INTEGRATION OF PROCESS PLANNING AND SCHEDULING USING MODIFIED PARTICLE SWARM OPTIMIZATION ALGORITHM

M. Petrović[1], Z. Miljković[1], N. Vuković[2], B. Babić[1], J. Petronijević[1]

1. University of Belgrade-Faculty of Mechanical Engineering, Production Engineering Department, Kraljice Marije 16, 11120 Belgrade 35, Republic of Serbia
2. University of Belgrade-Faculty of Mechanical Engineering, Innovation Center, Kraljice Marije 16, 11120 Belgrade 35, Republic of Serbia

ABSTRACT
Process planning and scheduling are two of the most important manufacturing functions which are usually performed sequentially in traditional approaches. Considering the fact that these functions are usually complementary, it is necessary to integrate them so as to improve performance of a manufacturing system. This paper conceptualizes a multi-agent methodology by considering four intelligent agents (job, machine, tool, and optimization agent) and presents developed modified particle swarm optimization (mPSO) algorithm to solve this combinatorial optimization problem effectively. In order to improve the search efficiency and increase ability to find global optimum, proposed mPSO algorithm has been enhanced with new crossover and mutation operators. Experimental results show applicability of the proposed approach in solving integrated process planning and scheduling problem.

KEYWORDS: process planning, scheduling, integrated process planning and scheduling, modified particle swarm optimization, multi-agent system.

1. INTRODUCTION

Computer aided process planning (CAPP) was developed at the end of the 20th century with the purpose of integrating computer aided design (CAD) and computer aided manufacturing (CAM). The aim of CAPP is to determine detailed methods for manufacturing a part economically and concurrently starting from the initial phase (drawing of the finished part) up to the final phase (the desired shape of the finished part). On the other hand, scheduling problem is defined as allocation of operation on machines in time. Scheduling plans receive process plans and output a sequence of operations on machines while satisfying the precedence relations given in process plans. 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.

Some of them applied artificial intelligence techniques for integration. Evolutionary algorithms, such as genetic algorithm (GA) and simulated annealing (SA), have recently been employed to generate optimal or nearly optimal plans satisfying the constraints and objectives of process planning and scheduling simultaneously. In /1/ GA based algorithm is developed to solve the integrated process planning and scheduling problem. The new symbiotic evolutionary algorithm, given in /2/, can simultaneously deal with the two problems of process planning and job shop scheduling. In /3/, a modified two-phase GA approach is used to optimize process planning and scheduling simultaneously. A unified representation model and a SA-based approach have been developed to facilitate the integration and optimization process /4/.

An agent-based approach can also be applied for integrating process planning and scheduling. An agent-based approach presented in /5/ has been developed to facilitate the integration of these two functions. In this approach, the two functions are carried out simultaneously and an optimization agent based on an evolutionary algorithm is used to manage the interactions and communications between agents. The development of an agent-based negotiation protocol for negotiations between the part agents and the machine agents is presented in /6/, and online hybrid agent-based negotiation multi-agent system to integrate process planning with scheduling/rescheduling is given in /7/. Dynamic flexible job shop scheduling problem with alternative process plans essentially involves deciding the order or priority for the jobs waiting to be processed on each machine. The concept of multi-agent systems is also applied to integrate dynamic process planning and dynamic production scheduling /8/.

In this paper, we focus on development and implementation of multi-agent system in order to obtain optimal process plans and optimal scheduling plans. The modified particle swarm optimization (mPSO) algorithm, introducing encoding/decoding method as well as crossover and mutation operators, is developed to solve this non-deterministic polynomial-time (NP-hard) combinatorial optimization problem. The network representation is adopted to describe various flexibilities including machine flexibility, tool flexibility, process flexibility, and sequence flexibility. The mathematical model for flexible process planning is described with the objective of minimizing the production time considering the alternative machine and alternative tool selection while objective for scheduling is minimization of the makespan.

2. MULTI-AGENT INTEGRATION METHODOLOGY

The concept of agent comes from artificial intelligence /9/. In the manufacturing domain, it is possible to define an agent as an intelligent entity that may represent physical manufacturing entity (job, machine, tool, robot, AGV, 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 four agents to make MAS and integrate manufacturing functions: job agent, machine agent, tool agent, and optimization agent. The job agent, machine agent and tool agent are used to represent jobs, machines and tools. The optimization agent is used to generate the alternative process plans and optimize scheduling plans.

3. BRIEF OVERVIEW OF PROPOSED AGENTS

Job agent

Job 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: machine, tool, sequence and process flexibility. Petri-net, AND/OR graphs and networks are some of the numerous methods used to describe these types of flexibilities. In this paper, a representation of flexible process plans in the form of a network is adopted, see Figure 1. 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 an operation number, a set of alternative machines and set of alternative tools that are used to perform the operation, and the processing time for the operation according to the selected machine and tool. All nodes are connected with arrows that represent the precedence relations between them.

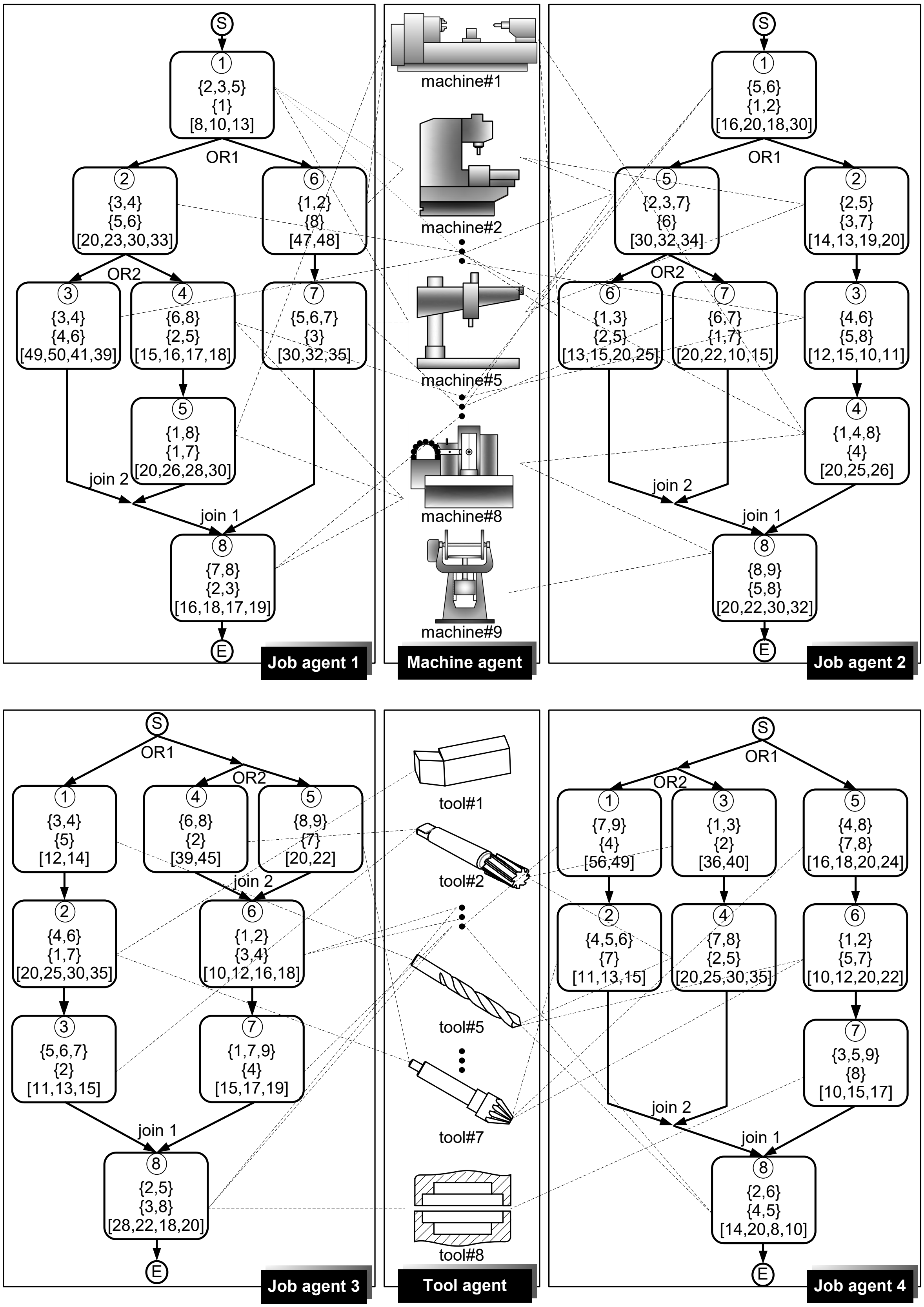


Figure 1: Alternative process plan network for four parts, nine machines and eight tools

For each job, every alternative path in network starts with OR-connector and ends with join-connector. OR-links are used to describe process flexibility and make decisions about alternative manufacturing process procedures to be selected. All links that are not connected by OR-connectors must be visited. Figure 1 visualizes an example of alternative process plan networks that models involved flexibilities and determines detailed representation of the machines and tools on which each operation of job 1, job 2, job 3 and job 4 are to be performed.

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. Because of the impact that alternative tools selection have on production time, besides impact of alternative machine selection, we additionally consider influence of alternative tools selection on production time. The notations used to explain mathematical model of operation sequencing problem is described as follows:

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

The production time to be minimized is formulated here as shown in equation (1):

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

Two constraints of machine and different process for one job are also taken into account. The first constraint is that each machine can handle only one operation at the time and the second one is that the operations of one job cannot be processed simultaneously. The objective function that defines the alternative process plans with the minimum production time $TP(i,g)$ is given in equation (2) as follows:

$$\text{objective: maximize } f(i,g) = \frac{1}{TP(i,g)}. \tag{2}$$

Optimization objective of the scheduling problem is to minimize makespan, which is calculated as in equation (3):

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

where c_{ij} is the earliest completion time of operation o_{ij} and s_{ij} is the earliest starting time of operation o_{ij}.

Machine agent

Each machine represented by machine agent contains the information about: machine ID, machine name, 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 negotiates with job and optimization agents to get necessary information.

Tool agent
Tool agents represent the tools used to manufacture the parts. Each tool agent contains information of a particular tool, which includes tool ID, tool name, and tool operations. Based on constraint that each tool can handle only one job at the time, each tool agent negotiate with job and optimization agents to get necessary information.

Optimization agent: Traditional PSO algorithm
Traditional PSO algorithm is initialized with a population of randomly generated candidate solutions known as particles. Each particle flies through the multidimensional search space of the optimization problem with a specific velocity searching for the optimal solution; its position represents a potential solution of the problem and its velocity is dynamically adjusted according to its own flying experience and according to the neighbouring flying experience. Particle position and particle velocity are updated iteratively by using equation (4) and equation (5):

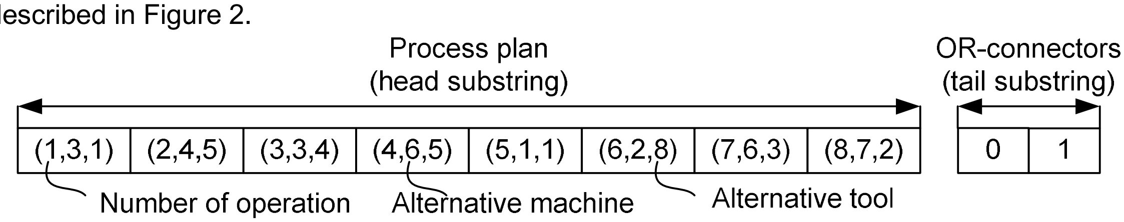
$$V_{id}^{t+1} = W \cdot V_{id}^{t} + C_1 \cdot rand() \cdot (P_{ld}^{t} - X_{id}^{t}) + C_2 \cdot Rand() \cdot (P_{gd}^{t} - X_{id}^{t}), \tag{4}$$

$$X_{id}^{t+1} = X_{id}^{t} + V_{id}^{t+1} \tag{5}$$

$$W = W_{\max} - \frac{W_{\max} - W_{\min}}{iter_{\max}} \cdot iter \tag{6}$$

where t is the iteration number; V_{id}^{t} and V_{id}^{t+1} represent the velocities of the particle i in generations t and $t+1$; X_{id}^{t} and X_{id}^{t+1} represent the positions of the particle i in generations t and $t+1$; P_{ld}^{t} is the local best solution ("pbest"); P_{gd}^{t} is the global best solution ("gbest"); W is inertia weight, set as in equation (6); C_1 and C_2 are positive acceleration constants; *rand()* and *Rand()* are two random numbers in the range [0,1].

Optimization agent: Encoding and decoding of the particles for process planning
One of the most important issues in applying PSO successfully is to develop particle encoding/decoding scheme in which process plan parameters are represented as a particle in search solution space. In this research, particles of modified PSO algorithm are seen as the chromosomes in the GA, where chromosome encoding and decoding is conducted as described in Figure 2.



<u>Figure 2:</u> Encoding scheme for flexible process planning

Optimization agent: Initial population for process planning
PSO algorithm starts with randomly generating an initial population of particles. After generating the individuals for an initial population, feasible operation sequence in a process plan is taken into account. Feasible operation-machine-tool sequence means that the order of elements in the encoding does not break constraints on precedence relations of operations, machines, and tools in network representation.

Optimization agent: Selection for process planning
After deciding on an encoding phase and generating an initial population, we need to make decision how to choose individuals in the population that will create offspring for the next generation. This phase is called selection and it is the process of selecting two parents from the population for crossover operation. We adopted fitness-proportional, roulette wheel selection, where the probability of selection is proportional to an individual's fitness.

Optimization agent: Crossover for process planning
According to the defined crossover probability p_c, some particles are picked out for crossover. For each pair of selected parent chromosomes, single crossover point is randomly generated and applied for the recombination of process planning individuals. Figure 3 shows how two offspring are produced from the parents' pair in terms of the crossover operation.

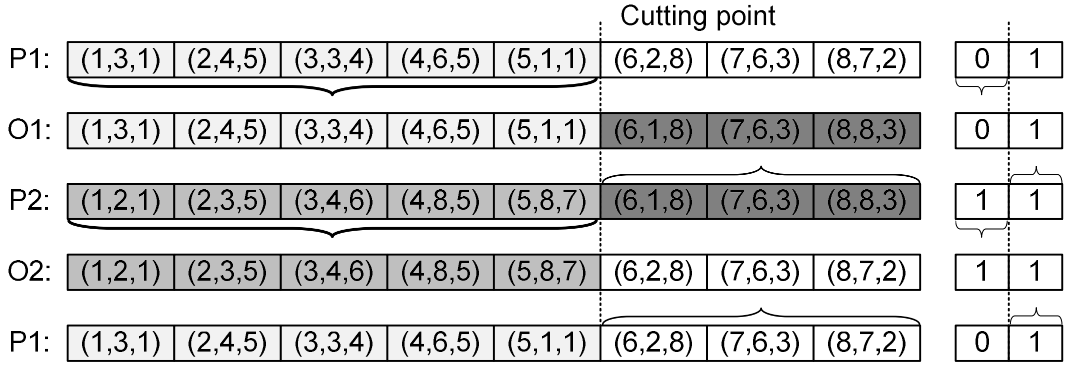


Figure 3: Crossover for flexible process planning

Optimization agent: Mutation for process planning
After crossover operation, according to the defined mutation probability p_m, some particles are randomly selected to be mutated. For each selected particle, a mutation point is randomly chosen, three mutation operators are applied, and, as a result of mutation, rearrangement of appropriate operation-machine-tool Gene is carried out. The examples of the three mutation operators for the particle are presented in Figure 4.

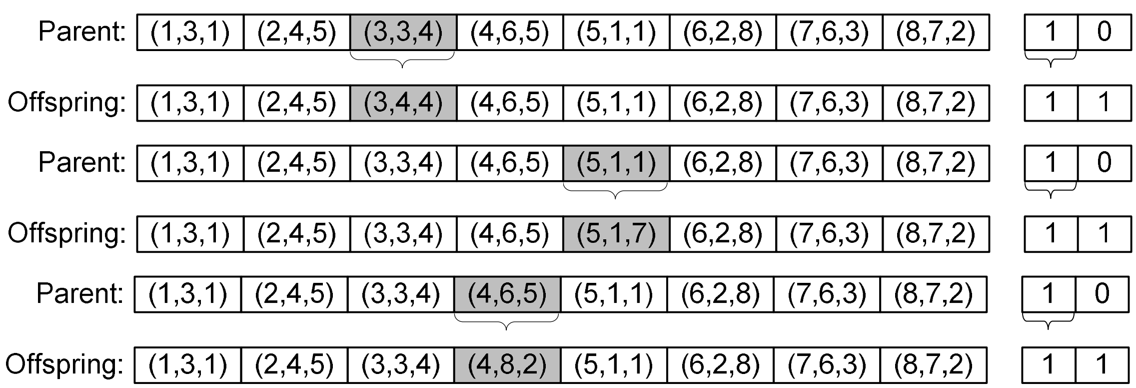


Figure 4: The first, the second and third mutation operation for process planning

Optimization agent: Encoding and decoding of the particles for scheduling
Each chromosome in for scheduling string consists of four parts: scheduling plan, process plan, machine string, and tool string. Particle encoding/decoding scheme for scheduling plan string is conducted as described in Figure 5.

Optimization agent: Initial population for scheduling
After selection of the alternative process plan generated in the process plans optimization phase and chromosome encoding/decoding phase, PSO algorithm for scheduling randomly generates an initial population of particles.

Optimization agent: Selection for scheduling
We adopted fitness-proportional, roulette wheel selection, where the probability of selection is proportional to an individual's fitness.

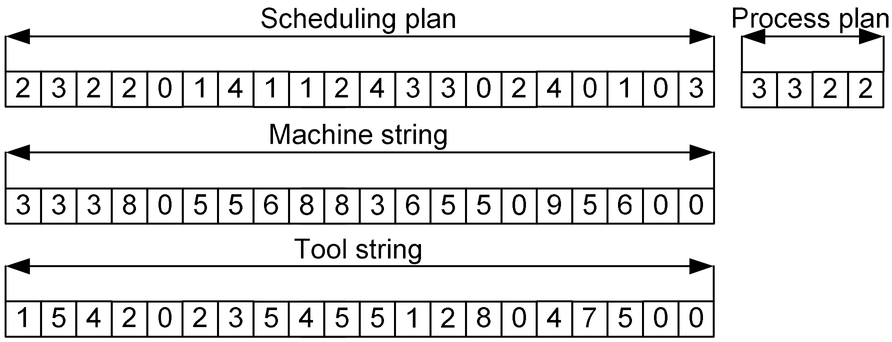


Figure 5: Individual for scheduling

Optimization agent: Crossover for scheduling

According to the defined crossover probability p_c, some particles are picked out for crossover. The crossover procedure for scheduling string is shown in Figure 6 and crossover procedure for machine string in Figure 7. The crossover for tool strings is performed analogously.

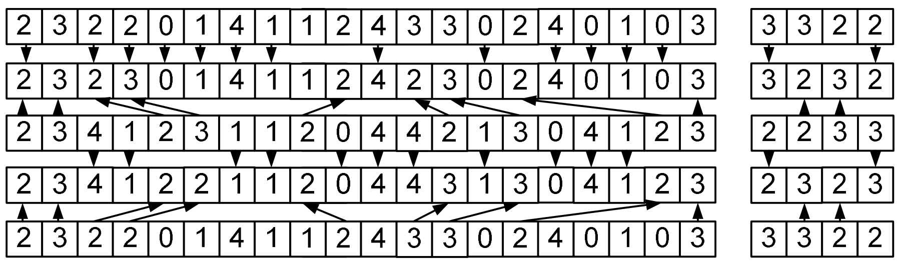


Figure 6: Crossover for scheduling plan

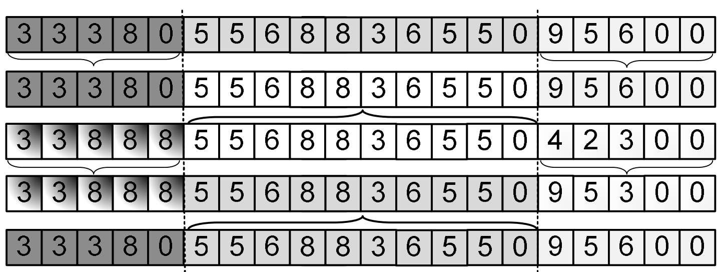


Figure 7: Crossover for machine string

Optimization agent: Mutation for scheduling

After crossover operation, according to the defined mutation probability p_m, some particles are randomly selected to be mutated. Mutation operator is used for generating new offspring by changing one job's alternative process plan.

Optimization agent: Modified PSO algorithm

In order to apply modified PSO algorithm, it is necessary to map operation-machine-tool sequence into mPSO particle on the following way: all the numbers from the first positions in the Genes are set in the operation particle, numbers from the second position are set in the machine position particle, and numbers from the third position are set in the tool position particle. According to encoding/decoding procedure described in previous section and mapping of the particles, particle position and particle velocity are expressed by the following formulas:

$$V_{id\_m}^{t+1} = W \cdot V_{id\_m}^{t} + C_1 \cdot rand() \cdot (P_{ld\_m}^{t} - X_{id\_m}^{t}) + C_2 \cdot Rand() \cdot (P_{gd\_m}^{t} - X_{id\_m}^{t}) \quad (7)$$

$$X_{id\_m}^{t+1} = X_{id\_m}^{t} + V_{id\_m}^{t+1} \quad (8)$$

$$V_{id\_t}^{t+1} = W \cdot V_{id\_t}^{t} + C_1 \cdot rand() \cdot (P_{ld\_t}^{t} - X_{id\_t}^{t}) + C_2 \cdot Rand() \cdot (P_{gd\_t}^{t} - X_{id\_t}^{t}) \quad (9)$$

$$X_{id\_t}^{t+1} = X_{id\_t}^{t} + V_{id\_t}^{t+1} \quad (10)$$

where i is the iteration number; $V_{id\_m}^{t}$ and $V_{id\_m}^{t+1}$ are the velocities for machine particle i in generations t and $t+1$; $V_{id\_t}^{t}$ and $V_{id\_t}^{t+1}$ are the velocities for tool particle i in generations t and $t+1$; $X_{id\_m}^{t}$ and $X_{id\_m}^{t+1}$ represent the positions for machine particle i in generations t and $t+1$; $X_{id\_t}^{t}$ and $X_{id\_t}^{t+1}$ represent the positions for tool particle i in generations t and $t+1$; $P_{ld\_m}^{t}$ and $P_{gd\_m}^{t}$ are the local best and global best positions for machine particle i in generation t; $P_{ld\_t}^{t}$ and $P_{gd\_t}^{t}$ are the local best and global best position for tool particle i in generation t; W, C_1, C_2, *rand()*, and *Rand()* are the same as explained earlier.

Finally, the workflow of the proposed mPSO algorithm is shown in the Figure 8 as follows:

Initialize swarm size, maximum number of generation, W_{max}, W_{min}, C_1, and C_2;
Initialize a swarm of particles with random positions for machine and tool particles;
Evaluate each particle's fitness;
Initialize the global and the local best position values for machines and tools;
Repeat
    generation = generation + 1;
    generate next swarm by updating the velocities and positions of the particles;
    apply the crossover and the mutation operations;
    evaluate swarm;
    compute each particle's fitness;
    find new "gbest"/"pbest" and update "gbest" of the swarm and "pbest" of each particle;
Until the maximum of generation is not met

Figure 8: The pseudocode of mPSO algorithm

4. EXPERIMENTAL RESULTS

The first phase of experiment considers optimization of flexible process planning with machine flexibility, tool flexibility, process flexibility, and sequence flexibility (see jobs in Figure 1). The transportation time matrix (the time units of transportation time are the same as the units of processing time) between the machines is given in Table 1 and parameters for mPSO algorithm are given in Table 2. Algorithm is implemented in Matlab environment and executed on a personal computer with a 3.10 GHz processor (2GB RAM). The objective of process planning optimization is to find optimal flexible process plans with the maximum fitness function *f(i,t),* equation (2). As a result of optimization, three alternative process plans for all four jobs are generated. The three near optimal alternative process plans, their fitnesses and production times obtained by using mPSO algorithm are gathered in Table 3.

Machine no.	1	2	3	4	5	6	7	8	9
1	0	50	79	36	99	106	130	116	102
2	50	0	31	16	51	56	78	67	54
3	79	31	0	47	20	27	63	48	26
4	36	16	47	0	67	70	90	84	70
5	99	51	20	67	0	7	55	40	22
6	106	56	27	70	7	0	62	47	29
7	130	78	63	90	55	62	0	15	37
8	116	67	48	84	40	47	15	0	22
9	102	54	26	70	22	29	37	22	0

Table 1: The transportation time between the machines for experiment

Parameters	Process planning	Scheduling
The size of the population, S	40	500
Total number of generation, M	30	100
The inertia weight *W*	1,2-4,0	1,2-4,0
Acceleration constants C_1 and C_2	2,0	2,0
Probability of crossover operation, p_c	0,60	0,80
Probability of mutation operation, p_m	0,10	0,10

Table 2: mPSO parameters for process planning and scheduling

Job	Alternative process plans	Fitness	Production time
1	(1,3,1)-(2,3,5)-(4,8,2)-(5,8,1)-(8,8,2)	0,0071	140
	(1,3,1)-(2,3,5)-(4,8,2)-(5,8,7)-(8,8,2)	0,0070	142
	(1,3,1)-(2,3,5)-(3,3,4)-(8,8,2)	0,0069	144
2	(1,5,1)-(2,5,3)-(3,6,5)-(4,8,4)-(8,8,5)	0,0069	145
	(1,5,1)-(2,5,3)-(3,6,5)-(4,8,4)-(8,8,8)	0,0068	147
	(1,5,2)-(2,5,3)-(3,6,5)-(4,8,4)-(8,8,5)	0,0067	149
3	(1,3,5)-(2,6,1)-(3,5,2)-(8,5,3)	0,0095	105
	(1,3,5)-(2,6,1)-(3,5,2)-(8,5,8)	0,0093	107
	(1,3,5)-(2,6,7)-(3,5,2)-(8,5,3)	0,0091	110
4	(1,9,4)-(2,5,7)-(8,6,2)	0,0101	99
	(1,9,4)-(2,5,7)-(8,6,5)	0,0099	101
	(5,4,7)-(6,2,5)-(7,3,8)-(8,6,2)	0,0078	128

Table 3: Experimental results of process planning

The second phase of experiment considers optimization of scheduling plans using parameters also given in Table 2. The optimization starts with randomly selecting one of the three alternative process plans for each job given in Table 3. Proposed mPSO algorithm for scheduling then generates optimal schedule plan i.e. job-machine sequence in accordance with objective function *object1,* equation (3). Figure 7 illustrates the convergence curve, which shows the search capability and evolution speed of mPSO algorithm. As it can be seen, the optimized schedule for a minimized makespan can be achieved after nearly 30 generations. Gantt chart for the optimal schedule solution is shown in Figure 10, where the maximum completion time of all the jobs in the schedule is 190.

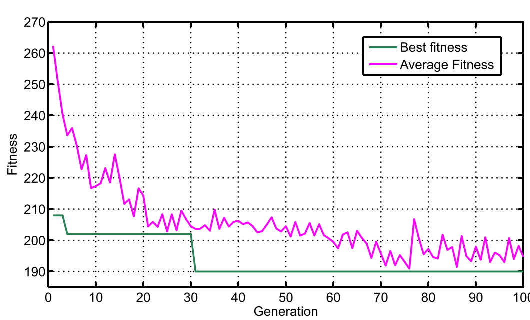


Figure 9: Convergence curve (best and average fitness values in each generations) for scheduling

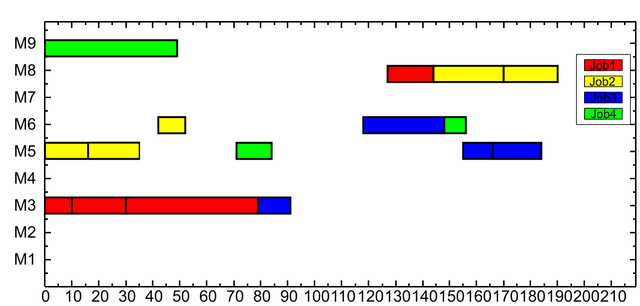


Figure 10: Gantt chart for experiment based on minimizing makespan values of the schedules (makespan = 190)

5. CONCLUSION

In this paper it has been presented methodology for integration of process planning and scheduling problem. The proposed methodology is based on multi-agent concept and modified particle swarm optimization (mPSO) algorithm. Four intelligent agents such as job agent, machine agent, tool agent and optimization agent collaborate together in order to obtain optimal solution of proposed combinatorial problem. The network representation for jobs is adopted to describe machine flexibility, tool flexibility, process flexibility as well as sequence flexibility. Solutions of the integration problem are encoded into PSO particles to intelligently search for the optimal solution for process plans and schedules. To explore the search space and make more effective information exchange mechanism for particles, new crossover and mutation operators were proposed and incorporated in a modified PSO algorithm. Optimal operation sequence is found in accordance with minimal production time as criteria (contains processing time and transportation time, where processing time depends on alternative machine and alternative tool selection) and optimal schedule sequence is found in accordance with minimal makespan as criteria. The experimental results show that the proposed method is a promising in the research of integration of process planning and scheduling.

6. 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* (2011-2014).

7. REFERENCES

1. Morad, N., Zalzala, A., Genetic algorithms in integrated process planning and scheduling*,* Journal of Intelligent Manufacturing, Vol. 2, No. 10, (1999), pp. 169–179.
2. Kim, Y.K., Park, K., Ko, J., A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling*,* Computers and Operations Research, Vol. 30, No. 8, (2003), pp. 1151-1171.
3. Shao, X.Y., Li, X.Y., Gao, L., Zhang, C.Y., Integration of process planning and scheduling-a modified genetic algorithm-based approach*,* Computers and Operations Research, Vol. 36, No. 6, (2009), pp. 2082-2096.
4. Li, W.D., 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, (2007), pp. 80-95.
5. Li, X., Zhang, C., Gao, L., Li, W., Shao, X., An agent-based approach for integrated process planning and scheduling, Expert Systems with Applications, Vol. 37, No. 2, (2010) pp. 1256-1264.
6. Wong, T.N., Leung, C.W., Mak, K.L., Fung, R.Y.K., An agent-based negotiation approach to integrate process planning and scheduling, International Journal of Production Research, Vol. 44, No. 7, (2006), pp. 1331-1351.
7. Wong, T.N., Leung, C.W., Mak, K. L., Fung, R.Y.K., Integrated process planning and scheduling/rescheduling-an agent-based approach*,* International Journal of Production Research, Vol. 18–19, No. 44, (2006), pp. 3627–3655.
8. Rajabinasab, A., Mansour, S., Dynamic flexible job shop scheduling with alternative process plans: an agent-based approach, The International Journal of Advanced Manufacturing Technology, Vol. 9-12, No. 54, (2011), pp. 1091-1107.
9. Monostori, L., Váncza, J., Kumara, S.R.T., Agent-based systems for manufacturing, Annals of the CIRP, Vol. 55, No. 2, (2006), pp. 672–720.