Journal of Universal Computer Science, vol. 15, no. 9 (2009), 1907-1923
submitted: 15/8/08, accepted: 24/5/09, appeared: 1/5/09

Application of Intelligent Strategies for Cooperative Manufacturing Planning

Weidong Li
(Coventry University, Coventry, UK
weidong.li@coventry.ac.uk)

Liang Gao, Xinyu Li
(Huazhong University of Science and Technology, Wuhan, China
gaoliang@mail.hust.edu.cn, lixy@smail.hust.edu.cn)

Abstract: Manufacturing planning is crucial for the quality and efficiency of product development. Process planning and scheduling are the most important and challenging tasks in manufacturing planning. These two processes are usually arranged in a sequential way. Recently, a significant trend is to make the processes to work more concurrently and cooperatively to achieve a globally optimal result. In this paper, several intelligent strategies have been developed to build up Cooperative Process Planning and Scheduling (*CPPS*). Three Game Theory-based strategies, i.e., Pareto strategy, Nash strategy and Stackelberg strategy, have been introduced to analyze the cooperative integration of the two processes in a systematic way. To address the multiple constraints in *CPPS*, a fuzzy logic-based Analytical Hierarchical Process (*AHP*) technique has been applied. Modern heuristic algorithms, including Particle Swarm Optimization (*PSO*), Simulated Annealing (*SA*) and Genetic Algorithms (*GAs*), have been developed and applied to *CPPS* to identify optimal or near-optimal solutions from the vast search space efficiently. Experiments have been conducted and results show the objectives of the research have been achieved.

Keywords: Collaborative system, Game Theory, Analytical Hierarchical Process, Particle Swarm Optimization, Simulated Annealing, Genetic Algorithms
Categories: I.1.2, I.1.4, I.2.1, I.2.4, J.6

1 Introduction

Product development is comprised of various stages, such as conceptual design, detailed design, prototyping, manufacturing planning, manufacturing and testing, etc. The task of manufacturing planning is to interpret product models created by a design team in terms of manufacturing processes, and associate the manufacturing equipments and resources in shop floors with the interpretation. The major tasks of manufacturing planning are process planning and scheduling. For a model of design (e.g., models for vehicles and aircraft), it needs a series of manufacturing operations (operations in the following content) to make it. For example, to make a hole, it could include a drilling operation and reaming operation. The task of process planning is more a product model-oriented. It interprets a model into some detailed operations, such as primary operations (e.g., forging or casting to generate the rough shape), rough machining, semi-finish machining, finish machining, surface treatment, painting, etc. When many models are made together, there is a competition of manufacturing resources such as machines, cutting tools, operators, etc. The task of scheduling is to assign suitable resources to a batch of models to achieve global optimization. A good schedule should utilize the full potential of resources while time and cost should be as short and low as possible. Process planning is usually arranged prior to scheduling in practice. The two functions could have different objectives and there might be many routes to choose from, especially when the number of models is large and the models are complex in terms of geometry and technical specifications. On the other hand, process planning and scheduling are naturally linked. It will simplify the multiple decision-making processes and provide a globally optimal viewpoint if the two functions are well integrated.

In this paper, a novel approach has been developed to establish Collaborative Process Planning and Scheduling (*CPPS*) in manufacturing planning. The Game Theory-based strategies, a fuzzy logic algorithm, and the modern heuristic algorithms, have been applied to solve the collaborative problem. Experiment results to verify the effectiveness of the approach are presented.

The rest of the paper is organized as follows. In Section 2, the related work is reviewed. In Section 3, the *CPPS* problem is modelled. In Section 4, discussions on the application of the game theory for the *CPPS* problem are given. Section 5 presents the constraint representation and handling. In Section 6, intelligent algorithms to solve the *CPPS* problem are introduced. Experiment results are presented in Section 7. Section 8 concludes the work.

2 Related Work

To develop a collaborative working environment is an important research area in computer-based applications. To enable it, modern computing and artificial intelligence technologies have been widely used [Schrum, 06] [Tomek, 01] [Lukosch, 08] [Wurdel, 08].

In the past decade, a few of research works have been reported to integrate process planning and scheduling to optimize decisions. Some earlier works of the integration strategy have been summarized in [Tan, 00]. The most recent works are summarized in [Zhang, 03] [Li, 07] according to two general categories: the enumerative approach and the simultaneous approach. In the enumerative approach ([Tonshoff, 89] [Huang, 95] [Aldakhilallah, 99] [Sormaz, 03] [Zhang, 03]), multiple alternative process plans are first generated for each part. A schedule can be determined by iteratively selecting a suitable process plan from alternative plans of each part to replace the current plan until a satisfactory performance is achieved. The simultaneous approach ([Li, 07] [Moon, 02] [Kim, 03] [Yan, 03] [Moon, 05] [Zhang, 05]) is based on the idea of finding a solution from the combined solution space of process planning and scheduling. In this approach, the process planning and scheduling are both in dynamic adjustment until specific performance criteria can be satisfied. Although this approach is more effective and efficient in integrating the two functions, it also enlarges the solution search space significantly. For this complex decision-making process, further studies are still required, especially in complex situations. To minimize the research gap, in this paper, research has been carried out from the following three aspects:

- It is imperative to develop a strategy to make the two functions to work together in a more cooperative way, that is, *CPPS*. With the *CPPS* strategy, different objectives can be prioritized flexibly, and the two functions can be adjusted in a cooperative way to meet both of the targets. In this research, game theory, which is the formal study of decision-making processes where several players (e.g., functions) make choices that potentially affect the interests of each other, has been introduced to analyze the cooperation of the functions in a systematic way;
- In practical situations, it might be impossible to satisfy all constraints in a process plan. For example, a high accuracy hole as a datum surface should be machined with a high priority according to the primary surfaces constraint, but it may be in conflict with the constraint of planes prior to holes and slots. Therefore, a fuzzy logic-based Analytical Hierarchical Process (*AHP*) technique has been applied to handle the complex constraints effectively;
- The complexity of manufacturing planning brings forth a vast search space when identifying good solutions. Three modern heuristic algorithms, i.e., Particle Swarm Optimization (*PSO*), Simulated Annealing (*SA*), and Genetic Algorithm (*GA*), have been developed and benchmarked in this research to facilitate the search process with optimal or near-optimal solutions. Essential performance criteria, such as makespan, the balanced level of machine utilization, job tardiness and manufacturing cost, have been defined in the algorithms to address the various practical requirements.

3 Modelling of *CPPS*

The *CPPS* problem can be defined as follows:

- Given a set of design models, each with a number of operations and set-up plans, to be processed on a set of manufacturing resources (machines and tools) in a job shop floor;
- Alternative process plans and schedules can be generated through process planning and scheduling flexibility strategies [2]. The processing planning flexibility refers to the possibility of performing an operation on alternative machines with alternative tools or set-up plans, and the possibility of interchanging the sequence in which the operations are executed. The scheduling flexibility corresponds to the possibility of generating alternative schedules for jobs by arranging the different sequences of parts to be machined [2];
- Through selecting suitable manufacturing resources and sequence the operations, process plans and schedules, in which constraints among operations are satisfied and pre-defined objectives are achieved, can be generated.

This problem is illustrated in Figure. 1. For example, there are 3 parts that can be machined by 3, 2 and 3 operations on 3 machines, respectively. For different parts, there are constraints among the operations to make them (Part1: Oper1 → Oper2 → Oper3; Part2: Oper4 → Oper5; Part3: Oper6 → Oper7 → Oper8). When all these 8 operations are sequenced as (Oper1 → Oper4 → Oper2 → Oper6 → Oper3 → Oper7 → Oper8 → Oper5 as shown in Figure 1) and manufacturing resources (machine, tool and set-ups) are specified, the schedule can be determined accordingly. The *CPPS* problem is to optimize the operation sequence and select the manufacturing resources so as to achieve the optimal or near-optimal process planning and scheduling objectives while maintain the manufacturing feasibility with the satisfactory of constraints.

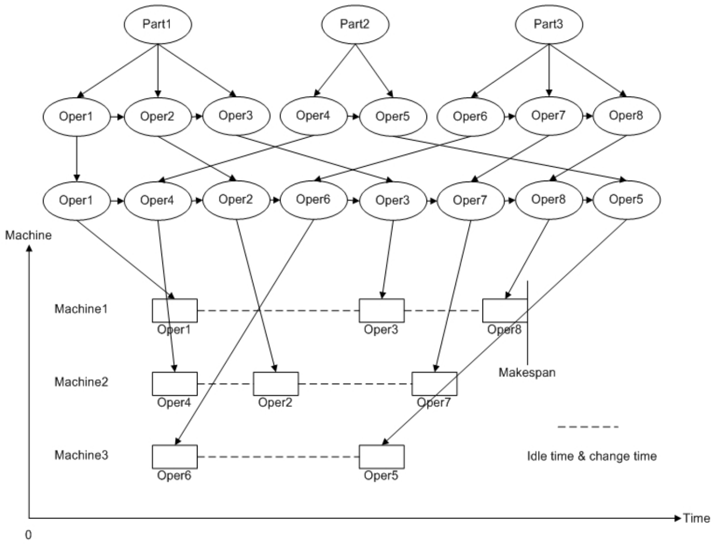


Figure 1: Illustration of the CPPS problem

The *CPPS* problem can be modeled as an extension of the operation sequencing optimization problem relating to a single model [Li, 02] [Guo, 06] into multiple models with the *CPPS* objectives. When the process plans of all models are generated and the manufacturing resources are specified, it is required to determine the schedule based on this information and calculate the makespan, total tardiness, etc. Here, four evaluation criteria of the *CPPS* problem can be calculated as follows.

(1) Makespan: $Makespan = \mathop{Max}_{j=1}^{m}(Machine[j].Available\_time)$.

(2) Total job tardiness: The due date of a part is denoted as DD, and the completion moment of the part is denoted as CM . Hence,

$$Part\_Tardiness = \begin{cases} 0 & if\ DD\ is\ later\ than\ CM \\ CM - DD & Otherwise \end{cases}$$

(3) Balanced level of machine utilization: the Standard Deviation concept is introduced here to evaluate the balanced machine utilization (assuming there are m machines, and each machine has n operations).

$$Machine[j].Utilization = \sum_{i=1}^{n}(Operation[i].Mac\_T) \text{ , } (j = 1,..,m)$$

$$\chi = \frac{\sum_{j=1}^{m}(Machine[j].Utilization)}{m}$$

$$Utilization\_Level = \sqrt{\sum_{j=1}^{m}(Machine[j].Utilization - \chi)^2}$$

(4) Manufacturing cost for the process plan of a part: In [Guo, 06], the manufacturing cost associated with the process plan of a part has been defined in terms of machine utilization, tool utilization, set-up changes, machine changes and tool changes. The relevant computations are elaborated in [Guo, 06].

4 Applications of Game Theory on *CPPS*

Game theory is a good tool to analyze the interaction and cooperation of decision makers with various objectives [Rasmusen, 01] [Xiao, 05]. For example, economists have used it as a tool to examine the actions of firms in a market. Recently, it has been applied to some complex engineering problems, such as communications and networks, power systems, collaborative product design, etc. Game theory consists of a series of strategies that are applicable for various situations. Here, three popular strategies in the game theory have been applied to *CPPS*, i.e., Pareto strategy, Nash strategy and Stackelberg strategy. The concepts for the three strategies are briefed below.

- Pareto strategy. A full cooperative solution between two players. Players in the game theory can represent a person, a team or a functional module. The strategy is to combine the objectives of two players as a single goal through weights.
- Nash strategy. Each player must make a set of decisions that is rational to him/her by assuming another player's reaction. If there is an overlap between these players' reactions, the result can be selected from the overlap.
- Stackelberg strategy. A leader-follower solution, which is well suitable for a situation in which one player dominates the decision-making process.

For the *CPPS* problem, the objectives of process planning and scheduling need to be considered from the cooperative point of view to achieve a balanced and overall target. In many cases, objectives from process planning and scheduling could be conflicted. For example, a lower manufacturing cost for making a part can be achieved through the intensive utilization of cheap machines, but it could be conflicted with the criterion for the balanced utilization of machines. Through applying the above three game theory-strategies, the solution of *CPPS* is flexible and adjustable according to various practical situations and users' specific requirements.

The application of the Pareto strategy is to combine the objectives from process planning and scheduling respectively with weights. The strategy is illustrated in Figure 2(a). The major characteristic of the strategy is that the objective of process planning is closely associated with that of scheduling. With the combined consideration, the strategy equals a single level decision-making process so that iteratively empirical process can be avoided. However, a serious problem is that it is difficult to determine a reasonable combination weight with engineering meanings.

Therefore, the strategy is more suitable for the purpose of comparison and trend studies.

A usual practice to use the Nash strategy in the *CPPS* problem is to apply the following procedure to the two functions. Process planning (or scheduling) is invoked to produce a number of alternative plans with the satisfaction of the process planning (or scheduling) objective and constraints, from which scheduling (or process planning) can choose and further decide a group of satisfactory solutions (denoted as Solutions) according to the scheduling (or process planning) objective. The overlapped set of the above two Solutions is the final solution of the *CPPS* problem. The process is illustrated in Figure 2(b). The strategy is characterized as a more independent decision-making process for each functional module, and both the objectives can be considered in a reasonable way. The Nash strategy is the same effect as the Pareto strategy when the objectives of process planning and scheduling are harmonious. When the objectives are contradictive, the results of the Nash strategy is more rational compared with that of the Pareto strategy, which much depends upon the setting of the weight.

In the application of the Stackelberg strategy to the *CPPS* problem, for the dominate function (process planning or scheduling), a number of alternative plans with the satisfaction of the function's objective and constraints are generated, from which another function can choose and further decide a satisfactory solution (illustrated in Figure 2(c)). This strategy is different from the Nash strategy in that the latter creates a larger computation space while the computation of the former is mainly constrained by the dominant function. The characteristic of the Stackelberg strategy is that it can fully satisfy the most important objective while the minimum conditions of other objectives can be met. However, the value of one function could be discounted in another module. For instance, to schedule parts based on generated process plans sometimes causes some machines to be overloaded to restrict the capabilities of the machines.

In this research, the *CPPS* model is equipped with the above three strategies for users to choose to meet their requirements.

5 Handling of Constraints in *CPPS*

Manufacturing processes are complex [Kalpakjian, 03]. There are many technical specifications and requirements. In *CPPS*, a number of constraints, which arise from geometric shapes of parts, technical restrictions, best practices, etc., are represented. A feasible solution of *CPPS* must comply with the constraints. These constraints can be summarized below [Ding, 05].

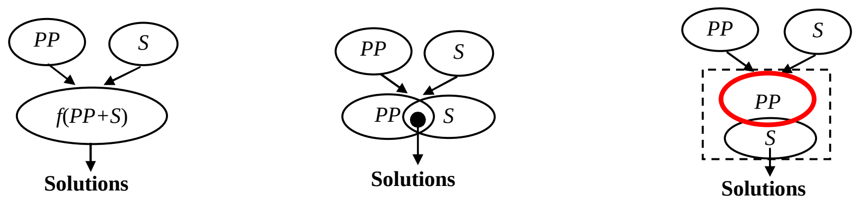
(1) Precedence constraints

- A parent feature should be processed before its child features.
- Rough machining operations should be done before semi-finish and finish machining operations.
- Primary surfaces should be machined prior to secondary surfaces. Primary surfaces are usually defined as surfaces with high accuracy or having a high

impact on the design specifications, such as a datum plane. The rest of the surfaces are regarded as secondary surfaces, e.g., a threaded hole.

- Planes should be machined prior to holes and slots.
- Edge cuts should be machined last.

PP – Process planning ; S - Scheduling



(a) Pareto strategy (b) Nash strategy (c) Stackelberg strategy

Figure 2: Illustration of the game theory strategies

(2) Successive constraints

- Features or operations, which can be machined within the same set-up should be machined successively.
- Features to be machined with the same cutting tool should be machined successively.
- Operations of the same type, such as rough, semi-finish and finish machining, should be executed successively.
- Features with similar tolerance requirements should be machined successively on the same machine tool.

(3) Auxiliary constraints

- Annealing, normalizing and ageing operations of ferrous metal component should be arranged before rough machining or between rough and semi-finish machining.
- Quenching for ferrous metal workpieces should be arranged between semi-finish and finish machining or between rough and semi-finish machining if it is followed by high temperature tempering.
- Quenching for non-ferrous metals should be arranged between rough and semi-finish machining or before rough machining.
- Carburizing would be arranged between semi-finish and finish machining.

To address the complexity of constraints, the *AHP* technique [Golden, 89], which specifies a set of fuzzy logic-based numerical weights to represent the relative importance of the constraints of *CPPS* with respect to a manufacturing environment, has been applied to evaluate the satisfaction degree of the constraints. The relevant computation is depicted below.

Step 1: The constraints are organized in a hierarchy structure, which includes an overall objective (Level 1), three general constraint groups (Level 2) and rules under each constraint group (Level 3). This situation is illustrated in Figure 3. For Level 2, a 3×3 pair-wise matrix (R^0*-matrix*) is created, where the number in the *i*th row and *j*th column, r_{ij}, specifies the relative importance of the *i*th group of constraints as compared with the *j*th group of constraints. For Level 3, three pair-wise matrices are created for each group of constraints (R^1*-matrix* (5×5) for Precedence constraints, R^2*-matrix* (4×4) for Succession constraints, and R^3*-matrix* (3×3) for Auxiliary constraints). Similarly, the number in the matrix (r_{ij}) specifies the relative importance of rules within each category of constraints. A *R-matrix* can be described as:

$$R = \begin{bmatrix} r_{11} & \cdot & r_{1i} & \cdot & r_{1m} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ r_{i1} & \cdot & r_{ii} & \cdot & r_{im} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ r_{m1} & \cdot & r_{mi} & \cdot & r_{mm} \end{bmatrix}$$

where $i = 1, 2, \ldots, m$ (m is the number of groups of constraints in Level 2 or the number of rules for each constraint group in Level 3),

$r_{ii} = 1$, and

$r_{ij} = 1/r_{ji}$.

Step 2: Evaluating criteria based on a 1-9 scale for the *R-matrices*, which are used to indicate the relative importance of two elements, are defined in Table 1. In order to get more neutral results, a group of experts is invited to fill in the four *R-matrices* according to their experience and knowledge.

For instance, considering two rules in the category of Precedence constraints - *Rule 2* and *Rule 4*:

Rule 2: Primary surfaces should be machined prior to secondary surfaces.
Rule 4: Planes should be machined prior to holes and slots.

From the perspective of an individual expert, if he thinks *Rule 2* is much more important than *Rule 4*, a weight of '7' is inserted in the juncture cell (r_{24}) of his filled R^1*-matrix*. On the contrary, the value in the juncture cell (r_{42}) is set to '1/7'.

Step 3: For Level 2 and Level 3, four weight vectors $W^0 - W^3$, which correspond to the four *R-matrices* respectively, are computed. The computation process consists of the following three steps.

(1) Multiplication (M) of all elements in each row of a *R-matrix* is computed as:

$$M_i = \prod_{j=1}^{n} r_{ij}$$

where j is the column index of elements, j =1, 2, …, n,
i is the index row of elements, i =1, 2, …, n, and
n is the number of the rows (columns) in a *R-matrix*.

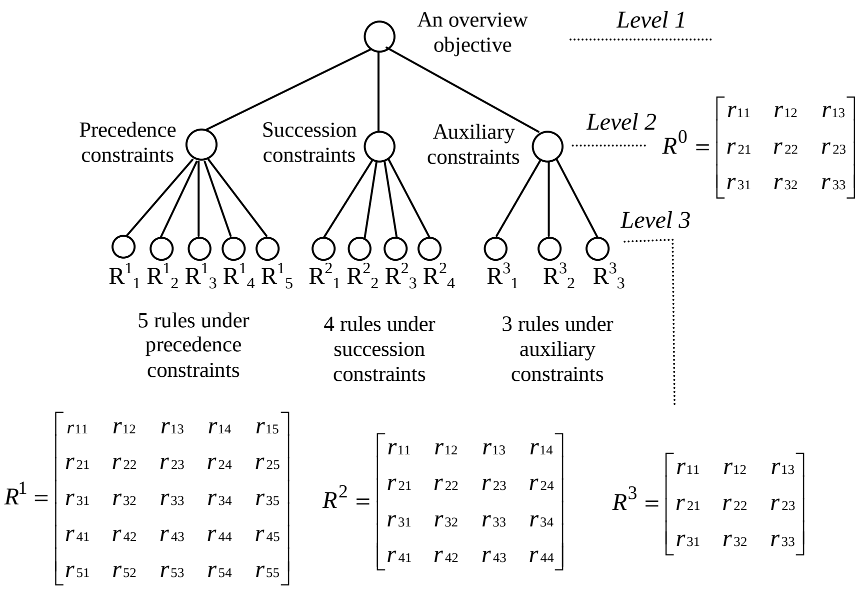


Figure 3: A three-level hierarchy structure for the constraints

Table 1: Evaluation criteria for R-matrices

Definition	Intensity of importance (r_{ij})	Intensity of importance (r_{ji})
The *i*th rule and the *j*th rule have equal importance	1	1
The *i*th rule is slightly more important than the *j*th rule	3	1/3
The *i*th rule is more important than the *j*th rule	5	1/5
The *i*th rule is much more important than the *j*th rule	7	1/7
The *i*th rule is absolutely more important than the *j*th rule	9	1/9
Intermediate values between adjacent scale values	2, 4, 6, 8	1/2, 1/4, 1/6, 1/8

(2) The *n*th root of M is calculated, that is:

$$\overline{w_i} = \sqrt[n]{M_i}$$

where *i* is the row (column) number in a *R-matrix*, and *i* =1, 2, …, *n*.

Therefore, the relative importance weight vector can be built as follows:

$$\overline{W} = \left|\overline{w_1},\quad \overline{w_2},\quad ...,\quad \overline{w_n}\right|$$

Each element of the weight vector W $(|w_1,\quad w_2,\quad ...,\quad w_n|)$ is finally generated through a normalization operation.

$$w_i = \frac{\overline{w_i}}{\sum_{j=1}^{n} \overline{w_j}}$$

For each W, it should be eventually denoted as $W^0 - W^3$ according to the individual computation process.

Step 4: There are totally 12 rules defined in this system (5 rules from Precedence constraints + 4 rules from Succession constraints + 3 rules from Auxiliary constraints). The element of a total weight vector for each rule - W^t $(|w^t{}_1,\quad w^t{}_2,\quad ...,\quad w^t{}_{12}|)$ can be generated as:

$$w^t{}_{1-5} = w^0{}_1 * w^1{}_{1-5}\ ,\quad w^t{}_{6-9} = w^0{}_2 * w^2{}_{1-4}\ ,\quad w^t{}_{10-12} = w^0{}_3 * w^3{}_{1-3}$$

Step 5: A series of *V-matrices* are designed to record the situation of violating constraints for a process plan. For instance, for *Rule k*, its *V-matrix* is defined as:

$$V_k = \begin{bmatrix} v_{k11} & \cdot & v_{k1i} & \cdot & v_{k1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ v_{ki1} & \cdot & v_{kii} & \cdot & v_{kin} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ v_{kn1} & \cdot & v_{kni} & \cdot & v_{knn} \end{bmatrix}$$

where n is the number of operations in a process plan,

$$v_{ij} = \begin{cases} 1 & \text{if Operation i prior to Operation j is against Rule k} \\ 0 & \text{if Operation i prior to Operation j obey Rule k} \end{cases}, \text{ and}$$

$$v_{ji} = 1 - v_{ij}\ .$$

Step 6: The value to evaluate the manufacturability of a process plan is determined. f_m is finally calculated as:

$$f_m = \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} w_k{}^t v_{kij}$$

where *m* is the total rule number of the constraints (here *m* =12).

6 Applications of Modern Heuristic Algorithms

The *CPPS* problem usually brings forth a vast search space. Conventional algorithms are often incapable of optimizing non-linear multi-modal functions. To address this problem effectively, some modern optimization algorithms, such as *GA* and *SA*, have been developed recently to quickly find a solution in a large search space through some evolutional or heuristic strategies. In this research, three modern algorithms, i.e., *PSO*, *SA* and *GA*, have been applied to facilitate the search process. In [Li, 02] [Guo, 06], the three algorithms have been successfully applied to process planning optimization problems. Here, the algorithms have been developed further to solve the *CPPS* problem. The application of an improved *PSO* process is explained here for illustration. More details of *SA* and *GA* can refer to [Li, 02] [Li, 07].

A standard *PSO* algorithm was inspired by the social behavior of bird flocking and fish schooling [Kennedy, 95]. Three aspects will be considered simultaneously when an individual fish or bird (particle) makes a decision about where to move: (1) its current moving direction (velocity) according to the inertia of the movement, (2) the best position that it has achieved so far, and (3) the best position that its neighbor particles have achieved so far. In the algorithm, the particles form a swarm and each particle can be used to represent a potential solution of a problem. In each iteration, the position and velocity of a particle can be adjusted by the algorithm that takes the above three considerations into account. After a number of iterations, the whole swarm will converge at an optimized position in the search space.

A traditional *PSO* algorithm can be applied to optimize *CPPS* in the following steps:

(1) Initialization
- Set the size of a swarm, e.g., the number of particles "*Swarm_Size*" and the max number of iterations "*Iter_Num*".
- Initialize all the particles (a particle is a *CPPS* solution) in a swarm. Calculate the corresponding criteria of the particles (a result is called *fitness* here).
- Set the local best particle and the global best particle with the best *fitness*.

(2) Iterate the following steps until *Iter_Num* is reached
- For each particle in the swarm, update its velocity and position values.
- Decode the particle into a *CPPS* solution in terms of new position values and calculate the *fitness* of the particle. Update the local best particle and the global best particle if a lower *fitness* is achieved.

(3) Decode global best particle to get the optimized solution

However, the traditional *PSO* algorithm introduced above is still not effective in resolving the operation sequencing problem. There are two major reasons for it:

- Due to the inherent mathematical operators, it is difficult for the traditional *PSO* algorithm to consider the different arrangements of machines, tools and set-ups for each operation, and therefore the particle is unable to fully explore the entire search space.
- The traditional algorithm usually works well in finding solutions at the early stage of the search process (the optimization result improves fast), but is less efficient during the final stage. Due to the loss of diversity in the population, the particles move quite slowly with low or even zero velocities and this makes it hard to reach the global best solution. Therefore, the entire swarm is prone to be trapped in a local optimum from which it is difficult to escape.

To solve these two problems and enhance the ability of the traditional *PSO* algorithm to find the global optimum, new operations, including mutation, crossover and shift, have been developed and incorporated in an improved *PSO* algorithm. Meanwhile, considering the characteristics of the algorithm, the initial values of the particles have been well planned. Some modification details are depicted below.

(1) New operators in the algorithm

- Mutation. In this strategy, an operation is first randomly selected in a particle. From its candidate machining resources (machines, tools, set-ups), an alternative set (machine, tool, set-up) is then randomly chosen to replace the current machining resource in the operation.
- Crossover. Two particles in the swarm are chosen as Parent particles for a crossover operation. In the crossover, a cutting point is randomly determined, and each parent particle is separated as left and right parts of the cutting point. The positions and velocities of the left part of Parent 1 and the right part of Parent 2 are reorganized to form Child 1. The positions and velocities of the left part of Parent 2 and the right part of Parent 1 are reorganized to form Child 2.
- Shift. This operator is used to exchange the positions and velocities of two operations in a particle so as to change their relative positions in the particle.

(2) Escape method

- During the optimization process, if the iteration number of obtaining the same best fitness is more than 10, then the mutation and shift operations are applied to the best particle to try to escape from the local optima.

7 Experimental Results

A group of 8 parts taken from [Li, 02] [Guo, 06] have been used for experiments. The relevant specifications of the parts are given in Table 2. The results of the following two conditions are taken first to demonstrate the performances of the chosen criteria:

(1) The criteria are manufacturing cost and makespan according to the Pareto strategy.
(2) The criteria are manufacturing cost and the balanced utilization of machines according to the Pareto strategy.

All of the results are prone to stabilization after several hundreds of iterations. Figure 4 indicates clearly that the manufacturing cost and the makespan follow the similar trends since the reduced numbers of set-ups, machine changes, and tool changes contribute to both of the lower manufacturing cost and the shorter makespan. Therefore, the effects of the three game theory strategies are the same. For the situation with conflicting objectives of *CPPS* like Figure 5, when the Stackelberg strategy is applied, the satisfactory results are within the highlighted region A (The balanced utilization of machines is the leader criteria. Higher values means unbalanced level) or B (Manufacturing cost is the leader criteria). When the Nash strategy is applied, the satisfactory results are within the highlighted region C, and both the objectives, i.e., manufacturing cost and the balanced utilization of machines, are discounted. Therefore, the developed method provides the flexibility to choose the suitable strategy according to the real practical requirement. The three optimization algorithms have been further compared under the same condition (the above Condition 1). To make the diagrams clearly, only the makespan has been chosen and the results are shown in Figure 6.

Table 2: The technical specifications for 8 parts

Part	Number of operations	Number of constraints
1	7 (9, 9, 27, 8, 8, 9, 36)	11
2	8 (9, 9, 36, 18, 27, 8, 27, 18)	11
3	7 (9, 9, 36, 36, 18, 6, 6)	10
4	9 (9, 9, 27, 6, 36, 36, 6, 18, 18)	18
5	7 (9, 9, 36, 36, 36, 18, 6)	13
6	9 (9, 9, 36, 27, 18, 6, 27, 6, 18)	20
7	5 (9, 27, 27, 18, 9)	5
8	7 (9, 9, 27, 36, 36, 6, 6)	13

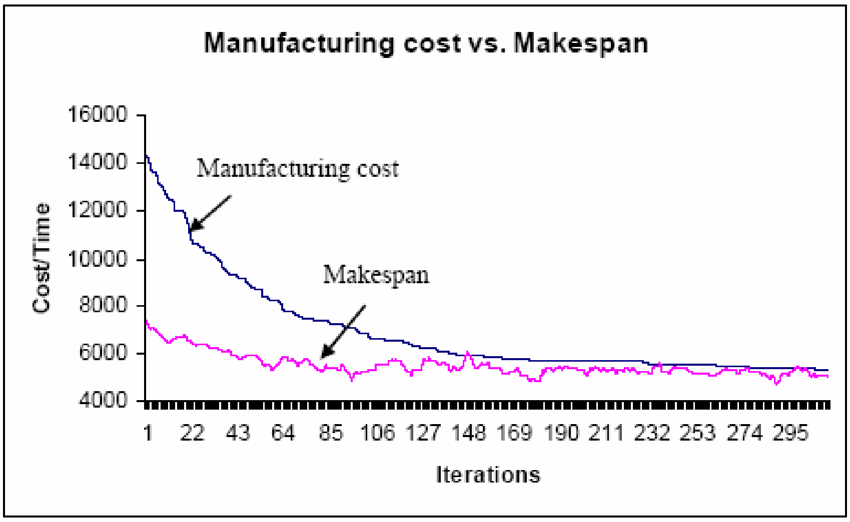


Figure 4: Case 1 of applying three strategies

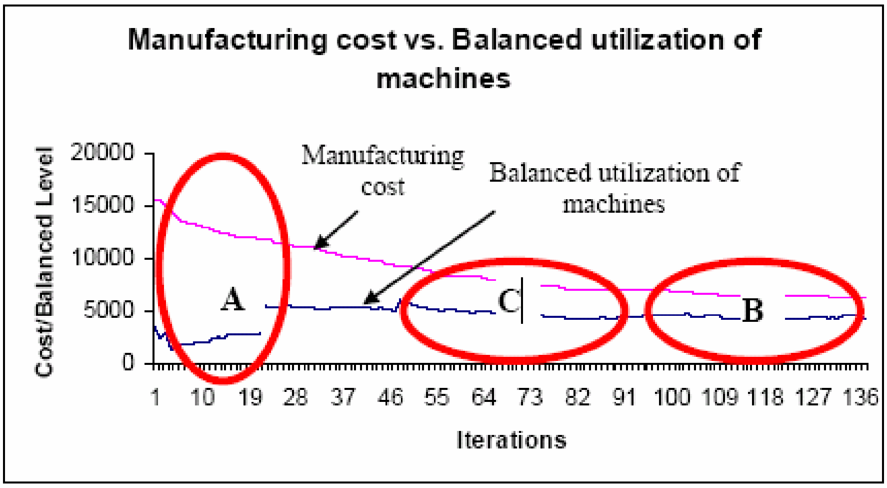


Figure 5: Case 2 of applying three strategies

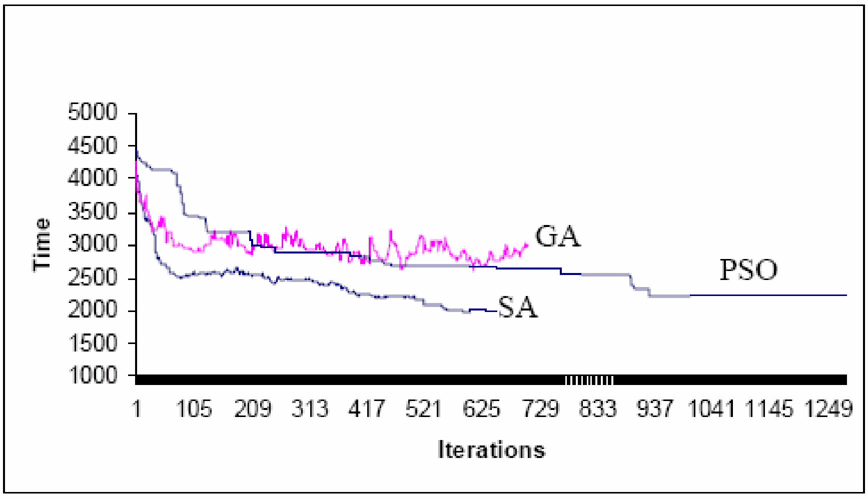


Figure 6: Comparisons of three algorithms

It can be observed that all of the approaches can reach good results, while there are different characteristics due to the inherent mechanisms of the algorithms. The *SA*-based algorithm usually takes shorter time to find good solutions but it is vigilant to its parameters (such as the starting temperature and the cooling parameter) and the problems to be optimized. The *GA*- and *PSO*-based algorithms are slow in finding good solutions but they are robust for optimization problems. Meanwhile, the SA-based approach is much "sharper" to find optimal or near-optimal solutions, and the common shortcoming of the *GA*- and *PSO*-based approaches is that they are prone to pre-maturity in some cases (converge too early and difficult to find the optimal solutions).

8 Conclusions

Manufacturing planning, which mainly include process planning and scheduling, is an important stage in product development. The decision will play a crucial role for the performance of the final products. Usually, process planning and scheduling are arranged in a sequential way. With this arrangement, it is difficult to adjust them in a cooperative way to achieve global optimization. To identify good solutions in manufacturing planning, in this research, *CPPS* has been developed. The contributions of this research include:

- To address *CPPS* effectively, three game theory-based strategies, i.e., Pareto strategy, Nash strategy and Stackelberg strategy, have been used to analyze and facilitate the cooperation of the two processes in a systematic way.
- *AHP* has been introduced to resolve the multiple constraints in the *CPPS* problem. The technique is effective in solving the complex and even conflicting constraints in manufacturing planning.
- To find optimal or near-optimal solutions from the vast search space efficiently, modern intelligent algorithms, including *PSO*, *SA* and *GAs*, have been developed and applied to the *CPPS* problem. Experiments have been conducted and computational results have shown the effectiveness of applying these intelligent strategies. Comparisons have been given to show the characteristics of the algorithms.

Acknowledgements

This research is supported by an applied research fellowship from the Coventry University for international collaboration, and the National Basic Research Program of China (973 Program) under Grant no.2004CB719405, the National High-Tech Research and Development Program of China (863 Program) under Grant nos.2007AA04Z107 and 2006AA04Z131 from the Huazhong University of Science and Technology.

References

[Aldakhilallah, 99] Aldakhilallah, K.A., Ramesh, R.: Computer-Integrated Process Planning and Scheduling (CIPPS): Intelligent Support for Product Design, Process Planning and Control, International Journal of Production Research, vol. 37, no. 3 (1999), 481-500.

[Ding, 05] Ding, L., Yue, Y., Ahmet, K., Jackson, M., Parkin, R.: Global optimization of a feature-based process sequence using GA and ANN techniques, International Journal of Production Research, vol. 43 (2005), 3247-3272.

[Golden, 89] Golden, B.L., Harker, P.T., Wasil, E.E.: The Analytic Hierarchy Process: Applications and Studies. Springer-Verlag, 1989.

[Guo, 06] Guo, Y.W., Mileham, A.R., Owen, G.W., Li, W.D.: Operation Sequencing Optimization Using a Particle Swarm Optimization Approach, IMechE, Journal of Engineering Manufacture, vol. 220, no. 12 (2006), 1945-1958.

[Huang, 95] Huang, S.H., Zhang, H.C., Smith, M.L.: A Progressive Approach for the Integration of Process Planning and Scheduling, IIE Trans., vol. 1 (1995), 456-464.

[Kalpakjian, 03] Kalpakjian, S., Schmid, S.R.: *Manufacturing* Processes for Engineering Materials, (4th Edition), Prentice Hall, 2003.

[Kennedy, 95] Kennedy, J., Eberhart, R.: Particle Swarm Optimization, Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, (IEEE Service Centre, Piscataway, New Jersey), IV (1995), 1942-1948.

[Kim, 03] Kim, Y.K., Park, K., Ko, J.: A Symbiotic Evolutionary Algorithm for the Integration of Process Planning and Job Shop Scheduling, Computers & Operations Research, vol. 30 (2003), 1151-1171.

[Li, 02] Li, W.D., Ong, S.K., Nee, A.Y.C.: A Hybrid Genetic Algorithm and Simulated Annealing Approach for the Optimization of Process Plan for Prismatic Parts, International Journal of Production Research, vol. 40, no. 8 (2002), 1899-1922.

[Li, 07] Li, W.D., McMahon, C.A.: A Simulated Annealing-based Optimization Approach for Integrated Process Planning and Scheduling, International Journal of Computer Integrated Manufacturing, vol. 20 , no. 1 (2007), 80-95.

[Lukosch, 08] Lukosch, S.: Seamless Transition between Connected and Disconnected Collaborative Interaction, Journal of Universal Computer Science, vol. 14, no. 1 (2008), 59-87.

[Moon, 02] Moon, C., Kim, J., Hur, S.: Integrated Process Planning and Scheduling with Minimizing Total Tardiness in Multi-plants Supply Chain, Computers & Industrial Engineering, vol. 43 (2002), 331-349.

[Moon, 05] Moon, C., Seo, Y.: Evolutionary Algorithm for Advanced Process Planning and Scheduling in a Multi-plant, Computers & Industrial Engineering, vol. 48 (2005), 311-325.

[Rasmusen, 01] Rasmusen, E.: Games and Information: An Introduction to Game Theory (3rd Edition), Oxford, 2001.

[Schrum, 06] Schrum, L., Lamb, T.A.: Groupware for Collaborative Learning: a Research Perspective on Processes, Opportunities, and Obstacles, Journal of Universal Computer Science, vol. 2, no. 10 (1996), 717-731.

[Sormaz, 03] Sormaz, D., Khoshnevis, B.: Generation of Alternative Process Plans in Integrated Manufacturing Systems, Journal of Intelligent Manufacturing, vol. 14 (2003), 509-526.

[Tan, 00] Tan, W., Khoshnevis, B.: Integration of Process Planning and Scheduling – a Review, Journal of Intelligent Manufacturing, vol.1 (2000), 53-61.

[Tomek, 01] Tomek, I.: Knowledge Management and Collaborative Virtual Environments, Journal of Universal Computer Science, vol. 7, no. 6 (2001), 458 – 471.

[Tonshoff, 89] Tonshoff, H.K., Beckendorff, U., Andres, N.: FLEXPLAN: A Concept for Intelligent Process Planning and Scheduling, Proceedings of the CIRP International Workshop, Hannover, Germany, 1989, 319-322.

[Wurdel, 08] Wurdel, M., Sinnig, D., Forbrig, P.: CTML: Domain and Task Modeling for Collaborative Environments, Journal of Universal Computer Science, vol. 14, no. 19 (2008), 3188-3201.

[Xiao, 05] Xiao, A., *et al*.: Collaborative Multidisciplinary Decision Making Using Game Theory and Design Capability Indices, Research in Engineering Design, vol. 16 (2005), 57-72.

[Yan, 03] Yan, H.S., Xia, Q.F., Zhu, M.R., Liu, X.L., Guo, Z.M.: Integrated Production Planning and Scheduling on Automobile Assembly Lines, IIE Transactions, vol. 35 (2003), 711-725.

[Zhang, 03] Zhang, Y.F., Saravanan, A.N., Fuh J.H.Y.: Integration of Process Planning and Scheduling by Exploring the Flexibility of Process Planning, International Journal of Production Research, vol. 41, no. 3 (2003), 611-628.

[Zhang, 05] Zhang, X.D., Yan, H.S.: Integrated Optimization of Production Planning and Scheduling for a Kind of Job-shop, International Journal of Advanced Manufacturing Technology, vol. 26 (2005), 876-886.