International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 22 (2017) pp. 11913-11919

Optimization Algorithm for Minimizing the Earliness/Tardiness of Automated Guided Vehicles Using Artificial Immune System

Mahalakshmi
Research Scholar, School of Applied Sciences,
REVA University, Rukmini Knowledge Park, Yelahanka, Bangalore, Karnataka, India.
Orcid 0000-0002-5989-6222

Murugesan R
Professor, School of Applied Sciences,
REVA University, Rukmini Knowledge Park, Yelahanka, Bangalore, Karnataka, India.
Orcid 0000-0002-0698-4763

Abstract
This paper regards with the job shop environment problem for multiple Automated Guided Vehicles (AGVs) in the manufacturing process. As the earliness of the AGVs results in waiting and tardiness of AGVs results in temporary storage of the products in the shop floor, it is essential to minimize the earliness and tardiness of AGVs. Therefore we are proposing a mathematical optimization program in order to minimize the total earliness and total tardiness of AGVs in the manufacturing system. Since it is difficult to solve the mathematical program by conventional method, an optimization technique namely Artificial Immune System (AIS) algorithm is used to obtain the optimal solution. The proposed algorithm is tested with the numerical examples and also compared with the other methods in the literature.

Keywords: Flexible Manufacturing System, Automated Guided Vehicles, Earliness/Tardiness, Artificial Immune System.

INTRODUCTION
Moving semi-finished products from one shop to another shop and fully finished products from shops to ware house may be done by using cranes trucks, employees etc., which causes increase in the cost function of the manufacturing process. The new trend is to use Automated Guided Vehicles (AGVs), which increases flexibility and efficiency in the manufacturing process by reducing the cost function and make span. Manufacturing systems containing the AGVs are called as Flexible Manufacturing Systems (FMS). AGVs also plays a vital role in warehousing systems, container terminals, service industries and hospital managements. AGVs are material handling machineries, which travels on the network of guided path with a constant speed. These AGVs can travel in both forward and backward positions. Since simultaneously many vehicles are travelling in the same guided path, the collisions have to be avoided. A general review of AGVs are given in the literature [1, 2, and 3]. The recent works on AGVs scheduling and routing problems is given in the survey of [4]. An extensive review on operational control of AGVs are addressed in [5]. The conflict free route of AGV in a directional network is found by [6], which is based on finding the shortest path using the concept of time window graph. The petri – net model is also used to find the conflict free routing of AGVs [7 and 8]. An intelligent framework for conflict free shortest time path is presented by [9]. A mathematical program for routing problem of AGVs using Lagrangian decomposition techniques is provided in [10]. The routing in a mesh topologies and conflict free scheduling is given by [11]. A heuristic method of two stage traffic control scheme of AGVs is given by [12]. The minimization of make span using an optimization approach by assuming assignment tasks are given to AGVs is done in [13]. The dispatching and routing problems using heuristic method separately by performing scheduling first and in the next step the sequential path generation (SPG) is done by [14]. The combined problem of dispatching and also the conflict free routing of AGVs with two vehicles is proposed in [15]. An extension this work with four vehicles was proposed by [16]. The conflict free routing using ant colony algorithm is given by [17]. By the above literature survey, it is found that there is gap of finding routing of AGVs along with minimization of Earliness and tardiness using Artificial Immune System (AIS) algorithm method. Artificial Immune System (AIS) [18, 19, 20 and 21] is the most interested new research field in artificial intelligent system. The major function of biological immune system is to save our body from the foreign microbes such as viruses, bacteria, pollen grains etc. These foreign microbes are called as Antigens. To fight with the antigens immune system will select the antibodies. The modern immunology tells that there are different cells which has various functions in our immune system. The antibody has idiotope, paratope and epitope. The idiotope acts as antigens. The function of paratope is to recognize the other antibodies and antigens. Epitope, which is recognized by other antigens and antibodies. Upon the selection of antibodies, the immune system best matches the antigen with the antibody to test the affinity of the antibody to

destroy the antigens. If the affinity value of the antibodies is less than the antigens, the antibodies undergo cloning of cells which is collectively called as clonal selection principle [22]. This methodology is applied to solve various problems in the research field. Therefore, in this paper we are also applying AIS algorithm to minimize earliness and tardiness of AGVs.

MATHEMATICAL MODEL

To study the proposed algorithm, we consider a job shop environment problem of multiple AGVs for the material handling. Sometimes the AGV guided path may be busy at the time of sending an AGV for material handling. Therefore, we need to obtain a free path for AGV movements. In the manufacturing process, the processing time of all jobs are planned clearly. If an AGV reports early to the shop floor, it has to wait until the processing is finished. This waiting time is regards with the due date of the jobs. This may delay the delivery of the product outlets in the manufacturing process. An essential driver of the manufacturing industry is to deliver the products in the correct time. Therefore, in this paper we are computing the manufacturing schedule of AGVs along with routing in order to minimize the total earliness/ total tardiness of AGVs in the job shop environment. The following assumptions are made before modelling the problem

- Number of jobs, number of AGVs and the number of shops are given earlier.
- Processing time of each job is specified clearly.
- At a time, an AGV can transport only a single load.
- The velocities of all AGVs are same

The mathematical formulation for the proposed algorithm is given as follows

Indices

i	Number of jobs, $i = 1, 2 \dots m$
j	Number of AGVs, $j = 1, 2 \dots n$
s, s'	Number of Shops, $s = 1, 2 \dots\dots o$ and $s' = 2, 3 \dots\dots o$

Parameters

WT_e	Weight of total earliness of the job, $WT_e \geq 0$
WT_d	Weight of total tardiness of the job, $WT_d \geq 0$
P_{is}	Processing time of the job i processed in the shop s
du_{is}	Due date for job i being processed in shop s
$d_{ss'}$	Distance between the shops s and s'
V_e	Velocity of the AGV (equal for all AGVs)
$t_{jss'}$	Travelling time for an AGV j between the shops s and s'
E_{is}	Earliness of the job i processed in shop s
T_{is}	Tardiness of the job i processed in shop s
C_{is}	Completion time of job i in shop s
A_{is}	Allocation of job i to shop s

$$A_{is} = \begin{cases} 1 & \text{if job } i \text{ is allocated to shop } s \\ 0 & \text{otherwise} \end{cases}$$

Decision Variables

$$x_{ijs} = \begin{cases} 1 & \text{if job } i \text{ is assigned to AGV } j \text{ in shop } s \\ 0 & \text{otherwise} \end{cases}$$

$$y_{jss'} = \begin{cases} 1 & \text{if an AGV } j \text{ moves between the shops } s \text{ and } s' \\ 0 & \text{otherwise} \end{cases}$$

Objective Functions

The objective function is to minimize

$$z = \sum_{s=1}^{o} \sum_{i=1}^{m} (WT_e . E_{is} + WT_d . T_{is}) \tag{1}$$

Where

$$WT_e = \sum_{s=1}^{o} \sum_{i=1}^{m} |P_{is} - du_{is}|^2 \quad and \quad WT_d = \sum_{s=1}^{o} \sum_{i=1}^{m} |P_{is} - C_{is}|^2 \tag{2}$$

Constraints

- For each s^{th} shop exactly one AGV is scheduled

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{s=1}^{o} x_{ijs} = 1 \tag{3}$$

- If an AGV is assigned to one shop, then the path between that shop and the immediate next shop is busy

$$x_{ijs} \cdot y_{jss'} = 1 \tag{4}$$

Where $i = 1, 2 \dots m, j = 1, 2 \dots n,$ $s = 1, 2 \dots\dots o$ and $s' = 2, 3 \dots\dots o$

- Only one AGV is assigned to each shop

$$\sum_{j=1}^{n} \sum_{s=1}^{o} \sum_{s'=2}^{o} y_{jss'} = 1 \tag{5}$$

- In a job shop allocation only one AGV is moved in the path between any two nodes

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{s=1}^{o} y_{jss'} \cdot A_{is} = 1, \qquad s' = s+1 \tag{6}$$

- In job shop allocation only one AGV is assigned

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{s=1}^{o} x_{ijs} \cdot A_{is} = 1, \qquad s' = s+1 \tag{7}$$

- The processing times of the jobs in the shops is given by

$$P_{is} = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{s=1}^{o}\sum_{s'=2}^{o} x_{ijs} \cdot t_{jss'} \tag{8}$$

- The relationship between completion time of the job in previous shop and processing time of the job in the next shop is given by

$$C_{is} \geq C_{is-1} + P_{is} \quad ,i = 1,2 \ldots m ,s = 1,2 \ldots\ldots o \tag{9}$$

- The tardiness of the job is given by

$$T_{is} = \max\{t_{jss'} , C_{is} - du_{is}\} \tag{10}$$

Where $i = 1,2 \ldots m, j = 1,2 \ldots n,$
$s = 1,2 \ldots\ldots o \text{ and } s' = 2,3 \ldots\ldots o$

- The Earliness of the jobs is given by

$$E_{is} = \max\{t_{jss'} , du_{is} - C_{is}\} \tag{11}$$

Where $i = 1,2 \ldots m, j = 1,2 \ldots n,$
$s = 1,2 \ldots\ldots o \text{ and } s' = 2,3 \ldots\ldots o$

Implementation of Artificial Immune System (AIS

In natural immune system antigens refers to the foreign elements such as bacteria, viruses, and pollen grains, which are collectively called as pathogens. And the antibodies are T-lymphocytes and B- lymphocytes cells within the human body which fights against the antigen. In the proposed algorithm antibody is the potential solution generated initially and the antigens are the best solution which is generated by the AIS. The building block of antibodies are genes. The antibody is represented by sequence of integers in AIS. Here each integer represents a gene of antigen or antibody in AIS. The total number of genes in a single antibody is equal to "number of jobs × number of AGVs × number of shops". The other component of AIS are antigen library, antibody library and final solution library. Antigen library is made by potential solutions which is can be considered for the next generation. After cloning and mutation of antigens, an antibody library is generated. The resulting best solutions are grouped as final solution library. From this final solution library, the required final schedule is selected. To recognize the best solution a threshold value is described for fitness values. If the fitness value is less than the threshold value of the solution, then the solution is retained otherwise the solutions are discarded from the system. Threshold is a specified value which separates good and bad solution. Fitness function and objective functions are defined as follows.

$$fitness = \max(z)$$

$$objective = \min(maz\ (z))$$

We have taken test problem to study the proposed algorithm which is given in the table 1

Table 1: Test problem for proposed algorithm

Parameters	**Values**
Number of jobs	45
Number of AGVs	12
Number of shops	17

The steps involved in the proposed AIS algorithm is given by the following steps.

Step 1: Define the problem size and threshold, initialize the random population of strings (represents the possible solution) up to the specified population which becomes an antigen library.

Step 2: We find the objective function value (OFV) and also the affinity value of each string using the relation

$$Affinity = 1/OFV$$

By the above equation affinity value increases as the OFV decreases. Depending on the value of affinity of each string the selection process is done.

Step 3: The rate of cloning of each string is calculated using the formula

$$Rate\ of\ cloning = affiniy\ value \times population\ size$$

Step 4: Each cloned cell undergoes mutation in terms of inverse mutation and pair wise mutation.

Step 4a: Inverse Mutation:

We have to generate some random positions between (1, n) of the original string, where n is the number of AGVs in the manufacturing process. Then we have to select two positions randomly for mutation, let it be 1 and 5. Then inverse mutation is obtained by reversing the order of the sequence between the positions 1 and 5 as given below

Original string →	5	1	6	2	7	9	4	8	3	10
Mutated string ←	7	2	6	1	5	9	4	8	3	10

After inverse mutation, OFV of the mutated string is calculated. If the OFV of the mutated string is less than the original string, then the original string is replaced by mutated string, otherwise pair wise mutation is done for the original string.

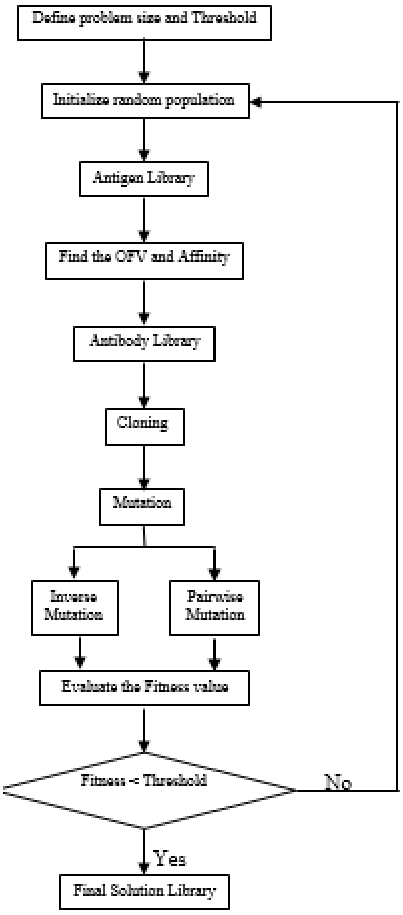

Figure 1: Flow chart of proposed algorithm

Step 4b: Pair wise Mutation:
We have to generate some random positions between (1, n) of the original string, where n is the number of AGVs in the manufacturing process. Then we have to select two positions randomly for mutation, let it be 1 and 5. Then pairwise mutation is obtained by reversing the order of the sequence between the positions 1 and 5 as given below

Original string → 5 1 6 2 7 9 4 8 3 10
Mutated string ← 7 1 6 2 5 9 4 8 3 10

After pair wise mutation, OFV of the mutated string is calculated. If the OFV of the mutated string is less than the original string, then the original string is replaced by mutated string, otherwise the original sequence is retained as it is.

Step 5: After cloning and mutation process, we reselect the improved strings to maintain the original population. The solution with highest OFV in the population is replaced by randomly generated solutions with lowest OFV, which forms an antibody library.

Step 6: Evaluate the fitness value of the antibodies.

Step 7: If the fitness value is less than the threshold value, select and move the solution to Final solution library, otherwise go to step 1.

Step 8: The Steps 1-7 is repeated for multiple runs till we get a desired solution.

Flow chart for the proposed algorithm is given in the figure 1.

RESULTS AND DISCUSSION

To study performance of the proposed algorithm, it must be simulated in MATLAB R2012a (7.14.0.739), 64–bit (win64). The entire tests were executed in Intel i5 processor under the Microsoft windows 7(64-bit) operating system. The obtained variables are given in the table 2 and a comparison analysis of the proposed algorithm with the other existing algorithm is given in the table 3 as well as in figure 2.

Table 2: Variable of the solution

x_{312}	x_{1191}	x_{1345}	x_{1536}	x_{3187}	x_{25610}	y_{234}	y_{3410}	y_{2212}	y_{5133}	y_{3102}	y_{71013}
x_{4415}	x_{526}	x_{1085}	x_{24126}	x_{23611}	x_{36111}	y_{1124}	y_{1233}	y_{3211}	y_{8103}	y_{12174}	y_{9812}
x_{2673}	x_{1264}	x_{22112}	x_{3177}	x_{1728}	x_{1910}	y_{457}	y_{10112}	y_{2113}	y_{677}	y_{81714}	y_{71614}
x_{7314}	x_{21813}	x_{35916}	x_{3719}	x_{341012}	x_{20210}	y_{4138}	y_{821}	y_{1117}	y_{6817}	y_{5911}	y_{6165}
x_{9102}	x_{6107}	x_{1438}	x_{16118}	x_{41916}	x_{38122}	y_{1169}	y_{1410}	y_{3159}	y_{8610}	y_{9154}	y_{16126}
x_{12123}	x_{8613}	x_{28517}	x_{4044}	x_{2739}	–	y_{456}	y_{4119}	y_{5616}	y_{6141}	y_{51516}	–
x_{2051}	x_{301214}	x_{39119}	x_{4445}	x_{18517}	–	y_{2135}	y_{732}	y_{1815}	y_{7145}	y_{81015}	–
x_{29114}	x_{32715}	x_{42210}	x_{4511}	x_{4553}	–	y_{9128}	y_{1178}	y_{1297}	y_{1076}	y_{1111}	–

Table 3: Comparison table of the proposed algorithm

Methods	Approaches	Size of the problem	Optimal Value	Run time(min)
Gerstl and Mosheiov [23]	Agent based	10*5*5	2327	145
		20*10*15	NFS	6000
		45*12*17	NFS	6000
Tanaka and Araki [24]	Lagrangian relaxation	10*5*5	2401	77
		20*10*15	6417	3401
		45*12*17	NFS	6000
Kianfar and Moshlehi [25]	Branch and bround	10*5*5	2278	66
		20*10*15	6532	2987
		45*12*17	91345	5467
Janiak et al. [26]	Dynamic programming	10*5*5	2205	43
		20*10*15	6347	197
		45*12*17	8765	4986
Elyashi and Salmasi [27]	Chance constrained programming	10*5*5	2491	54
		20*10*15	6691	81
		45*12*17	NFS	6000
Droubouchevitch and Sidney [28]	Two stage heuristic algorithm	10*5*5	2709	238
		20*10*15	6987	4671
		45*12*17	NFS	6000
Hamed and Mohammad [29]	Heuristics	10*5*5	2114	Negligible
		20*10*15	6323	Negligible
		45*12*17	75438	Negligible
Proposed algorithm	Artificial immune system	10*5*5	2054	Negligible
		20*10*15	6032	Negligible
		45*12*17	65342	Negligible

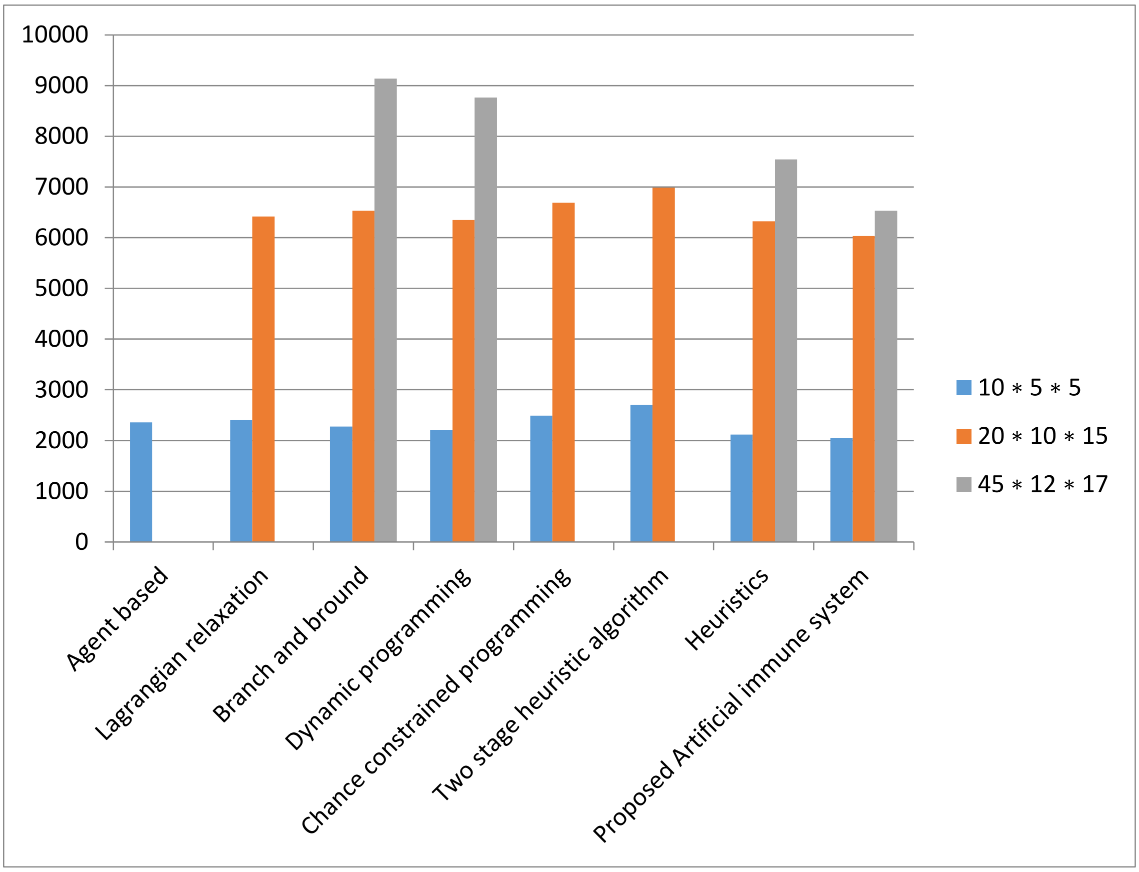

Figure 2: Comparison of the proposed algorithm with other algorithms

CONCLUSION

An essential driver to the manufacturing industries is to deliver the products on time to the customer. This is possible only if we minimize the make span of the manufacturing process. As a part of it, in this paper we are minimizing the total earliness and tardiness of AGVs to reduce the make span. The proposed algorithm is modelled by assuming number of jobs, number of AGVs, number of shops and processing time of each jobs are given. Along with that it is assumed that all AGVs moves with the same speed and each AGV can carry a unit load at a time. Scheduling, conflict free routes and optimal values are found using proposed algorithm. The obtained result is compared with the other existing models in the literature. This shows that the proposed algorithm yields a better result compared to others. We can solve the same model using other hybrid algorithms.

REFERENCES

[1] C.G. Co, J.M.A. Tanchoco, A review of research on AGVs vehicle management, Eng. Costs Prod. Econ. 21 (1991) 35–42.

[2] R.E. King, C.A. Wilson, A review of automated guided vehicle system design and scheduling, Prod. Plan. Control 2 (1991) 44–51.

[3] T. Ganesharajah, N.G. Hall, C. Sriskandarajah, Design and operational issues in AGV-served manufacturing systems, Ann. Oper. Res. 76 (1998) 109–154.

[4] L. Qiu, W.-J. Hsu, H. Wang, Scheduling and routing algorithms for AGVs: a Survey, Int. J. Prod. Res. 40 (2002) 745–760. I.F.A. Vis, Survey of research in the design and control of automated guided vehicle systems, European J. Oper. Res. 170 (2006) 677–709.

[5] I.F.A. Vis, Survey of research in the design and control of automated guided vehicle systems, European J. Oper. Res. 170 (2006) 677–709.

[6] C.W. Kim, J.M.A. Tanchoco, Conflict-free shortest time bi-directional AGV routing, Int. J. Prod. Res. 29 (1991) 2377–2391.

[7] M. Dotoli, M.P. Fanti, Coloured timed Petri net model for real-time control of automated guided vehicle systems, Int. J. Prod. Res. 9 (2004) 1787–1814.

[8] N. Wu, M. Zhou, Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles, IEEE Trans. Syst. Man Cybern. B 35 (2005) 1193–1201.

[9] S.P. Singh, M.K. Tiwari, Intelligent agent framework to determine the optimal conflict-free path for an automated guided vehicle system, Int. J. Prod. Res. 40(2002) 4195–4223.

[10] T. Nishi, M. Ando, M. Konishi, Distributed route planning for multiple mobile robots using an augmented Lagrangian decomposition technique, IEEE Trans. Robot. 22 (2005) 1191–1200.

[11] S. Ghasemzadeh, E. Behrangi, M.A. Azgomi, Conflict-free scheduling and Routing of automated guided vehicles in mesh topologies, Robot. Auton. Syst.30 (2009) 738–748.

[12] J.H. Lee, B.H. Lee, M.H. Choi, A real-time traffic control scheme of multiple AGV systems for collision free minimum time motion: a routing table approach, IEEE Trans. Syst. Man Cybern. A 28 (1998) 347–358.

[13] N.N. Krishnamurthy, R. Batta, M.H. Karwan, Developing conflict-free routes for automated guided vehicles in a flexible manufacturing system, Oper. Res. 41(1993) 1077–1090.

[14] C. Oboth, R. Batta, M. Karwan, Dynamic conflict-free routing of automated guided vehicles, Int. J. Prod. Res. 37 (1999) 2003–2030.

[15] A. Langevin, D. Lauzon, D. Riopel, Dispatching, routing and scheduling of two automated guided vehicles in a flexible manufacturing system, Int. J. Flexible Manuf. Syst. 8 (1996) 246–262.

[16] G. Desaulniers, A. Langevin, D. Riopel, B. Villeneuve, Dispatching and conflict free routing of automated guided vehicles: an exact approach, Int. J. Flexible Manuf. Syst. 15 (2003) 309–331.

[17] Mohammad Saidi-Mehrabad, Saeed Dehnavi-Arani, Farshid Evazabadian, Vahid Mahmoodian. An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs, Computers & Industrial Engineering. 86 (2015) 2–13.

[18] D. Dasgupta and F. Gonzalez, Artificial immune systems (AIS) research in the last five years, in Proc. Conf. Evolutionary Computation, 2003, PP. 123130.

[19] De Castro L. N., Timmis J. Artificial Immune Systems: A New Computational Intelligence Paradigm, London, 2002, UK: Springer-Verlag.

[20] Jiaquan Gao, jun Wang, WBMOAIS: A novel artificial immune system for multiobjective optimization, Computers& Operation Research, 2010, 37, pp 50-61.

[21] Leandro Nunes de Castro, Fernando Jose Von Zuben,"Artificial Immune System: Part I Basic Theory and Applications", Technical Report, TRDCA 01/99

[22] Waiel F. Abd El-Wahed, Elsayed M. Zaki, Adel M. El-Refaey' "Artificial immune system based neural networks for solving multi-objective programming problems" Egyptian Informatics Journal (2010) 11, 59–

65.

[23] E. Gerstl, G. Mosheiov, Scheduling problems with two competing agents to minimized weighted earliness–tardiness, Comput. Oper. Res. 40 (1) (2013) 109–116.

[24] S. Tanaka, M. Araki, An exact algorithm for the single-machine total weighted tardiness problem with sequence-dependent setup times, Comput. Oper. Res. 40 (1) (2013) 344–352.

[25] K. Kianfar, G. Moslehi, A algorithm for single machine scheduling with quadratic earliness and tardiness penalties, Comput. Oper. Res. 39 (12) (2012) 2978–2990.

[26] A. Janiak, W. Janiak, M.Y. Kovalyov, E. Kozan, E. Pesch, Parallel machine scheduling and common due window assignment with job independent earliness and tardiness costs, Inform. Sci. 224 (2013) 109–117.

[27] A. Elyasi, N. Salmasi, Stochastic scheduling with minimizing the number of tardy jobs using chance constrained programming, Math. Comput. Modelling(2012).

[28] I.G. Drobouchevitch, J.B. Sidney, Minimization of earliness, tardiness and due date penalties on uniform parallel machines with identical jobs, Comput. Oper. Res. 39 (9) (2012) 1919–1926.

[29] Hamed F, Mohammad S, Jaydeep B, Mathematical optimization for earliness/tardiness minimization in a multiple automated guided vehicle manufacturing system via integrated heuristic algorithms, Robotics and Autonomous Systems. 72 (2015) 131–138