution is solution y, while in the second phase initial solution is given by the solution x_b. Perturbation moves executed in the org2Phase algorithm are 2-OPT, Exchange and Relocate (Fig. 1) and they are used in the given order.

Implemented improvement strategy defines how improved solutions from a local search algorithm are selected. Namely, in the case of the first improvement strategy (FI) each time a better solution is found it is accepted as a new the best solution of the search and the local search algorithm is restarted with that solution as the initial one. Oppositely, the best improvement strategy (BI) implies that all moves from the local search algorithm are executed and that the best solution found in that search is accepted as the new the best solution of the search. Of course, local search algorithm is restarted only in the case when the new best solution is better than the previous. If all moves of the local search algorithm are executed without finding a solution that is better than the initial one search ends.

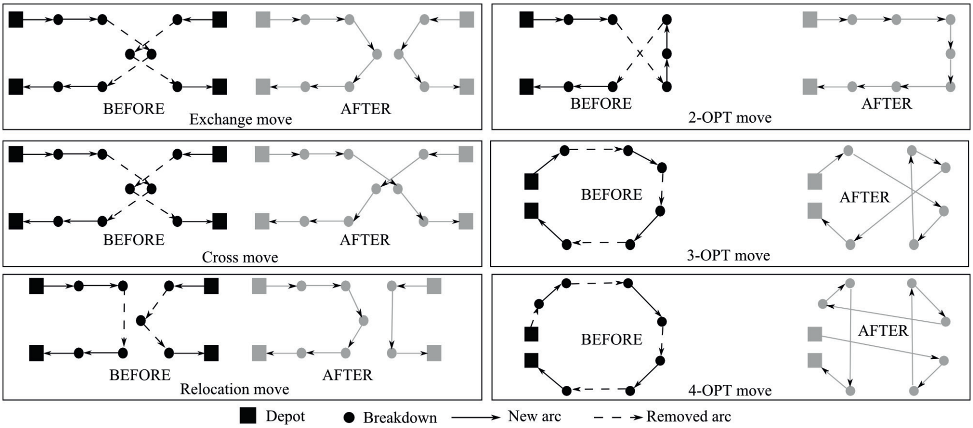

Fig. 1.

Perturbation Moves Used in Org2phase Algorithm

Concerning differences between CCVRP and hetTRPTW it is obvious that different perturbation moves improve or worsen performances of proposed algorithm, both in senses of obtained objective values and execution time. Therefore, because of the importance efficient local search algorithm has on performances of org2Phase algorithm we changed given local search algorithm with the variable neighborhood descent (VND) algorithm given in Bjelić et al. (2013b). This algorithm is called mod2Phase and its performances, together with performances of org2Phase algorithm, are tested in solving HDAP on numerical examples presented the next section.

5. Computational Experiments

Efficiency of org2Phase and mod2Phase algorithms, and the comparison with the efficiency of VNS based algorithms from Bjelić et al. (2013b) in solving HDAP is tested on a set of 30 randomly generated instances consisting of 10, 25 and 50 tasks created according to the example of the Belgrade gravel serving area.

Fig. 2 shows the 20 unloading stations of the Belgrade serving area with the distances between adjacent gravel unloading locations (in kilometers), depot location and the corresponding incomplete graph.

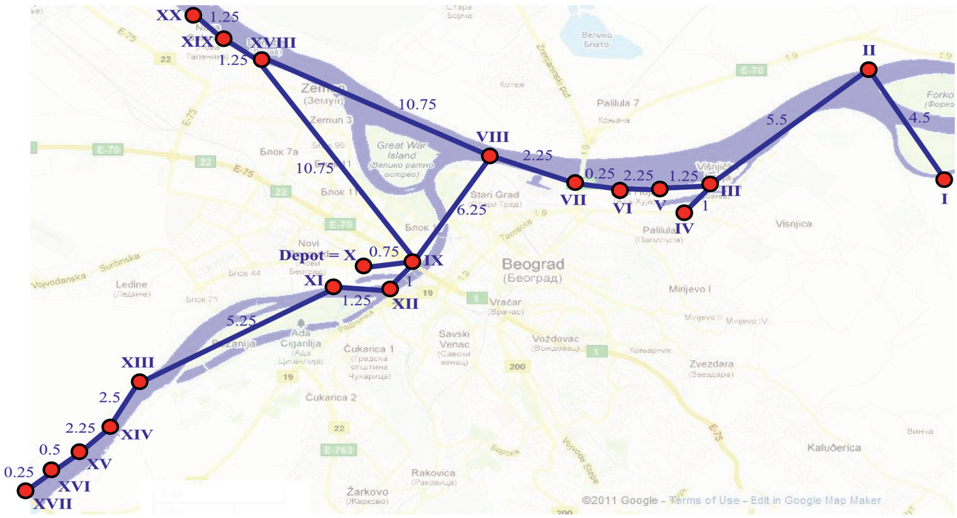

Fig. 2.
Belgrade Serving Area

We considered cases with an equal chance of task arrivals at each location and with a uniform distribution of task inter-arrival times over the planning horizon. Each task implies unloading of 1000 tons of gravel. Devices used for gravel unloading belong to one of three different classes ("I", "II" and "III"). Devices of class "I" have transfer speed of 6 km/h, and unloading capacity of 100 t/h; of class "II" have transfer speed of 5 km/h, and unloading capacity of 150 t/h; while devices of the class "III" have transfer speed of 4 km/h, and unloading capacity of 200 t/h. The following series of device classes {"I", "II", "III", "II", "III", "II", "III", "I", "II", "III"} corresponds to devices that make fleets of 2, 5 and 10 devices. It is assumed that all of the devices are available for the service at the beginning of the planning horizon.

All of the test runs were executed on a Windows XP OS powered by an AMD Phenom II 2.61 GHz processor with 2GB of RAM, while all of the coding was done in Python 2.5. Every instance of the problem has been solved five times and overall average values of objective functions and running times for all pairs (number of tasks, number of unloading devices) are given in Tables 1 and 2, respectively. Parameters used in executing org2Phase and mod2Phase algorithms were Max_{iter} = 10 and T = 5.

Table 1

Average Objective Function Values of Test Instances

	m=2				m=5				m=10			
Number of tasks	VNS	RdVNS	orig2Phase	mod2Phase	VNS	RdVNS	org2Phase	mod2Phase	VNS	RdVNS	orig2Phase	mod2Phase
10	187.3373	188.2016	187.6915	187.3321	65.34234	65.38666	65.38282	65.31876	54.72144	54.70108	54.72446	54.7539
25	837.924	842.8145	841.8041	838.6069	157.0026	156.9921	157.0115	157.4803	131.3228	131.3397	131.3478	131.4067
50	2955.785	2973.431	2980.204	2960.791	333.8942	333.867	336.1993	336.925	269.3232	269.1586	269.1572	269.19738

Table 2

Average Running Times of Algorithms (in Seconds)

	m=2				m=5				m=10			
Number of tasks	VNS	RdVNS	orig2Phase	mod2Phase	VNS	RdVNS	org2Phase	mod2Phase	VNS	RdVNS	orig2Phase	mod2Phase
10	7.5505	2.1956	0.06582	0.14338	7.7731	4.7908	0.1646	0.09286	11.4041	7.9177	0.36302	0.14976
25	250.974	77.4698	1.97948	5.69968	140.7488	67.8415	2.82436	2.24074	156.227	83.6353	4.11374	2.42336
50	2940.712	1516.273	28.43868	94.94366	1558.489	692.8986	34.63734	32.26988	1281.065	677.4177	38.87604	27.63836

From the data presented in the Table 1 it is evident that differences between org2Phase and mod2Phase algorithms, and VNS based algorithms from Bjelić et al. (2013b) in terms of average objective functions are minimal and acceptable form practical implementation point of view. Nevertheless, it has to be noted that VNS based algorithms, especially VNS algorithm, obtained lover values of the objective functions. However, data from Table 1 indicates that 2Phase based algorithms are more robust on the increase in problem's complexity. Namely, with the increase of the number of tasks in the planning horizon, or the number of unloading devices in the fleet, difference of average objective function values of the 2Phase based algorithms and the VNS based algorithms are minimized or even on the side of the 2Phase based algorithms. In the case of the time efficiency of the 2Phase based algorithms, form the Table 2 it is obvious that they are significantly efficient that the VNS based algorithms. It is also noticeable that the effectiveness of the org2Phase and mod2Phase algorithms, in terms of the average objective function values, is in close correlation with the running time of an algorithm.

6. Conclusion

According to the results from numerical experiments of solving HDAP, it is obvious that algorithms for solving hetTRPTW, which are presented in the paper, provide solutions of good quality in practically applicable running time. In that sense, this research should be considered just as a beginning step in more detailed study of determining the importance that different perturbation moves have on a solution quality and running times of both entire algorithm and the local search heuristics. Moreover, the study must examine necessity of second phase in the algorithm because, although study of Ke and Feng (2013) showed that the second phase improves efficiency of the two-phase algorithm, it does not have to stand for the case of solving hetTRPTW.

Acknowledgement

This paper is partially supported by the Ministry of education, science and technological development of the Government of the Republic of Serbia trough the project TR36006 in the period 2011-2014.

References

Bjelić, N.; Vidović, M.; Miljuš, M. 2010. Two mathematical modells of the Handling Devices Allocation Problem. In *Proceedings of the XXXVII operations research symposium - SYMOPIS 2010*, 349-352 (In Serbian).

Bjelić, N.; Vidović, M. 2011. Memetic algorithm for Dynamic Handling Device Allocation Problem. In *Proceedings of the XXXVIII operations research symposium - SYMOPIS 2011*, 359-362 (In Serbian).

Bjelić, N.; Vidović, M.; Popović, D.; Ratković, B. 2013a. Genetic algorithm for solving travelling repairman problem with time windows. In *Proceedings of the XL operations research symposium - SYMOPIS 2013*, 509-514 (In Serbian).

Bjelić, N.; Vidović, M.; Popović, D. 2013b. Variable neighborhood search algorithm for heterogeneous traveling repairmen problem with time windows, *Expert Systems with Applications*. DOI: http://dx.doi.org/10.1016/j.eswa.2013.05.036, 40(15): 5997-6006.

Chaudhuri, K.; Godfrey, B.; Rao, S.; Talwar, K. 2003. Paths, trees, and minimum latency tours. In *Proceedings of 44th symposium on foundations of computer science (FOCS)*, 36-45.

Fischetti, M.; Laporte, G.; Martello, S. 1993. The delivery man problem and cumulative matroids, *Operations Research*. DOI: http://dx.doi.org/10.1287/opre.41.6.1055, 41(6): 1055-1064.

García, A.; Jodrá, P.; Tejel, J. 2002. A note on the traveling repairman problem, *Networks*. DOI: http://dx.doi.org/10.1002/net.10031, 40(1): 27-31.

Heilporn, G.; Cordeau, J.; Laporte, G. 2010. The delivery man problem with time windows, *Discrete Optimization*. DOI: http://dx.doi.org/10.1016/j.disopt.2010.06.002, 7(4): 269-282.

Ke, L.; Feng, Z. 2013. A two-phase metaheuristic for the cumulative capacitated vehicle routing problem, *Computers & Operations Research*. DOI: http://dx.doi.org/10.1016/j.cor.2012.08.020, 40(2): 633-638.

Lysgaard, J.; Wøhlk, S. 2014. A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem, *European Journal of Operational Research*. DOI: http://dx.doi.org/10.1016/j.ejor.2013.08.032, 236(3): 800-810.

Mladenović, N.; Urošević, D.; Hanafi, S. 2013. Variable neighborhood search for the traveling deliveryman problem, *4OR: A Quarterly Journal of Operations Research*. DOI: http://dx.doi.org/10.1007/s10288-012-0212-1, 11(1): 57-73.

Ngueveu, U.S.; Prins, C.; Wolfler-Calvo, R. 2010. An effective memetic algorithm for the cumulative capacitated vehicle routing problem, *Computers and Operations Research*. DOI: http://dx.doi.org/10.1016/j.cor.2009.06.014, 37(11): 1877-1885.

Ribeiro, G.M.; Laporte, G. 2012. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem, *Computers and Operations Research*. DOI: http://dx.doi.org/10.1016/j.cor.2011.05.005, 39(3): 728-735.

Salehipour, A.; Sorensen, K.; Goos, P.; Braysy, O. 2011. Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem, *4OR: A Quarterly Journal of Operations Research*. DOI: http://dx.doi.org/10.1007/s10288-011-0153-0, 9(2): 189-209.

Silva, M.; Subramanian, A.; Vidal, T.; Ochi, S. 2012. A simple and effective metaheuristic for the minimum latency problem, *European Journal of Operational Research*. DOI: http://dx.doi.org/10.1016/j.ejor.2012.03.044, 221(3): 513-520.

Tsitsikils, J. 1992. Special cases of traveling salesman and repairman problem with time windows, *Networks*, 22: 263-282.

Van der Meer, R. 2000. *Operational control of internal transport*. PhD thesis, University of Rotterdam.

Wu, Y.; Huang, Z-N.; Zhan, F-J. 2004. Exact algorithm for the minimum latency problem, *Information Processing Letters*. DOI: http://dx.doi.org/10.1016/j.ipl.2004.09.009, 92(6): 303-309.