Int J Adv Manuf Technol (2012) 62:1219–1232
DOI 10.1007/s00170-011-3870-9

ORIGINAL ARTICLE

An effective hybrid graph and genetic algorithm approach to process planning optimization for prismatic parts

Weijun Huang · Yujin Hu · Ligang Cai

Received: 19 July 2011 / Accepted: 14 December 2011 / Published online: 30 December 2011

Abstract Computer-aided process planning is an important interface between computer-aided design and computer-aided manufacturing in computer-integrated manufacturing environments. In this paper, the complicated process planning is modeled as a combinatorial optimization problem with constraints, and a hybrid graph and genetic algorithm (GA) approach has been developed. The approach deals with process planning problems in a concurrent manner by simultaneously considering activities such as sequencing operations, selecting manufacturing resources, and determining setup plans to achieve the global optimal objective. Graph theory accompanied with matrix theory, as the basic mathematical tool for operation sequencing, is embedded into the main frame of GA. The precedence constraints between operations are formulated in an operation precedence graph (OPG). The initial population composed of all feasible solutions is generated by an elaborately designed topologic sort algorithm to the OPG. A modified crossover operator guaranteeing only feasible offspring generated is used, two types of mutation strategies are adopted, and a heuristic algorithm is applied to adjust the infeasible plan generated by the mutation operator to the feasible domain. A case study has been carried out to demonstrate the feasibility and efficiency of the proposed approach.

Keywords Process planning · Operation sequencing · Genetic algorithm · Operation precedence graph · Optimization

1 Introduction

Computer-aided process planning (CAPP) has received much attention in both the academe and industry during the last three decades [1]. Process planning, as an essential component for linking design and downstream manufacturing processes, deals with the selection of necessary manufacturing processes and determination of their sequences to "transform" a designer's ideas (namely the designed part) into a physical component economically and competitively [2]. For a part with complex structures and numerous features, process planning is well known as a complicated decision-making process involving selecting machining operations for every feature and sequencing the aggregate of all the operations considering precedence constraints, choosing available manufacturing resources, determining setup plans, and machining parameters, etc. These activities must be carried out simultaneously to achieve an optimal plan against a predetermined criterion such as minimum processing time or minimum machining cost, so the manufacturing efficiency could be largely increased or the production cost could be decreased.

In the past two decades, many optimization approaches, such as the genetic algorithm (GA) [3–13], simulated annealing (SA) algorithm [14–16], particle swarm optimization (PSO) [17, 18], and artificial neural networks [13, 19], have been developed and widely applied for solving complex manufacturing system problems, e.g., job shop scheduling and process planning, and significant improvements have been achieved. However, there still remains potential for further improvement. For example, optimization algorithms need to

W. Huang (✉) · Y. Hu
School of Mechanical Science & Engineering,
Huazhong University of Science & Technology,
Wuhan, China
e-mail: wjhuang529@yahoo.com.cn

L. Cai
College of Mechanical Engineering and Applied Electronics
Technology, Beijing University of Technology,
Beijing, China

be improved to be more efficient, and a more reasonable constraint modeling and handling mechanism need to be developed; also, the approach should provide the multiple alternative optimal plans, and some practical manufacturing environment should be considered.

This paper developed a hybrid graph theory and GA approach to process planning for a prismatic part within the context of CAPP. In this approach, operations and the precedence constraints among all the operations are formulated in an operation precedence graph (OPG), the decision making of selecting alternative manufacturing resources and tool approach direction (TAD) for every operation, determining in what order to perform a set of selected operations such that the resulting sequence satisfies the precedence constraints established by both features and operations, is considered concurrently to achieve the optimal plan.

2 Related research work

The potential for application graph theory to operation sequencing was addressed by Prabhu et al. [20]. Irani et al. [21] developed the Hamiltonian path (HP) analogy for the process planning problem based on the precedence graph and operation cost matrix, and the Latin multiplication method for constrained enumeration of all feasible HPs was implemented. The optimal process plan is an HP that corresponds to the least number of setups required for machining each feature once and only once from a feature graph. Though the Hamiltonian method can enumerate all the paths, it is inefficient due to a large number of precedence constraints. Lin et al. [22] designed a graph-search strategy for operation sequence planning in a prismatic part with interacting features. The search graph is built by considering alternative machining operations in converting a blank stock into the final part configuration. The high-quality process plan is generated by the graph-search process considering several heuristic rules concerning machining practices. Since the approaches are based on heuristic rules and reasoning, the final solution may be only feasible, and the optimal plans might be lost during the reasoning process.

The global search techniques such as the GAs and PSO have been successfully applied to solve the combinatorial optimization problems. Dutta et al. [3] have used GAs for the sequencing of operations in process planning for a parallel machining environment where combinations of interacting work-holding and tool-holding devices are used. A new coding method to translate between an operation sequence and its string representation is developed in their work. But the operation features with multiple parents are not considered. The evaluation criterion is minimum processing time. Zhang et al. [4] present a novel CAPP model for parts to be machined in a job shop manufacturing environment. GA is used to carry out the selection of machining resources and sequencing operations simultaneously. The dynamic status of machining resources in the job shop and alternative optimal plans are not taken into account. Reddy et al. [5] proposed a GA-based operation sequencing optimization approach to identify optimal or near-optimal operation sequences in a dynamic planning environment. A feature precedence graph (FPG) is defined to identify the feasible sequences, and the size of the solution space in operation sequencing can be reduced with the FPG .The minimum production cost is used as the evaluation criterion. Li et al. [6] developed a hybrid genetic algorithm and a simulated annealing approach for optimizing process plans for prismatic components. They modeled the process planning as a combinatorial optimization problem with constraints. The evaluation criterion was the combination of machine costs, cutting tool costs, machine change costs, tool change, and setup costs.

Automated processing planning based on GA and/or SA also has been reported by Qiao et al. [7], Lee et al. [8], Rocha et al. [9], Alam et al.[10], Li et al. [11, 12, 16], Ding et al. [13], Ma et al. [14], and Ong et al. [15]. Ding et al. [13] present a global optimization strategy incorporating the GA, neural network, and analytical hierarchical process (AHP) for process sequencing. A global fitness function was defined including the evaluation of manufacturing rules using AHP, calculation of cost and time, and determination of relative weights using neural network techniques.

Guo et al. [17] proposed a PSO approach to operation sequencing problem. The initial process plan solutions randomly generated are encoded into particles of the PSO algorithm. To avoid being trapped into local optima and improve the particles' movements, several new operators have been developed. Penalty strategy is used considering the evaluation of infeasible particles (process plans).

Though significant improvements have been achieved, several problems of these approaches still exist, including the following aspects:

1. Feature is usually used as the basic unit for process planning, i.e., supposing each feature is machined out in one setup and process routing of a component is represented by a features sequence. In practice, the roughing and finishing operations of a feature must be assigned to different setups sometimes. If a component is with high-precision requirements, or a large deformation occurs during the machining, and heat treatment processes are required, a feature is usually machined in more than one setup.
2. The generating of an initial population in GA or PSO is usually done randomly, which makes some of the produced solutions infeasible because of their violation of the precedence constraints. And during the implementation of algorithms, new infeasible solutions are likely to be produced by the operator. The penalty strategy employed is

not effective in performance. So it is essential to develop a general methodology for allowing the generation of only feasible solutions or adjusting infeasible solutions to a feasible domain.

3. Most of the proposed method for process planning can only generate a single process plan in a simple or limited machining environment without regard to resource availability on the shop floor; due to the limitation, most of the process plans generated by these methods are likely to be too inefficient to implement.

3 Process planning problem description

To conduct process planning, parts are commonly described by features with technological attributes such as tolerances and surface finishes, which are geometric forms having machining meanings, e.g., planes, slots, and holes. The CAD information about features of a part is recognized and extracted by the CAPP system before the process planning is carried out. Then the operation methods for each feature are selected according to the requirements of different features. An operation method refers to an operation in name without any attachment of corresponding machines, tools, and TADs, e.g., turning, reaming, and milling. When the specific machine, tools, and TAD are assigned to a related operation method, the operation can be executed. Each operation can be executed by several alternative plans if different machines, cutting tools, or setup plans are chosen for this operation [23, 24]. In this paper, a setup is defined as a group of operations owning the same TAD executed on a machine continuously, and a TAD is defined as a direction from which a cutting tool can access a feature [25]. The features and their valid TADs can be recognized using a geometric reasoning approach [6, 13]. In our study, the machining environment is a workshop layout with both conventional machines and CNC 3-axis vertical milling machines without rotatable fixtures, so one setup can only machine features in one TAD.

However, even though the determination of operation methods for features has been completed, it is impossible to specify any individual machine or tool for each operation method. This is owing to the fact that machining resources and setup plan decision are subject to process planning, which can be accomplished only in a simultaneous manner with the activities of sequencing operations. This leads to the dilemma that in process planning, machining resources information is available, but this information cannot be specified until operation sequencing is performed.

The quality measurement of a process plan for a part is based on two considerations: (1) the optimum selection of machine, cutting tool, and TAD for every operation and (2) the optimum sequence of the operations for machining the part. Hence, the proposed approach to address these two aspects is carried out.

3.1 Knowledge-based representation of a process plan

For a part, a process plan provides detailed information about the sequence of the operations, applicable candidate manufacturing resources, setup plans, machining parameters, etc. Since knowledge-based representation could produce good performance results, the operations and their relevant machines, cutting tools, and TADs are represented as a chromosome for the application of GA. Each operation has a set of candidate machines, cutting tools, and TADs by which the operation can be executed, and the details are listed in Table 1.

Therefore, a process plan is represented using a string which consists of n bits, each bit represents an operation once and only once, and the order of all those bits within the string determines the machining sequence. Any sequence of all bits is a possible solution (chromosome) for a process plan in the solution space. In the paper, a process plan is formulated in a vector: ***Oper***[n], n is the total number of operations.

Figure 1 shows the representation of a process plan, a sequence with six machining operations: ***Oper***[6]. “op5” represents operation 5; m-01, t-02, and +x in the second column represent the machine, tool, and TAD that will be used to execute operation 5, respectively; so are the other columns. Here, the index of vector ***Oper*** represents the operation's executing order in the sequence, i.e., the machining route is starting with ***Oper***[0] (op5), then ***Oper***[1] (op4),…, the last is ***Oper***[5] (op2).

3.2 Precedence constraints analyzing and modeling

The preliminary precedence constraints among operations come from the consideration of geometrical and manufacturing interactions between features as well as technological requirements in a part [6, 26]. The constraints imply precedence relationships (PRs) to determine in what order to

Table 1 Class definition of an operation

Class_operation type: an operation	
Variable	Description
Operation_id	The ID of the operation
Machine_id	The ID of the machine to execute the operation
Machine_list[]	The candidate machine list for executing the operation
Tool_id	The ID of the tool to execute the operation
Tool_list[]	The candidate tool list for executing the operation
TAD_id	The ID of a TAD to apply the operation
TAD_list[]	The candidate TAD list for executing the operation

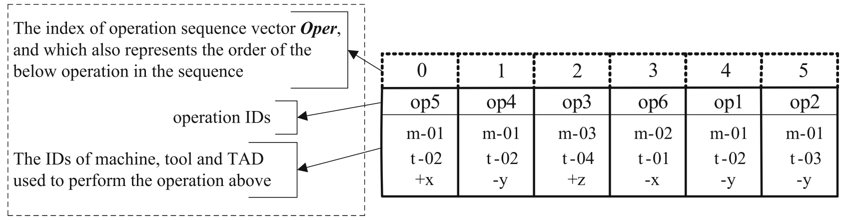

Fig. 1 Representation of a process plan

perform a set of selected operations. It is necessary for a process plan not to violate any of the precedence constraints.

Supposing there is a total of eight operations required for a test part to be machined, i.e., the operation aggregation is {op1, op2,⋯, op8}, and after the geometrical, manufacturing interactions, and technological requirements analysis, the PRs between operations are identified, as listed in Table 2.

Here, all the operation and PRs are formulated in an OPG, e.g., considering a set of PRs among operations (listed in Table 2) of a part, a directed graph, as an effective precedence constrained model, can be conveniently constructed, as shown in Fig. 2. To represent these PRs, every element of the operation aggregation is represented as the corresponding vertex of the OPG, and the PR between every two operations is represented as a directed edge which the two vertices lie on, i.e., one vertex that the arrow points to must be executed after the other. An example is given in Fig. 2, where the directed edge starts from vertex "op1" and the arrow points to vertex "op3," which is equivalent to the operation of op1 must be prior to op3. Complying with the PRs among vertices, traversing every vertex of the OPG once and only once can obtain a Hamiltonian path, which is also a feasible operation sequence.

Because computers are more adept at manipulating numbers than at recognizing pictures, it is standard practice to communicate the specification of a graph to a computer in matrix form [25]. The adjacency matrix is frequently used to represent the relationships among vertices of a graph. The adjacency matrix of a given OPG in the paper is formulated by a matrix **P**, and there is:

$$\mathbf{P} = (pij)n \times n \tag{3.1}$$

where n is the total number of operations for a part, and the value of p_{ij} is calculated in accordance with the following rules:

- When there is a directed edge connecting the vertex op$i$ and op$j$, and the arrow comes from op$j$ and points to op$j$, then p_{ij}=1 and p_{ji}=0; otherwise, p_{ij}=0, (i≠j);
- $p_{ii} = 0(i = 1, 2, \cdots, n)$

In Fig. 2, the OPG can be represented with an *adjacency matrix*, as illustrated in Fig. 3.

3.3 Process plan evaluation criterion

The criterion of minimum production cost is generally used for process plan evaluation. The detailed analysis of the cost is based on the consideration of the time required to complete every machining operation, which consists of the following aspects: unproductive time, preparation time, tool change time, and cutting time [27, 28]; for more details on the machining time calculation, see [27, 28]. Because detailed information on tool paths and machining parameters has not been determined so far in the paper, instead of accurate cost and time calculation, we use the estimated production cost to evaluate process plans in the macro-planning stage, which comprises five factors: machine utilization cost, tool utilization cost, machine change cost, tool change cost, and setup cost [6, 11, 17]; the calculation procedures of these cost factors is described in detail below.

1. Total machine cost (MC). MC is the total costs of the machines used to accomplish a process plan, and it can be calculated as:

Table 2 The precedence relationships between operations

Operation ID	Precedence relationship description
op3	op1 and op2 must be performed prior to op3
op4	op3 must be performed prior to op4
op5	op3 must be performed prior to op5
op6	op2 must be performed prior to op6
op7	op4 must be performed prior to op7
op8	op5 and op6 must be performed prior to op8

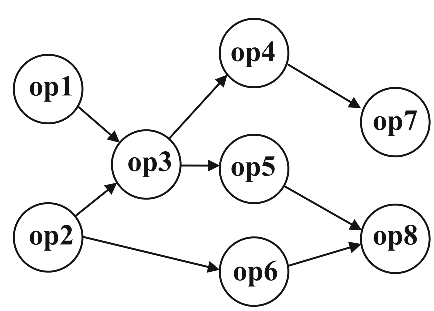

Fig. 2 An operation precedence graph

$$\begin{array}{c|cccccccc} & \text{op1} & \text{op2} & \text{op3} & \text{op4} & \text{op5} & \text{op6} & \text{op7} & \text{op8} \\ \hline \text{op1} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \text{op2} & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ \text{op3} & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ \text{op4} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \text{op5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \text{op6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \text{op7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{op8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Fig. 3 The adjacency matrix

$$\text{MC} = \sum_{i=0}^{n-1} \text{MCI}j, j = Oper[i].\text{machine\_id} \tag{3.2}$$

where n is the total number of operations (the length of ***Oper***), MCI_j is the machine cost index for using machine-j, a constant for a specific machine.

2. Total tool cost (TC). TC is the total costs of the cutting tools used to accomplish a process plan, and it can be calculated as:

$$\text{TC} = \sum_{i=0}^{n-1} \text{TCI}j,\ j = Oper[i].\text{tool\_id} \tag{3.3}$$

where TCI_j is the tool cost index for using tool-j, a constant for a specific cutting tool.

3. Number of machine changes (NMC) and total machine change cost (MCC). A machine change occurs when two adjacent operations in ***Oper*** are executed on different machines.

$$\text{NMC} = \sum_{i=0}^{i=n-2} \Omega 1(Oper[i].\text{machine\_id}, Oper[i+1].\text{machine\_id}) \tag{3.4}$$

$$\text{MCC} = \text{NMC} \times \text{MCCI} \tag{3.5}$$

$$\Omega 1(X, Y) = \begin{cases} 1, & X \neq Y \\ 0, & X = Y \end{cases} \tag{3.6}$$

where MCCI is the machine change cost index. ***Oper***[i].machine_id is the ID of the machine used to perform operation–***Oper***[i].operation_id

4. Number of tool changes (NTC) and total tool change cost (TCC): The definition of a tool change is described in Table 3, so NTC and TCC are calculated as follows:

$$\text{NTC} = \sum_{i=0}^{i=n-2} \Omega 2(\Omega 1(Oper[i].\text{machine\_id}, Oper[i+1].\text{machine\_id}), \Omega 1(Oper[i].\text{tool\_id}, Oper[i+1].\text{tool\_id})) \tag{3.7}$$

$$\text{TCC} = \text{NTC} \times \text{TCCI} \tag{3.8}$$

$$\Omega 2(X, Y) = \begin{cases} 0, & X = Y = 0 \\ 1, & \text{otherwise} \end{cases} \tag{3.9}$$

where TCCI is the tool change cost index and is a constant.

5. Number of setup changes (NSC), the number of setups (NS), and the total setup cost (SC).

   The definition of a setup change is depicted in Table 4, and NSC is calculated as follows:

$$\text{NSC} = \sum_{i=0}^{i=n-2} \Omega 2(\Omega 1(Oper[i].\text{machine\_id}, Oper[i+1].\text{machine\_id}), \Omega 1(Oper[i].\text{TAD\_id}, Oper[i+1].\text{TAD\_id})) \tag{3.10}$$

The corresponding NS and SC can be calculated as:

$$\text{NS} = \text{NSC} + 1 \tag{3.11}$$

$$\text{SC} = \text{NS} \times \text{SCI} \tag{3.12}$$

Where SCI is the setup change cost index.

6. Total production cost (PC)

$$\text{PC} = \text{MC} + \text{TC} + \text{SCC} + \text{MCC} + \text{TCC} \tag{3.13}$$

4 Hybrid graph and GA approach

The application of GA-based approaches for optimization generally includes these steps: (1) generating the initial population comprising a specified number of process plans, (2) fitness calculation for individuals of the

Table 3 Definition of a tool change

Conditions of machining two consecutive operations	Tool change
Same tool and same machine	No
Same tool and different machines	Yes
Different tools and same machine	Yes
Different tools and different machines	Yes

Table 4 Definition of a setup change

Conditions of machining two consecutive operations	Setup change
Same TAD and same machine	No
Same TAD and different machines	Yes
Different TADs and same machine	Yes
Different TADs and different machines	Yes

population, (3) reproducing chromosomes using a certain selection strategy, and (4) applying the crossover and mutation operators. The steps (2–4) are repeated until the number of iterations.

4.1 The technique to initializing populations

Initialization is an important procedure that generates a certain number of chromosomes with which the implementation of GA begins. The production of the initial population in GA is usually done randomly. But, since the existence of precedent constraints, some operation sequences generated randomly are likely to be infeasible because of the violation of constraints, which makes the performance of GA reduced. So, it is necessary to propose an approach that can guarantee that the initial population consists of all the feasible chromosomes.

To discuss the approach to produce the initial population of all feasible solutions, the process planning problem mentioned in chapter 3.2 is used as an example. Now there is an operation aggregation {op1, op2,⋯, op7, op8}, and each operation is initialized to a legal value based on the available machining resources in the workshop, i.e., randomly select one machine, tool, and TAD from the corresponding candidates and assign them to the correlative operation. Initially, the OPG ***G*** is established according to the PRs list in Table 1. $\mathbf{P}_{n\times n}$ (n=8 here), as the adjacency matrix correlative with ***G***, is generated by the mentioned rules.

To eliminate the infeasible chromosomes, a randomly topologic sorting algorithm for operation sequencing is designed, and the number of initialized chromosomes is prescribed as N; the procedures of initializing populations are given as follows:

Firstly, several variables required in the implementation of the algorithm mentioned below are defined.

Inedge For any vertex op$i$ in ***G***, calculate the total number of edges with an arrowpoint to op$i$, and this number is denoted by the variable "*inedge*," obviously, which is equal to the number of operations prior to op$i$. As the calculating of any vertex in *inedge*, there is a formula:

$$inedge(\text{op}i) = \sum_{j=1}^{n} pji. \tag{4.1}$$

G' A graph with the same forms as G, and G' is initialized to an empty graph.

$\mathbf{M}_{n\times n}$ A variable with the same forms as $\mathbf{P}_{n\times n}$, and $\mathbf{M}$ is initialized to an empty matrix, i.e., all the element of $\mathbf{M}$ is set to 0, $m_{ij} \leftarrow 0,(i,j = 1, 2, \cdots, n)$.

q An integral variable q is defined.

1. Copy G to G', and copy $\mathbf{P}$ to $\mathbf{M}$, i.e., $m_{ij} = p_{ij},(i,j = 1, 2 \ldots, n)$, set $q \leftarrow 0$.
2. Calculating the *inedge* value of all vertices of G': $inedge(\text{op}i) = \sum_{j=1}^{n} mji,\ i = 1, 2, \cdots, n$
3. Randomly select one vertex (here, which is expressed as op$_k$) among the operation vertices with *inedge* equal to 0. Delete op$_k$ in digraph Gx$'$, and delete all the edges that start from op$_k$, which can be achieved on operating $\mathbf{M}$ by setting all elements of the ith row as 0.
4. Calculating the *inedge* of the remaining vertices of G', from the remaining operations, randomly select one (expressed as op$_l$)with inedge equal to 0, delete op$_l$ in digraph G', and delete all the edges that start from op$_l$.
5. Repeat 3 until every vertex has been selected for once from G', then a sequential listing of all the vertices (a toplogical order for G') is obtained.
6. Revisit the sequential listing of all the vertices (***Oper***) in ascending sequence; the current position index is q.
7. Randomly select a machine, tool, and TAD from the candidates that can be used for performing ***Oper***[q] and assign them to the operation-***Oper***[q].
8. $q \leftarrow q+1$;
9. Repeat 6, 7, and 8 until each operation has been reassigned a machine, tool, and TAD (i.e., q is equal to n).
10. Repeat steps 1–10 until the number of prescribed chromosome is reached.

Applying the initialization approach, the initial solutions are produced in the feasible domain.

4.2 Fitness function of GA

Fitness function is used to connect the problem and the algorithm and assess the capability of a chromosome. Whether a chromosome is relatively better or not and whether a chromosome should be reserved or not is decided directly by the fitness function. Since the objective of GA is often to achieve the maximum, the fitness calculation function f can be defined as:

$$f = \text{UL} - \text{PC} \tag{4.2}$$

PC is the total production cost of a process plan, as in Eq. 3.13, and UL is the upper limit constant of PC.

4.3 The GA operators

The design of appropriate genetic operators, including the reproduction, crossover, and mutation, plays a crucial role for the successful implementation of GA.

4.3.1 Reproduction

The strategies of "elitism" and the "tournament selection" are adopted as reproduction operators in this paper. This works in two steps: firstly, to guarantee the astringency, it applies "elitism" by copying the optimal individual of the population in the current generation to the next generation; secondly, other remaining individuals of the current generation are selected to the next generation by using the "tournament selection" operator. Assuming there are k (k=N−1) individuals to be selected, select two individuals randomly from the population and keep the better one for the next generation. Repeating this procedure k times, all individuals of the next generation will be obtained consequently.

4.3.2 Crossover

Crossover operator is applied, at a given probability p_c, to the chromosomes that resulted from reproduction by splitting and recombining between two parent chromosomes to create new chromosomes for the next generation. A modified crossover operator, which is similar to Zhang et al. [4], is adopted in this paper. Since the existence of precedence constraints, the operator must ensure that PRs are maintained and only feasible plans are generated. The procedure of the operator is described as follows:

1. Randomly select two chromosomes from the current population as parent chromosomes.
2. Based on the chromosome length, a crossover point is randomly selected, and each parent chromosome is divided into left and right segments from the crossover point.
3. Copy the left segment of parent 1 to form the left segment of child 1. Find the operations in the right segment of parent 1 and copy them to the right segment of child 1 according to their sequence orders in parent 2.
4. The role of parents 1 and 2 will then be exchanged in order to produce another offspring, child 2.

This procedure is illustrated with an example shown in Fig. 4.

4.3.3 Mutation

The mutation operator is applied with a small probability to investigate some of the unvisited points in the search space and introduce some genetic diversity; thus, being trapped at a local optimum could be avoided.

In the proposed GA, two types of mutation strategies are applied. The first strategy is to randomly choose two genes (operations) in a chromosome and exchange them at the prescribed probability p_{m1}. An example of the first mutation process is shown in Fig. 5.

After mutation 1, the newly produced chromosome may be infeasible because of the violation of precedence constraints, e.g., shown in Fig. 6, the operation sequence violates the *PR* list in Table 1. Thus, a heuristic algorithm aimed at a specific problem will be applied to adjust them to the feasible domain and is described as follows:

Firstly, a chromosome is represented by a vector ***Oper***[n], the OPG is G, and its adjacency matrix is $\mathbf{P}_{n\times n}$; the variable pt is the vector's index and initialize pt=n−1; calculate the *inedge* of operation vertex $Oper[i], i = 0, 1, 2, \cdots, n-1$.

Algorithm for adjusting the infeasible chromosome:

```
Repeat (until pt is equal to 0)
{
  if ( inedge of Oper[pt] is not equal to 0)
  // Indicate that Oper[pt] cannot be treated as the last among all the current operations which have
     not been finished.
   { p = pt ;
     Repeat (until the inedge of Oper[p] is equal to 0) { p = p-1;}
        Exchange the position of Oper[pt] and Oper[p] in the chromosome Oper[n];
        The inedge of all Oper[p]'s predecessor minus 1; //Corresponding to delete all the edges
        that point to node Oper[p] of G , and synchronously the vertex Oper[p] is removed}
  else { pt = pt-1; The inedge of Oper[p]'s predecessors minus 1; }
 }
```

The algorithm is illustrated with an example shown as follows:

The second mutation strategy refers to machine, tool, and TAD mutation of an operation. In practice, machine change is an important factor in considering processing time and production cost. So here, three mutation operators were developed based on the heuristic knowledge of process planning.

1. *Machine mutation* is used to change the current machine of an operation if there is any candidate machine for it.

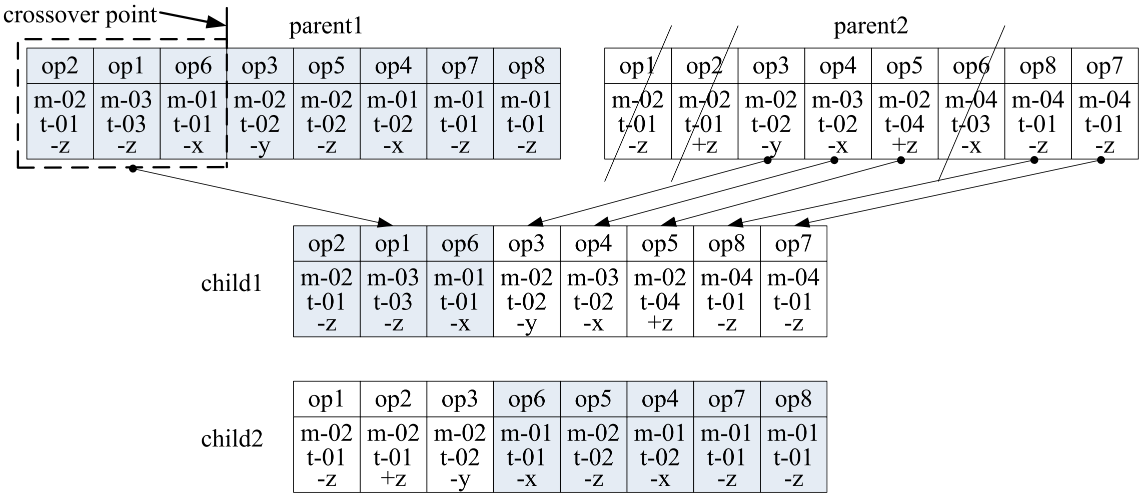

Fig. 4 An example of applying the crossover operator

The algorithm for machine mutation operator is described as follows:

(a) Select an operation randomly from a chromosome and use the probability p_{m2} to determine whether the machine needs to be changed.
(b) Randomly choose a machine (m-*j*) from the candidates to replace the current machine (m-*i*) of the operation.
(c) Identify all the other operations with m-*i* as current machine in the one chromosome. If any operation has m-*j* as a candidate, replace machine (m-*i*) with machine (m-*j*).

The procedure is illustrated with an example shown in Fig. 7.

2. *Tool mutation* occurs after machine mutation. The mechanism is similar to machine mutation.
3. *TAD mutation* occurs after tool mutation. The mechanism is also similar to machine mutation.

4.3.4 The GA's termination criterion

There are several termination criteria for the search process of GA. Usually, the user pre-set maximum iteration times. Once the iteration times for the program have reached the maximum, then stop. The final obtained solution is regarded as the optimal or suboptimal result satisfying the optimization objective.

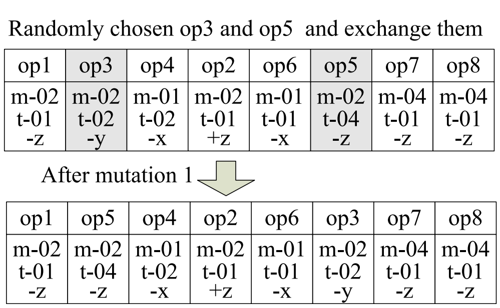

Fig. 5 An example of mutation 1 application

5 Process planning for a sample part

5.1 Problem description

A sample part taken from the work of Guo et al. [17] is used here to test the developed approach (Fig. 8). The part, which is assumed to be manufactured in a job shop manufacturing environment, consists of 14 manufacturing features, including planes, holes, pockets, etc. These features can be machined with 20 operations (n=20). The relevant information of features, operations, manufacturing resources, and precedence constraints of the part are given, respectively, in Tables 5 and 6.

Complying with the PRs among operations in Table 6, the corresponding OPG of the sample part can be established easily, as shown in Fig. 9. The adjacency matrix of the OPG, $\mathbf{P}_{20\times20}$, can be formulated by the mentioned rules in chapter 3.2.

5.2 Determination of parameters in GA approach

In order to realize the dual goals of maintaining diversity in the population and sustaining the convergence capacity of the GA, the usage of adaptive probabilities of crossover and mutation is recommended in this paper; the detailed settings of GA parameters were as follows:

1. Population size, N=100.
2. In the fitness calculation function of Eq. 4.2, determine the upper limit UL=5,000.

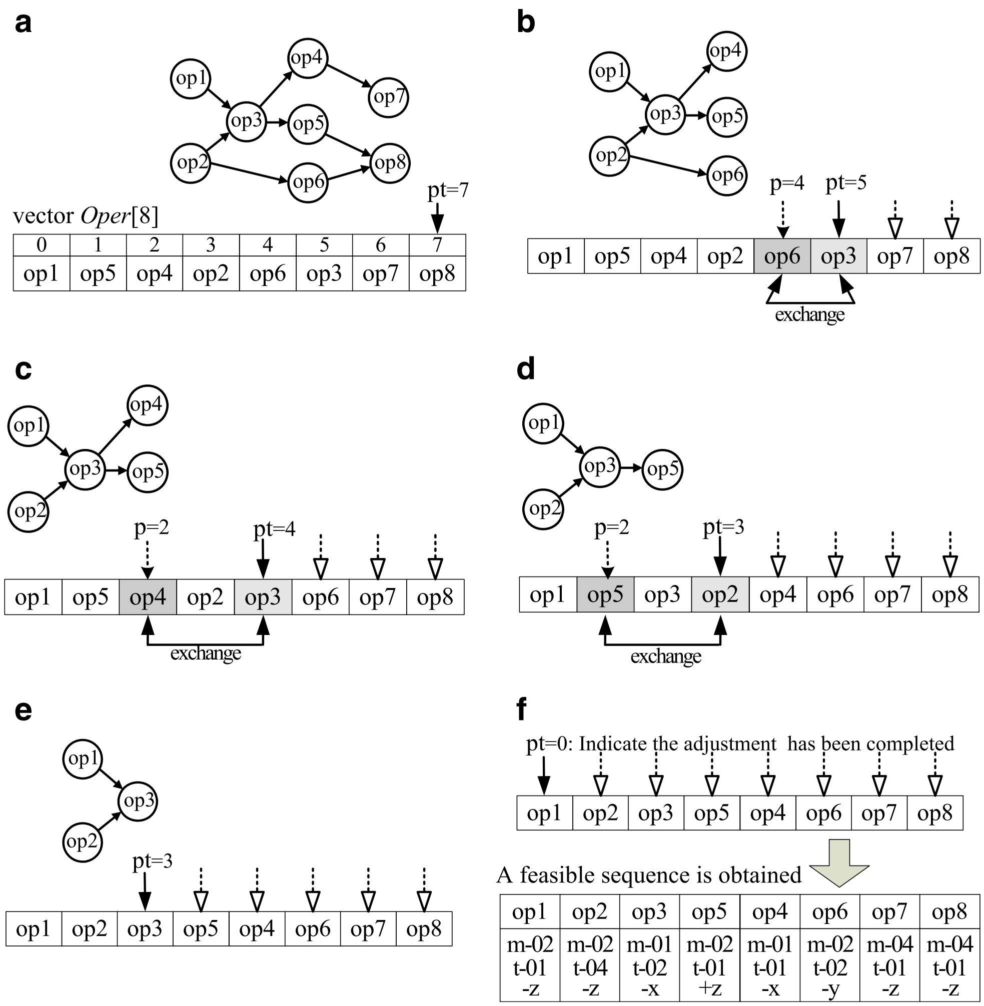

Fig. 6 An example of adjusting an infeasible process plan to feasible domain

3. An adaptive crossover probability is designed by improving the calculation method in the adaptive GA (AGA) proposed by Srinvivas et al. [29], and p_c is calculated by Eq. 5.1:

$$P_c = \begin{cases} P_{c1} - \dfrac{(P_{c1}-P_{c2})\left(f'-f_{avg}\right)}{\left(f_{max}-f_{avg}\right)}, & f' \geq f_{avg} \\ P_{c1}, & f' < f_{avg} \end{cases} \tag{5.1}$$

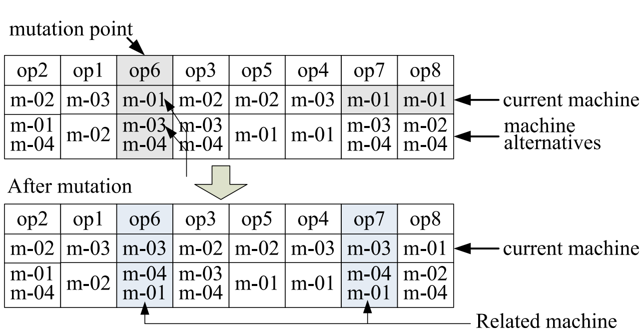

Fig. 7 An example of mutation 2 application

In Eq. 5.1, P_{c1}=0.9, P_{c2}=0.6, f_{max}, and f_{avg} represent maximal and average fitness of the individuals of the population in each generation, respectively; f' represents the larger fitness of the two individuals for crossover operation.

4. The adaptive probability p_{m1} of the first mutation strategy is calculated by Eq. 5.2:

$$P_{m1} = \begin{cases} P^0_{m1} - \dfrac{\left(P^0_{m1}-P^1_{m1}\right)\left(f-f_{avg}\right)}{\left(f_{max}-f_{avg}\right)}, & f \geq f_{avg} \\ P^0_{m1}, & f < f_{avg} \end{cases} \tag{5.2}$$

In Eq. 5.2, $P^0_{m1} = 0.1$, $P^1_{m1} = 0.001$, f represents the fitness of the individual for the first mutation.

5. The second mutation strategy probability p_{m2}. The availability of alternative machines, tools, and *TAD*s must be adequately traversed in the optimizing process, and the three mutation rates for machines, tools, and TADs are determined by Eq. 5.3.

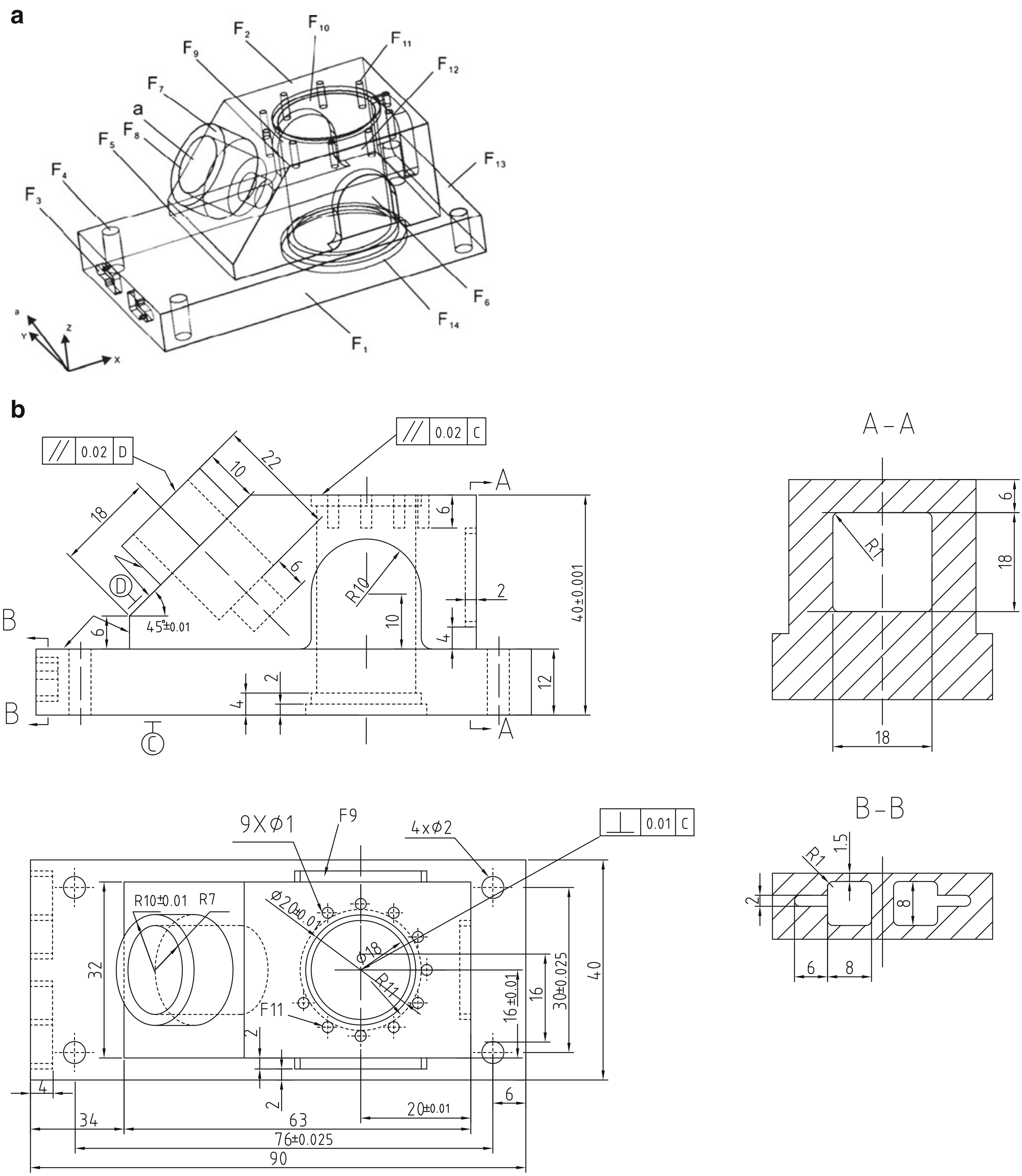

Fig. 8 A sample part with 14 machining features

$$P_{m2} = \begin{cases} P_{m2}^0 - \frac{(P_{m2}^0 - P_{m2}^1)(f - f_{avg})}{(f_{max} - f_{avg})}, & f \geq f_{avg} \\ P_{m2}^0, & f < f_{avg} \end{cases} \quad (5.3)$$

In Eq. 5.3, $P_{m2}^0 = 0.6$, $P_{m2}^1 = 0.1$.

6. Terminating criterion. The maximal iteration time of 1,500 generations is selected.

Table 5 The features, operations, and manufacturing resources information of the sample part

Features	Feature descriptions	Operation (operation_id)	TAD candidates	Machine candidates	Tool candidates	Available machining resources in a workshop environment
F1	Planar surface	Milling (op1)	+*z*	m-02, m-03	t-06, t-07, t-08	m-01 (10): drilling press
F2	Planar surface	Milling (op2)	−*z*	m-02, m-03	t-06, t-07, t-08	m-02 (40): 3-axis vertical milling machine
F3	Two pockets arranged as a replicated feature	Milling (op3)	+*x*	m-02, m-03	t-06, t-07, t-08	m-03 (100): CNC 3-axis vertical milling machine
F4	Four holes arranged as a replicated feature	Drilling (op4)	+z, −z	m-01, m-02, m-03	t-02	m-04(60): boring machine
F5	Planar surface	Milling (op5)	+*x*, −z	m-02, m-03	t-06, t-07	t-01 (7): drill 1
F6	Planar surface	Milling (op6)	+*y*, −z	m-02, m-03	t-07, t-08	t-02 (5): drill 2
F7	Planar surface	Milling (op7)	−*a*	m-02, m-03	t-07, t-08	t-03 (3): drill 3
F8	A compound hole	Drilling (op8)	−*a*	m-01, m-02, m-03	t-02, t-03, t-04	t-04 (8): drill 4
		Reaming (op9)		m-01, m-02, m-03	t-09	t-05 (7): tapping tool
		Boring (op10)		m-02, m-03	t-10	t-06 (10): milling 1
F9	A protrusion (rib)	Milling (op11)	−*y*, −z	m-02, m-03	t-07, t-08	t-07 (15): milling 2
F10	A compound hole	Drilling (op12)	−*z*	m-01, m-02, m-03	t-02, t-03, t-04	t-08 (30): milling 3
		Reaming (op13)		m-01, m-02, m-03	t-09	t-09 (15): reamer
		Boring (op14)		m-03, m-04	t-10	t-10 (20): boring tool
F11	Nine holes arranged as a replicated feature	Drilling (op15)	−*z*	m-01, m-02, m-03	t-01	MCCI=160
		Tapping (op16)		m-01, m-02, m-03	t-05	SCI=100
F12	A pocket	Milling (op17)	−*z*	m-02, m-03	t-07, t-08	TCCI=20
F13	A step	Milling (op18)	−*x*, −*z*	m-02, m-03	t-06, t-07	Note: values in brackets are cost indices
F14	a compound hole	reaming(op19)	+*z*	m-01,m-02,m-03	t-09	
		Boring (op20)		m-03, m-04	t-10	

Table 6 The PRs between operations

Features	Operation (operation_id)	Precedence constraint descriptions
F1	Milling (op1)	F1 (op1) is the datum and supporting face for the part; hence, it is machined before all features
F2	Milling (op2)	F2 (op2) is before F10 (op12, op13, op14) and F11(op15, op16)
F3	Milling (op3)	
F4	Drilling (op4)	
F5	Milling (op5)	F5 (op5) is before F4 (op4) and F7 (op7) for the datum and material removal interactions
F6	Milling (op6)	F6 (op6) is before F10 (op12, op13, op14) for the datum interaction
F7	Milling (op7)	F7 (op7) is before F8 (op8, op9, op10) for the datum and material removal interactions
F8	Drilling (op8) Reaming (op9) Boring (op10)	op8 is before (op9 and op10); op9 is before opo10 for the fixed order of machining operations
F9	Milling (op11)	F9 (op11) is before F10 (op12, op13, op14) for the datum interaction
F10	Drilling (op12) Reaming (op13) Boring (op14)	op12 is before op13 and op14; op13 is before op14; F10 (op12, op13, op14) is before F11 (op15, op16) for the datum interaction; op12 is before F14 (op19, op20)
F11	Drilling (op15) Tapping (op16)	op15 is before op16 for the fixed order of operations
F12	Milling (op17)	
F13	Milling (op18)	F13(op18) is before op4 and op17 for the material removal interaction
F14	Reaming (op19) Boring (op20)	op19 is before op20 for the fixed order of machining operations

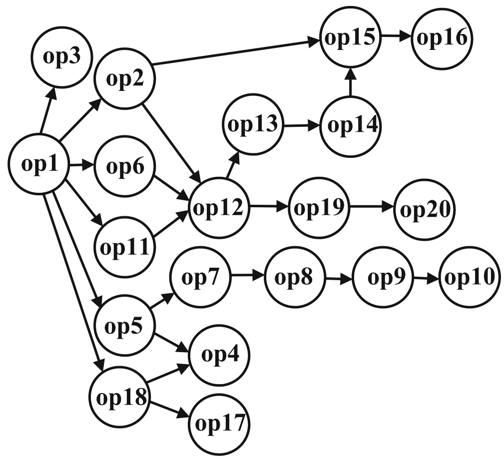

Fig. 9 The OPG of the sample part

5.3 Computational results under different conditions and criteria

To test the capability and flexibility of the proposed optimization approach in a dynamic workshop environment, we carried out the process planning under three different conditions. Conditions 1 and 2 assume that all the machining resources are available, while condition 3 assumes that vertical milling machine m-02 and tool t-08 are down.

Since the performance of GA is not guaranteed and can never be evaluated on the basis of a single run, the program for each condition will be repeatedly run 10 times with the same parameter settings and different random initialization seeds. The computational results under different conditions and criteria are presented as follows:

Condition 1: All machines and tools are available. By repeatedly running the program 10 times, the optimal process plan in each run is generated with the machining cost varying from 2,527 to 2,585, and the process plans with the cost of 2,527 appeared five times in the 10-time run, therefore they are selected as the final optimal process plans and are listed in Table 7.

Condition 2: All machines and tools are available. Considering the machine costs, the numbers of setups, and the number of machine changes are mainly factors in calculating the total machining cost, we ignore secondary factors such as tool cost and tool change cost to simplify the calculation, and PC=MC+SC+MCC. By repeatedly running the program 10 times, the multiple optimal process plans with cost 2,120 are obtained as listed in Table 8, and they can be selected as the final optimal solutions.

Condition 3: Suppose machine m-02 and tool t-08 are down, we set PC=MC+SC+MCC. In a dynamic workshop environment, some machines or tools may be in the state of bottleneck usage or breakdown. Some optimal or suboptimal process plans are listed in Table 9 when m-02 and tool t-08 are down and only certain aspects of costs are considered, so process plan 1 and 2 can be selected as the final optimal solutions now.

The optimal process plan for the same part reported in the work of Guo et al. [17] is with the production cost 2,535, while ours is 2,527 with the same evaluation criterion in condition 1. Through comparison with the computational result reported in [17], it can be seen that our hybrid graph and GA approach can generate higher-quality solutions. In addition, the more significant improvement of the proposed

Table 7 Two optimal process plans generated under condition 1

Process plan 1																				
Order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Operation ID	1	5	3	18	6	2	11	12	13	17	7	8	9	19	14	20	10	4	15	16
Machine ID	2	2	2	2	2	2	2	2	2	2	2	2	2	2	4	4	4	1	1	1
Tool ID	6	6	6	6	6	6	7	3	9	7	7	2	9	9	10	10	10	2	1	5
TAD	+*z*	+*x*	+*x*	−*z*	−*z*	−*z*	−*z*	−*z*	−*z*	−*x*	−*a*	−*a*	−*a*	+*z*	−*z*	+*z*	−*a*	−*z*	−*z*	−*z*
NMC=2, NS=10, NTC=9, MCC=320, SC=1,000, TCC=200, MC=770,TC=237; total cost, 2,527																				
Process plan 2																				
Order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Operation ID	1	3	5	2	18	6	11	12	13	17	7	8	9	19	14	20	10	15	16	4
Machine ID	2	2	2	2	2	2	2	2	2	2	2	2	2	2	4	4	4	1	1	1
Tool ID	6	6	6	6	6	6	7	3	9	7	7	2	9	9	10	10	10	1	5	2
TAD	+*z*	+*x*	+*x*	−*z*	−*z*	−*z*	−*z*	−*z*	−*z*	−*x*	−*a*	−*a*	−*a*	+*z*	−*z*	+*z*	−*a*	−*z*	−*z*	−*z*
NMC=2, NS=10, NTC=9, MCC=320, SC=1,000, TCC=200, MC=770,TC=237; total cost, 2,527																				

Table 8 Two process plans generated under condition 2

Process plan 1																				
Sequence Order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Operation ID	1	11	2	6	12	13	14	18	15	16	3	5	4	19	20	7	8	9	10	17
Machine ID	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Tool ID	6	7	7	7	2	9	10	6	1	5	6	7	2	9	10	7	3	9	10	7
TAD	+z	−z	−z	−z	−z	−z	−z	−z	−z	+x	+x	+z	+z	+z	+z	−a	−a	−a	−a	−x
NMC=1, NS=8, MCC=160, SC=800, MC=1160; total cost, 2,120																				
Process plan 2																				
Sequence order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Operation ID	1	18	17	5	6	4	11	2	12	13	7	8	9	3	10	19	20	14	15	16
Machine ID	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
Tool ID	7	6	7	7	8	2	7	6	2	9	7	2	9	8	10	9	10	10	1	5
TAD	+z	−x	−x	−z	−z	−z	−z	−z	−z	−z	−a	−a	−a	+x	−a	+z	+z	−z	−z	−z
NMC=1, NS=8, MCC=160, SC=800, MC=1,160; total cost, 2,120																				

approach is that it can produce multiple optimal or suboptimal process plans considering different conditions and criteria in a dynamic environment.

6 Conclusion

A hybrid graph and GA approach were proposed to solve the optimization problem of process planning for prismatic parts by simultaneously considering the assignment of machining resources, determining sequencing operation and setup plans. The approach presented here has several advantages in the following aspects:

1. The precedence constraints are formulated in an OPG manipulated in the approach, and the operation sequencing is always conducted in a feasible solution domain so the search space of the GA can be reduced and the efficiency of optimization algorithm can be improved.

Table 9 Some optimal and suboptimal process plans generated under condition 3

Process plan 1																				
Sequence order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Operation ID	1	18	17	2	6	11	12	13	5	14	7	8	9	10	3	19	20	4	15	16
Machine ID	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	1	1
Tool ID	7	6	7	6	7	7	2	9	6	10	7	2	9	10	6	9	10	2	1	5
TAD	+z	−x	−x	−z	−z	−z	−z	−z	−z	−z	−a	−a	−a	−a	+x	+z	+z	−z	−z	−z
NMC=1, NS=7, MCC=160, SC=700, MC=1,730; total cost, 2,590																				
Process plan 2																				
Sequence order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Operation ID	1	6	2	5	11	12	13	14	17	18	7	8	9	10	19	20	3	4	15	16
Machine ID	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	1	1
Tool ID	6	6	7	7	7	2	9	10	7	6	7	2	9	10	9	10	6	2	1	5
TAD	+z	−z	−z	−z	−z	−z	−z	−z	−x	−x	−a	−a	−a	−a	+z	+z	+x	−z	−z	−z
NMC=1, NS=7, MCC=160, SC=700, MC=1,730; total cost, 2,590																				
Process plan 3																				
Sequence order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Operation ID	1	11	2	6	12	13	14	18	15	16	3	5	4	19	20	7	8	9	10	17
Machine ID	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Tool ID	6	7	7	7	2	9	10	6	1	5	6	7	2	9	10	7	3	9	10	7
TAD	+z	−z	−z	−z	−z	−z	−z	−z	−z	+x	+x	+z	+z	+z	+z	−a	−a	−a	−a	−x
NMC=0, NS=6, MCC=0, SC=100, MC=2,000; total cost, 2,600																				

2. The approach can generate multiple optimal process plans. The availability of alternative optimal process plans can provide the production scheduling module with the flexibility to select different plans depending on the status of machining resources and make the scheduling module implement the scheduling algorithm under a more relaxed set of constraints.

Future work includes the development of the algorithm for the automatic identification of the precedence relationships between operations, further elaborating the evaluation criterion of machining costs to adapt well to more complex and practical machining environments.

Acknowledgments This work was financially supported by the National Natural Science Foundation of China (grant number 50675078). The authors are appreciative of the referee's careful and valuable comments that helped to improve the manuscript.

References

1. Cay F, Chassapis C (1997) An IT view on perspectives of computer aided process planning research. J Comput Ind 34:307–337
2. Xun X, Wang LH, Newmanc ST (2011) Computer-aided process planning—a critical review of recent developments and future trends. Int J Comput Integr Manuf 24(1):1–31
3. Dutta D, Yip HD (1998) A genetic algorithm application for sequencing operations in process planning for parallel machining. IIE Trans 28:55–68
4. Zhang F, Zhang YF, Nee AYC (1997) Using genetic algorithms in process planning for job shop machining. IEEE Trans Evolutional Comput 1(4):278–289
5. Reddy SVB, Shunmugam MS et al (1999) Operation sequencing in CAPP using genetic algorithms. Int J Prod Res 37(5):1063–1074
6. Li WD, Ong SK, Nee AYC (2002) Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts. Int J Prod Res 40(8):1899–1922
7. Qiao L, Wang XY, Wang SC (2000) A GA based approach to machining operation sequencing for prismatic parts. Int J Prod Res 38(14):3283–3303
8. Lee H, Kim SS (2001) Integration of process planning and scheduling using simulation based genetic algorithms. Int J Adv Manuf Technol 18:586–590
9. Rocha J, Ramos C, Vale Z (1999) Process planning using a genetic algorithm approach. Proceedings of the 1999 IEEE International symposium on Assembly and Task Planning. Porto, Portugal, July. pp 338–343
10. Alam MR, Lee KS, Rahman M, Zhang YF (2003) Process planning optimization for the manufacture of injection moulds using a generic algorithm. Int J Comput Integr Manuf 16(3):181–191
11. Li L, Fuh JYH, Zhang YF, Nee AYC (2005) Application of genetic algorithm to computer-aided process planning in distributed manufacturing environments. Robot Comput Integr Manuf 21:568–578
12. Li XY, Shao XY, Gao L (2008) Optimization of flexible process planning by genetic programming. Int J Adv Manuf Technol 38:143–153
13. Ding L, Yue Y, Ahmet K, Jackson M, Parkin R (2005) Global optimization of a feature-based process sequence using GA and ANN techniques. Int J Prod Res 43(15):3247–3272
14. Ma GH, Zhang YF, Nee AYC (2000) A simulated annealing-based optimization for process planning. Int J Prod Res 38(12):2671–2687
15. Ong SK, Ding J, Nee AYC (2002) Hybrid GA and SA dynamic set-up planning optimization. Int J Prod Res 40(18):4697–4719
16. Li WD, McMahon CA (2007) A simulated annealing-based optimization approach for integrated process planning and scheduling. Int J Comput Integr Manuf 20(1):80–95
17. Guo YW, Mileham AR, Owen GW, Li WD (2006) Operation sequencing optimization using a particle swarm optimization approach. Proc IMechE B J Eng Manuf 220:1945–1958
18. Jerald J, Asokan P, Prabaharan G, Saravanan R (2005) Scheduling optimization of flexible manufacturing systems using particle swarm optimization algorithm. Int J Adv Manuf Technol 25:964–976
19. Amaitik SM, Engin Kiliç S (2007) An intelligent process planning system for prismatic parts using STEP features. Int J Adv Manuf Technol 31:978–993
20. Prabhu P, Elhence S, Wang H (1990) An operations network generator for computer aided process planning. J Manuf Syst 9:283–291
21. Irani SA, Koo HY, Raman S (1995) Feature-based operation sequence generation in CAPP. Int J Prod Res 38(17):4285–4300
22. Lin AC, Lin SY, Lu WF (1998) An integrated approach to determining the sequence of machining operations for prismatic parts with interacting features. J Mater Process Technol 73:234–250
23. Case K, Wan H (2000) Feature-based representation for manufacturing planning. Int J Prod Res 38(17):4285–4300
24. Maropoulos PG, Baker RP (2000) Integration of tool selection with design (part 1. Feature creation and selection of operations and tools). J Mater Process Technol 107:127–134
25. Zhang HC, Lin E (1999) A hybrid graph approach for automated set-up planning in CAPP. Robot Comput Integr Manuf 15:89–100
26. Faheem W, Hayes CC, Castano JF, Gaines DM (1998) What is manufacturing interaction? Proceedings of the 1998 ASME Design Engineering Technical Conference, Atlanta, GA, USA, 1–6
27. Ciurana J, G. Romeu ML, Ferrer I, Casades M (2008) A model for integrating process planning and production and control planning in machining processes. Robot Comput Integr Manuf 24(4):532–544
28. Blanch R, Pellicer N, G-Romeu ML, Ciurana J (2011) Machining processes time calculating tool integrated in computer aided process planning (CAPP) for small and medium enterprises (SMEs). Int J Comput Integr Manuf 24(1):40–52
29. Srinivas M, Patnaik LM (1994) Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans Syst Man Cybern 24(4):656–667