GENETIC ALGORITHM APPROACH FOR SOLVING TRUCK SCHEDULING PROBLEM WITH TIME ROBUSTNESS

Nenad Bjelić*
University of Belgrade, Faculty of Transport and Traffic Engineering, n.bjelic@sf.bg.ac.rs
Dražen Popović
University of Belgrade, Faculty of Transport and Traffic Engineering, d.popovic@sf.bg.ac.rs
Branislava Ratković
University of Belgrade, Faculty of Transport and Traffic Engineering, b.ratkovic@sf.bg.ac.rs

***Abstract**: Problem considered in this paper regards inbound truck to dock door assignments in a cross-dock terminals with temporal constraints included in decision process. Because of stochastic nature of terminal related activities, modifications of already established operational plans are relatively frequent. These disturbances are mainly induced by inaccurate assessment of inbound trucks' arriving times, caused by unpredictable traffic conditions, as well as by stochastic nature of unloading operation within a terminal. We present a new approach for solving this type of problem by utilizing genetic algorithm framework. Eventually, we compare efficiency of presented algorithm with the algorithms already used for solving the problem.*

Keywords: *cross-dock, operational planning, genetic algorithm.*

* Corresponding author

1. INTRODUCTION

Cross-dock is a term used in logistics practice to refer to a process implemented mainly in hub terminals. Generally, cross-dock implies unloading of incoming transportation means at inbound doors, and loading of outbound trucks or trailers at outbound doors, with little or no storage in between. More precisely, general framework of a cross-dock terminal implies that after incoming trucks arrive at the inbound doors of the terminal, each arrived item is sorted in accordance with its destination and moved onto the shipping door where outgoing truck waits for making delivery to the designated destination. The movements of items from the inbound to the outbound doors are usually made either by fork-lift trucks or conveyors. Efficiency of this type of terminals is dependent on the quality of numerous cross-docks' strategic, tactical and operational problems.

Solving a strategic, tactical or operational problem implies making a decision, which, on the other hand is formalized by generating a plan. Based on that plan a problem is solved.

Accordingly, operational decisions result in operational plans which contain detailed information about realization of considered activities. Numerous operational decisions that have to be made in a cross-dock terminal imply generation of numerous operational plans. One of those decisions is considered in this paper.

Despite the relatively precise information on which operational plans are made, because of stochastic nature of terminal related activities, modifications of already established operational plans are relatively frequent. These disturbances are mainly induced by inaccurate assessment of inbound trucks' arriving times, caused by unpredictable traffic conditions, as well as by stochastic nature of unloading operation within a terminal. High intensity of transshipment processes within a cross-dock terminal implies that any deviation from original operational plans has multiplicative impact of system performances by influencing all activities after the disorder point.

Because it is practically impossible to influence on traffic conditions in order to reduce truck arrival time inaccuracies, and because stochastic nature of unloading activities is practically inevitable, the best way to reduce operational plan disturbances is generation of robust operational plans. Such plans are capable of absorbing some level of variations in arrival and unloading truck times.

In this paper we present new metaheuristic approach of generating robust operational plans for one problem in serving inbound trucks in cross-dock terminals. After that we compare effectiveness and efficiency of two variations of presented metaheuristic approach, and compare them to approaches and metaheuristic algorithms already presented in the literature. Based on the results of the computational experiments it is obvious that the main contribution of this paper is formalization of a solution methodology, based on the genetic algorithm framework. This algorithm provides solutions of large scale size problems in practically applicable times and with reasonable quality.

The rest of the paper is organized in such way that in the following section we give the problem description and brief literature review; in section three we present new metaheuristic approach for solving the problem; in section four we compare proposed solution procedures to existing solution procedures. Finally, in section five we give conclusion on obtained results.

2. PROBLEM DESCRIPTION AND LITERATURE REVIEW

Problem considered in this paper regards inbound truck to dock door assignments in a cross-dock terminals with temporal constraints included in decision process. This problem is known in literature as the truck scheduling problem (TSchP). The problem very similar to the TSchP is the dock assignment problem (DAP). Because of the similarity, the DAP is commonly misplaced with the TSchP. The main difference between these two problems is that DAP is considered only when set of inbound trucks is to be allocated on a set of available dock doors in such way that not more than one truck is assigned to each dock. The reason for such allocation is that DAP does not consider time as a constraint.

On the other hand, the TSchP considers time aspect of the problem and consequentially it allocates more then one truck on a dock door. In other words, TSchP considers the dock doors as resources (used by the trucks) that have to be scheduled over time. The problem decides on the succession of inbound trucks at the dock doors of a cross-dock: where and when the trucks should be processed [1].

More precisely, operational plan obtained by solving TSchP determines which trucks will be served on which dock door, as well as order of service for trucks served at the same door.

Because of its importance on effective realization of cross-dock operations, the TSchP has been subject of numerous research papers. As a result of problem complexity different cross-dock structures are considered in the literature, all with intention to get insight into the problem by reducing the size of a problem. Some authors decided to consider only inbound or outbound parts of a cross-dock terminal, while others reduced a number of available dock doors in a terminal. For the reader interested in comprehensive literature review regarding the TSchP we recommend papers [1] and [2].

Another differentiate feature of the TSchP models is observed objective function. By classification used in [2] authors presented six the most widely used objective functions and leave place for some surrogate goals of a model.

Objective function considered in this research is firstly used by Acar in [3] and by Acar et al. in [4] with the goal to increase the robustness of the inbound truck schedule in a cross-dock to deviations in arrival times and unloading times. Therefore in the rest of the paper we refer to the problem as the TSchP with Time Robustness (TSchPTR). Aforementioned goal has been accomplished by using objective function that aims at spreading the slack time (time that remains after subtracting inbound trucks' service times from the time available for service at a dock door) as evenly as possible in order to create buffers to absorb variability in scheduled arrival and service times. This is achieved by minimizing the square of the slack time since this is equivalent to minimizing the variance.

Therefore if by N we denote cardinality of set of incoming trucks to be served on M available dock doors, and if $S_{j,k}$ denotes idle time before task j served on dock door k, then objective function used in TSchPTR can be written as (1).

$$\min \sum_{k=1}^{M} \sum_{j=1}^{N+M} S_{j,k}^{2} \qquad (1)$$

It should be noted that values of index j larger then N refer to idle times after the last trucks served at a dock door, as well as that values of $S_{j,k}$ are equal to zero for all trucks j not served at dock door k.

In [3] and [4] authors define mathematical model for obtaining optimal solution of the problem. The model consists of linear constraints, quadratic objective function and mixed, binary and continuous, variables. Because of combinatorial complexity of the problem, presented MIQP (mixed integer quadratic programming problem) can be useful only for small size problem instances. For large size problem instances authors presented the Door Assignment Heuristics (DAH). In [4] authors tested efficiency of the DAH in different dock utilization rates which showed that average deviation of DAH result from optimal ones ware around 9% in the case of high dock utilization, and around 4.5% in low utilized systems.

Metaheuristic approach in solving TSchP is not new. So far one can find several papers that already used it, and for example authors in [5] used it for solving complex case of the TSchP. However, the first metaheuristc approach for solving TSchPTR can be found in [6] where authors presented algorithm based on the Variable Neighborhood Search (VNS) metaheuristic framework. Tests conducted in the research showed that the VNS algorithm outperformed the DAH in terms of the objective function values. Nevertheless, VNS's running times seemed to increase exponentially with the growth of problem size. Therefore, in this research we present another metaheuristic algorithm whose running time increases at significantly lower rate, but which gives solution of satisfying quality.

For a detailed insight into the mathematical formulation of the TSchPTR, the DAH, the Insertion and the VNS algorithm, the reader is referred to [4] and [6].

3. GENETIC ALGORITHM

Genetic algorithm (GA) is well known and widely used metaheuristic framework designed for solving different combinatorial optimization problems. The GA exploits of the main principles of the evolution of living organisms which eventually results in survival and future development of the best individuals in a population. Accordingly, realization of every GA consists of the following procedures: generation of initial population of problem solutions (aka chromosomes), selection of chromosomes for reproduction process, and reproduction process itself. Reproduction process is made of crossover, and mutation procedures. Procedures of chromosome selection and reproduction are repeated as long as one of algorithm's termination criteria is not fulfilled. During the algorithm's run initial solutions are successively improved toward a solution which is relatively close to the optimal solution of a problem. Closer insight to the GA's procedures used for solving the TSchPTR is the subject of this chapter.

One of the main characteristics of the GA is a solution encoding. Encoding used in this research is a consequence of the problem's nature. Namely, because the goal of the problem is to optimally distribute slack times between successive tasks on a dock, an optimal solution has to have increasing order of tasks on each dock in terms of truck incoming times. Therefore, in a problem of *n* truck arrivals that are to be served on *m* docks, solution encoding is an array of *n* positions. Each position is related to the task whose ordinal number is equal to the position in the array, starting from the left hand side of the array, and it contains an information about the dock on which task is served. Size of the so encoded solution space is equal to n^m. For example, in problem with 10 tasks and 4 docks, the solution in which tasks 1, 4, and 7 are served on dock 1, tasks 3 and 5 on dock 2, tasks 2, 6, and 10 on dock 3, and tasks 8 and 9 on dock 4, is represented as 1,3,2,1,2,3,1,4,4,3. Illustration of this example is given on Figure 1.

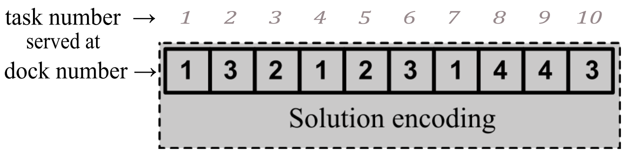

Figure 1. Solution encoding used in the research

Each chromosome in the initial population of chromosomes is generated in a random manner.

Selection of chromosomes for reproduction process is realized by roulette wheel procedure. Probability for selecting a chromosome is obtained by implementing fitness function $\phi(x)$ on value of chromosomes objective function x. Expression used for fitness function is given by (2),

$$\phi(x) = \frac{e^{f(x)-\mu/5}}{1+e^{f(x)-\mu/5}} \qquad (2)$$

where μ represents average value of population objective function.

Chromosome pairs for crossover procedure are selected randomly with consideration that two same chromosomes cannot be selected. In such case one chromosome is replaced with some other chromosome from selected population. Eventually, if there are no two different chromosomes in a remaining crossover population, one of them is replaced with randomly generated one, as in initial population.

As crossover operators we used two simple but widely used ones. The first one is 1-point crossover operator where one break point on both parents is randomly selected. The first offspring is generated by combining the part before the break point on the first parent and the part after the break point on the second parent. The second offspring is generated accordingly, but with parents' counterparts. Illustrative example of the used 1-point crossover is presented on Figure 2.

The second crossover operator is 2-point crossover. This operator implies selection of two randomly selected break points for both parents. The first offspring is generated by replacing first parent's part of chromosome between breaking points by second parent's part of chromosome between break points. The second offspring is generated accordingly but with reverse positions of parents. Illustrative example of 2-point crossover is presented on Figure 3.

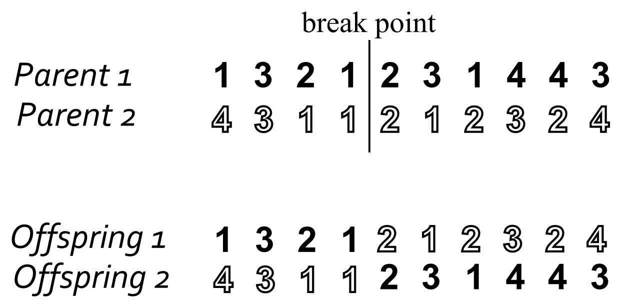
break point
Parent 1 1 3 2 1 2 3 1 4 4 3
Parent 2 4 3 1 1 2 1 2 3 2 4
Offspring 1 1 3 2 1 2 1 2 3 2 4
Offspring 2 4 3 1 1 2 3 1 4 4 3

Figure 2. An example of one point crossover used in the GA algorithm

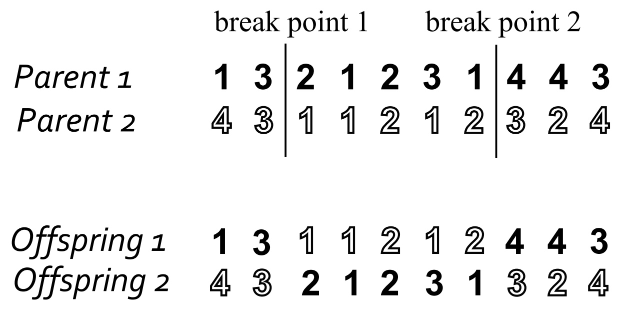
break point 1
break point 2
Parent 1 1 3 2 1 2 3 1 4 4 3
Parent 2 4 3 1 1 2 1 2 3 2 4
Offspring 1 1 3 1 1 2 1 2 4 4 3
Offspring 2 4 3 2 1 2 3 1 3 2 4

Figure 3. An example of two point crossover used in the GA algorithm

Mutation of the chromosomes is realized by selecting a random position in the array and changing its original value with a randomly selected value from a set of remaining docks.

4. COMPUTATIONAL EXPERIMENTS

In order to test this algorithm's effectiveness and efficiency for solving the TSchPRT we exploit the set of problem instances used in [6]. Three sets of problem instances are considered, each with different problem complexity. The first set consists of 10 trucks (small problem size), the second of 25 (medium problem size), and the third of 50 (large problem size) trucks. Each truck has randomly generated arrival time according to uniform distribution on the [0, 8] h interval, as well as randomly generated duration of service time uniformly distributed on the [0.4, 2.4] h interval. It is supposed that service time of a truck is the same for all docks. The number of available docks changes with the number of trucks in the planning horizon. Accordingly, there are four, six and ten available dock doors for the small, medium and large problem instances, respectively.

We observed planning horizon of 16 hours during which all, except first four docks, are available for service during [0, 16] h interval. The first dock is supposed to be available during [12, 16] h interval, the second during [8, 12] h interval, the third during [9, 16] h interval, and the fourth during [4, 14] h interval. Notice that first four docks are always used in serving trucks, regardless of problem size. Additional docks are used with the problem size increase according to the previously explained strategy.

Algorithm's termination criteria implied that search procedure is finished if after 100 steps there were not solution improvement. In every step we considered population of 100 solutions. Besides that, we supposed that probability of mutation is 0.1, and of crossover it is 0.95.

Because of the problem complexity only small size problem instances are solved to the optimality by using CPLEX 12.2's quadratic optimizer. In the cases of deterministic heuristics (DAH and Insertion) every instance is solved once, while in the cases of the VNS and GA every instance is solved ten times. All of the test runs were executed on a Windows XP OS powered by an AMD Phenom II 2.61 GHz processor with 1GB of RAM, while all of the coding was done in Python 2.5.

Results of conducted tests are summarized in Table 1 and Table 2. First three columns describe instances, precisely, ordinal number of an instance, number of inbound trucks to be allocated, and number of available dock doors. Column "Opt" in table 1 contain data about the objective value obtained for an instance by implementing CPLEX's quadratic optimizer. Column "DAH" contains data about the minimal objective function of an instance obtained by implementing DAH algorithm, proposed by Acer in [3]. Column "Insertion" in table 1 contain information on the minimal values of objective function for an instance obtained by implementing insertion algorithm, presented in [6]. Similarly, column "VNS" contains best out of ten solution objective values of implemented VNS algorithm, proposed also in [6]. Columns, "GA – 1opt" and "GA – 2opt" contain data about the best objective values related to the implementation of GA based algorithms presented in this paper.

Table 2 holds information on average time an algorithm requires for an execution. First three columns in table 2 have the same meaning as in the case of the table 1. Following six columns refer to the same algorithms, but now containing data about the average time required for obtaining a solution of the TschPTR.

Table 1. Minimal values out of ten obtained solutions

Inst	N° of trucks	N° of docks	Opt	DAH	Insertion	VNS	GA 1point	GA 2point
1	10	4	7.33	20.54	8.05	7.33	7.33	7.33
2	10	4	7.44	21.57	8.38	7.44	7.44	7.44
3	10	4	8.38	23.35	8.53	8.38	8.38	8.38
4	10	4	5.65	20.29	6.04	5.65	5.65	5.66
5	10	4	6.52	20.28	6.88	6.52	6.54	6.53
6	10	4	8.12	21.25	8.41	8.12	8.12	8.12
7	10	4	8.49	24.16	9.93	8.49	8.49	8.50
8	10	4	9.62	23.19	10.19	9.62	9.62	9.62
9	10	4	11.52	25.22	11.99	11.52	11.53	11.52
10	10	4	5.30	18.17	5.32	5.30	5.30	5.30
11	25	6		39.34	20.58	19.84	19.88	19.86
12	25	6		43.23	21.91	20.65	20.66	20.68
13	25	6		30.62	17.23	17.07	17.12	17.11
14	25	6		43.52	23.03	22.14	22.17	22.18
15	25	6		28.47	14.85	13.82	13.84	13.86
16	25	6		32.93	18.86	18.27	18.30	18.28
17	25	6		39.58	20.16	19.21	19.26	19.21
18	25	6		36.22	18.39	17.21	17.27	17.25
19	25	6		37.42	16.31	14.94	14.95	14.96
20	25	6		45.17	22.97	22.31	22.33	22.33
21	50	10		59.29	43.61	43.01	43.11	43.22
22	50	10		57.30	43.54	42.14	42.54	42.53
23	50	10		59.50	46.26	45.20	45.40	45.40
24	50	10		138.11	137.89	54.81	58.13	58.24
25	50	10		246.68	235.54	54.43	60.78	60.96
26	50	10		187.09	189.18	42.19	45.05	43.68
27	50	10		161.19	156.36	44.05	44.61	45.17
28	50	10		166.37	165.88	45.86	49.89	49.73
29	50	10		161.24	155.01	50.93	55.19	53.87
30	50	10		148.88	142.15	45.03	53.71	54.16

On the other side, time robustness of the GA caused decrease in the solution quality, compared to the VNS. Precisley, while the differences in solutions' objective values in the case of small and medioum problem instances are negligible, they significantly increased in the case of the large problem instances.

Table 2. Average algorithm run time in seconds

Inst	N° of trucks	N° of docks	Opt	DAH	Insertion	VNS	GA 1point	GA 2point
1	10	4	2426.2	<0.01	<0.1	0.30	1.37	1.35
2	10	4	3161.3	<0.01	<0.1	0.27	1.17	1.10
3	10	4	2719.4	<0.01	<0.1	0.26	0.97	1.12
4	10	4	2804.2	<0.01	<0.1	0.26	1.14	1.29
5	10	4	1867.7	<0.01	<0.1	0.23	1.34	1.08
6	10	4	2191.5	<0.01	<0.1	0.26	1.04	1.09
7	10	4	4441.9	<0.01	<0.1	0.29	1.27	1.12
8	10	4	2840.0	<0.01	<0.1	0.31	0.93	0.96
9	10	4	3241.1	<0.01	<0.1	0.35	0.98	1.13
10	10	4	2914.8	<0.01	<0.1	0.25	1.37	1.12
11	25	6		<0.01	<0.1	9.82	1.80	2.16
12	25	6		<0.01	<0.1	9.12	2.48	2.15
13	25	6		<0.01	<0.1	9.08	2.44	2.07
14	25	6		<0.01	<0.1	10.89	2.47	2.87
15	25	6		<0.01	<0.1	9.90	2.16	2.07
16	25	6		<0.01	<0.1	8.73	2.57	2.49
17	25	6		<0.01	<0.1	10.50	1.77	2.07
18	25	6		<0.01	<0.1	10.22	2.76	2.28
19	25	6		<0.01	<0.1	11.58	2.39	2.45
20	25	6		<0.01	<0.1	11.36	2.35	2.32
21	50	10		<0.01	<0.1	221.14	5.55	6.70
22	50	10		<0.01	<0.1	300.09	7.55	8.07
23	50	10		<0.01	<0.1	236.69	6.91	7.18
24	50	10		<0.01	<0.1	224.78	10.17	9.08
25	50	10		<0.01	<0.1	313.02	11.43	10.17
26	50	10		<0.01	<0.1	223.79	11.81	9.70
27	50	10		<0.01	<0.1	200.45	9.06	8.98
28	50	10		<0.01	<0.1	215.51	12.14	8.75
29	50	10		<0.01	<0.1	246.69	9.08	8.81
30	50	10		<0.01	<0.1	266.70	12.85	10.25

Therefore, based on obtained results is can be concluded that implemented encoding may be very usefull for solving large instances of the TSchPTR. However, in order to improve solution quality of large instances an effort has to be made in direction of finding more appropriate crossover operators. Accordingly, the following research related to the TSchPTR will be pointed in that direction.

ACKNOWLEDGMENT

This work was supported by the Ministry of Education and Science of the Government of the Republic of Serbia through the project TR36006, for the period 2011-2014.

REFERENCES

[1] Van Belle J, Valckenaers P, Cattryse D, 2012 *Cross-docking: State of the art*, Omega 40, 827-846.

[2] Boysen N., Fliedner M., 2010. *Cross dock scheduling: Classification, literature review and research agenda*, Omega 38, 413-422.

[3] Acar M K., 2004. Robust dock assignments at less-than-truckload terminals. Master's thesis, University of South Florida, 2004.

[4] Acar M K., Yalcin A., Yankov D., 2012, *Robust door assignment in less-than-truckload terminals*, Computers & Industrial Engineering 63,729–738.

[5] Soltani R, Sadjadi SJ, 2010. Scheduling trucks in cross-docking systems: A robust meta-heuristics approach, Transportation Research Part E 46, 650-666.

[6] Bjelić N., Radivojević G., Popović D., and Ratković B., 2012, *Analysis of Neighborhood Structures in the Variable Neighborhood Search Algorithm for the Truck Scheduling Problem with Time Robustness*, Proceedings of The XX International Conference on Material Handling, Constructions and Logistics – MHCL'12, Faculty of Mechanical Engineering, 3.-5. October 2012.,Belgrade, Serbia, 231-236.