Integration of Process Planning, Scheduling, and Mobile Robot Navigation Based on TRIZ and Multi-Agent Methodology

Milica Petrović
Research and Teaching Assistant
University of Belgrade
Faculty of Mechanical Engineering

Zoran Miljković
Full Professor
University of Belgrade
Faculty of Mechanical Engineering

Bojan Babić
Full Professor
University of Belgrade
Faculty of Mechanical Engineering

This paper presents methodology for development of software application for integration of process planning, scheduling, and the mobile robot navigation in manufacturing environment. Proposed methodology is based on the Russian Theory of Inventive Problem Solving (TRIZ) and multi-agent system (MAS). Contradiction matrix and inventive principles are proved as effective TRIZ tool to solve contradictions during conceptual phase of software development. The proposed MAS architecture consists of six intelligent agents: job agent, machine agent, optimization agent, path planning agent, machine learning agent and mobile robot agent. All agents work together to perform process plans optimization, schedule plans optimization, optimal path that mobile robot follows and classification of objects in a manufacturing environment. Experimental results show that developed software can be used for proposed integration in order to improve performance of intelligent manufacturing systems.

Keywords: *TRIZ, multi-agent system, process planning, scheduling, integrated process planning and scheduling, optimization, genetic algorithm, mobile robot navigation.*

1. INTRODUCTION

The advanced manufacturing paradigms like computer integrated manufacturing (CIM) and intelligent manufacturing systems (IMS) may provide more integrated manufacturing environments developed on the basis of modern software architectures and information technologies [1]. Some of the components that participate in integration of various phases of manufacturing are computer-aided process planning (CAPP), scheduling, material transport system (conveyers, automatic guided vehicles (AGV), intelligent mobile robots), etc.

The CAPP, as an important interface between computer-aided design (CAD) and computer-aided manufacturing (CAM), is an essential component of CIM system. The purpose of CAPP is to determine and optimize process plans for part to be manufactured economically. Scheduling problem is defined as allocation of operation on machines in time and the output of scheduling is a sequence of operations on machines. In traditional approaches, process planning and scheduling were carried out sequentially. Because of the fact that process planning and scheduling are complementary functions, many researchers proposed their integration to achieve global optimization of product development and manufacturing. In [2] is proposed particle swarm optimization (PSO) approach; in [3] a simulated annealing (SA) based approach; in [4] improved genetic algorithm (GA); in [5] symbiotic evolutionary algorithm; in [6] modified genetic algorithm; in [7,8] agent-based approach to integrate these functions.

The intelligent material transport, within integrated manufacturing environment, implies solving a path generation problem and controlling the movement of an intelligent agent - a mobile robot. The path that a mobile robot tracks directly depends on process planning and scheduling and can be generated and optimized in many ways, for example, using algorithms such as Dijkstra, A*, D*, E*, or optimization algorithms like PSO and GA [9].

While executing transportation task, the intelligent mobile robot must be capable of sensing its environment. Using sensor readings and artificial intelligence, the objects in manufacturing environment (machines and obstacles) can be classified and according to classification appropriate mobile robot actions can be made [10].

In this paper, methodology for development of software application for integrating all aforementioned components is presented. The first part of paper focuses on TRIZ methodology as an effective tool for conceptual design of software application and the second part of the paper focuses on development and implementation of multi-agent system in order to obtain optimal process plans, optimal scheduling plans, optimal path and classified objects in a manufacturing environment.

The reminder of this paper is organized as follows. Section 2 gives introduction of TRIZ methodology used for software development. In Section 3 is presented multi-agent methodology. In Section 4 are reported experimental results obtained by using own developed

Received: September 2012, Accepted: February 2013
Correspondence to: Milica Petrović, M.Sc.
University of Belgrade - Faculty of Mechanical
Kraljice Marije 16, 11120 Belgrade 35, Serbia
E-mail: mmpetrovic@mas.bg.ac.rs

software named Skynet. Section 5 provides concluding remarks.

2. TRIZ METHODOLOGY FOR SOFTWARE DEVELOPMENT

The Theory of Inventive Problem Solving (Russian acronym: TRIZ) has been widely used as effective tool to define, analyse and solve problems, especially at the conceptual design phase. The TRIZ method was developed by Genrich Altshuller in 1946 and it is a systematic approach for generating innovative solutions based on the extensive study of more than two million worldwide patents and other inventions. The TRIZ approach to innovative problem solving is given in Figure 1. The first step of TRIZ problem solving is identification of specific problem and then follows its classification into generic problem in terms of technical or physical contradictions. The next step is formulating problem using TRIZ language and searching for analogous solutions. At the end, by using engineering knowledge, generic solution can be transformed into desired specific solution.

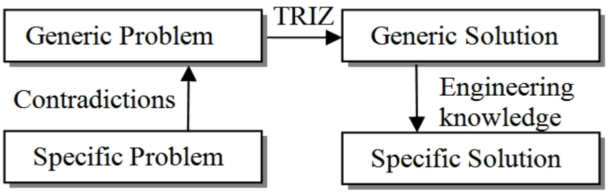

Figure 1. TRIZ way of innovative problem solving

The most commonly used TRIZ tool is the contradiction matrix, which is composed of 39 engineering parameters and 40 inventive principles [11]. There is a dependency relationship between the mentioned engineering parameters and while improving some parameters with positive effects, some of the other parameters might have negative effects. This results in a contradiction. Altshuller asserts that conflict resolving occurs when a contradiction between parameters is solved using 40 contradiction principles. In that way, the ideality of the design increases while a parameter is improved without worsening the other parameter. The 39 engineering parameters and 40 inventive contradiction principles are given in Table 1.

In terms of TRIZ for software, in literature [12,13] are presented analogies of TRIZ inventive principles in the context of software development. For example, principle “#1-Segmentation” means “dividing an object into independent parts” and can be interpreted as “divide a system into autonomous components-agents that can operate independently of each other, achieving a common goal” [12]. The other inventive principles that are applied to solve contradictions in this software development are 7, 10, 15, 16 and 35 (Table 2). When “#38-Level of automation” is selected as an improving feature and “#36-Complexity of a device” as a worsening feature, respectively, the corresponding matrix obtains suggested inventive principles as 15, 7, and 10. Using inventive principle “#15-Dynamicity”, we optimize work of all agents without increasing “#36-Complexity of a device”. Also, using principle “#7-Nesting” we improve “#38-Level of automation” by setting more sub-functions in each main function of the software application code without increasing “#36-Complexity of a device”.

Table 1. List of 39 parameters and 40 inventive principles

No.	Engineering parameters	Inventive principles
1	Weight of mobile object	Segmentation
2	Weight of stationary object	Extraction
3	Length of mobile object	Local quality
4	Length of a stationary object	Asymmetry
5	Area of a mobile object	Consolidation
6	Area of a stationary object	Universality
7	Volume of a mobile object	Nesting
8	Volume of a stationary object	Counterweight
9	Speed	Prior counteraction
10	Force	Prior action
11	Tension/Pressure	Cushion in advance
12	Shape	Equipotentiality
13	Stability of composition	Do it in reverse
14	Strength	Spheroidality
15	Time of action of a moving object	Dynamicity
16	Time of action of a stationary object	Partial or excessive actions
17	Temperature	Another dimension
18	Brightness	Mechanical vibration
19	Energy spent by a moving object	Periodic action
20	Energy spent by a stationary object	Continuity of useful action
21	Power	Rushing through
22	Loss of energy	Convert harm into benefit
23	Loss of substance	Feedback
24	Loss of information	Mediator
25	Loss of time	Self-service
26	Amount of substance	Copying
27	Reliability	Dispose
28	Accuracy of measurement	Mechanics substitution
29	Accuracy of manufacturing	Pneumatics or hydraulic construction
30	Harmful factors acting on an object from outside	Thin and flexible
31	Harmful factor developed by an object	Porous materials
32	Manufacturability	Changing the colour
33	Convenience of use	Homogeneity
34	Repairability	Rejecting and regenerating parts
35	Adaptability	Transformation properties
36	Complexity of a device	Phase transitions
37	Complexity of control	Thermal expansion
38	Level of automation	Accelerated oxidation
39	Capacity/Productivity	Inert environment
40	Engineering parameters	Composite materials

Table 2. Contradiction matrix and corresponding TRIZ principles

No.	Feature to improve	Conflicting feature	TRIZ principles	
1	Level of automation (38)	Complexity of a device (36)	15	Dynamicity
			7	Nesting
			10	Prior action
2	Adaptability (35)	Convenience of use (33)	1	Segmentation
			15	Dynamicity
			16	Partial or excessive actions
3	Reliability (27)	Productivity (39)	1	Segmentation
			35	Transformation properties

3. MULTI-AGENT INTEGRATION METHODOLOGY

The concept of agent comes from artificial intelligence [14]. In the manufacturing domain, it is possible to define an agent as an intelligent entity that may represent physical manufacturing entity (machine, robot, AGV, tool, cell, etc.) or computational entity (algorithm, soft-computing technique, etc. that can be implemented in a manufacturing system such as learning agent, optimization agent, path planning agent).

A multi-agent system (MAS) is an artificial intelligence system composed of a population of autonomous agents that are able to cooperate in order to reach an overall goal, while simultaneously pursuing individual objectives. In this research, we applied following six agents to make MAS and integrate manufacturing functions: job agent, machine agent, optimization agent, path planning agent, machine learning agent and mobile robot agent. The job agent and machine agent are used to represent jobs and machines. The optimization agent is used to generate the alternative process plans and optimize scheduling plans. Path planning agent is used to generate optimal path that mobile agent follows and machine learning agent is used for classifying object (machines and obstacles) during robot movement through manufacturing environment.

3.1 Job agent

Jobs agents represent the jobs (parts) that are manufactured in the manufacturing system. Each job agent contains information of a particular job, which includes job ID, job name, job operations and information about alternative process plans.

In order to adopt representation for alternative process plans, many types of flexibilities in process planning are considered: operation, sequencing, and processing flexibility [6]. Petri-net, AND/OR graphs and networks are some of the numerous methods used to describe these types of flexibilities. In this paper, a network representation method is adopted. Generally, there are three node types in the network representation: the starting node, the intermediate node and the ending node. The starting and the ending node indicate the beginning and the end of the manufacturing process of a job and an intermediate node represents an operation. The intermediate node contains a set of alternative machines that are used to perform the operation and the processing time for the operation according to the machines. All nodes are connected with arrows that represent the precedence between them. For each job, every alternative path in the network starts with an OR-connector and ends with a join-connector. OR-links are used for making decisions as to which of the alternative manufacturing process procedures will be selected. All links that are not connected by OR-connectors must be visited. Figure 2 shows alternative process plan networks for two jobs, job 1 and job 2.

In this research, the optimization objective of the flexible process planning problem is to minimize the production time which consists of processing time and transportation time. The notations used to explain mathematical model of operation sequencing problem is described as follows [6]:

- n – total number of jobs;
- G_i – total number of process plans of the i-th job;
- t – 1, 2, 3,..., M generations;
- o_{ijl} – j-th operation in the l-th process plan of the i-th job;
- P_{il} – number of operations in the l-th process plan of the i-th job;
- k – Alternative machine corresponding to o_{ijl};
- $TW(i,j,l,k)$ – processing time of operation o_{ijl} on the k-th alternative machine;
- $TP(i,t)$ – production time of i-th job in the t-th generation;
- $TT(i,l,(j,k_1),(j+1,k_2))$ – the transportation time between the k_1-th and the k_2-th alternative machine

The production time TP is calculated as shown in (1).

$$TP(i,t) = \sum_{j=1}^{P_{il}} TW(i,j,k,l) + \sum_{j=1}^{P_{il}-1} TT(i,l,(j,k_1),(j+1,k_2)), \quad i \in [1,n],\ j \in [1,P_{il}],\ l \in [1,G_i]. \tag{1}$$

The first constraint used here is that each machine can handle only one job (operation) at the time and the second is that the operations of one job cannot be processed simultaneously. The objective function is given as follows:

$$\max f(i,t) = \frac{1}{TP(i,t)}, \tag{2}$$

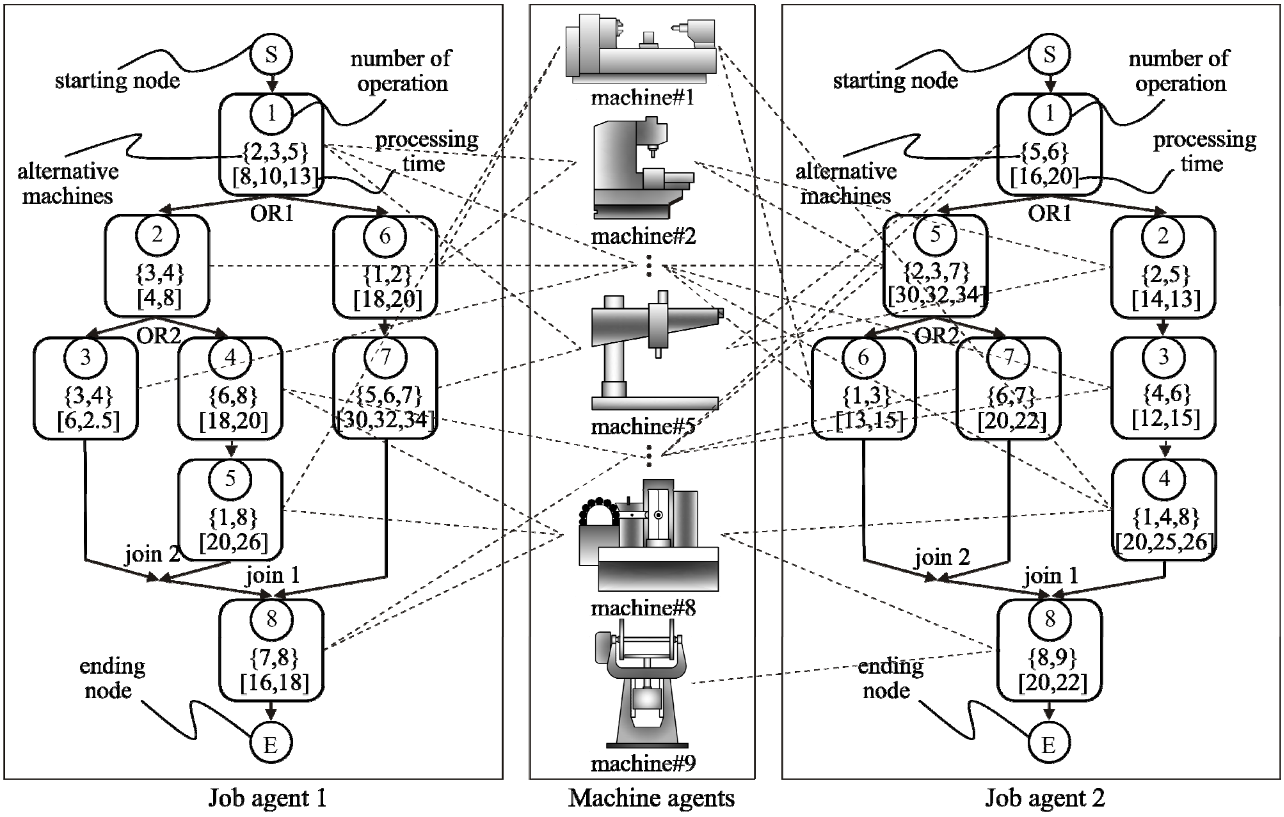

Figure 2. Alternative process plan network for two parts and nine machines

and it defines the alternative process plan with the minimum production time *TP*.

3.2 Machine agent

Each machine represented by machine agent contains the information about: machine ID, machine name, the manufacturing features which machines can process, the processing times and transportation times between machines. For each machine agent, status is available: idle or in use for manufacturing operation. Based on constraint that each machine can handle only one job at the time, each machine agent negotiate with job and optimization agents to get necessary information.

3.3 Optimization agent

Optimization agent is a part of the proposed MAS. It can optimize the process plans and scheduling plans using GA approach. The major steps of the GA approach for process plans optimization are described as follows:

Step 1: Generate the individuals for an initial population and initialize the parameters of the GA for the process plans optimization. Chromosome encoding and decoding is conducted as described in [6].

Step 2: Evaluate the fitness of the individuals as described by (1) and (2).

Step 3: Selection. We adopted the fitness-proportional, roulette wheel selection, where the probability of selection is proportional to an individual's fitness.

Step 4: Crossover. According to the defined crossover probability p_c, some individuals are picked out for crossover. For each pair of selected parent chromosomes, a single crossover point is randomly generated and applied for the recombination of process planning individuals.

Step 5: Mutation. According to the defined mutation probability p_m, some individuals are randomly selected to be mutated and for each selected chromosome a mutation point is randomly chosen.

Step 6: Repeat steps 2-5 for number of generations set in the step 1.

Step 7: Save s optimal or near optimal alternative process plans.

The scheduling optimization agent uses optimal or near optimal process plans and generates optimal scheduling plans. The major steps of the GA approach for generating optimal scheduling plans are described as follows:

Step 8: Select the alternative process plan generated in the process plans optimization phase.

Step 9: Generate the individuals for an initial population and initialize the parameters of the GA for the optimization of scheduling plans. Chromosome encoding and decoding is conducted as described in [6].

Step 10: Evaluate the fitness of the individuals and initialize the parameters of the GA for scheduling. The fitness function for each individual is calculated by using these equations:

$$object1 = \max(c_{ij})(c_{ij} \in T_d(s_{ij}, c_{ij})), \tag{3}$$

$$object2 = object1 + \sum_{a=1}^{m} \left| \sum p_{ij} - avgmt \right| (o_{ij} \in M_a), \tag{4}$$

where c_{ij} is the earliest completion time of operation o_{ij}, s_{ij} the earliest starting time of operation o_{ij}, $\sum p_{ij}$ is the total processing time for a machine and *avgmt* is the average processing time of all machines.

Step 11: Selection: use the same selection operator as used in step 3.

Step 12: Crossover. According to the defined crossover probability p_c, some individuals are picked out for crossover. The crossover procedure for scheduling string is described in [6].

Step 13: Mutation. According to the defined mutation probability p_m, some individuals are randomly selected to be mutated. In this paper, two types of mutation operators are used: two point swapping mutation and mutation for changing one job's alternative process plan.

Step 14: Repeat steps 8-13 for number of generations set in the step 9.

Step 15: Output the optimal scheduling plan as sequence of operations on machines.

After applying previous steps 1-7, the optimization agent generates optimal or near optimal process plans of each job according to minimum production time and applying steps 8-15 the optimal scheduling plan is obtained according to *object 1* or *object 2*.

3.4 Path planning agent

Path planning agent has a task to select paths with the assigned task before movement commences. For the applications in the known static environment, path planning is usually solved in two steps: (1) building a graph to represent the geometric structure of the environment and (2) perform a graph search to find path between the start and goal points based on certain criteria. Path planning agent can use algorithms such as Dijkstra, A*, D*, E*, or optimization algorithms like PSO and GA [9].

For purposes of the mobile robot path planning task, the A* search algorithm is used for finding the shortest path between the start and goal points. Description of A* path planning algorithm and its implementation is given in [15].

3.5 Machine learning agent

The intelligent mobile robot must be capable of sensing its environment while executing transportation task. Using sensing devices, mobile robot first gets perception of its environment and then moves through environment. It is very important to mobile robot classify objects in the environment based on measurements of the distance to an object in environment. Machine learning agent based on artificial neural networks [16] is used for classification of characteristic objects (machines and obstacles) in laboratory model of manufacturing environment. Detailed description of development and implementation of this algorithm is given in [10].

3.6 Mobile robot agent

Mobile robot agent represents mobile robot with appropriate sensors that are used for getting sensor readings as well as testing paths obtained by optimization and path planning agents. LEGO Mindstorms NXT configuration with one intelligent control unit, two servo motors and one ultrasonic sensor is used. For controlling the mobile robot, RWTH Mindstorms NXT Toolbox is utilized.

4. EXPERIMENTAL RESULTS

In order to illustrate the effectiveness and performance of the proposed TRIZ methodology in terms of software development and multi-agent approach in integration task, developed software application named Skynet is used. The platform for software development is Matlab software package, Java Development Kit and Microsoft Visual C++.

Each agent in the software application described in previous section can operate in two ways: independently and in cooperation with other agents. For example, when agent operates independently, we can use only optimisation agent to get optimal process plans without using path planning or machine learning agents. Also, we can use only path planning agent and get optimal trajectory from start to goal point without using any other agent. On the other hand, we can use agents all together in generating integrated intelligent manufacturing system. How agents are organised together and how they solve problems by working together is shown by MAS architecture in Figure 3. Communication between agents is done by using agent platform with host PC. Data about machines, jobs and sensor readings are stored in Matlab database and communication between the Lego NXT mobile robot and PC are also achieved with Matlab via USB or Wireless protocol.

Figure 4 shows the procedure of creating alternative process plan network for job 2 using job agent. On the left part of Figure 4 are places where user can enter job name, operation number, alternative machines for each operation, time for each operation, and OR links if it is necessary for that job. Also, here is button for join connector used to end links of alternative process plans and button for the end node used after the last job operation to indicate end of the manufacturing process of a job. The maximum number of jobs that Skynet software supports is 10, the maximum number of operations for one job is 16 and the maximum number of OR links for one job is 5. In this experiment, we use four representative jobs and each job consists of 8 operations and two OR connectors.

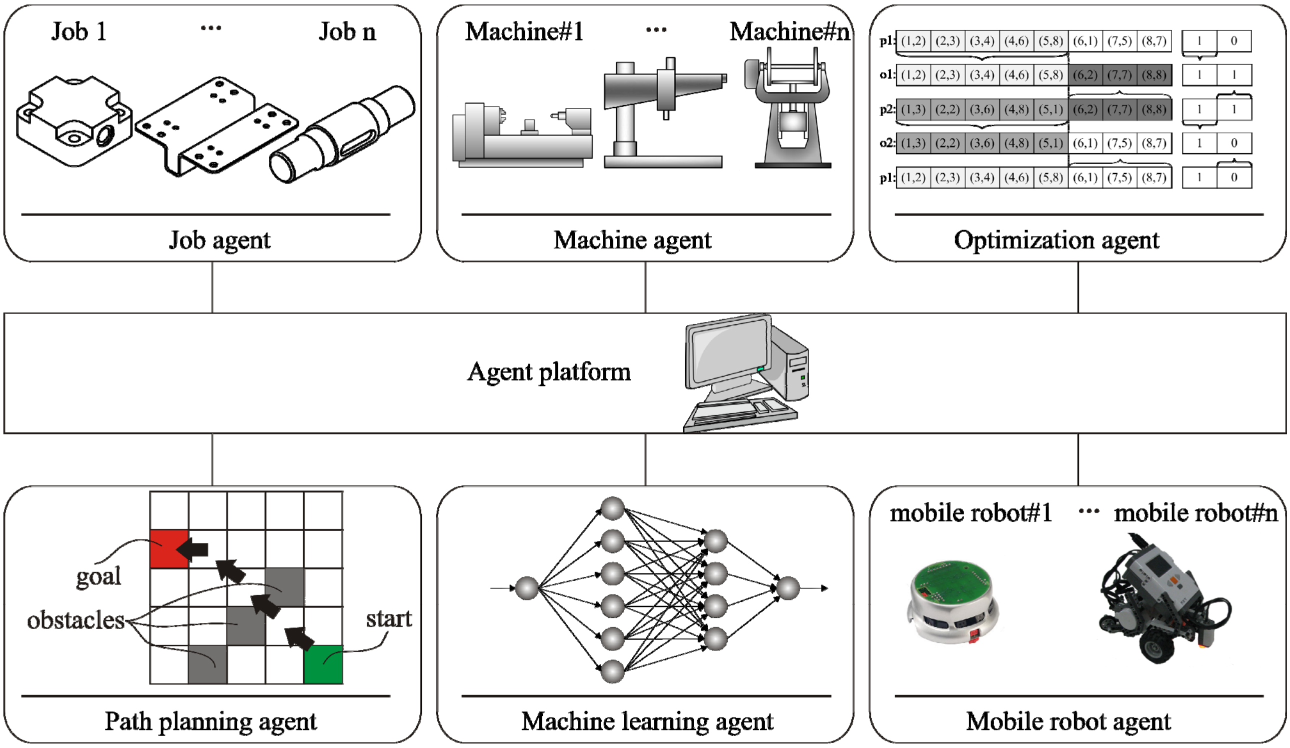

Figure 3. Multi-agent architecture for six agents

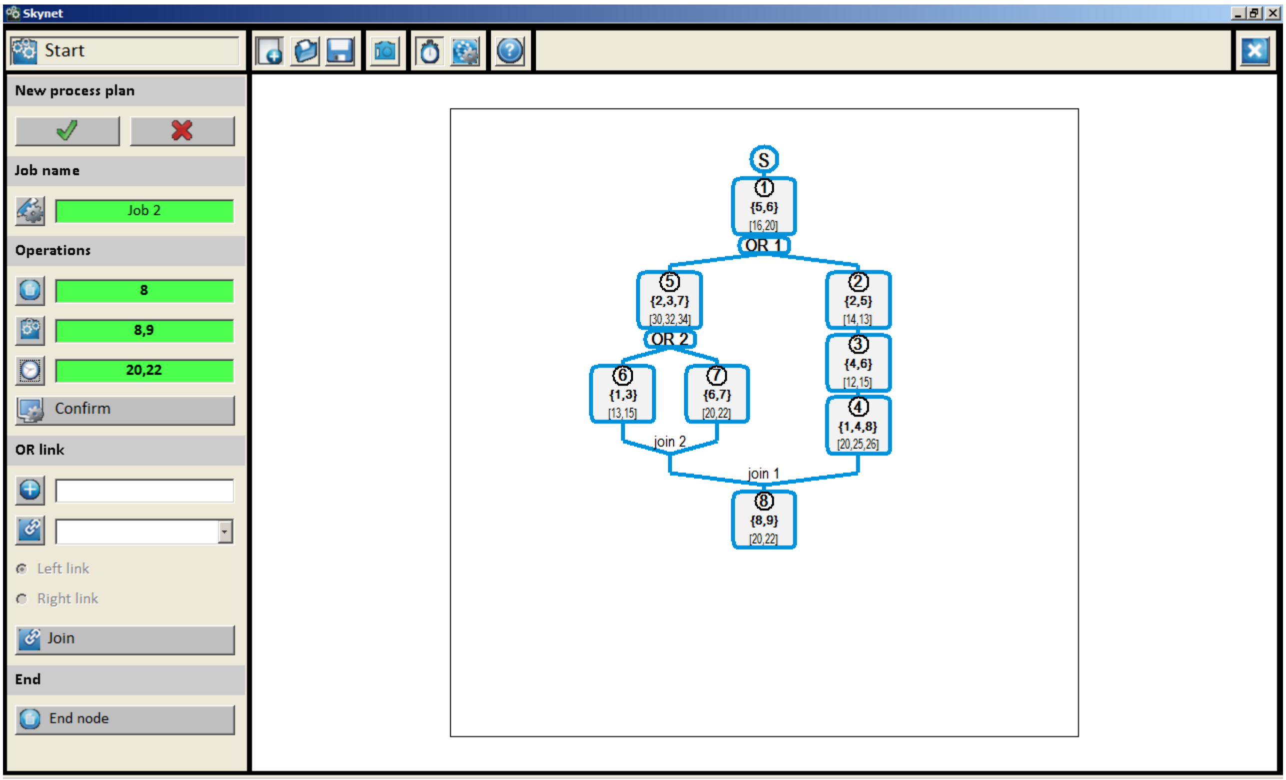

Figure 4. Job agent: Generation process plan network for job 2

Machine agent is presented in Figure 5. Using this agent, user can enter number of machines in considered manufacturing environment as well as machine ID, machine name, the manufacturing features which machines can process, the processing time and transportation times between machines. The maximum number of machines in this software is 15.

Using optimization agent for process plans optimization, firstly, user can set following GA parameters: the size of population is 40, the probability of crossover is 0.60, the probability of mutation is 0.10 and the number of generations is 30. After setting parameters, this agent generates desired entered number of alternative process plans for all jobs. For each job three alternative process plans with production time and fitness function are given in Figure 6. According to the results of process planning, optimization agent generates optimal scheduling plans. User can set GA

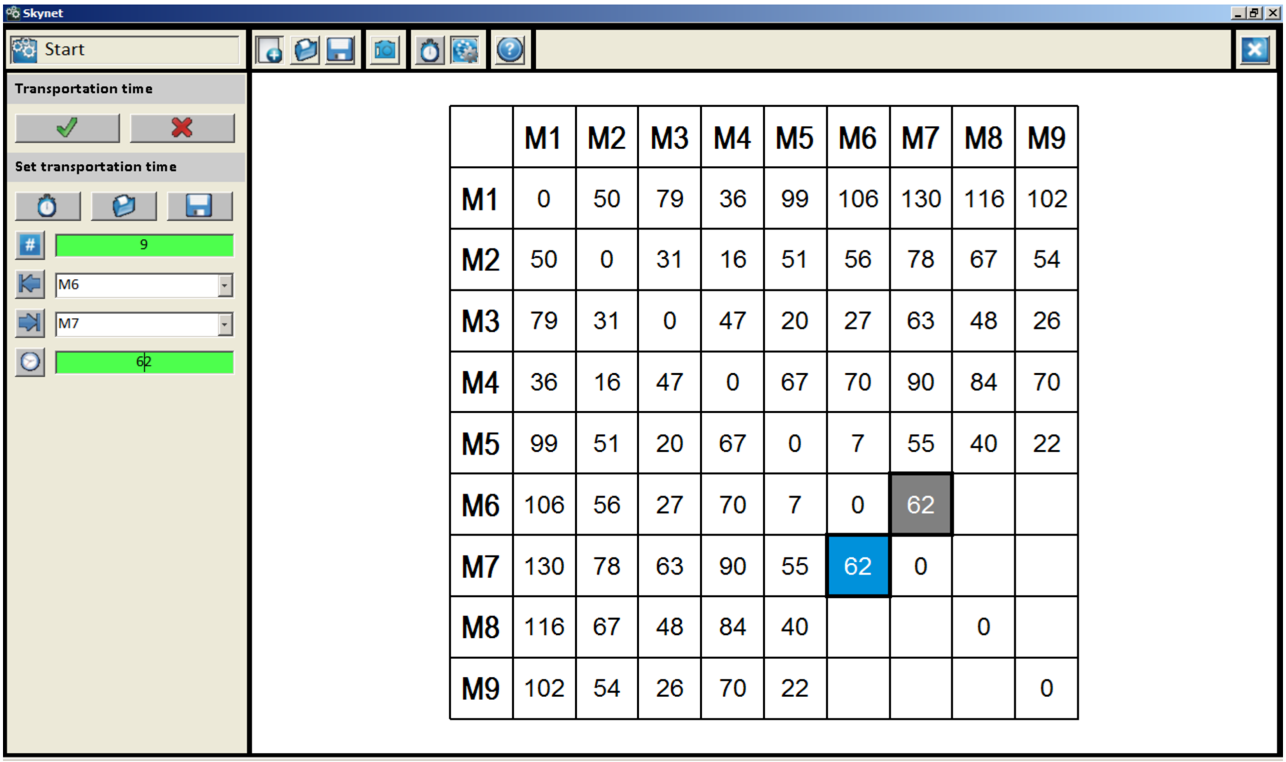

Figure 5. Machine agent: Setting transportation times between machines

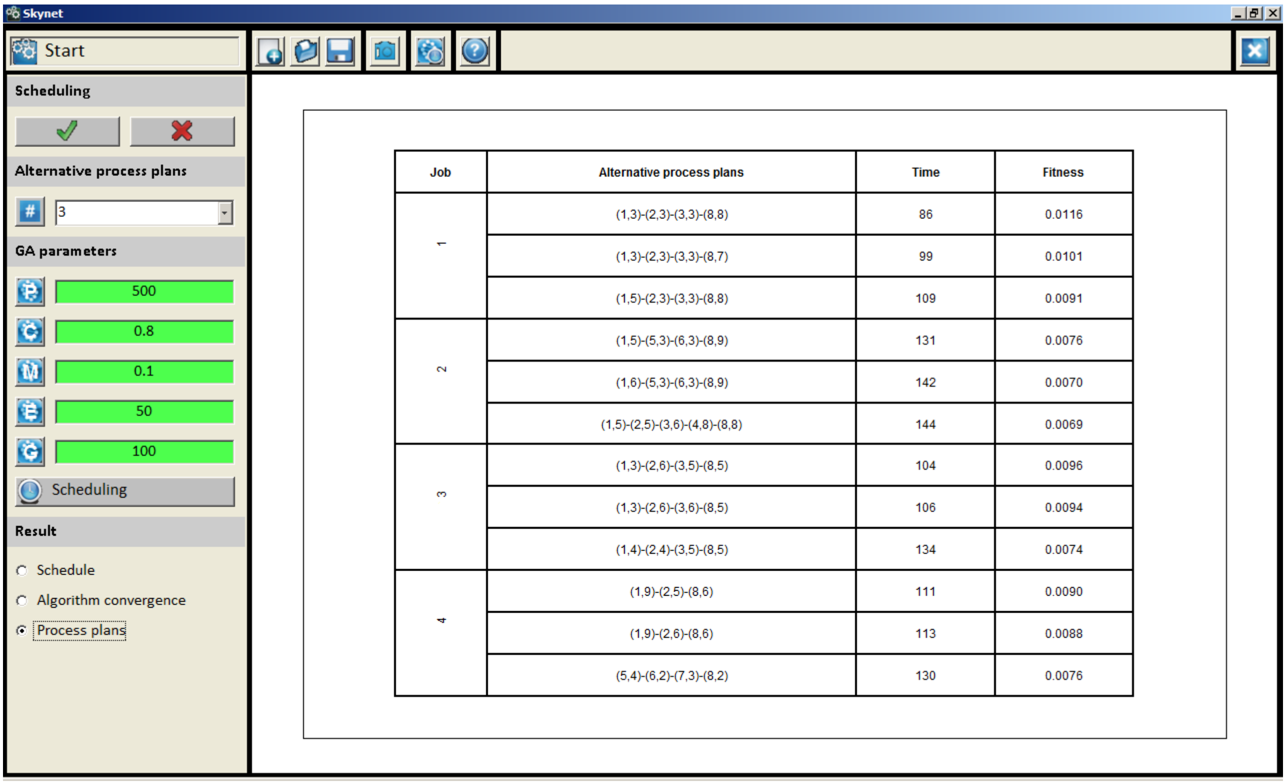

Figure 6. Optimization agent: Experimental results of process planning

parameters for scheduling as follows: the size of population is 500, the probability of crossover is 0.80, the probability of mutation is 0.10 and the number of generations is 100 (left part of Figure 7). Scheduling results for four jobs and 9 machines are given in a form of Gantt chart (right part of Figure 7) and convergence curves of the scheduling algorithm is given in Figure 8.

In learning agent module, it is possible to collect data from ultrasonic sensor while mobile robot moves through manufacturing environment and train neural networks to classify objects (machines and obstacles) in manufacturing environment. On the left part of Figure 9 is shown how we can set parameters for neural network learning process. The name of neural network is net, the neural network architecture is $[10\text{-}20\text{-}10]_3$, the activation function is tangent sigmoid and learning algorithm is Levenberg-Marquardt. On the right side of Figure 9 we can see convergence of neural network.

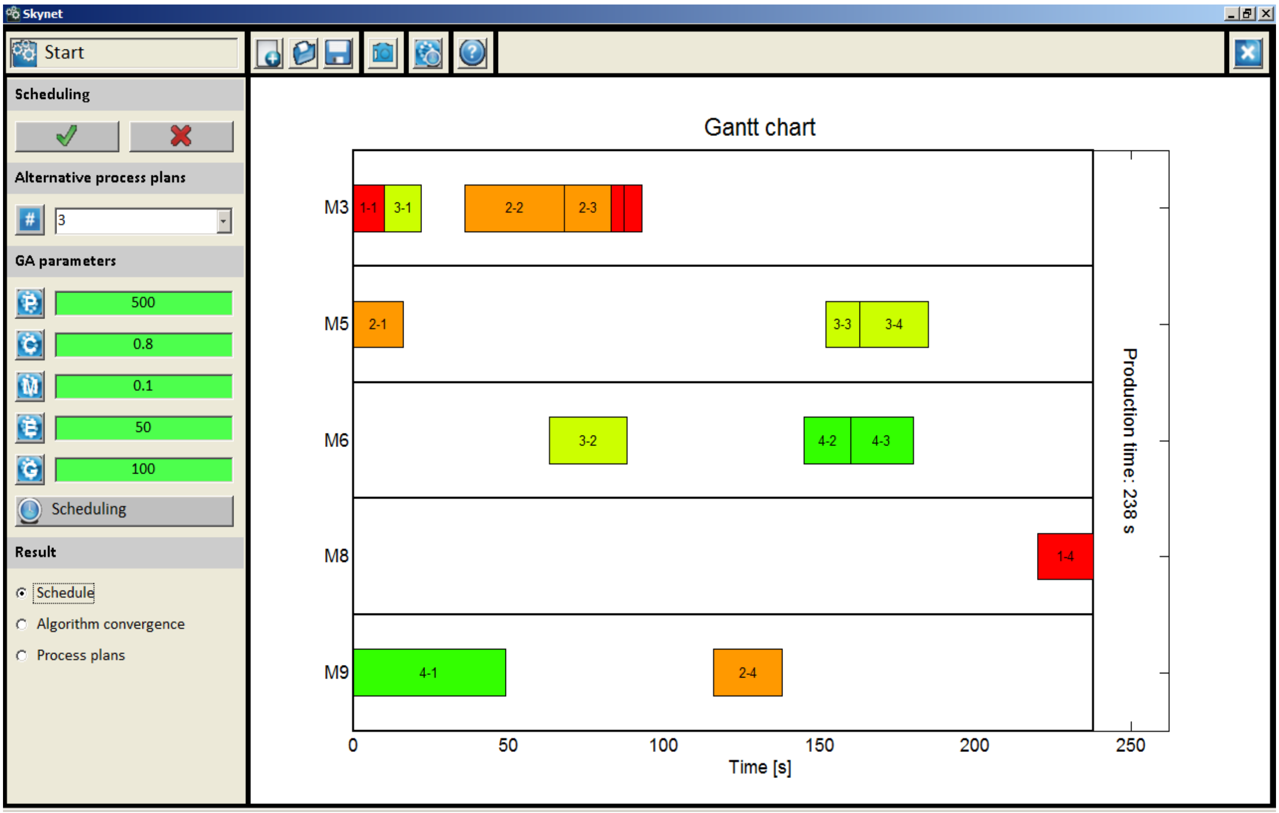

Figure 7. Optimization agent: Scheduling results for 4 jobs and 9 machines. Gantt chart is based on object 1 (Makespan=238)

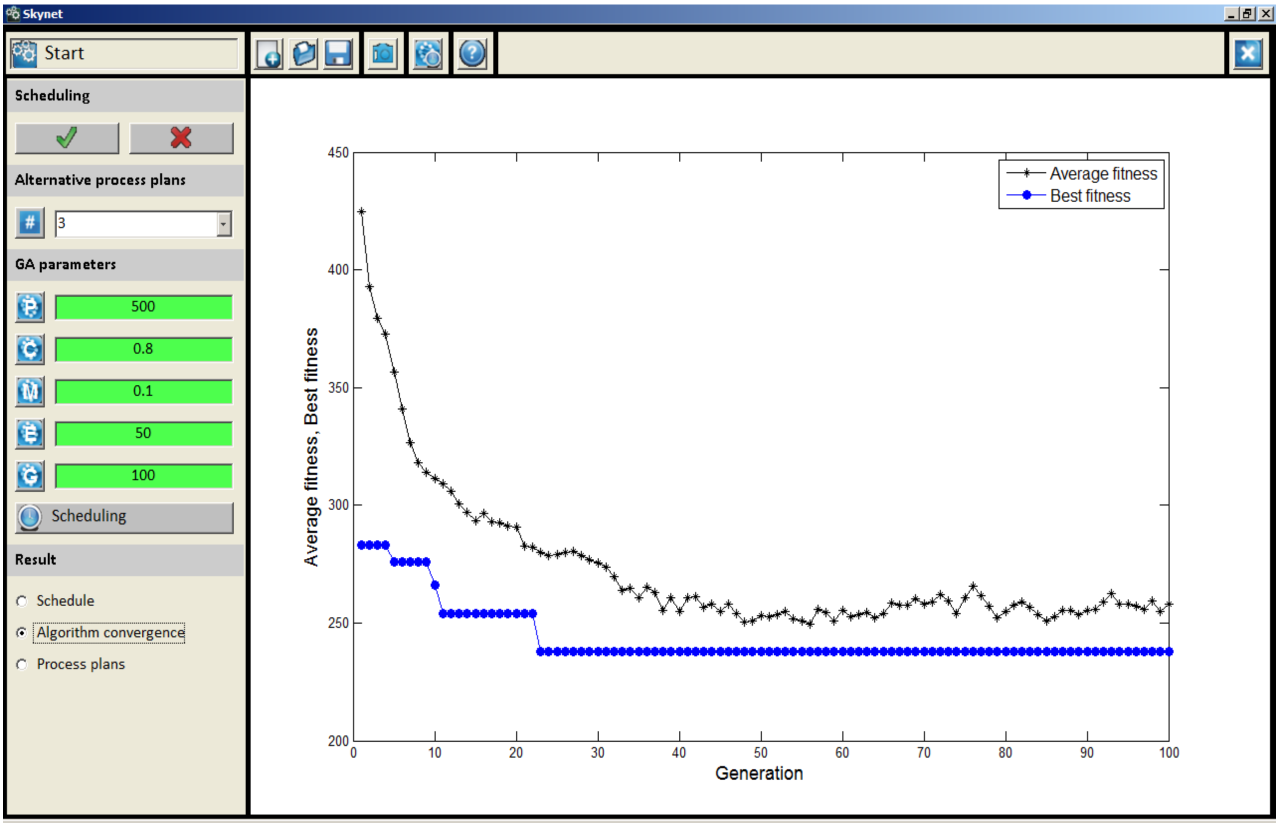

Figure 8. Optimization agent: Convergence curves of the proposed scheduling algorithm

In path planning agent, A* search algorithm is used for generating optimal path between the machines. If we want to use path planning agent independently from other agents, on the left part of Figure 10 in part “custom path” we can choose machines for optimization as well as Euclidean or Manhattan norm. On the other hand, we can utilize a sequence of machines generated by optimization agent and use A* algorithm to optimize the path between machines. Results for optimal path from machine 1 to machine 6, generated using Euclidean distance norm, are shown on the right part of Figure 10.

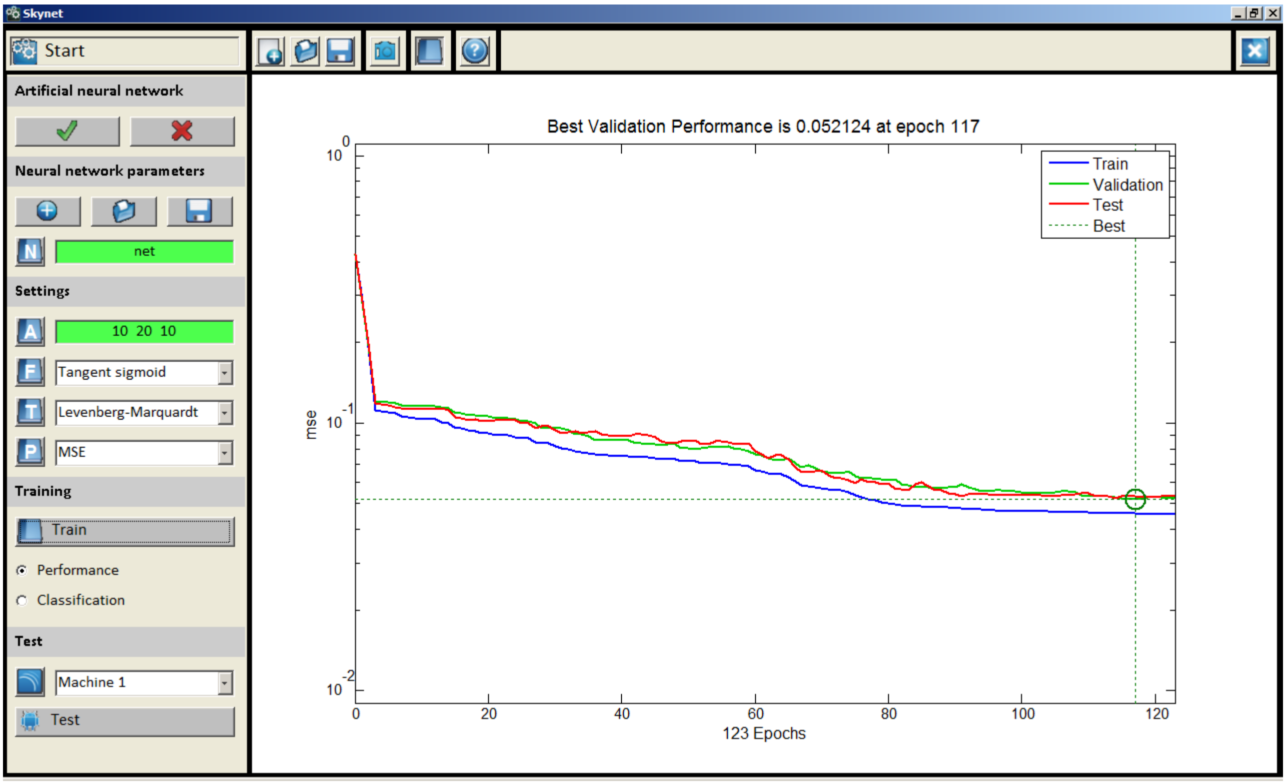

Figure 9. Learning agent: Convergence (MSE vs. learning iterations) of the selected ANN architecture [10-20-10]$_3$

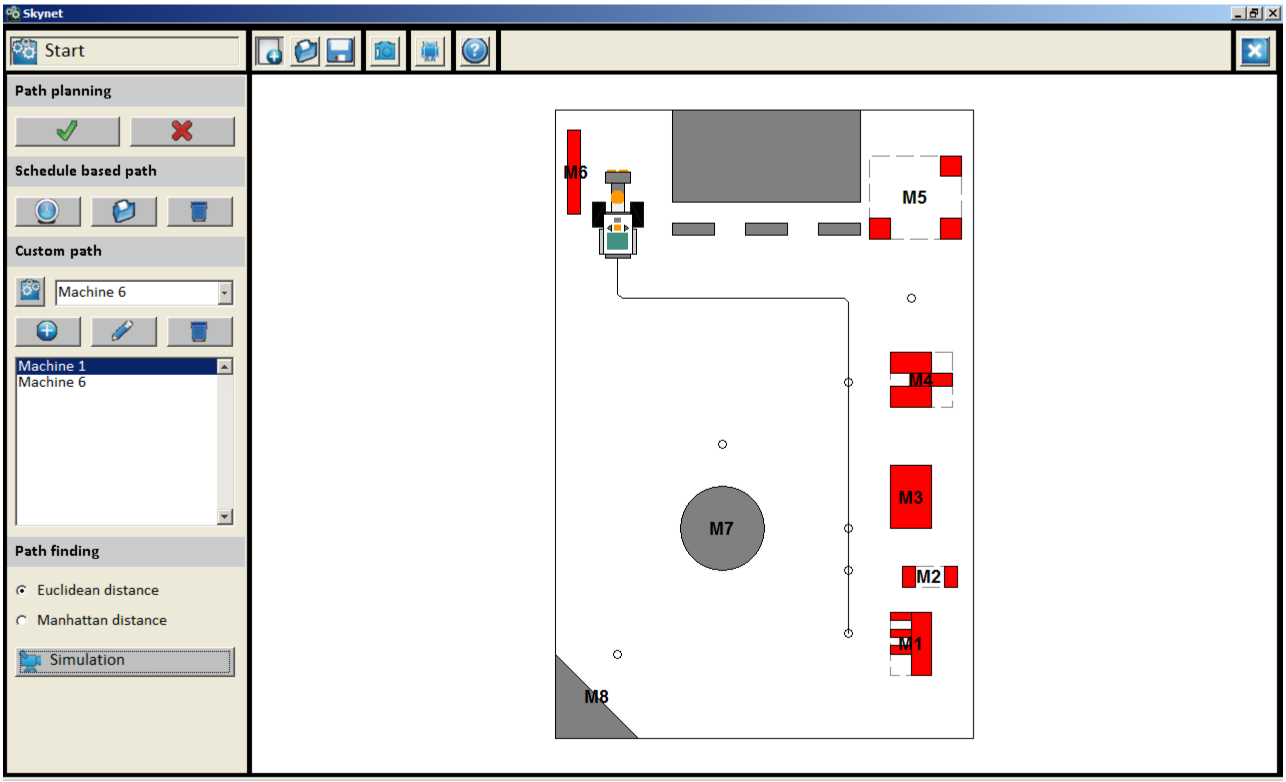

Figure 10. Path planning agent: Optimal path from machine 1 to machine 6 that mobile robot follows

5. CONCLUSION

In this paper was presented methodology for own software application development for integration of process planning, scheduling, and mobile robot navigation in a manufacturing environment. The proposed methodology is based on the Theory of Inventive Problem Solving (TRIZ) and multi-agent system (MAS) methodology. It is shown that TRIZ tools such as contradiction matrix and inventive principles can be used for solving contradictions in software application development. Using MAS based methodology integration of manufacturing functions was facilitated. Six agents that present proposed MAS are: job agent, machine agent, optimization agent, path planning agent, machine learning agent and mobile robot agent. An optimization agent is based on GA, path planning agent uses A* algorithm, learning agent is based on neural networks and mobile robot agent belongs to LEGO Mindstorms NXT technology. Although experimental verification was done in one manufacturing environment, the developed software application Skynet can also be used for Lego mobile robot navigation in different manufacturing environments, which implies other layouts, jobs, machines and technological processes.

ACKNOWLEDGMENT

This paper is part of the project: *An innovative, ecologically based approach to the implementation of intelligent manufacturing systems for the production of sheet metal parts*, financed by the Ministry of Education, Science and Technological Development of the Serbian Government, Grant *TR-35004*.

REFERENCES

[1] Miljković, Z.: *Systems of Artificial Neural Networks in Production Technologies*, Faculty of Mechanical Engineering, Belgrade, 2003, (in Serbian).

[2] Guo, Y.W., Li, W.D., Mileham, A.R. and Owen, G.W.: Applications of particle swarm and optimization in integrated process planning and scheduling, Robotics and Computer-Integrated Manufacturing, Vol. 25, No. 2, pp. 280-288, 2009.

[3] Li, W.D. and McMahon, C.A.: A simulated annealing-based optimization approach for integrated process planning and scheduling, International Journal of Computer Integrated Manufacturing, Vol. 20, No.1, pp. 80-95, 2007.

[4] Lihong, Q. and Shengping, L.: An improved genetic algorithm for integrated process planning and scheduling, The International Journal of Advanced Manufacturing Technology, Vol. 58, No. 5-8, pp. 727-740, 2012.

[5] Kim, Y.K., Park, K. and Ko, J.: A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling, Computers and Operations Research, Vol.30, No.8, 1151-1171, 2003.

[6] Shao, X.Y., Li, X.Y., Gao, L. and Zhang, C.Y.: Integration of process planning and scheduling-a modified genetic algorithm-based approach, Computers and Operations Research, Vol. 36, No. 6, pp. 2082-2096, 2009.

[7] Wong, T.N., Leung, C.W., Mak, K.L. and Fung, R.Y.K.: An agent-based negotiation approach to integrate process planning and scheduling, International Journal of Production Research, Vol. 44, No. 7, pp. 1331-1351, 2006.

[8] Li, X., Zhang, C., Gao, L., Li, W. and Shao, X.: An agent-based approach for integrated process planning and scheduling, Expert Systems with Applications, Vol. 37, No. 2, pp. 1256-1264, 2010.

[9] Liu, D.K. and Kulatunga, A.K.: Simultaneous planning and scheduling for multi-autonomous vehicles, in: *Studies in Computational Intelligence*, Springer-Verlag, pp. 437-464, 2007.

[10] Vuković, N. and Miljković, Z.: New Hybrid Control Architecture for Intelligent Mobile Robot Navigation in a Manufacturing Environment, FME Transactions, Vol. 37, No. 1, pp. 9-18, 2009.

[11] Kaplan, S.: An introduction to TRIZ: The Russian Theory of Inventive Problem Solving, Ideation International Inc., USA, 1996.

[12] Rea, K.: TRIZ and software-40 principles analogies part 1, TRIZ Journal, 2001.

[13] Rea, K.: TRIZ and software-40 principles analogies part 2, TRIZ Journal, 2001.

[14] Monostori, L., Váncza, J., and Kumara, S.R.T.: Agent-based systems for manufacturing, Annals of the CIRP, Vol. 55, No. 2, pp. 672–720, 2006.

[15] Babić,B., Miljković,Z., Vuković,N. and Antić,V.: Towards Implementation and Autonomous Navigation of an Intelligent Automated Guided Vehicle in Material Handling Systems, Iranian Journal of Science and Technology–Transaction B: Engineering, Vol. 36, No. M1, pp. 25-40, 2012.

[16] Miljković, Z. and Aleksendrić, D.: *Artificial Neural Networks – solved examples with theoretical background*, University of Belgrade - Faculty of Mechanical Engineering, 2009, (in Serbian).

ИНТЕГРАЦИЈА ПРОЈЕКТОВАЊА ТЕХНОЛОШКОГ ПРОЦЕСА, ТЕРМИНИРАЊА ПРОИЗВОДЊЕ И УПРАВЉАЊА МОБИЛНОГ РОБОТА БАЗИРАНА НА ТРИЗ И МУЛТИАГЕНТСКОЈ МЕТОДОЛОГИЈИ

Милица Петровић, Зоран Миљковић, Бојан Бабић

У раду је представљена методологија за развој софтверске апликације за интеграцију пројектовања технолошког процеса, терминирања производње и навигације мобилног робота у технолошком окружењу. Предложена методологија је базирана на примени теорије инвентивног решавања проблема и мултиагентске методологије. Матрица контрадикције и инвентивни принципи су се показали као ефективан алат за отклањање контрадикторности у концепцијској фази развоја софтвера. Предложена мултиагентска архитектура садржи шест агената: агент за делове, агент за машине, агент за оптимизацију, агент за планирање путање, агент за машинско учење и агент мобилни робот. Сви агенти заједно учествују у оптимизацији технолошког процеса, оптимизацији планова терминирања, генерисању оптималних путања које мобилни робот прати и класификацији објеката у технолошком окружењу. Експериментални резултати показују да се развијени софтвер може користити за предложену интеграцију, а све у циљу побољшања перформанси интелигентних технолошких система.