APEM journal
Advances in Production Engineering & Management
Volume 11 | Number 2 | June 2016 | pp 77–92
http://dx.doi.org/10.14743/apem2016.2.211
ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

A simulation approach to the process planning problem using a modified particle swarm optimization

Wang, J.F.[a,*], **Kang, W.L.**[a], **Zhao, J.L.**[a], **Chu, K.Y.**[a]

[a]Department of Mechanical Engineering, North China Electric Power University, Baoding, China

A B S T R A C T

Due to the complexity and variety of practical manufacturing conditions, computer-aided process planning (CAPP) systems have become increasingly important in the modern production system. In CAPP, the process planning (PP) problem involves two tasks: operation determining and operation sequencing. To optimize the process plans generated from complex parts, the traditional particle swarm optimization (PSO) algorithm is modified. Efficient encoding and decoding population initialization methods have been developed to adapt the PP problem for the PSO approach. In addition, to avoid the proposed approach becoming trapped in local convergences and achieving local optimal solutions, parameters are set to control the iterations. Several extended operators for the different parts of the particles have been incorporated into the traditional PSO. Simulation experiments have been run to evaluate and verify the effectiveness of the modified PSO approach. The simulation results indicate that the PP problem can be more effectively solved by the proposed PSO approach than other approaches.

A R T I C L E I N F O

Keywords:
Process planning
Operation determining
Operation sequencing
Particle swarm optimization
Extended operator

**Corresponding author:*
wjf266@163.com
(Wang, J.F.)

Article history:
Received 29 January 2016
Revised 27 March 2016
Accepted 10 May 2016

1. Introduction

In the modern computer-integrated manufacturing system (CIMS), the CAPP system plays an important role [1]. Generally, process planning involves two activities: operation determining and operation sequencing. In CAPP systems, these two activities must be executed simultaneously to achieve a good solution. Thus far, some effort has been made to address this problem, for instance, by designing a more feasible mathematical model or developing a more efficient approach. The application of some artificial intelligence approaches in the PP problem has especially promoted the development of CAPP technology. These approaches can be categorized as the genetic algorithm (GA) [1, 2], simulated annealing (SA) [2, 3], tabu search (TS) [4, 5], ant colony optimization (ACO) [6-8], particle swarm optimization (PSO) [9-13], honey bees mating optimization [14], and hybrid approaches [2, 15].

In 1995, Kennedy and Eberhart proposed the PSO algorithm [16]. The PSO is a new swarm optimization approach that can optimize engineering problems in aspects such as turning process modelling [17], assembly sequence planning [18], and process parameter optimization (TSP) [19].

The search for applications of PSO in the PP problem was first introduced by Guo et al. [9]. The process plan particle encoded/decoded strategy and some modified operators were designed. Kafashi et al. [10] optimized the setup planning using cost indices based on constraints such as the TAD (tool approach direction), the tolerance relation between features, and the feature precedence relations. Wang et al. [11] proposed an innovative process plan representation in which operation selection and operation sequencing were introduced simultaneously in a particle. Two local search operators are incorporated into the traditional PSO to achieve the optimal solution. Li et al. [12] modified the traditional PSO for the process planning problem. Miljković et al. [13] adopted the AND/OR network representation to describe the flexibility of the machine, tool, TAD, process and sequence and a performed multi-objective optimization procedure for the minimization of the production time and production cost using a modified PSO algorithm on this representation.

Zhang et al. [1] proposed a GA for a novel CAPP model. Li et al. [2] incorporated an SA into a GA to improve its searching efficiency. Ma et al. [3] modeled the PP problem as a combinational optimization problem with constraints. An entire solution space is constructed in reference to precedence constraints among operations. An SA algorithm is then proposed to address the PP problem. Li et al. [4] applied a TS-based approach to address the PP problem. In this approach, a mapping relationship is established between process constraints among features and precedence constraints among operations. Lian et al. [5] proposed a multi-dimensional tabu search (MDTS) approach to address the PP problem. Some local search strategies for different parts have been integrated into this TS approach. Liu et al. [6] constructed a mathematical model for the PP problem by considering the process constraints and optimization objectives. The ACO approach has been developed to optimize the PP problem based on this mathematical model. Wang et al. [7] represented the PP problem by an improved directed/undirected graph. A two-stage approach based on ACO was developed to optimize the process plans on the directed/undirected graph. Wen et al. [14] proposed a new method based on the honey bees mating optimization (HBMO) approach to optimize the PP problem. The solution encoding, crossover operator, and local search strategies were developed according to the characteristics of the PP problem. Huang et al. [15] designed a hybrid algorithm combining a graph and the GA to optimize process plans. The precedence constraints are mapped to an operation precedence graph on which an improved GA was applied to solve the PP problem.

Although there have been some significant improvements in solving the PP problem, there still remains the potential for further improvement [20]. Up to now, some heuristics or evolutionary approaches have been applied to optimize the PP problem, but the major difficulty is that the search space is too large for parts with complex features to find optimal solutions efficiently. To address this problem, a traditional PSO approach is modified to solve the PP problem in this paper. The main work includes the following two aspects:

- The representation of a process plan mapped to a particle is modified to facilitate the discrete PP problem. A new particle encoding/decoding strategy is adapted to make the search more efficient.
- A diverse searching mechanism has been adopted to improve the performance of the PSO approach. To avoid local convergence, some modifications to the traditional PSO approach have been adopted to better explore the solution space. Several operators for the different parts of the particles have been incorporated into the traditional PSO.

Section 2 describes the process planning model. Section 3 introduces the particle swarm optimization approach. Section 4 introduces an application of the modified PSO approach to the PP problem. Section 5 presents the simulation results of the proposed PSO algorithm. Finally, some conclusions and outlook are given in Section 6.

2. Problem modelling

2.1 Problem description

In the PP problem, two tasks have to be performed, namely, operation determining and operation sequencing. For operation determining, the method of mapping from features to operations is widely used in the PP problem. The attributes of each feature determines the corresponding machining methods, which can be expressed by the alternative operations. The combination of machines, cutting tools, and tool approach directions (TAD) comprise all of the different operation types (OPTs) for a feature of a part, which will be selected to determine the final process plans [1, 11].

For an operation, there are a set of OPTs under which the operation can be executed. Accordingly, an integrated process plan can be represented as follows.

$$PP = \{OP_1, OP_2, \dots, OP_i, \dots, OP_n\} \tag{1}$$

$$OP_i = \{OPT_{i1}, OPT_{i2}, \dots, OPT_{ij}, \dots, OPT_{im}\} \tag{2}$$

$$OPT_{ij} = \{M_{ij}, T_{ij}, TAD_{ij}\} \tag{3}$$

OP_i is the i-th operation, and OPT_{ij} is the j-th alternative selection of the operation OP_i. M_{ij}, T_{ij} and TAD_{ij} are the ID of the selected machine, tool and TAD for operation OPT_{ij}, respectively.

An example part in Fig. 1 is used to demonstrate the representation of a process plan [7]. The candidate operations are listed in Table 1.

In addition to operation determining, operation sequencing is another task in the PP problem. The operations should be sequenced under the conditions of satisfying the precedence constraints among operations [4, 6, 12, 15]. The constraints of Part 1 are shown in Table 2.

According to the precedence constraints in Table 2, an available process plan for Part 1 is shown in Table 3.

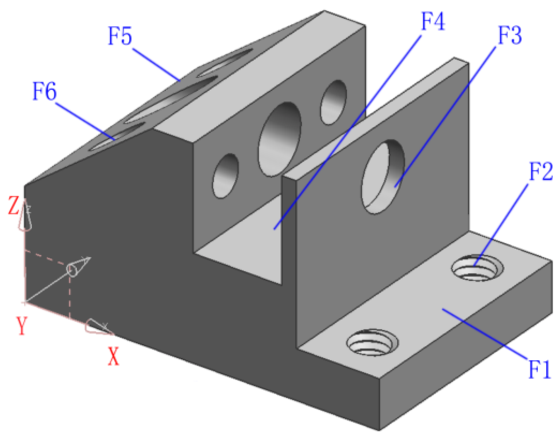

Fig. 1 An example part – Part 1

Table 1 Candidate operations for Part 1

Feathers	Operations	Operation types	Machines	Tools	TADs	Description
F1	Milling (OP_1)	OPT_{11}, OPT_{12}	M2	T1	+X, +Z	M1: Drilling press M2: Vertical milling machine T1: Milling cutter T2: Drill1 T3: Tapping tool T4: Drill2 T5: Reamer1 T6: Slot cutter T7: Chamfer cutter T8: Drill3 T9: Reamer2
F2	Drilling (OP_2)	OPT_{21}, OPT_{22}	M1, M2	T2	−Z	
	Tapping (OP_3)	OPT_{31}, OPT_{32}		T3		
F3	Drilling (OP_4)	OPT_{41}, OPT_{42}	M1, M2	T4	−X	
	Reaming (OP_5)	OPT_{51}, OPT_{52}		T5		
F4	Milling (OP_6)	OPT_{61}	M2	T6	+Z	
F5	Milling (OP_7)	OPT_{71}, OPT_{72}	M2	T7	+Y, −Z	
F6	Drilling (OP_8)	OPT_{81}, OPT_{82}	M1, M2	T8	+X	
	Reaming (OP_9)	OPT_{91}, OPT_{92}		T9		

Table 2 Constraints for Part 1 [7]

Operations	Precedence constraint description	Constraint types
OP_1	OP_1 is prior to OP_2 and OP_3.	Hard
	OP_1 is prior to OP_4 and OP_5.	Soft
OP_2	OP_2 is prior to OP_3.	Hard
OP_4	OP_4 is prior to OP_5.	Hard
OP_4, OP_5	OP_4 and OP_5 are prior to OP_6.	Hard
OP_6	OP_6 is prior to OP_2 and OP_3.	Hard
OP_8	OP_8 is prior to OP_9.	Hard
OP_8, OP_9	OP_8 and OP_9 are prior to OP_7.	Hard

Table 3 Available process plan for Part 1

Operation	Machine	Tool	TAD
OP_1	M2	T1	+X
OP_8	M1	T8	+X
OP_9	M1	T9	+X
OP_4	M1	T4	−X
OP_5	M1	T5	−X
OP_6	M2	T6	+Z
OP_7	M2	T7	−Z
OP_2	M1	T2	−Z
OP_3	M1	T3	−Z

2.2 Mathematical model

The criterion of minimizing production costs (CP) is usually used to evaluate the process plan. The CP includes the machine cost (CM), cutting tool cost (CT), machine-changing cost (CMC), cutting tool changing cost (CTC), and set-up cost (CSC) [2, 3, 6, 8, 11, 14, 15].

The machine cost is

$$CM = \{cm_1, cm_2, \dots, cm_i, \dots, cm_{N_M}\} \tag{4}$$

where N_M is the number of machines.

The cutting tool cost is

$$CT = \{ct_1, ct_2, \dots, ct_i, \dots, ct_{N_T}\} \tag{5}$$

where N_T is the number of cutting tools.

As shown in Eq. 2 and Eq. 3, an operation is selected from several alternative OPTs. The machine cost for an operation varies according to the alternative OPTs, so the machine cost CM_{ij} for an OPT OPT_{ij} can be given as

$$CM_{ij} = cm_{M_{ij}} \tag{6}$$

where M_j is explained in Eq. 3.

The cutting tool cost TC_{ij} for an OPT OPT_{ij} can be given as

$$CT_{ij} = ct_{T_{ij}} \tag{7}$$

where T_{ij} is explained in Eq. 3.

The machine changing cost $CMC_{iji'j'}$ between OPT OPT_{ij} and OPT $OPT_{i'j'}$ can be given as

$$CMC_{iji'j'} = \Phi(M_{ij}, M_{i'j'}) \times C^{cm} \tag{8}$$

where C^{cm} is the cost of machine changing, which is considered to be the same for each machine change. $\Phi(X,Y)$ can be calculated as follows:

$$\Phi(X,Y) = \begin{cases} 1 & X \neq Y \\ 0 & X = Y \end{cases} \tag{9}$$

The cutting tool changing cost $CTC_{iji'j'}$ between OPT OPT_{ij} and OPT $OPT_{i'j'}$ can be given as

$$CTC_{iji'j'} = \Omega\left(\Phi\left(M_{ij}, M_{i'j'}\right), \Phi\left(T_{ij}, T_{i'j'}\right)\right) \times C^{ct} \tag{10}$$

where C^{ct} is the cutting tool changing cost, which is considered to be the same for each cutting tool change. $\Omega(X,Y)$ can be calculated as follows.

$$\Omega(X,Y) = \begin{cases} 0 & X = Y = 0 \\ 1 & otherwise \end{cases} \tag{11}$$

The set-up cost $CS_{iji'j'}$between OPT OPT_{ij} and OPT $OPT_{i'j'}$ can be given as

$$CS_{iji'j'} = \Omega\left(\Phi\left(M_{ij}, M_{i'j'}\right), \Phi\left(TAD_{ij}, TAD_{i'j'}\right)\right) \times C^{cs} \tag{12}$$

where TAD_{ij} is explained in Eq. 3, and C^{cs} is the cost for a set-up, which is considered to be the same for each set-up.

The definitions of machine changing, tool changing, and set-up changing have been explained in reference [2]. Based on the above analysis, the mathematical model of the PP problem is formulated as follows [6]:

Objectives:

(i) A combination of *CM, CT, CMC, CTC*, and *CS* will be considered as *CP*, and minimizing *CP* is the objective of PP.

$$\text{Min } CP = \sum_{i=1}^{n}\sum_{j=1}^{m} u_{ij}.(\omega_1.CM_{ij} + \omega_2.CT_{ij}) + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\substack{i'=1 \\ i'\neq i}}^{n}\sum_{\substack{j'=1 \\ j'\neq j'}}^{m} v_{iji'j'}.\left(\omega_3.CMC_{iji'j'} + \omega_4.CS_{iji'j'} + \omega_5.CTC_{iji'j'}\right) + C^{sc} \tag{13}$$

Subject to:

(ii) *n* operations have to be selected while machining a part.

$$\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ij} = n \tag{14}$$

(iii) For an operation, one and only one OPT can be selected from its *m* alternative OPTs.

$$\sum_{j=1}^{m} u_{ij} = 1 \ \ (i = 1, 2, \dots, n) \tag{15}$$

(iv) For a process plan consisting of *n* operations, the numbers of operation changes is *n* – 1, and the changes of combinations of machines, cutting tools, and set-ups are accompanied by operation changes.

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\substack{i'=1 \\ i'\neq i}}^{n}\sum_{\substack{j'=1 \\ j'\neq j'}}^{m} v_{iji'j'} = n - 1 \tag{16}$$

(v) For each OPT, only one adjacent operation is lined up before it.

$$\sum_{i=1}^{n}\sum_{j=1}^{m} v_{iji'j'} \leq 1 \tag{17}$$

(vi) For each OPT, only one adjacent operation is lined up after it.

$$\sum_{i'=1}^{n}\sum_{j'=1}^{m} v_{iji'j'} \leq 1 \tag{18}$$

(vii) u_{ij} is an integer enumeration variable.

$$u_{ij} = \begin{cases} 1 & \text{if } \mathrm{OPT}_{ij} \text{ is selected for operation } OP_i \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

(viii) $v_{iji'j'}$ is an integer enumeration variable.

$$v_{iji'j'} = \begin{cases} 1 & \text{if } \mathrm{OPT}_{i'j'} \text{is selected after } \mathrm{OPT}_{ij} \text{ is executed} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

Eq. 13 means that the objective function of the PP is to minimize the total production cost. The constraints are in Eq. 14 to Eq. 20. Constraints in Eq. 14 and Eq. 15 ensure that all the operations are carried out. The constraint in Eq. 16 ensures that $n - 1$ changing costs of machines, cutting tools, and set-ups are added into the PC among n operations. Constraints in Eq. 17 and Eq. 18 ensure that every operation except the first and the last have only one adjacent operation on each side.

3. Introduction of the particle swarm optimization approach

PSO is a swarm optimization approach [16]. Every particle in the population represents an N-dimensional solution that constructs a search space for every particle. The particles fly freely to search for the optimal position at a given velocity. Hence, for the particle i, the vectors X_i^t and V_i^t at the t-th iteration can be denoted as $X_i^t = \{X_{i1}^t, X_{i2}^t, \dots, X_{ik}^t, \dots, X_{iN}^t\}$ and $V_i^t = \{V_{i1}^t, V_{i2}^t, \dots, V_{ik}^t, \dots, V_{iN}^t\}$, respectively. With the emergence of the optimal position at iteration t + 1, the vectors X_i^t and V_i^t can be updated as follows:

$$V_{ik}^{t+1} = w * V_{ik}^t + c_1 * Rand(\quad) * \left(P_{ik}^t - X_{ik}^t\right) + c_2 * Rand(\quad) * \left(P_{gk}^t - X_{ik}^t\right) \tag{21}$$

$$X_{ik}^{t+1} = X_{ik}^t + V_{ik}^{t+1} \tag{22}$$

In Eq.21 and Eq.22, V_{ik}^{t+1} is the velocity on dimension k, and X_{ik}^{t+1} is the position on dimension k. P_{ik}^t is the local optimal position on dimension k, and P_{gk}^t is the global optimal position on dimension k. The weight w is used to control the iteration. The constants c_1 and c_2 control the balance between a local optimal position and the global optimal position. Rand () is limited in [0, 1]. The traditional PSO approach is carried out in four steps:

Step 1: Population initialization.
Step 2: Update the vectors X_i^t and V_i^t according to Eq. 21 and Eq. 22.
Step 3: Update P_{ik}^t and P_{gk}^t according to the performance of the population
Step 4: Loop to Step 2 until a termination condition is met.

4. The proposed PSO approach

4.1 Solution representation

It is necessary to modify the traditional PSO for the PP problem to include, for example, particle representation and the particle movement strategy. In applying the PSO approach to the PP problem, three tasks have to be performed, namely, particle encoding, particle validation and particle decoding. The first task is to encode a process plan to an appropriate particle. Because operation determining and operation sequencing have to be performed simultaneously in process planning, the particle structure should be considered to comprise the information of determining and sequencing operations. The details of a particle are listed in Table 4.

According to the definition of a particle, we modified a particle i of a $2 \times n$ matrix to represent a process plan at iteration t [11], i.e.

$$X_i^t = \begin{bmatrix} x_{i11}^t, x_{i12}^t, \dots, x_{i1n}^t \\ x_{i21}^t, x_{i22}^t, \dots, x_{i2n}^t \end{bmatrix} \tag{23}$$

The first row x_{i1n}^t is the Operation Determining (OD) part and represents the operation selection for each feature. The second row x_{i2n}^t is the Operation Sequencing (OS) part and represents the priority among operations. x_{i2n}^t is initialized randomly in the range of 0 and 1 according to the priority among operations, and x_{i1n}^t can be calculated by Eq. 24.

$$X_{i1n}^t = (a^2 * ix\_m + a * ix\_c + ix\_t)/a^3 \tag{24}$$

In Eq. 24, *ix_m*, *ix_c*, and *ix_t* are generated randomly from the candidate machine set, cutting tool set, and TAD set for executing the operation. a can be given by the equation

$$a = Max(N_M, N_T, N_S) + 1 \tag{25}$$

where N_S is the number of TADs.

To illustrate the particle encoding, the process plan in Table 3 is taken as an example and the corresponding encoding of x_{i1n}^t is shown in Fig. 2. For instance, the values of *ix_m*, *ix_c*, and *ix_t* are set to 1 if the OPTs of M2, T1 and +X, respectively, are selected to process operation OP_1. If the OPTs of M2, T1 and +Z are selected to process operation OP_1, the values of *ix_m*, *ix_c*, and *ix_t* will be 1, 1, and 3, respectively.

For the process plan in Table 3, a feasible particle is listed in Table 5. In Table 5, the first row means the operations. The second row represents the calculated value of its assigned OPT, and the third row represents the priority among the operations. The second task is to validate the particles to accord with the precedence constraints. A particle represents a process plan. Nevertheless, process plans generated by particles flying freely in solution space are usually invalid against the precedence constraints. To validate each particle, an $n \times n$ constraint matrix P is proposed to incorporate the precedence constraints among the operations into the particles.

Table 4 Detail of a particle

Data type	Variable	Description
Integer	*ix_o*	Id of an operation, which corresponds to the index of the operation set.
Integer	*ix_m*	Id of a machine, which corresponds to the index of the machine set.
Integer	*ix_c*	Id of a cutting tool, which corresponds to the index of the cutting tool set.
Integer	*ix_t*	Id of a TAD, which corresponds to the index of the TAD set.

Operation	OP_1	OP_8	OP_9	OP_4	OP_5	OP_6	OP_7	OP_2	OP_3
x_{i1n}^t	0.111	0.181	0.191	0.144	0.154	0.163	0.176	0.126	0.136
(*ix_m*, *ix_c*, *ix_t*)	(1, 1, 1)	(1, 8, 1)	(1, 9, 1)	(1, 4, 4)	(1, 5, 4)	(1, 6, 3)	(1, 7, 6)	(1, 2, 6)	(1, 3, 6)

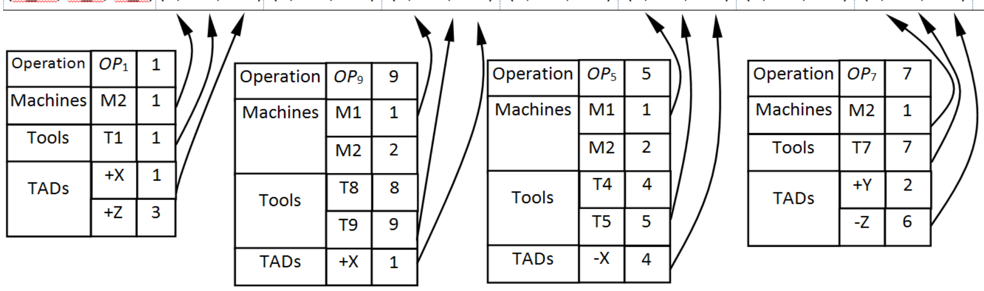

Fig. 2 Encoding of x_{i1n}^t for the process plan in Table 3

Table 5 A feasible particle for the process plan in Table 3

Operation	OP_1	OP_2	OP_3	OP_4	OP_5	OP_6	OP_7	OP_8	OP_9
x_{i1n}^t	0.111	0.126	0.136	0.144	0.154	0.164	0.176	0.181	0.191
x_{i2n}^t	1	0.50	0.40	0.85	0.75	0.70	0.60	0.95	0.90

The second task is to validate the particles to accord with the precedence constraints. A particle represents a process plan. Nevertheless, process plans generated by particles flying freely in solution space are usually invalid against the precedence constraints. To validate each particle, an $n \times n$ constraint matrix P is proposed to incorporate the precedence constraints among the operations into the particles.

$$P = (p_{ij})n \times n \tag{26}$$

$$P_{ij} = \begin{cases} 1 \;\; OP_i \text{ is prior to } OP_j \text{ or } i = j \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{27}$$

The precedence constraint matrix for the precedence constraints in Table 2 is shown as follows:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{28}$$

The third task is to decode a particle into a solution. According to the input of encoded position matrix, a particle can be decoded to obtain a process plan. The particle decoding includes the following two steps.

First, determine the machine, cutting tool and TAD according to the value x_{i1n}^t. The decoding approach for x_{i1n}^t is

$$y_{in} = x_{i1n}^t * a^3 \tag{29}$$

$$ix\_m_{in} = \lfloor y_{in}/a^2 \rfloor \tag{30}$$

$$ix\_c_{in} = \lfloor (y_{in} - ix\_m_{in} * a^2)/a \rfloor \tag{31}$$

$$ix\_t_{in} = y_{in} - ix\_m_{in} * a^2 - ix\_c_{in} * a \tag{32}$$

where y_{in} is an integer.

Second, sequence these determined operations according to the precedence value x_{i2n}^t. If the sequence of operations violates the precedence constraints, the operations should be sequenced again according to the precedence value x_{i2n}^t with the help of the precedence matrix P_m.

$$x_{i2n}^t = x_{i2n}^t \times P_m \tag{33}$$

Similarly, for particle i, a velocity matrix at iteration t can be represented as

$$V_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{ik}^t, \dots, v_{in}^t] \tag{34}$$

where v_{in}^t is initialized randomly in the range of –1 to 1.

4.2 Population initialization

The particle swarm is initialized in three steps:

(i) Set the population size P_{max} and the maximum iteration number N_{max}.
(ii) Initialize each particle. The initial position and velocity of each of the particles in the population are generated.
(iii) Decode every particle to the process plan according to Eq. 29 to Eq. 33, and then calculate CP. Get the local optimal position P_l^0 and the global optimal position P_g^0.

4.3 Iteration and control

For every selected particle, the position and velocity can be updated according to Eq. 21 and Eq. 22. To ensure the process plan validity, decoding every newly generated particle to the process plan is necessary. If the new process plans violate the precedence constraints, the corresponding particles will be normalized by using the constraint matrix P. For each valid particle, the CP of the corresponding process plan will be calculated. If a lower CP is achieved, the local optimal position P_l and the global optimal position P_g will be updated.

When the traditional PSO is applied in the PP problem, a quick local convergence at an early stage of the PSO usually has to be faced. The quick local convergence will make further exploration difficult and can generate an undesirable solution. To solve this problem, some modifications are suggested to enhance the performance of the traditional PSO algorithm [9, 10, 13]. Four operators are incorporated into the traditional PSO approach. Because the position of a particle is expressed as a 2 × n matrix, in which the first row is the OD part and the second row is the OS part, the operators for the different rows vary independently.

For the OD part, two types of mutation operator are designed to generate a new solution.

Mutation operator 1

One particle in the swarm is chosen for a mutation operation with a predefined probability (p_{ms}). First, for the OD part of this particle, one position point L is randomly selected. Second, decode the particle to a process plan, and obtain the machine, cutting tool, and TAD (M, T, TAD) of the Lth operation in this process plan. Third, from the machine set, cutting tool set, and TAD set of the Lth operation, an alternative selection of machine, cutting tool, and TAD is determined to replace the current machine, cutting tool, and TAD.

Mutation operator 2

One particle is chosen for a mutation operation with a predefined probability (p_{ss}). For the OD part of this particle, two adjacent position points L_1 and L_2 are randomly selected. Decode the particle to a process plan, and obtain the machines, cutting tools, and TADs (Ms, Ts, TADs) of the L_1-th and L_2-th operations in this process plan. If $M_{L_2} = M_{L_1}$, let $T_{L_1} = T_{L_2}$ or $TAD_{L_1} = TAD_{L_2}$, or otherwise the position points L_1, L_2 will be reselected.

With respect to the OS part, two operators are employed, crossover and shift.

Crossover operator

Two particles, A and B, are selected to execute a crossover operation with a predefined probability (p_{cq}). For the OS part of those two particles, one position point L is randomly determined. The OS part is divided into two parts. Subsequently, the value of the front part of OS_A is taken out and inserted before the cutting point of OS_B, and the value of the left part of OS_B is taken out and inserted before the cutting point of OS_A.

Shift operator

One particle is chosen for a shift operation with a predefined probability (p_{sq}). For the OS part of this particle, two position points L_1, L_2 are randomly selected, and their values $x_{i1L_1}^t$, $x_{i1L_2}^t$ exchanged.

4.4 Termination

If the max number of iterations N_{max} is reached, the iteration will be terminated. Decode the obtained particle position P_g^t to achieve the final process plan.

The flowchart is described in Fig. 3.

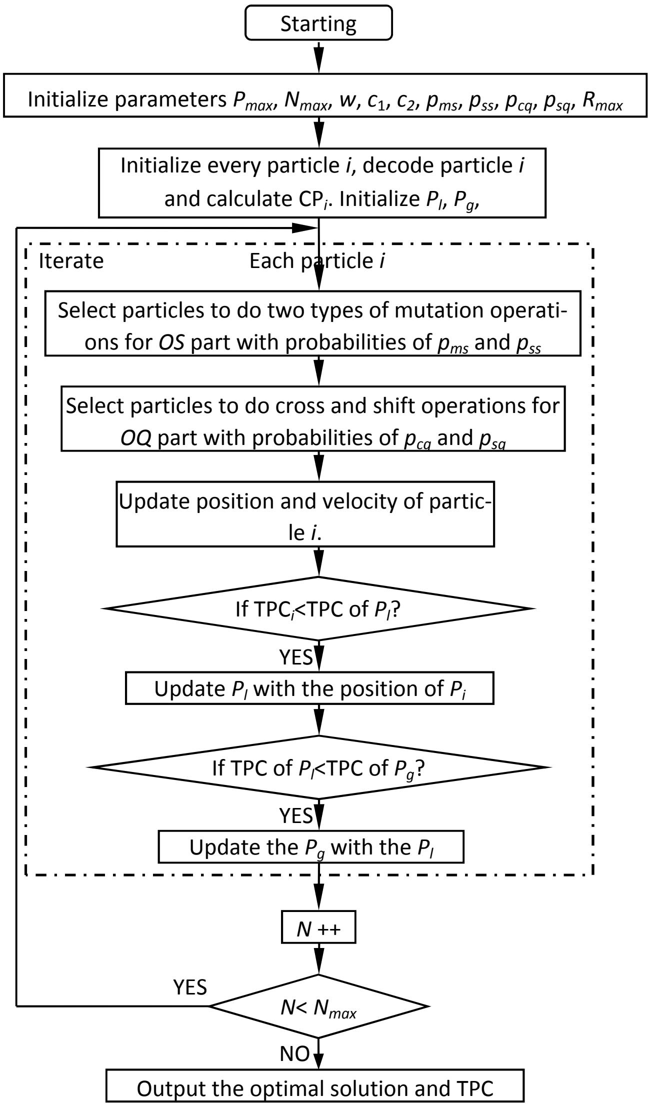

Fig. 3 Flowchart of the modified PSO approach

5. Experiments and results

Two characteristic parts are used for the simulation experiments. The first part is shown in Fig. 4 [1] (Part 2), and the second part is shown in Fig. 5 [2] (Part 3). The detailed information on Part 2 and Part 3 is introduced in the research of Li et al. [4]. All of the simulation experiments will be performed on a PC with 2.8 GHz and the Windows 7 operating system.

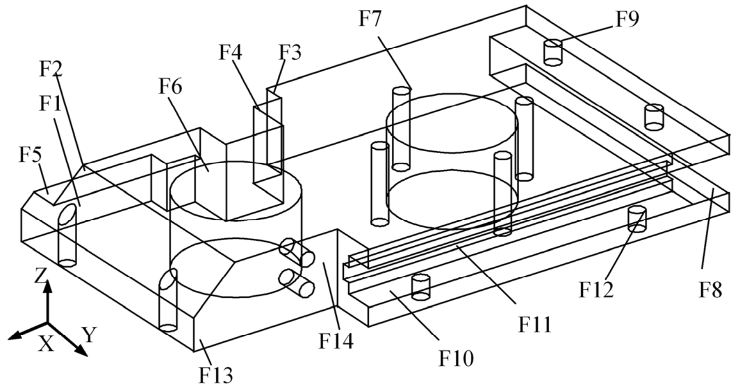

Fig. 4 An example part – Part 2

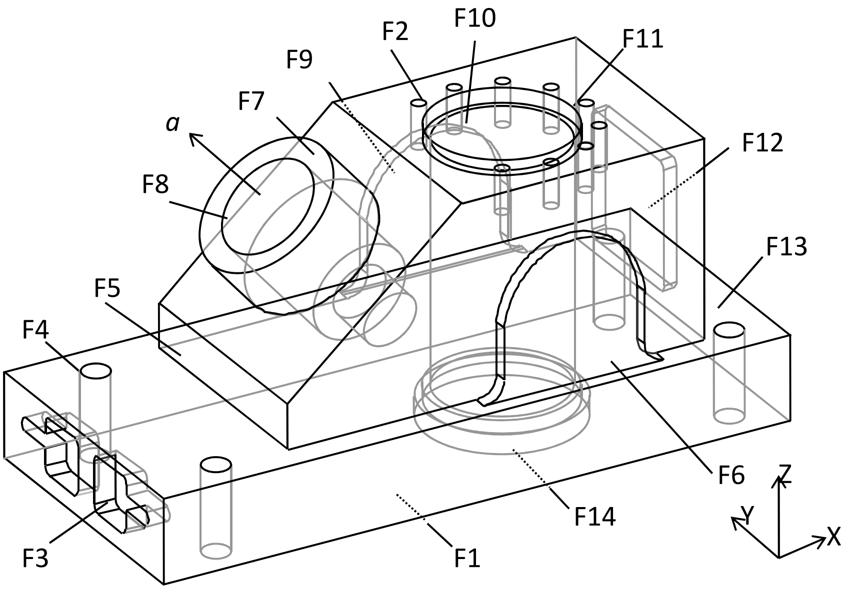

Fig. 5 An example part – Part 3

5.1 Simulation experiments

While applying PSO to solve the process planning problem, the key parameters have to be determined to facilitate the performance of the modified PSO. Accordingly, many preliminary simulation experiments have to be carried out to determine those parameters. The process planning problem for Part 2 is used to illustrate how the key parameters are determined. It is assumed that ω_1 to ω_5 in Eq. 13 are set as 1.

The key parameters may be analysed from two aspects, namely, the swarm characteristic parameters of PSO (P_{max}, N_{max}, w, c_1, c_2) and the problem data (p_{ms}, p_{ss}, p_{cq}, p_{sq}). The swarm size P_{max} and iteration number N_{max} will be increased with the increased complexity of the part. For Part 2, after many trials, P_{max} and N_{max} are fixed at 2000 and 300, respectively.

The constants c_1 and c_2 are used to balance the velocity tendency to the local optimal position P_l and the global optimal position P_g. If c_1 and c_2 are too large, the search space of the particles will be expanded, which may even lead to no convergence of the PSO. If c_1 and c_2 are too small, slow convergence may cause the computation time to be very long. In the case of problems with P_{max} = 2000, N_{max} = 300, w = 0.75, p_{ms} = 0.6, p_{ss} = 0.6, p_{cq} = 0.2, p_{sq} = 0.2, 50 trials were separately conducted by varying the values of $c_1 = c_2 \in \{1, 1.5, 2\}$. The average results are summarized in Fig. 6.

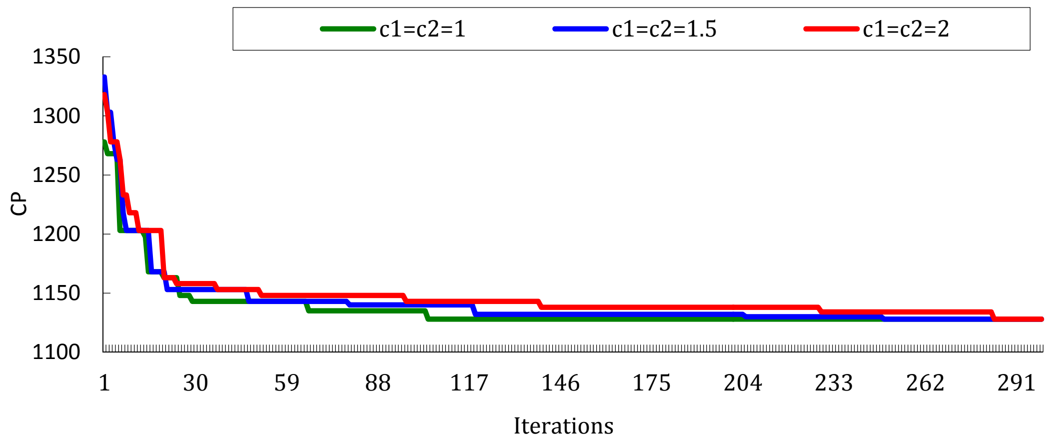

Fig. 6 *CP* of the proposed PSO with different constants $c_1$and c_2

From Fig. 6, when c_1 and c_2 are both set to be 1, the PSO algorithm shows fast convergence and good computational efficiency.

The inertia weight w is set to coordinate the local exploration and the global exploration. If w is too large, there may be a quick local convergence at an early stage of the PSO. If w is too small, the computation time for each iteration will be long, and the optimization rate will become very slow. In the case of problems with P_{max} = 2000, N_{max} = 300, c_1 = c_2 = 1, p_{ms} = 0.6, p_{ss} = 0.6, p_{cq} = 0.2, p_{sq} = 0.2, 50 trials were separately conducted by varying the values of $w \in \{0.75, 1, 1.25\}$. The average results are summarized in Fig. 7.

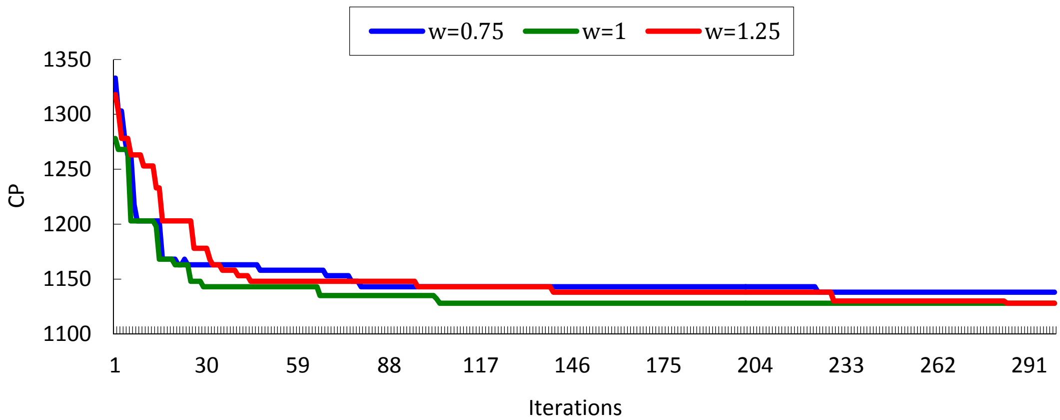

Fig. 7 *CP* of the proposed PSO with different inertia weights *w*

From Fig. 7, the optimal efficiency and stability are achieved under the condition of w = 1. The problem data (p_{ms}, p_{ss}, p_{cq}, p_{sq}) are determined to help the approach escape from local convergences. Fifty trials are conducted in8 group combinations of the four parameters p_{ms}, p_{ss}, p_{cq}, p_{sq} to illustrate the selection of these parameters. The average *CP*s for the 50 trials are listed in Table 6. It is shown that the combination of p_{ms} = 0.6, p_{ss} = 0.6, p_{cq} = 0.2 and p_{sq} = 0.2 can yield the best performance. In conclusion, the performance of the modified PSO for Part 2 is good, and P_{max} = 2000, N_{max} =300, w = 1, c_1 = c_2 = 1, p_{ms} = 0.6, p_{ss} = 0.6, p_{cq} = 0.2, p_{sq} = 0.2. The best simulation result and average simulation result are shown in Table 7 and Table 8, respectively.

Table 6 Determination of four probabilities of the modified PSO

	(Mutation1: p_{ms}, Mutation2: p_{ss}, Crossover: p_{cq}, Shift: p_{sq})							
	(0.6, 0.4, 0.2, 0.4)	(0.4, 0.6, 0.4, 0.2)	(0.6, 0.6, 0.4, 0.4)	(0.6, 0.6, 0.2, 0.2)	(0.4, 0.4, 0.2, 0.2)	(0.4, 0.4, 0.4, 0.4)	(0.4, 0.6, 0.2, 0.4)	(0.6, 0.4, 0.4, 0.2)
Mean	1198.5	1198.8	1308.4	1131.8	1136.6	1541.3	1206.1	1186.7
Maximum	1263	1318	1713	1163	1188	1998	1318	1263
Minimum	1143	1143	1148	1128	1128	1158	1143	1143

Table 7 Best simulation result for Part 2

Operation	Machine	Tool	TAD	Result
6	2	2	−Z	CM = 490
1	2	1	−Z	CT = 98
7	2	1	−Z	CTC = 60
9	2	1	−Z	CS = 480
12	2	1	−Z	CP= 1128
5	2	5	−Z	
3	2	5	+Y	
4	2	5	+Y	
8	2	5	+X	
10	2	5	−Y	
11	2	5	−Y	
13	2	5	−Y	
14	2	1	−Y	
2	2	8	−Y	

Table 8 Average simulation result of 50 trials for Part 2

Type	Mean	Maximum	Minimum	Standard deviation
CM	490	490	490	0
CT	98.2	103	98	0.98
CMC	0	0	0	0
CTC	60.9	75	60	3.56
CS	480	480	480	0
CP	1128.9	1143.0	1128	3.56

The above method of determining the key parameters is based on Part 2. The method of choosing parameters for Part 3 is same as for Part 2. In the case of problems with P_{max} = 2000, N_{max} = 500, w = 1.25, $c_1 = c_2$ = 1, p_{ms} = 0.6, p_{ss} = 0.6, p_{cq} = 0.3, p_{sq} = 0.3, 50 trials were separately conducted. The best process plans are listed in Table 9, and the average results are listed in Table 10.

Table 9 Best simulation result for Part 3

Operation	Machine	Tool	TAD	Result
1	2	6	+Z	CM = 770
3	2	6	+X	CT = 235
5	2	6	+X	CMC = 320
6	2	6	−Z	CTC = 200
2	2	6	−Z	CS = 1000
18	2	6	−Z	CP = 2525
11	2	7	−Z	
12	2	2	−Z	
13	2	9	−Z	
17	2	7	−X	
7	2	7	−a	
8	2	2	−a	
9	2	9	−a	
19	2	9	+Z	
14	4	10	−Z	
20	4	10	+Z	
10	4	10	−a	
4	1	2	−Z	
15	1	1	−Z	
16	1	5	−Z	

Table 10 Average simulation result of 50 trials for Part 3

Type	Mean	Maximum	Minimum	Standard deviation
CM	770	770	770	0
CT	240	267	235	10.13
CMC	320	320	320	0
CTC	197.2	180	200	6.94
CS	1000	1000	1000	0
CP	2527.2	2535.0	2525.0	3.28

5.2 Extensive comparative experiments

The following three conditions are used to verify the modified PSO approach for the example parts [2, 4, 6]:

(a) All machines and cutting tools are available, and ω_1 to ω_5 in Eq. 13 are set as 1.
(b) All machines and cutting tools are available, and ω_1 = ω_5 = 0, ω_1 = ω_3 = ω_4 = 1.
(c) M_2 and T_7 are down, ω_2 = ω_5 = 0, ω_1 = ω_3 = ω_4 = 1.

For Part 2, 50 trials were performed under conditions (a) and (b). A penalty cost is included in the *CP* to facilitate the comparison with other approaches, which is 200 for Part 2 [4, 7]. A comparison with the results obtained using the GA and SA approaches [2], TS [4], and HBMO [14], as well as the two-stage ACO [7], is provided in Table 11.

Under condition (a), this approach is same as SA, TS, HBMO, and two-stage ACO and is better than GA using the minimum machine costs. Using the maximum machine costs, the *CP* (1328.0) is the same as that of HBMO, and it is better than the other four approaches. This approach has the best performance in the mean machine cost (1328). It is obvious that the same CP (1328.0) is achieved 50 times and is superior to all of the other approaches.

Under condition (b), the performance of this approach is the same as those of HBMO and the two-stage ACO. Using the minimum machine costs, this approach is better than GA, and it is the same as the other four approaches. Using the maximum and mean machine costs, this approach is better than GA, SA, and TS.

For Part 3, 50 trials were carried out under conditions (a), (b), and (c). A comparison of the results with those of TS [4], PSO [9], and HBMO [14], as well as the two-stage ACO [7] is provided in Table 12.

Table 11 Results compared to other approaches

Condition	Proposed approach	GA	SA	TS	HBMO	Two-stage ACO
(a)						
Mean	1328	1611.0	1373.5	1342.6	1328	1329
Maximum	1328	1778	1518	1378	1328	1348
Minimum	1328	1478	1328	1328	1328	1328
(b)						
Mean	1170	1482	1217	1194	1170	1170
Maximum	1170	1650	1345	1290	1170	1170
Minimum	1170	1410	1170	1170	1170	1170

Table 12 Results compared to other approaches

Condition	Proposed approach	TS	PSO	HBMO	Two-stage ACO
(a)					
Mean	2527.2	2609.6	2680.5	2543.5	2552.4
Maximum	2535	2690	-	2557	2557
Minimum	2525	2527	2535	2525	2525
(b)					
Mean	2093.0	2208.0	-	2098.0	2120.5
Maximum	2120	2390	-	2120	2380
Minimum	2090	2120	-	2090	2090
(c)					
Mean	2593.2	2630.0	-	2592.4	2600.8
Maximum	2600	2740	-	2600	2740
Minimum	2590	2580	-	2590	2590

Under condition (a), the minimum machine cost is the same as that of HBMO and the two-stage ACO, and it is better than TS and PSO. Among the 50 trial results, *CP* (2525) occurs 23 times, *CP* (2527) occurs 20 times, and *CP* (2535) occurs 7 times. Accordingly, this approach is superior to all of the other approaches on the mean machine cost. Under condition (b), the minimum machine cost (2090) is the same as that of HBMO and the two-stage ACO. *CP* (2090) occurs 45 times, and *CP* (2120) occurs 5 times in 50 trials; the mean machine cost is the best among all of the approaches. Under condition (c), *CP* (2590) occurs 34 times, and *CP* (2600) occurs 16 times in 50 trials. Generally, the performance of this approach is similar to that of HBMO under condition (c) and is superior to those of the two-stage ACO and TS.

6. Conclusions

A traditional PSO approach is modified to solve the PP problem. Efficient encoding and decoding, population initialization, and iteration and control methods have been designed. Meanwhile, to avoid local convergence, several new operators for the different parts of the particles have been designed and incorporated into the traditional PSO to improve the particles' movements. Simulation experiments show that the modified PSO algorithm can perform the process plan optimization competently and consistently and generate better solutions compared with other approaches.

In the simulation experiment, a small change in the parameters induces computational result saltation. Hence, a deep discussion of selecting the modified PSO approach parameters will be conducted. Additionally, integrated process planning and scheduling that considers the better performance of manufacturing systems may be a direction for further study [21, 22].

References

[1] Zhang, F., Zhang, Y.F., Nee, A.Y.C. (1997). Using genetic algorithms in process planning for job shop machining, *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 4, 278-289, doi: 10.1109/4235.687888.

[2] Li, W.D., Ong, S.K., Nee, A.Y.C. (2002). Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts, *International Journal of Production Research*, Vol. 40, No. 8, 1899-1922, doi: 10.1080/00207540110119991.

[3] Ma, G.H., Zhang, Y.F., Nee, A.Y.C. (2000). A simulated annealing-based optimization algorithm for process planning, *International Journal of Production Research*, Vol. 38, No. 12, 2671-2687, doi: 10.1080/002075400411420.

[4] Li, W.D., Ong, S.K., Nee, A.Y.C. (2004). Optimization of process plans using a constraint-based tabu search approach, *International Journal of Production Research*, Vol. 42, No. 10, 1955-1985, doi: 10.1080/00207540310001652897.

[5] Lian, K.L., Zhang, C.Y., Shao, X.Y., Zeng, Y.H. (2011). A multi-dimensional tabu search algorithm for the optimization of process planning, *Science China Technological Sciences*, Vol. 54, No. 12, 3211-3219, doi: 10.1007/s11431-011-4594-7.

[6] Liu, X.-J., Yi, H., Ni, Z.-H. (2013). Application of ant colony optimization algorithm in process planning optimization, *Journal of Intelligent Manufacturing*, Vol. 24, No. 1, 1-13, doi: 10.1007/s10845-010-0407-2.

[7] Wang, J.F., Wu, X., Fan, X. (2015). A two-stage ant colony optimization approach based on a directed graph for process planning, *The International Journal of Advanced Manufacturing Technology*, Vol. 80, No. 5, 839-850, doi: 10.1007/s00170-015-7065-7.

[8] Hu, Q., Qiao, L., Peng, G. (2015). An ant colony approach to operation sequencing optimization in process planning, (In press), *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, doi: 10.1177/0954405415616786.

[9] Guo, Y.W., Mileham, A.R., Owen, G.W., Li, W.D. (2006). Operation sequencing optimization using a particle swarm optimization approach, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 220, No. 12, 1945-1958.

[10] Kafashi, S., Shakeri, M., Abedini, V. (2012). Automated setup planning in CAPP: a modified particle swarm optimisation-based approach, *International Journal of Production Research*, Vol. 50, No. 15, 4127-4140, doi: 10.1080/00207543.2011.592157.

[11] Wang, Y.F., Zhang, Y.F., Fuh, J.Y.H. (2012). A hybrid particle swarm based method for process planning optimisation, *International Journal of Production Research*, Vol. 50, No. 1, 277-292, doi: 10.1080/00207543.2011.571459.

[12] Li, X., Gao, L., Wen, X. (2013). Application of an efficient modified particle swarm optimization algorithm for process planning, *The International Journal of Advanced Manufacturing Technology*, Vol. 67, No. 5, 1355-1369, doi: 10.1007/s00170-012-4572-7.

[13] Miljković, Z., Petrović, M. (2016). Application of modified multi–objective particle swarm optimisation algorithm for flexible process planning problem, (In press), *International Journal of Computer Integrated Manufacturing*, doi: 10.1080/0951192X.2016.1145804.

[14] Wen, X.-Y., Li, X.-Y., Gao, L., Sang, H.-Y. (2014). Honey bees mating optimization algorithm for process planning problem, *Journal of Intelligent Manufacturing*, Vol. 25, No. 3, 459-472, doi: 10.1007/s10845-012-0696-8.

[15] Huang, W., Hu, Y., Cai, L. (2012). An effective hybrid graph and genetic algorithm approach to process planning optimization for prismatic parts, *The International Journal of Advanced Manufacturing Technology*, Vol. 62, No. 9, 1219-1232, doi: 10.1007/s00170-011-3870-9.

[16] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization, In: *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, WA, IEEE Press, Vol. 4, 1942-1948, doi: 10.1109/icnn.1995.488968.

[17] Hrelja, M., Klancnik, S., Irgolic, T., Paulic, M., Jurkovic, Z., Balic, J., Brezocnik, M. (2014). Particle swarm optimization approach for modelling a turning process, *Advances in Production Engineering & Management*, Vol. 9, No. 1, 21-30, doi: 10.14743/apem2014.1.173.

[18] Wang, Y., Liu, J.H. (2010). Chaotic particle swarm optimization for assembly sequence planning, *Robotics and Computer-Integrated Manufacturing*, Vol. 26, No. 2, 212-222, doi: 10.1016/j.rcim.2009.05.003.

[19] Malik, J., Mishra, R., Singh, I. (2011). PSO-ANN approach for estimating drilling induced damage in CFRP laminates, *Advances in Production Engineering & Management*, Vol. 6, No. 2, 95-104.

[20] Xu, X., Wang, L., Newman, S.T. (2011). Computer-aided process planning – A critical review of recent developments and future trends, *International Journal of Computer Integrated Manufacturing*, Vol. 24, No. 1, 1-31, doi: 10.1080/0951192X.2010.518632.

[21] Dai, M., Tang, D., Xu, Y., Li, W. (2015). Energy-aware integrated process planning and scheduling for job shops, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 229, No. 1, 13-26, doi: 10.1177/0954405414553069.

[22] Wang, J., Fan, X., Zhang, C., Wan, S. (2014). A graph-based ant colony optimization approach for integrated process planning and scheduling, *Chinese Journal of Chemical Engineering*, Vol. 22, No. 7, 748-753, doi: 10.1016/j.cjche.2014.05.011.