Optimization of Logistic Distribution Centers

Process Planning and Scheduling

Jan Karasek, Radim Burget, Vaclav Uher
Faculty of Electrical Engineering and Communication,
Brno University of Technology
Brno, Czech Republic.
{karasekj, burgetrm}@feec.vutbr.cz
vaclav.uher@phd.feec.vutbr.cz

Malay Kishore Dutta, Yogesh Kumar
Amity School of Engineering and Technology
Amity University,
Noida, Uttar Pradesh, India.
mkdutta@amity.edu, ykumar2@amity.edu

***Abstract*— This paper describes a novel method for solving the problem of automatic planning and scheduling of work-plans in logistic distribution centers. The solution of the problem is based on well-known scheduling problems such as Job-Shop Scheduling Problems or Vehicle Routing Problems. By the time of writing this article, the key representatives of the logistics and warehousing industry do not use fully automated processes for work scheduling. The purpose of this paper is to connect the scientific result with demands of the companies in logistics and warehousing industry. The main contribution of this paper is a) to describe the motivation for solving the problem of logistic and warehousing companies, b) to describe a set of benchmarks and to give the reference layout of the warehouse, and c) to present a baseline results obtained by a genetic programming.**

Keywords*— benchmark definition, genetic programming, logistic warehouse, optimization, process planning & schedullin.

I. INTRODUCTION

The operational level of management competences such as planning and scheduling of routine tasks are important skills for each logistic and warehousing company. The problem in its principle is not that difficult and can be handled by skilled operational manager or with help of any method of mathematical programming. The problem arises when the company has dozens of employees and big amount of commodities and goods to handle. The problem can become very large and complex, which disqualifies the methods of mathematical programming from getting the competitive result in reasonable amount of time. By the time of writing this article, the significant representatives of logistics and warehousing industry do not use the fully automated method of process scheduling, but they are aware of the necessity of automated method. The implementation of such method could rapidly increase the competitiveness of the company. Therefore, this work has been created in cooperation with the prominent consultants of one significant EU logistic company.

The planning is defined as the process of identifying all activities necessary to complete the project. The scheduling is defined as the process of determining the sequential order of tasks, assigning planned duration and determining the start and finish of each task. Moreover, we consider the scheduling process also as the assigning the tasks to the concrete employee. The problem of warehouse process planning and scheduling is also related to Job-Shop Scheduling and Vehicle Routing, but the problem described in this paper is far more complex, because it combines these two problems and contains many other parameters of logistic warehouse such as attributes related to performance of the employees and/or attributes which describes the equipment, logistic warehouse layout, processed tasks and others. The literature [1] refers to the time spent on each operation in warehouse, which can be divided into a) travelling (50%), searching (20%), picking (15%), setup (10%), and other unpredictable circumstances (5%). This suggests that travelling is the most significant part of time and it is potentially the best place for further optimization.

The main contribution of this paper is a) to describe the problem of logistic and warehousing companies, b) to describe a set of benchmarks and to give the reference layout of the warehouse, and c) to present a baseline results obtained by a genetic programming algorithm.

The rest of the paper is structured as follows. The section II describes the related work for this paper. Furthermore, the section II describes the motivation of this work. The set of benchmarks are described in the section III as well as the reference layout of the warehouse. The section IV presents the genetic programming method. The section V presents baseline results and conclusion can be found in section V.

II. RELATED WORK

A. Problem Motivation

The problem described in this paper deals with planning and scheduling of complex logistic warehouses and distribution centers. There are two main motivation factors of this work: a) there is a significant demand from industry to solve the problem by fully automated method, and b) to set the strong cooperation between the science in the Brno University of Technology and significant representatives of logistics and warehousing industry.

B. Related Work

The problems addressed within the logistics and warehousing industry are related mostly to the optimization of some part of the warehouse layout, design of receiving and shipping areas and design of other manipulation areas in the warehouse. The commodities and goods stored in the warehouse are also often a subject for optimization, e. g. optimization of product groups, product classes and/or zoning for placing the commodities and goods. Please see the [2] and [3] for more references and information.

There are two main approaches for problem solving. The first approach is to use some method of mathematical programming [4], which is good for smaller optimization cases. The second approach is to use some method of heuristics, which is better when the problem becomes more complex, and it is too time-demanding (nearly impossible) to try every possible solution to get the global optima. The heuristic approaches represent the methods which are not trying every possible solution, but they go to specific direction which could lead to the optimal or at least sub-optimal solution. Heretofore, the large amount of methods has been tested, e.g. shifting bottleneck [5], dispatching rules [6], simulated annealing [7], particle swarm optimization methods [8], tabu search methods [9], and of course evolutionary algorithms [10].

The most prevalent are evolutionary algorithms. Genetic algorithms, as the representative of the group of evolutionary algorithms have been used the most. The problem of use of Genetic algorithm in our case is that it is needed to build the structure of chromosome from scratch. The structure is not given by any prescription. That is why the Genetic programming seems to be the best way to get the optimal result. Genetic programming has demonstrated its potential for problem solving in complex and dynamic situations where other algorithms fail. In recent years, the Genetic programming has been successfully applied in many areas, e.g. in [11] image detector for common carotid artery in ultra sound images was successfully designed. The paper [12] presents the use of genetic programming in optimal feature selection to classify emotions from textual data, and [13] was designed the non-cryptographic hash function by genetic programming.

III. LAYOUT AND BENCHMARK DEFINITON

This section describes the reference layout of logistic warehouse which models the real-world situation. Besides, this section also describes the benchmark definition which is used as a metric for performance measurement.

A. Layout Definition

The layout of the warehouse represents the model of real-world situation. The model is based on consultations with expert from logistic industry. The warehouse described is 2D matrix of size 21 x 21 cells. There are 10 columns of racks and each column of rack has 19 racks standing next to each other. Every single rack has 10 shelves one above another to store pallets of commodities and goods. The level 0 indicates that the pallet is standing on the floor. The warehouse is divided into equal cells (as mentioned 21 x 21). Each cell has the same width and height as a single rack. The place before the rack (aisle) is also divided into same cells. This is done, because it is essential to know how many cells have to be exceeded from the beginning of the tasks to its finish.

The model of the warehouse is depicted in Fig. 1. The model consists of several parts, such as a) trucks importing and exporting commodities, goods, b) the receiving and shipping areas of warehouse, c) bidirectional warehouse gates, d) space and offices of employees, e) hand pallet trucks (fork-lift hand) which are able to store a commodity only to the level 0, f) fork-lift low truck which is able to store a commodity from the level 0 to level 2, g) fork-lift high truck which is able to store a commodity from the level 0 to the level 9 (all shelves).

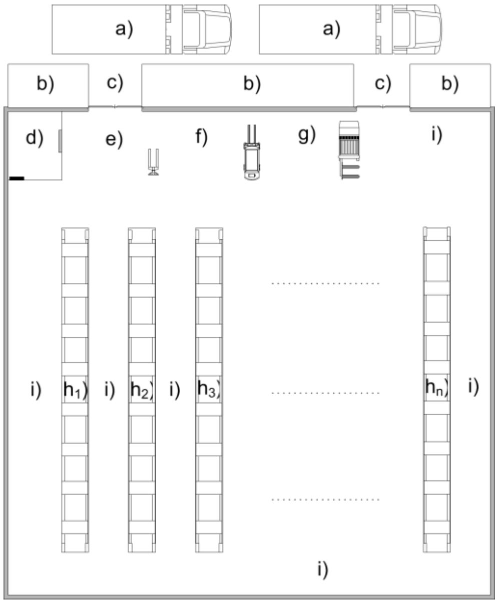

Fig. 1. Reference Layout Definition of Logistic Warehouse.

B. Benchmark Definition

The initial results have been obtained by measurements of two sets of benchmarks. The first set of benchmarks consists of 10 simple scenarios. Each scenario processes several tasks spread among 2-4 employees. The main difference between scenarios is in the tasks processed, in coordinates of start and end position, in paths used to process the task, in potential collision places, and in that how they are spread among the employees. Each employee has associated an equipment to process the task, which is the hand pallet truck in these cases. Each employee has to fulfill the task from the beginning to the end. Another 10 examples represent the same scenarios. The difference here is that employees can use hand pallet truck and fork-lift low truck as well. The fork-lift low truck has different speed which means that the scenario will be processed differently. The collisions of trucks, performance of the employees are not taken into account during the evaluation.

The *task* in this context represents the single assignment given to the employee by operational manager e.g. the employee has to unload the pallet from a lorry, go through the warehouse and store it in the shelf. The task is composed of jobs. The *job* represents a single operation of task e.g. unloading, moving and/or storing.

The fitness function is the minimization function of the time needed to process whole buffer of tasks. The final time (fitness value) is given by the finish time of last task in scenario. Time T is calculated as the quotient of path length in cells S and speed of vehicle in units of speed R.

IV. GENETIC PROGRAMMING METHOD

Genetic programming is a domain-independent problem solving approach in which computer programs are evolved to solve, or approximately solve, problems. The basic element of genetic programming data structure is *gene*. Gene in this context is a set of tasks, also called work-plan, meant for one single employee. The *chromosome* is composed of bunch of genes. It means that the chromosome represents work-plans for all employees – the buffer of all tasks assigned to be processed by concrete employees.

The proposed genetic programming algorithm is depicted in Fig. 2. The figure represents the computational core of the proposed automated system for process planning and scheduling. The input of algorithm is the buffer of tasks, and set of employees with equipment assigned (fork-lift trucks). The first step of algorithm is to create the initial population (task and particular jobs are used for *gene* creation, genes are then used for *chromosome* composing and then the whole initial population is created).

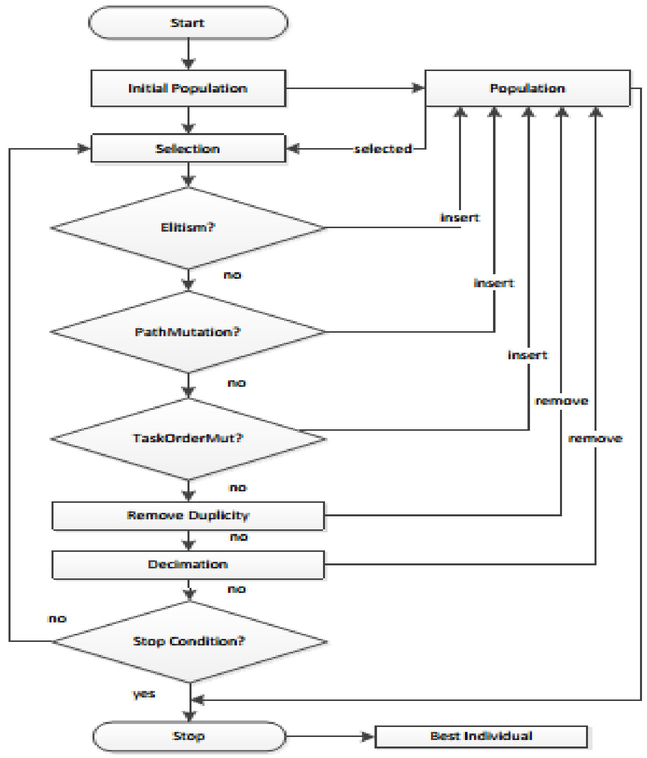

Fig. 2. Genetic Programming Algorithm.

The next step is the evolution process which is driven by parameter which determines the number of evolution steps, which is also considered as a stop condition. The first part of evolution process is a tournament selection, which is used for selecting individuals for further processing. Then a few conditions follow. The first condition represents the elitism used. If the condition is true, the certain number of individuals is copied to the next population. The second condition is the use of path mutation operator which mutates the paths of trucks through the warehouse. The third condition is the use of task order mutation which makes the schedule for every single employee. Mutation operators are basic operators which help to automated system find an optimal solution. The paths are mutated because of collisions of vehicles and the task order mutation is used due to schedule creation.

V. BASELINE RESULTS & DISCUSSION

The baseline results depicted in the following tables have been obtained by testing of proposed algorithm in previous section on each scenario in benchmark set. The algorithm has been using the path mutation, task order mutation and their combination. The results measured are compared with human operator design solution for each scenario.

The initial setting of genetic programming algorithm has been set as follows:

- *Number of individuals* = 10
- *Number of generations* = 10
- *Rate of mutations* has been tested (see the Tables I-VI)
- *Tournament selection* has been used in all cases

Table I and Table II present the results of a system which has been using only one genetic operator – path mutation. Table III and Table IV present the results of a system which has been using only task order mutation operator. Table V and Table VI present the results of a system which has been using both of the operators with the same rate of mutation in each measurement. The measurements have been carried out for mutation rates form 5% to 30% with step of 5%, and from 30% to 90% with step of 10%. The benchmark scenarios are quite simple, which means that the improvement in performance cannot be the significant value, but still the performance improvement is measurable and obvious in particular scenarios. It means that the proposed genetic programming algorithm is competitive to human operator design of scheduling scenario.

TABLE I. RESULTS FOR PATH MUTATION ONLY, CASE 1-10

Rate of Mutation	**# of scenario**									
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
Human operator	13.00	16.50	13.00	12.50	14.50	15.00	15.00	09.00	13.00	16.50
5%	13.00	16.50	**29.50**	**16.50**	12.50	**26.50**	15.00	08.00	12.50	16.00
10%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	12.50	16.00
15%	**15.50**	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	**14.00**	16.00
20%	13.00	16.50	**31.50**	**16.50**	12.50	**26.50**	15.00	08.00	13.00	16.00
25%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	**14.00**	16.00
30%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	13.00	16.00
40%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	12.50	16.00
50%	13.00	16.50	**28.50**	**16.50**	12.50	**28.00**	15.00	08.00	12.50	16.00
60%	13.00	16.50	**28.50**	**16.50**	12.50	**28.00**	15.00	08.00	12.50	16.00
70%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	**14.00**	16.00
80%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	12.50	16.00
90%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	12.50	16.00

TABLE II. RESULTS FOR PATH MUTATION ONLY, CASE 11-20

Rate of Mutation	# of scenario 1	2	3	4	5	6	7	8	9	10
Human operator	08.00	14.00	13.00	14.00	11.00	14.50	15.00	08.50	13.00	16.50
5%	08.00	**14.50**	**17.88**	14.00	11.00	**15.00**	11.50	08.00	12.00	12.13
10%	08.00	**14.50**	13.00	12.50	11.00	**15.00**	11.50	08.00	12.00	12.13
15%	08.00	**14.50**	**14.00**	13.25	**11.50**	**15.00**	11.50	08.00	12.00	12.13
20%	08.00	**14.50**	**20.88**	12.25	11.00	**16.50**	11.50	08.00	12.00	13.00
25%	08.00	**14.50**	**14.00**	12.50	11.00	**16.50**	11.50	08.00	12.00	12.13
30%	08.00	**14.50**	13.00	12.50	11.00	**15.00**	11.50	08.00	12.00	13.00
40%	08.00	**14.50**	13.00	12.50	11.00	**15.00**	11.50	08.00	12.00	12.13
50%	08.00	**14.50**	**13.88**	12.50	11.00	**15.00**	11.50	08.00	12.00	13.00
60%	08.00	**14.50**	**13.88**	12.50	11.00	**15.00**	11.50	08.00	12.00	13.00
70%	08.00	**14.50**	13.00	13.25	11.00	**16.50**	11.50	08.00	**13.13**	13.50
80%	08.00	**14.50**	**15.50**	12.50	11.00	**15.00**	11.50	08.00	12.00	13.00
90%	08.00	**14.50**	**14.63**	12.50	11.00	**15.00**	11.50	08.00	12.00	12.13

Other important parameter used for measurements presented in this paper is speed of truck. The fork lift hand speed has been set to 2 [speed units], and fork lift low truck speed has been set to 8 [speed units]. The path is defined in cells. The time of processing is in [time units]. Specific time units as seconds or minutes have not been used. The common time units used in this measurement can be transformed, but it is not relevant for the purpose of this paper.

All the scenarios have been measured 10 times and the values in Table I – Table VI represent the average value of all measurements. The scenarios designed by human operator are highlighted by dark gray row. These scenarios are designed to be optimal as much as possible when it is designed by human, which is why there is no significant progress by automated system. In spite of that there is almost no space for optimization because human operator is the expert in scheduling, the automated system proved its quality by giving the better solution in some cases at least in small units of time measurement. On the basis of presented baseline results it can be supposed that the proposed automated system can give the better solutions for more complex scenarios in the future.

Values in the tables highlighted by bold font represent the cases which showed the worst performance than human operator. The worst performance is given by the fact that only 10 individuals has been used and the diversity of population decreased rapidly even if it has been used only 10 generations. This is suggestion for further measurements to set up the optimal number of individuals and generations. The second suggestion is to avoid the number of generations and instead of this end condition use the demanded accuracy of final solution. The end condition could be set as the number of scenarios which proved the better performance by automated system then the human operator. This could be set e.g. to 90%. So, in the case of Table I it could be set to 9 scenarios designed by automated system with better performance than human.

The values in tables which are not highlighted show the same performance as the human operator. Even though that there is no performance increase, the solution is successful, because the automated system reached the same performance as the operator. Values highlighted by light gray shows the better performance than human operator. The rate of mutation highlighted by light gray shows the best rate used for further processing. The best rate was selected as the first percentage which gave at least one better result than human operator and previous rate, e.g. in Table I was selected 10% rate, because the 4th case has better performance than 5% rate. Besides that, the 5% rate shows the worst performance in case no. 3.

TABLE III. RESULTS FOR TASK ORDER MUTATION ONLY, CASE 1-10

Rate of Mutation	# of scenario 1	2	3	4	5	6	7	8	9	10
Human operator	13.00	16.50	13.00	12.50	14.50	15.00	15.00	09.00	13.00	16.50
5%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	**15.00**	08.00	12.50	16.50
10%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	**15.00**	08.00	12.50	16.00
15%	13.00	16.50	**31.50**	**16.50**	12.50	**26.50**	**15.00**	08.00	12.50	16.00
20%	13.00	16.50	**37.00**	**16.50**	12.50	**26.50**	**15.00**	08.00	12.50	16.00
25%	13.00	16.50	**31.50**	**16.50**	12.50	**26.50**	**15.00**	08.00	12.50	16.00
30%	13.00	16.50	**28.50**	**18.50**	12.50	**26.50**	**15.00**	08.00	12.50	16.00
40%	**15.50**	16.50	**28.50**	**16.50**	12.50	**26.50**	**15.00**	08.00	**14.00**	16.00
50%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	**15.00**	08.00	12.50	16.00
60%	**15.50**	16.50	**28.50**	**16.50**	12.50	**26.50**	**15.00**	08.00	12.50	16.00
70%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	**15.00**	08.00	12.50	16.00
80%	**15.50**	16.50	**28.50**	**16.50**	12.50	**26.50**	**15.00**	08.00	**14.50**	16.00
90%	13.00	16.50	**29.50**	**16.50**	12.50	**26.50**	**15.00**	08.00	**14.00**	16.00

TABLE IV. RESULTS FOR TASK ORDER MUTATION ONLY, CASE 11-20

Rate of Mutation	# of scenario 1	2	3	4	5	6	7	8	9	10
Human operator	08.00	14.00	13.00	14.00	11.00	14.50	15.00	08.50	13.00	16.50
5%	08.00	**14.50**	13.00	12.75	11.00	**16.50**	11.50	08.00	12.00	12.13
10%	08.00	**14.50**	**14.00**	12.25	11.00	**15.00**	11.50	08.00	12.50	12.13
15%	08.00	**14.50**	13.00	12.75	11.00	**15.00**	11.50	08.00	12.00	13.00
20%	08.00	**14.50**	**20.88**	12.25	11.00	**16.50**	11.50	08.00	12.00	13.00
25%	08.00	**14.50**	**14.00**	14.00	11.00	**15.00**	11.50	08.00	12.00	13.00
30%	08.00	**14.50**	**15.50**	12.75	11.00	**15.00**	11.50	08.00	12.00	12.13
40%	08.00	**14.50**	**16.50**	12.25	11.00	**15.00**	11.50	08.00	12.00	13.00
50%	08.00	**14.50**	**14.63**	12.75	11.00	**15.00**	11.50	08.00	12.00	12.13
60%	08.00	**14.50**	**16.50**	12.25	11.00	**15.00**	11.50	08.00	12.50	12.13
70%	**11.00**	**14.50**	13.00	12.25	11.00	**16.50**	11.50	08.00	12.00	12.13
80%	08.00	**14.50**	**14.63**	12.25	11.00	**15.00**	11.50	08.00	12.50	12.13
90%	08.00	**14.50**	**15.50**	12.25	11.00	**15.00**	11.50	08.00	12.00	12.13

TABLE V. RESULTS FOR PATH & TASK ORDER MUTATION, CASE 1-10

Rate of Mutations	**# of scenario**									
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
Human operator	13.00	16.50	13.00	12.50	14.50	15.00	15.00	09.00	13.00	16.50
5%	13.00	16.50	**34.50**	**16.50**	12.50	**26.50**	15.00	08.00	**14.50**	16.00
10%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	12.50	16.00
15%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	12.50	16.00
20%	**15.50**	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	13.00	16.00
25%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	13.00	16.00
30%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	**14.00**	16.00
40%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	**14.50**	16.00
50%	13.00	16.50	**28.50**	**16.50**	12.50	**26.50**	15.00	08.00	**14.00**	16.00
60%	13.00	16.50	**28.50**	**18.50**	12.50	**26.50**	15.00	08.00	13.00	16.00
70%	13.00	16.50	**31.50**	**16.50**	12.50	**26.50**	15.00	08.00	12.50	16.00
80%	**15.50**	16.50	**31.50**	**16.50**	12.50	**26.50**	15.00	08.00	13.00	16.00
90%	13.00	16.50	**31.50**	**16.50**	12.50	**26.50**	15.00	08.00	12.50	16.00

TABLE VI. RESULTS FOR PATH & TASK ORDER MUTATION, CASE 11-20

Rate of Mutations	**# of scenario**									
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
Human operator	08.00	14.00	13.00	14.00	11.00	14.50	15.00	08.50	13.00	16.50
5%	08.00	**14.50**	**18.00**	12.75	11.00	**16.50**	11.50	08.00	12.00	14.00
10%	08.00	**14.50**	**13.38**	12.25	11.00	**15.00**	11.50	08.00	12.00	13.00
15%	08.00	**14.50**	**18.25**	14.00	11.00	**15.00**	11.50	08.00	12.00	12.13
20%	08.00	**14.50**	**13.88**	12.25	11.00	**17.13**	11.50	08.00	12.00	13.00
25%	08.00	**14.50**	**13.88**	12.25	11.00	**15.00**	11.50	08.00	12.50	13.00
30%	**11.00**	**14.50**	**16.50**	12.25	11.00	**15.00**	11.50	08.00	12.00	13.00
40%	**11.00**	**14.50**	**16.50**	12.25	11.00	**15.00**	11.50	08.00	12.00	13.00
50%	**11.00**	**14.50**	**13.38**	12.25	11.00	**16.50**	11.50	08.00	12.50	12.13
60%	08.00	**14.50**	**14.00**	12.25	11.00	**15.00**	11.50	08.00	12.50	12.13
70%	08.00	**14.50**	**14.00**	12.25	11.00	**15.00**	11.50	08.00	12.00	12.13
80%	08.00	**14.50**	**14.00**	12.25	11.00	**15.00**	11.50	08.00	12.50	12.13
90%	08.00	**14.50**	**14.63**	12.75	11.00	**15.00**	11.50	08.00	12.00	12.13

VI. CONCLUSION

This paper presents an automated approach to job-shop scheduling in logistic warehouses and distribution centers based on genetic programming algorithms. The main motivation of this work is to present the experimental results of genetic programming algorithm and to show the better performance of the proposed system on the simple scenarios extracted from real situations in warehouse. The optimal setting for mutation rates seems to be 10%. The level 10% is selected because it is the first level producing the better performance than operational manager and at least same performance as the other mutation rates tested. Other percentages also gained good or same results, but the problem is when the rate of mutation is increasing the more individuals have to be generated, which could cause the memory overflow problem. The further research will be focused on examination of the number of individuals, generations and other end conditions for genetic programming algorithm as well as investigation of other genetic programming operators which could help to improve the algorithm and reach the optimal solution faster.

ACKNOWLEDGMENT

This research work is funded by projects SIX CZ.1.05/2.1.00/03.0072, SPLab team CZ.1.07/2.3.00/20.0094, MPO FR-TI1/444, and project FEKT-S-11-17.

REFERENCES

[1] J. A. Tompkins, J. A. White, Y. A. Bozer, and J. M. A. Tanchoco, Facilities Planning, John Wiley & Sons, 4th edition, p. 846, 2010.

[2] R. de Koster, T. Le-Duc, and K. J. Roodbergen, “Design and control of warehouse order picking: A literature review,” European Journal of Operational Research, vol. 182, pp. 481–501, 2007.

[3] C. A. S. Garaldes, M. S. F. Carvalho, and G. A. B. Pereira, “A Warehouse design decision model – Case study,” IEEE International Engineering Management Conference,IEMC Europe,” pp.397–401, 2008.

[4] K. Bülbül, P. Kaminsky, “A linear programming-based method for job shop scheduling,” Journal of Scheduling, vol. 16, no. 2, pp. 161–183, 2013.

[5] J. Adams, E. Balas, D. Zawack, “The shifting bottleneck procedure for job shop scheduling,” Management Science, vol. 34, no. 3, pp. 391–401, 1988.

[6] K. Bhaskaran, and M. Pinedo, Dispatching, Handbook of Industrial Engineering, Wiley, 1992.

[7] P. J. M. van Laarhoven, E. H. L. Aarts, and J. K. Lenstra, “Job shop scheduling by simulated annealing,” Operations Research, vol. 40, pp. 113–125, 1992.

[8] M. Tasgetiren, Y.-C. Liang, M. Sevkli, and G. Gencyilmaz, “A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem,” European Journal of Operational Research, vol. 177, no. 3, pp. 1930–1947, 2007.

[9] E. Nowicki, and C. Smutnicki, “An advanced tabu search algorithm for the job shop problem,” Journal of Scheduling, vol. 8, pp. 145–159, 2005.

[10] M. Koksalan, and A. B. Keha, “Using genetic algorithm for single-machine bicriteria scheduling problems,” European Journal of Operational Research,” vol. 145, no. 3, pp. 543–556, 2003.

[11] R. Benes, J. Karasek, R. Burget, and K. Riha, “Automatically designed machine vision system for the localization of CCA transverse section in ultrasound images,” Computer Methods and Programs in Biomedicine, vol. 109, no. 1, pp. 92–103, 2013.

[12] R. Burget, J. Karasek, and Z. Smekal, “Recognition of Emotions in Czech Newspaper Headlines,” Radioengineering, vol. 20, no. 1, pp. 39–47, 2011.

[13] J. Karasek, R. Burget, and O. Morsky, “Towards an Automatic Design of Non-Cryptographic Hash Function,” 34th International Conference on Telecommunications and Signal Processing, Budapest, pp. 19–23, 2011.