International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

Optimal process plan selection in networked based manufacturing using game-theoretic approach

V.K. Manupati [a] , Sujay Deo [b] , N. Cheikhrouhou [c] & M.K. Tiwari [a]

[a] Department of Industrial Engineering and Management, Indian Institute of Technology Kharagpur, Kharagpur-721302, West Bengal, India

[b] Department of Mechanical Engineering, Indian Institute of Technology Kharagpur, Kharagpur-721302, West Bengal, India

[c] EPFL, Ecole Polytechnique Federale De Lausanne, France

Available online: 14 May 2012

International Journal of Production Research
2012, 1–20, iFirst

Optimal process plan selection in networked based manufacturing using game-theoretic approach

V.K. Manupati[a], Sujay Deo[b], N. Cheikhrouhou[c] and M.K. Tiwari[a]*

[a]*Department of Industrial Engineering and Management, Indian Institute of Technology Kharagpur, Kharagpur-721302, West Bengal, India;* [b]*Department of Mechanical Engineering, Indian Institute of Technology Kharagpur, Kharagpur-721302, West Bengal, India;* [c]*EPFL, Ecole Polytechnique Federale De Lausanne, France*

(*Received 5 November 2011; final version received 30 March 2012*)

This paper proposes a scheme for generating optimal process plans for multi jobs in a networked based manufacturing system. Networked manufacturing offers several advantages in the current competitive atmosphere such as reducing short manufacturing cycle time and maintaining the production flexibility, thereby achieving several feasible process plans. An *N*-person non-co-operative game with complete information is proposed and a mathematical model has been developed to generate the payoff functions. To be part of a game, we divided the game into two sub-games such as games to address sub-game (GASG) and games to solve sub-game (GSSG) which try to interact with each other and achieve the Nash equilibrium (NE). Consequently, a hybrid dynamic-DNA (HD-DNA) based evolutionary algorithm approach has been developed for more effective solutions of the game and also for finding the perfect NE points. The objective of this game is to generate the optimal process plans to minimise the makespan. Finally, three cases having different job complexities are presented to demonstrate the feasibility of the approach. The proposed algorithm is validated and results are analysed to benefit the manufacturer.

Keywords: networked manufacturing; Nash equilibrium; payoff; makespan; process planning; scheduling

1. Introduction

Rapid developments in information technology and network technology have profoundly influenced manufacturing research and its application. However, the products functionality and complexity are intensifying and organisations need to sustain advantage of huge competitiveness in the market. Some of the advantages provided by the global economic competition are short manufacturing cycle time, in-time information, consistent knowledge flow, short supplier times, etc. Hence, computer-based manufacturing is essential for effective decisions to improve the business market. In order to meet quickly expanding demands of global customers the traditional manufacturing systems have to transform themselves. It can be attained through the adoption of a new manufacturing paradigm known as network-based manufacturing or networked manufacturing. As a new and advanced manufacturing paradigm, the networked manufacturing pattern suits the global trends towards a knowledge-based economy and global manufacturing.

Anuj *et al*. (2010) addressed a complex scheduling problem in a flexible manufacturing system (FMS) and it was solved by a novel approach called knowledge-based genetic algorithm (KBGA). The knowledge base has been used on four different stages of simple genetic algorithm (SGA) and the performance measures such as mean flow time and throughput have been measured. The proposed algorithm has been tested and proved with different data sets to measure the efficacy, robustness and flexibility of the algorithm. Henry *et al*. (2002) proposed a scheme for integrated product and process design for generating optimal process plans based on the product raw data. Through this approach, the authors found a significant improvement in terms of resource allocation. Jiang *et al*. (1999) designed an automatic process planning system (APPS) which can generate process plans directly from CAD drawings. Here, with the help of eight modules the APPS has been used to implement the knowledgebase for intelligent interfaces and to generate appropriate process plans. Moreover, network-based manufacturing is born out of industrial demands as well as the emergence of new information technologies (Fan *et al*. 2003). Liu *et al*. (2002) defines networked manufacturing as a set of manufacturing activities, including market control,

*Corresponding author. Email: mkt09@hotmail.com

ISSN 0020–7543 print/ISSN 1366–588X online

http://dx.doi.org/10.1080/00207543.2012.682181
http://www.tandfonline.com

manufacturing technologies and manufacturing systems, that can help enterprises improve business management and enhance their competitiveness in the market. Out sourcing and leagile strategies are the important functions for today organisations. The productivity and profitability of modern manufacturing companies depends on the competitive factor. Chan *et al*. (2001a) proposed a new hybrid 'enhanced shift converging simulated algorithm' (ESCSA) to deal with the above-mentioned strategies in terms of their reduced makespan. In networked manufacturing, the mode of production has been changed from make-to-stock to make-to-order, in which the active customer's participation of submitting manufacturing jobs in manufacturing process is emphasised.

Process planning and scheduling are the two most important functions used to process the jobs in a manufacturing system. These functions specify what manufacturing resources, operations and routes are needed to produce a product and schedule the operations of all the jobs on machines, while the precedence relationships in the process plans are satisfied. Generally in the traditional manufacturing system, process planning and scheduling were carried out in a sequential way. Although, the monolithic approach of traditional manufacturing has its own advantage, it is not sufficient in today's dynamic manufacturing environment. Subsequently, some researchers have realised that there is a greater need to integrate both activities and have found the basis in the context of networked manufacturing environment. Chryssolouris and Chan (1985) and Chryssolouris *et al*. (1984) were the first who introduced the basic idea of the integration of process planning and scheduling. Khoshnevis and Chen (1991) used the concept of integration and introduced a dynamic feedback concept for effective co-ordination. Larsen and Zhang (1993) proposed the integration model and implemented the dynamic feedback and alternative process plans concepts. Usher and Fernandes (1996) presented a computer-aided process planning system, PARIS, which integrated process planning and scheduling. Here, the authors have implemented dynamic process planning by dividing the tasks into static and dynamic phases. Moreover, with the help of an example the overall approach of the dynamic process planning has been implemented. Saygin and Kilic (1999) proposed a framework and implemented the concept of integrating process planning and scheduling in a flexible manufacturing environment for flexible process plans. Chan *et al*. (2001b) developed a framework of an agent-based system for the integration of process planning and production scheduling. Through the implementation of an integrated, distributed and co-operative process planning system (IDCPPS) the performance of manufacturing systems has been improved. A simulated annealing-based and modified genetic algorithm-based approach for the integration of process planning and scheduling to optimise the performances was developed by Li and McMahon (2007) and Shao *et al*. (2009). Wong and Leng (2006) and Mak and Fung (2006) developed an online hybrid agent-based negotiation multi-agent system to integrate process planning with scheduling. Premaratne *et al*. (2010) presented the integration approach for a manufacturing and distribution network within the supply chain of a global car company. The authors found that through the integration approach, interfacing of individual networks was eliminated. Furthermore, they also stated that for further improvement of a supply chain network the integration approach is capable of providing flexibility, visibility and maintainability. Li and Chaoyong (2010) developed an agent-based system with an optimisation agent for improving the generated process plans and scheduling plans. Shukla *et al*. (2008) presented a bidding-based multi-agent system for solving integrated process planning and scheduling. The proposed architecture consists of various autonomous agents capable of communicating with each other and making decisions based on their knowledge. Zhang *et al*. (2003) described the architecture of holon and applied it to process planning and shop floor scheduling. Due to the advantage of the integration approach over traditional approaches, the authors have implemented a holon architecture-based integration approach for process planning and shop floor scheduling. Finally, the performance measures and the flexibility of the system have been improved with proposed architecture. Anuj *et al*. (2011) discussed a problem on process planning in a CIM context for the minimisation of cost of the finished product. Due to the complex nature of the problem the authors have proposed a noble search algorithm, known as knowledge-based artificial immune system (KBAIS). To demonstrate the efficacy of the problem a bench mark problem has been considered and computational experiments are performed. Finally, the authors have shown that the proposed algorithm performs better over other existing algorithms.

The job scheduling problem in a networked manufacturing system is distinct from the monolithic approach. Therefore, the concept of traditional job scheduling is extended and updated and the following characteristics are considered for effective production and scheduling. Primarily, in traditional job scheduling, the objective of traditional process planning and scheduling is to acquire overall optimal results for all jobs under some manufacturing constraints. However, this is not the case for the network-based manufacturing approach. Here, in networked manufacturing, the jobs are submitted by different customers whose process planning and scheduling integration helps for every individual job which strives for their own optimal result, such as minimisation of makespan, etc. Second, in the traditional manufacturing approach the machines associated to jobs are always limited and they are distributed in a single workshop or an enterprise. However, in network-based manufacturing the machines associated with conducting various job operations are geographically distributive, ranging from intra-shop to inter-shop and even inter-enterprise. Moreover, in the networked manufacturing pattern because of its flexibility it has the ability to generate the optimal process plans for each job conforming to dynamic constraints and it has become a challenging and new problem according to today's status.

Considering these characteristics and requirements of networked manufacturing, it has become urgent to develop a suitable decision approach which is crucial for manufacturing enterprises to support and generate the optimal process plans for jobs in the context of network based manufacturing. However, whether it can be in traditional manufacturing environments or in networked manufacturing ones, process planning and scheduling plays an important role in production by Brandimarte and Calderini (1995). At present, in the area of integrated process planning and scheduling, game theory has captured the interest of a number of researchers. Seredynski *et al.* (2001) and Seredynski (1997, 1998) paid a great deal of attention to using game theory for scheduling problems and analysis of the nature of parallel computing systems, such as typical distributed scheduling problems. Subsequently, the author proposed a multi-agent-based representation with game theory models for parallel computing systems. Here, for the game theory model a non-cooperative N-person game with limited interaction was introduced and a genetic algorithm-based evolutionary algorithm was implemented for finding out the Nash equilibrium point with better performance. Kim *et al.* (2003) proposed a framework for maintenance scheduling of generating units problem in competitive electricity markets. Here, the authors have implemented a game theory-based approach of a dynamic non-co-operative with complete information for analysing the strategic behaviour of generating companies. Jiang *et al.* (2007) presented an e-service-driven networked manufacturing mode and also the tools and techniques, which are mainly useful for extended enterprises, were discussed. Zhou *et al.* (2009, 2010) introduced a game theory approach for job scheduling in a networked manufacturing pattern for generating optimal process plans of each job. The authors proposed an N-person, non-co-operative mathematical model with complete information and the Nash equilibrium point with evolutionary algorithms, such as genetic algorithm and hybrid adaptive genetic algorithm (HAGA), has been implemented with better performance. As efficiency becomes a significant part for network-based manufacturing, particularly job scheduling problems, the recent research has tried to combine the game theory approach with other techniques of nature-inspired evolutionary algorithms such as genetic algorithm and modified genetic algorithm. There is still a lack of development and implementation of efficient evolutionary algorithms for optimal process plans, thus creating a gap in the research on networked manufacturing. In this article, we have developed a hybrid dynamic-DNA (HD-DNA) algorithm by using a game theoretic approach to integrate process planning and scheduling in a networked manufacturing environment. The performance of a HD-DNA algorithm is compared with DNA as well as the HA-GA algorithm (Zhou *et al.* 2010). It was found that the proposed algorithm obtained lower makespan in less computational time. Thus, we were able to develop and implement an efficient evolutionary algorithm for improving the generated process plans and scheduling plans.

The remainder of this paper is organised as follows. In Section 2, we describe a concrete example and its notations. In Section 3, we develop the integration approach and mathematical representation of the proposed model and a game theory-based solution approach is presented. Section 4 explains the DNA and hybrid dynamic-DNA algorithms for solving the proposed game. The experimentation with different cases and their results are introduced in Section 5. In Section 6 the results and their discussions are detailed. The paper concludes with Section 7 which suggests the direction of future work.

2. Problem description

We consider a series of jobs on order submitted by different customers denoted as n. Each job has a set of strategies which corresponds to its alternative process plans and each process plan contains a series of sequential operations. Consequently, the jobs with alternative process plans are processed for different operations on a set of alternative machines. In other words, each operation corresponds to alternative process plans. However, in networked manufacturing the machines are distributed geographically to perform different operations on the jobs; this can be one of the complex tasks of the present problem, although the transportation time between two corresponding machines plays a crucial role for process planning and scheduling tasks.

Moreover, flexibility in networked manufacturing, such as integration of process planning and scheduling, gives us potential to generate the effective optimal process plan. Therefore, generation of optimal process plans for each job corresponding to constraints is a challenging task and is considered a new problem according to today's manufacturing scenario. In this paper the objective that we consider is to minimise the makespan. The objective function is described as follows:

$$\bar{F}_i = \min(T_i) = \min(stt_{i,j}^{OP_{ij}}(mac_1) + prt_{i,j}^{OP_{ij}}(mac_1)) \tag{1}$$

Here, T_i is the makespan of the ith job. The makespan of each individual job is minimised which will result in reduction in makespan of all jobs. Makespan includes starting time and processing time. The starting time for ith job consists of transportation time between one machine to another and idle time for the job at any machine. Some assumptions required to solve Equation (1) are described as follows:

Assumptions:

(1) Job pre-emption is not allowed.
(2) When an operation of a job is being processed on a machine, it cannot be interrupted until finished.
(3) Each machine can handle only one job at a time.
(4) Transportation time is considered. The system is designed in such a way that, after an immediate completion of the operation of a job on a machine, the job is immediately transported to the succeeding machine on its process.
(5) All jobs and machines are simultaneously available at the time zero.

Based on these assumptions and the proposed model, the mathematical model representation and their notations are described in Table 1.

3. The game

3.1 *Proposed integration model*

As reviewed earlier, the issues of job scheduling problems in networked manufacturing have been extensively studied in the literature. In order to respond to the above challenges and issues, it is imperative to design a different approach which can meet the requirements in the present era of manufacturing methods. In recent years, with the advent of better tools, game theory has been profoundly used to tackle the challenges present in advanced

Table 1. The notations used to explain the proposed model.

n	Total number of jobs.
M	Total number of machines.
P_i	Payoff value of ith job.
P_i^v	Payoff value of ith job and vth generation for process plan game.
F_v	Fitness value for vth generation.
T_i	Makespan of ith job from the set of possible process plans.
Prt	Processing time.
Trt	Transportation time.
mac_1	Any machine.
v	Number of generation for process planning game.
w	Number of generation for job scheduling game.
$prt_{i,j}^k(mac_1)$	Processing time of ith job and jth process plan and kth operation on machine 1 (one of the machines on which kth operation of jth process plan of ith job is done).
$trt_{i,j}\ (mac_1, mac_2)$	Transporting time for ith job and jth process plan from machine1 to machine 2.
$OP_{i,j}$	Maximum number of operations of ith job and jth process plan.
F'_w	Fitness value for wth generation for job scheduling game.
ST_i	All the strategies of ith job (i.e. all machining sequences available of all process plans for ith job).
$ST_{i,j}$	All the strategies for ith job and jth process plan.
$stt_{i,j}^{OP_{ij}}(mac_1)$	Starting time for $OP_{i,j}$ operation (i.e. the last operation) of ith job and jth process plan.
$\mathrm{PP}_{i,j}$	Process plan of ith job and jth process plan.
κ	Tolerance value for process plan sub-game.
κ'	Tolerance value for scheduling sub-game.
NG	Number of generations.
CT	Computational time.

manufacturing in a totally new way. This section presents a different kind of game theoretic-based integration approach, which has special features to achieve the best objective and to generate the optimal process plans for multiple jobs. Moreover, the proposed integrated approach is used to generate effective and robust schedules. The game theoretical mathematical model and a framework for an *N*-person non-cooperative game with complete information is constructed and expatiated. This game is broadly divided into two kinds of sub-games named as games-to-address sub-game (GASG) and games-to-solve sub-game (GSSG). However, these two sub-games are capable enough for planning and scheduling of jobs which enables the interaction and co-operation with each other for optimal process plans of each job. The logical framework and the relationship between the two sub-games are illustrated in Figure 1.

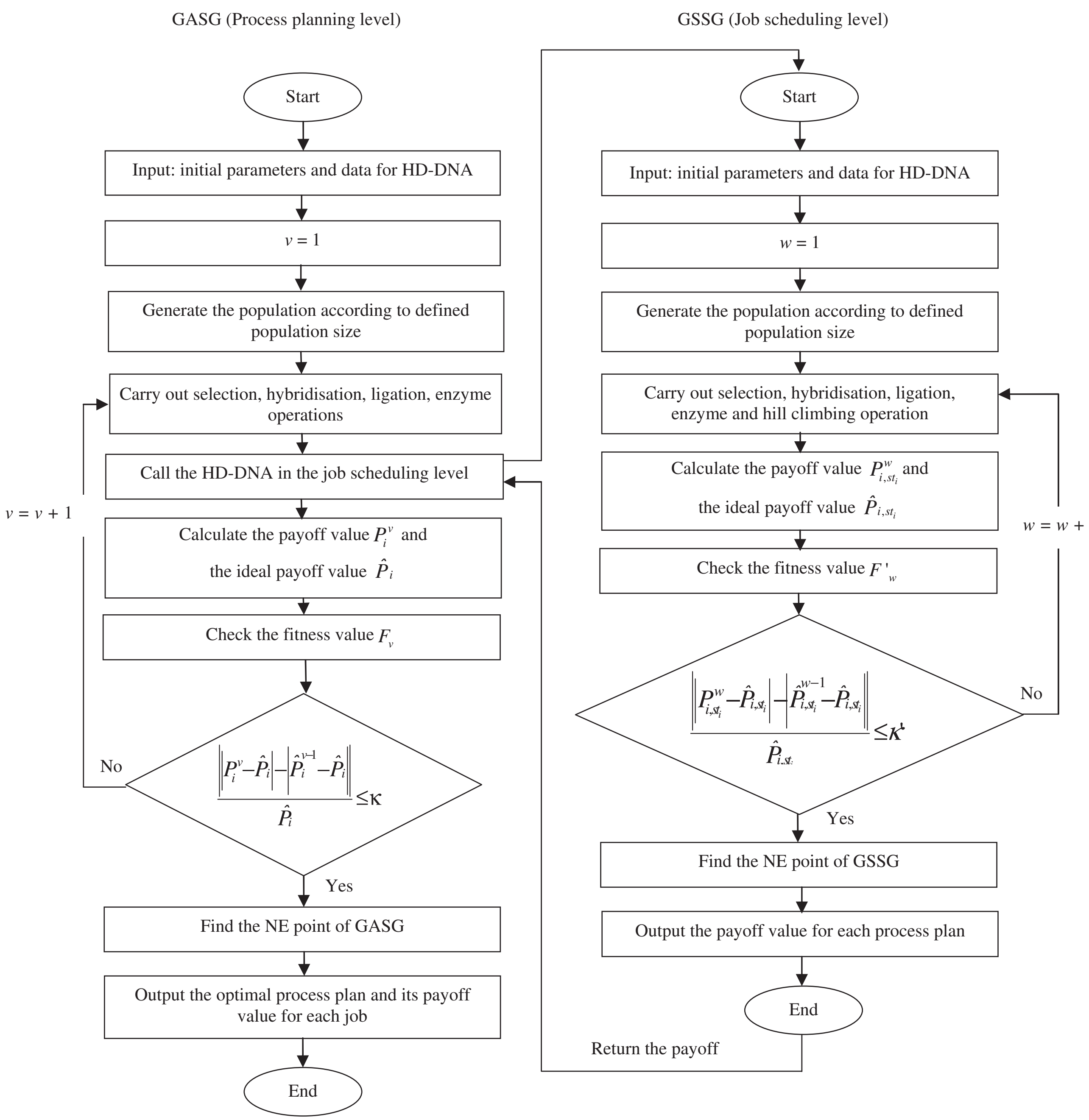

Figure 1. Flowchart for process planning and job scheduling of the game.

3.2 *Games to address sub-game (GASG)*

GASG encapsulates the information of the process plans generation and their decisions, represented here as players submitted by different customers. During the process plan generation, the players play worthwhile strategies through which the alternative process plans of each job can be achieved and the payoff of each job is defined as its makespan. The mathematical model representation, notations and definitions are expressed in detail as follows. We designate this game as games to address sub-game, which is expressed as a tupple: $SG = (N, ST_i, P_i)$. Here, N represents the set of jobs, ST_i represents the set of strategies for ith job to get the corresponding alternative process plans and P_i represents a payoff function for job i depend upon the strategies for jobs that cannot be known in advance. Moreover, we define the payoff of job i as a function of strategies set (st). The mathematical model representation of the above-mentioned notations is depicted in following equations.

$$SG = (N, ST_i, P_i) \tag{2}$$

where

$$ST_i = \{PP_{i,j} | 1 \leq j \leq U_i\}$$

where $U_i =$ max number of process plan of ith job.

$$P_i(st) = F(PP_{i,j}, P\widehat{P}_{x,y}) \tag{3}$$

where

$$P\widehat{P}_{x,y} = \left(PP_{1,y_1}, \ldots, PP_{i,y_{i-1}}, PP_{i+1,y_{i+1}}, \ldots, PP_{n,y_n}\right).$$

The function F in Equation (3) guarantees the optimal process plan, it implies that the job tries its best to generate the minimal makespan. We considered the Nash equilibrium (NE) as the solution for the N-person non-cooperative games and the function value is derived from the GSSG.

3.3 *Games to solve sub-game (GSSG)*

Job scheduling in networked manufacturing can be implemented in GSSG. In this sub-section the process plans provided by GASG to GSSG acts as players and the payoff of each process plan is defined as its makespan. The strategies played for players to its alternative machines, which are distributed geographically, and payoff of each process plan further leads to an optimal process plan for each job. In this context, the transportation time between the machines plays a crucial role for process planning and scheduling. The optimal result of each job can be derived from the NE point of the GSSG. We designate this game as GSSG, which is expressed as a tupple: $SG' = (N', ST_{i,j}, P_{i,j})$. Here, N' represents a set of players corresponding to process plans provided by the games to address sub-game. $ST_{i,j}$ represents strategies set of alternative machine sequences for sequential operations. $P_{i,j}$ denotes payoff function, which can be useful for choosing feasible machine sequence. Moreover, the mathematical representation of the above-mentioned notations is as follows:

$$SG' = (N', ST_{i,j}, P_{i,j}) \tag{4}$$

$$P_{i,j}(st') = stt_{i,j}^{OP_{ij}}(mac_1) + prt_{i,j}^{OP_{ij}}(mac_1) \tag{5}$$

where

$$st' = \left(st'_1, \ldots, st'_i, \ldots, st'_n\right), st'_i \in ST_{i,j}$$

and subject to

$$stt_{i,j}^{k+1}(mac_2) \geq stt_{i,j}^{k}(mac_1) + prt_{i,j}^{k}(mac_1) + trt_{i,j}(mac_1, mac_2) \tag{6}$$

and

$$stt_{i,j}^{k}(mac_1) + prt_{i,j}^{k}(mac_1) \geq stt_{a,b}^{k'}(mac_1)$$

or

$$stt_{a,b}^{k'}(mac_1) + prt_{a,b}^{k'}(mac_1) \geq stt_{i,j}^{k}(mac_1) \tag{7}$$

where, $b = 1, \ldots, n$; $b = 1, \ldots, U_x$; $k' = 1, \ldots, OP_{i,j}$; $i \neq x$; $mac_1 \in ST_{i,j} \cap ST_{a,b}$.

Here, Equation (5) represents the payoff function for job scheduling in networked manufacturing. It is the summation of starting time and processing time for different machining operations followed by the constraints. The formulae (6) operations constraint ensures that each machine can only process one operation at a time; the successive operation on that particular machine has been carried out only after the completion of the previous operation. Consequently, the machine constraint portrayed in formulae (7) expresses that each machine can only process one operation at a time. Moreover, we obtained Equation (8) by transforming Equation (3) and finally from Equations (5) and (8) we draw a conclusion that the objective for generating optimal process plans of multiple jobs is obtained through minimising the makespan of each job. Since all the jobs try to minimise their own makespan simultaneously, by maximising the opponent jobs as it tries to balance the profits for obtaining the optimal results among all the jobs, is the bottleneck.

$$P_i(st) = F(PP_{i,j}, \widehat{PP}_{x,y}) = \min_{st=(st_1,\ldots,st_n);\, st_i \in ST_i;\, st'=(st'_1,\ldots,st'_n);\, st'_i \in ST_{i,j}} (P_{i,j}(st')) \tag{8}$$

Nash equilibrium has been broadly considered and it acts as a solution for N-person non- co-operative games with a complete solution. Zhou *et al.* (2009) proposed that the NE point has N-tupple of strategies for each player if any player who deviates from this point cannot regain the expected payoff. We adopt NE to analyse the strategic behaviour of players to obtain the final objective.

$$P_i\left(st_i^{\text{Nash}}, st_{-i}^{\text{Nash}}\right) \leq P_i\left(st_i^{\text{Nash}}, st_{-i}^{\text{Nash}}\right) \tag{9}$$

$$st_{-i}^{\text{Nash}} = \left(st_1^{\text{Nash}}, \ldots, st_{i-1}^{\text{Nash}}, st_{i+1}^{\text{Nash}}, \ldots, st_n^{\text{Nash}}\right)$$

$$P_{i,j}\left(st_i'^{\text{Nash}}, st_{-i}'^{\text{Nash}}\right) \leq P_{i,j}\left(st_i', st_{-i}'^{\text{Nash}}\right) \tag{10}$$

$$st_{-i}'^{\text{Nash}} = \left(st_1'^{\text{Nash}}, \ldots, st_{i-1}'^{\text{Nash}}, st_{i+1}'^{\text{Nash}}, \ldots, st_n'^{\text{Nash}}\right)$$

Equations (9) and (10) presented above represent the solution profiles of games to address sub-game for process planning of the players and games to solve sub-game for job scheduling scheme respectively. NE point is characterised by a unique set of strategies for each job. Change in any one strategy of any one job will result in deviation from the NE point which is shown in Equations (9) and (10). In this paper, we developed DNA and hybrid-dynamic DNA (HD-DNA) based algorithms for searching of NE points efficiently and more effectively and also we compared the most effective one. The detailed description of the algorithms and their implementation with the model is described in the next section

4. HD-DNA (hybrid dynamic DNA) based algorithm

The concept of Nash point has been implemented to obtain the optimal solution in a networked manufacturing environment. There are two layers of sub-games, namely process plan sub-game and job scheduling sub-game for which NE point should be determined. After finding the NE point of process plan sub-game, the optimal process plans for the jobs can be realised. We have used DNA-based two level nested heuristic algorithm to obtain the NE point. To improve the convergence capabilities of DNA, we have developed HD-DNA which incorporates tournament selection, hill climbing search, dynamic hybridisation, ligation and enzyme operators.

In order to determine the NE point of GASG, we need to determine the NE points of each individual process plan which is present in the population of the sub game. Job scheduling sub-games in GSSG of each process plan are used to obtain the NE point for the individual process plan and its job scheduling sub-games. The implementation of HD-DNA at process plan layer and job scheduling layer is done in parallel. Although the parallel computing helps in reducing the computation time of the solution, it is not the focus of this research work. The HD-DNA at both layers co-operate with each other to obtain the optimal solution. The flowchart of the proposed HD-DNA algorithm is shown in Figure 2.

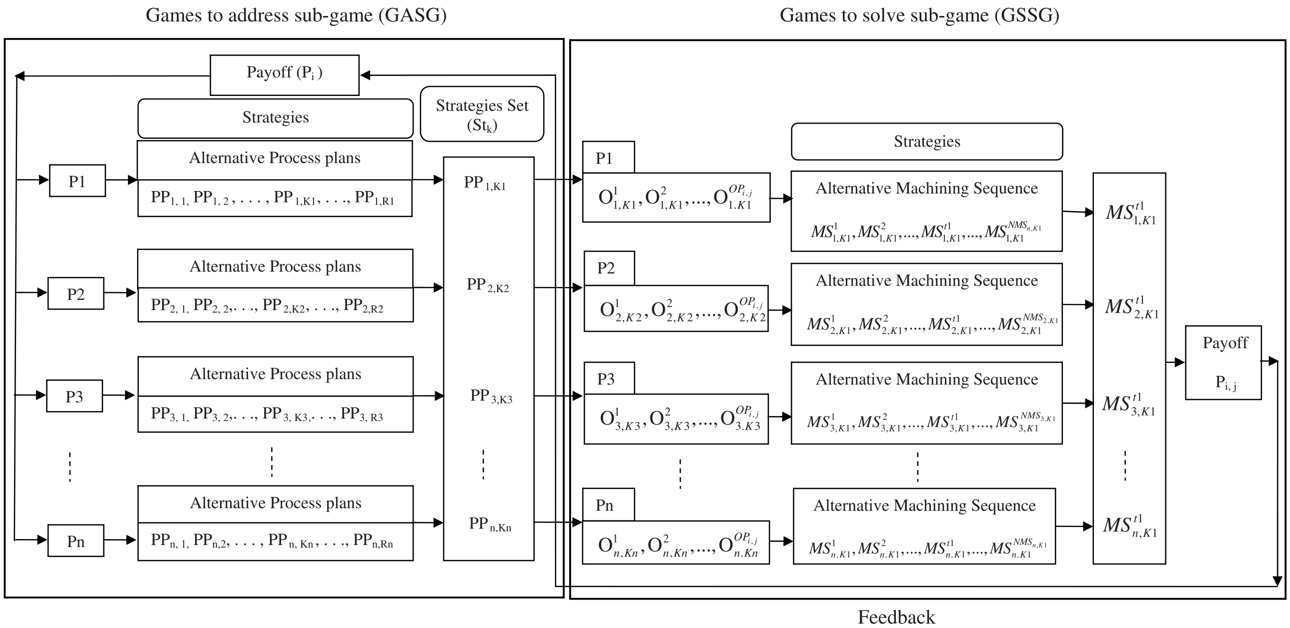

Figure 2. Logic for solving the game.

4.1 *Basics of DNA algorithm*

The DNA molecule plays a vital role in DNA-based computing. Generally, the DNA molecule exists in a single strand structure, i.e. a thread like shape and are known as ligonucleotides. The oligonucleotide represents a possible solution to the optimisation problem. The DNA molecule comprises of a collection of four nucleic acid bases such as: Adenine (A), Guanine (G), Thymine (T) and Cytosine (C). These protein synthesis molecules have the tendency to form chains called oligonucleotides by the formation of hydrogen bonds between the two complementary DNA strands. The Guanine (G) combines with the Cytosine (C) of the complementary DNA strand whereas Adenine (A) from one DNA strand combines with the Thymine (T) of the complementary DNA strand. The major chemical reactions that drastically affect the existing population of oligonucleotides are hybridisation, ligation, and action of restriction enzymes.

4.2 *The encoding of oligonucleotide*

Encoding the problem under consideration as a problem of DNA computing is an important step and requires proper assessment so that unproductive hybridisations, such as mismatched hybridisation, hairpins or shifted hybridisations are avoided. The present paper considers each oligonucleotide as individual in order to reduce the time complexity. The complexity of the problem demands two different types of schemes for process plan layer and job scheduling layer. The structure of oligonucleotides$_1$ for process plan sub-game has been explained in Figure 3(a). This oligonucleotide consists of a string of numbers where the ith number represents the feasible process plan for the ith job. Each job can have one and only one process plan in one of oligonucleotide$_2$ for job scheduling sub-game depends on the oligonucleotide$_1$ which has process plans of each job. Machine id and the time required to complete the operation on that machine are available in oligonucleotide$_2$. The oligonucleotide$_2$ consists of a long string where machine sequence and processing time corresponding to each process plan of oligonucleotide$_1$ are concatenated end to end. The structure of oligonucleotide$_2$ is shown in Figure 3(b).

4.3 *Payoff and fitness function*

The fitness value of oligonucleotides decides the quality of the solution. High fitness value implies a better solution. In HD-DNA, fitness value helps to decide whether an oligonucleotide will survive the generation or not. Hence, formulation of an appropriate fitness function is critical for obtaining a global optima and early convergence.

(a)

3	1	2	1	2	3
J1	J2	J3	J4	J5	J6

(b)

J1				J2						J3		J4		J5		J6	
Mac	Time	Mac	Time	Mac	Time	Mac	Time	Mac	Time	Mac	Time	Mac	Time	Mac	Time	Mac	Time

Figure 3. (a) Oligonucleotide$_1$ for process plan sub-game. (b) Oligonucleotide$_2$ for scheduling of process plans.

The fitness function has been designed to obtain the NE point at the process planning layer and job scheduling layer. The NE point at the process plan layer is a strategic combination of process plans of all jobs for which the makespan is minimum. Similarly, the NE point at the job scheduling layer is a strategic combination of machine sequences of all jobs for which the makespan is least. The objective is not to optimise the makespan of any one job but to optimise the makespan of all jobs simultaneously. We will have to optimise all the jobs simultaneously to obtain minimum makespan. A standard fitness function cannot serve this purpose so we have designed a fitness function which will cater to our needs. The fitness function is shown in Equation (11).

$$F_v = \left\{ \frac{1}{\sqrt{\sum_{i=1}^{n} (P_i^v - \widehat{P}_i)^2}} \right\} \tag{11}$$

Here, F_v is the fitness value at vth generation. P_i^v is the payoff value of J_i at vth generation. $\widehat{P}_i$ is the ideal payoff for J_i which is obtained when there is no idle time. Every ideal payoff value is calculated by considering zero waiting time. $\widehat{P}_i$ is defined in Equation (12).

$$\widehat{P}_i = \min_{1 \le j \le T_i} \left(\min \left(\sum_{k=1}^{OP_{ij}} prt_{i,j}^k(mac_1) + \sum trt_{i,j}(mac_1, mac_2) \right) \right) \tag{12}$$

The ideal payoff for J_i implies that the payoff is calculated by assuming that J_i is the only job present in the manufacturing environment. Due to this assumption, there is no component of waiting time in the ideal payoff.

A stopping criterion was set to stop the iterations at an appropriate point. This criterion ensures the generation of NE point for process plan layer. Stopping criterion is shown in Equation (13)

$$\frac{\left| \left| P_i^v - \widehat{P}_i \right| - \left| \widehat{P}_i^{v-1} - \widehat{P}_i \right| \right|}{\widehat{P}_i} \le \kappa \tag{13}$$

Here, κ is the tolerance value which is critical in finding the NE point. $\widehat{P}_i^{v-1}$ is the minimum payoff of J_i at $(v-1)$th generation. If P_i^v satisfies the criteria explained in Equation (13) it will become $\widehat{P}_i^v$. Every job will have to satisfy this condition for the termination of the algorithm.

$$\widehat{P}_i^{v-1} = \arg\min \sum_{i=1}^{n} P_i^{v-1} \tag{14}$$

$\widehat{P}_1^{v-1}, \widehat{P}_2^{v-1}, \ldots, \widehat{P}_n^{v-1}$ is obtained from one oligo which has least payoff among process plan level population during $(v-1)$th generation.

To maintain equal performance level in both the sub-games, similar fitness function and stopping criterion was chosen for job scheduling sub-game. The fitness function is shown in Equation (15)

$$F'_w = \left\{ \frac{1}{\sqrt{\sum_{i=1}^{n} (P_{i,st_i}^w - \widehat{P}_{i,st_i})^2}} \right\} \tag{15}$$

where, P^{w}_{i,st_i} is the payoff of J_i for ith strategy during wth generation. $\widehat{P}_{i,st_i}$ is the ideal payoff of J_i for ith strategy which is characterised by the sequence of the process plan. $\widehat{P}_{i,st_i}$ is defined in Equation (16)

$$\widehat{p}_{i,st_i} = \min\left(\sum_{k=1}^{OP_{ij}} prt^{k}_{i,j}(mac_1) + \sum trt_{i,j}(mac_1, mac_2)\right) \tag{16}$$

The ideal payoff $\widehat{P}_{i,st_i}$ is the payoff obtained by implementing strategy st_i under the assumption that J_i is the only job in the manufacturing environment. The stopping criterion is shown in Equation (17).

$$\frac{\left|\left|P^{w}_{i,st_i} - \widehat{P}_{i,st_i}\right| - \left|\widehat{P}^{w-1}_{i,st_i} - \widehat{P}_{i,st_i}\right|\right|}{\widehat{P}_{i,st_i}} \leq \kappa' \tag{17}$$

Similar to κ, κ' is the tolerance value which is critical in finding NE point for strategy st_i. $\widehat{P}^{w-1}_{i,st_i}$ Stands for ideal payoff of J_i during $(w-1)$th generation by utilising strategy st_i.

$$\widehat{P}^{w-1}_{i,st_i} = \arg\min \sum_{i=1}^{n} P^{w-1}_{i,st_i} \tag{18}$$

$\widehat{P}^{w-1}_{1,st_i}, \widehat{P}^{w-1}_{2,st_i}, \ldots, \widehat{P}^{w-1}_{n,st_i}$ belong to one oligo which has minimum payoff among a population of job scheduling level during $(w-1)$th generation while implementing strategy st_i of process planning level.

4.4 *HD-DNA operators*

We have used various DNA operators to increase the diversity of population and retaining better oligonucleotides. The HD-DNA algorithm includes tournament selection, hybridisation, ligation and enzyme operations. The framework of these operators is as follows.

4.4.1 *Selection*

Tournament selection is used to decide which oligonucleotide will undergo hybridisation. In this selection, several tournaments are played among a few randomly chosen oligonucleotides (oligo) of the population. The number of tournaments to be played is equal to the population size (PS), where the winner of each tournament is selected for hybridisation. However, the oligonucleotide with a higher fitness value is the winner of the tournament. The number of oligonucleotides randomly selected from the population, i.e. tournament size is 3 in HD-DNA algorithm. This tournament size will not only help us in retaining the stronger oligonucleotides but also maintains the diversity of the population by including some of the weaker oligonucleotides.

4.4.2 *Hybridisation*

Hybridisation is a chemical process that unified two complementary single strands into a double strand via hydrogen bonds. Thus, it conduces to fast maturation which depicts the exploration of search space. The selected oligonucleotides from tournament selection are added to the mating pool in which two randomly selected oligonucleotides from the mating pool are hybridised to form double stranded oligonucleotide. The hybridisation process is carried out until we have 'PS' double stranded oligonucleotides. The maximum of two fitness values of constituent oligos of double stranded oligo is assigned as the fitness value to double stranded oligo. No two double stranded oligo will have the same constituent oligos.

4.4.3 *Ligation*

Diversification of the population is carried out by ligation process; where diversity increases the probability of finding global optima instead of local optima. During diversification of the population, previously unexplored parts of search space are scavenged for possible solutions. Diversification may result in destruction of strong oligo and preservation of weak oligo. To address this issue, we have incorporated the probability of ligation p_l which will enhance the utility of ligation. Higher value of p_l will increase the probability of ligation. On the other hand, lower

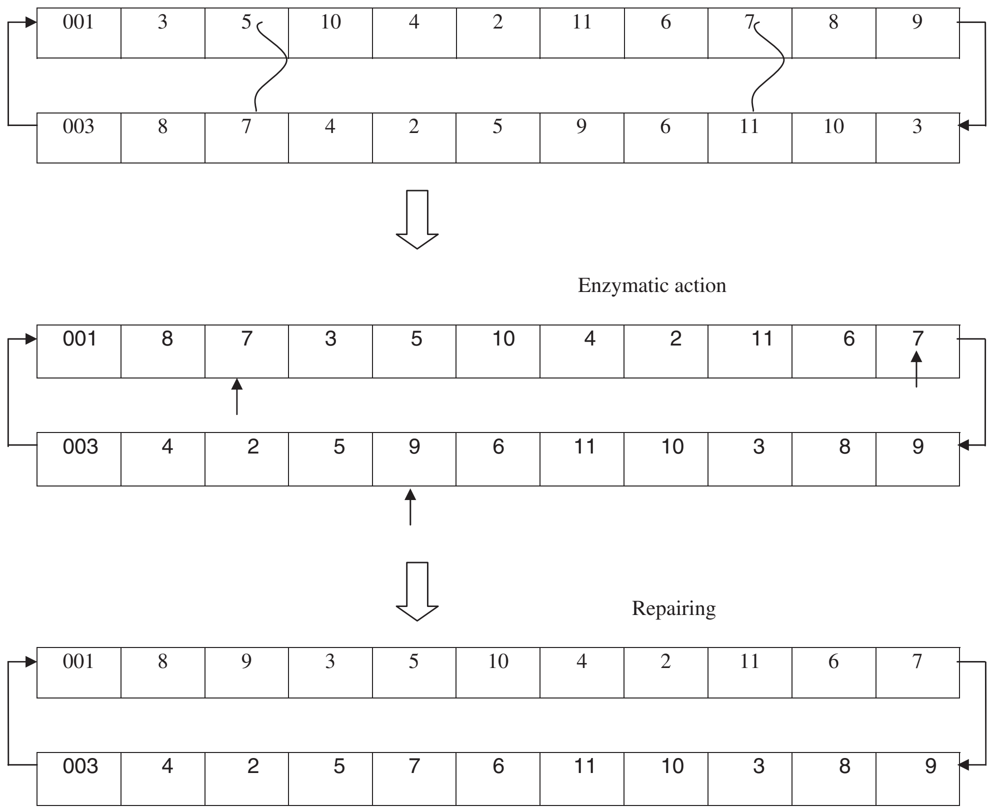

Figure 4. Action of restriction enzymes and repairing.

value of p_l will reduce the probability of ligation. The formulation of p_l is explained in Equation (19)

$$P_{lig} = \begin{cases} \dfrac{(p_{l1}+p_{l0})}{2} + \dfrac{(p_{l1}-p_{l0})}{2}\cos\left(\dfrac{F-F_{\text{avg}}}{F_{\max}-F_{\text{avg}}}\pi\right) & F \geq F_{\text{avg}} \\ \dfrac{(p_{l1}+p_{l2})}{2} + \dfrac{(p_{l1}-p_{l2})}{2}\cos\left(\dfrac{F_{\text{avg}}-F}{F_{\text{avg}}-F_{\min}}\pi\right) & F < F_{\text{avg}} \end{cases} \tag{19}$$

Here, p_{l0}, p_{l1}, p_{l2} are constants defined in HD-DNA and $p_{l0} > p_{l1} > p_{l2}$. $F_{\max}$ is the highest fitness value in the population. $F_{\min}$ is the smallest fitness value in the population. F_{avg} is the average fitness value of the population. F is the fitness value of the oligo. This formulation ensures that the probability of ligation of oligo with the higher fitness value will be lower compared with the probability of ligation of oligo with the lower fitness value. On the basis of p_l, we select some oligos and add them into the ligation pool. The unselected oligos are not ligated and preserved in the population. From the ligation pool, two randomly selected oligos are ligated. Number of ligation operation (NL) is given by Equation (20)

$$NL = PS - \text{number of unselected oligo} \tag{20}$$

In ligation, two double stranded oligo are ligated to form one single double stranded oligo. To form a double stranded oligo, some nucleobases (like gene in GA) are taken from one double stranded oligo and others are taken from second double stranded oligo. We selected a nucleobase of stronger double stranded oligo with a probability of 0.65 and nucleobase of weaker double stranded oligo with a probability of 0.35. The ligated individuals are added to the population to maintain the size of the population. The similarity with stronger oligo and appropriate diversity has resulted in better convergence rate.

4.4.4 *Restriction enzymes*

After ligation, restriction enzymes are applied on the oligos. Enzymes such as EcoRI are critical in introducing variation in the oligos. The prime objective of these enzymes is to break down the double-stranded oligos into single-stranded oligos. The effect of application of enzymes is shown in Figure 4.

In order to protect the stronger oligos from diversification and to amend the weaker oligos, we have incorporated the probability of enzymatic action p_{enz} into the framework. High p_{enz} value will increase the probability of enzymatic action on an oligo and low p_{enz} value will decrease the probability of enzymatic action on an oligo. The formulation of p_{enz} is explained in Equation (21)

$$P_{\text{enz}} = \begin{Bmatrix} \dfrac{(p_{e1}+p_{e0})}{2} + \dfrac{(p_{e1}-p_{e0})}{2}\cos\left(\dfrac{F-F_{\text{avg}}}{F_{\max}-F_{\text{avg}}}\pi\right) & F \geq F_{\text{avg}} \\ \dfrac{(p_{e1}+p_{e2})}{2} + \dfrac{(p_{e1}-p_{e2})}{2}\cos\left(\dfrac{F_{\text{avg}}-F}{F_{\text{avg}}-F_{\min}}\pi\right) & F < F_{\text{avg}} \end{Bmatrix} \tag{21}$$

Here, p_{e0}, p_{e1}, p_{e2} are constants defined in HD-DNA and $p_{e0} > p_{e1} > p_{e2}$. $F_{\max}$ is the highest fitness value in the population. $F_{\min}$ is the smallest fitness value in the population. F_{avg} is the average fitness value of the population. F is the fitness value of an oligo. This formulation ensures that the probability of enzymatic action on an oligo with higher fitness value will be lower compared with probability of enzymatic action on oligo with lower fitness value. On the basis of p_{enz}, we select some oligos on which enzymatic action is performed. From the unselected double stranded oligos, single stranded oligo is extracted. During the extraction of single stranded oligo, we selected nuclobase of stronger single stranded oligo with a probability of 0.65 and nucleobase of weaker single stranded oligo with a probability of 0.35. The output of enzymatic action is included with that obtained from unselected oligos.

4.4.5 *Hill climbing search*

Hill climbing search is a prominent method to improve the convergence rate and quality of solution of evolutionary algorithm. In hill climbing search the following steps are taken.

Step 1: Oligo with highest fitness ($F_{\max}$) is selected from current population.

Step 2: The encoded form of the selected oligo is generated (ex. ATGTATTTGTCCAT). The oligo comprises of multiple nucleobases like A, T, G, C.

Step 3: Let the length of oligo be L and i is initialised to 1.

Step 4: The ith nucleobase is selected and it is mutated into other possible nucleobase. Fitness value of mutated oligo is calculated. Let it be F_i.

Step 5: If $F_i > F_{\max}$, the mutated oligo survives and original oligo is removed. Else the mutated oligo is removed and original oligo is retained. ($i = i + 1$).

Step 6: If $i \leq$ L, go back to step 4. Else hill climbing search is over.

5. Experimental study

HD-DNA algorithm has been implemented in three case studies which express different manufacturing scenarios and cases input data proposed by Zhou *et al*. (2010) and Shao *et al*. (2010) is used here. To establish the superiority of HD-DNA algorithm, we have compared it with DNA algorithm and HA-GA algorithm. The DNA algorithm does not utilise the dynamic ligation, enzyme action operators and hill climbing search which are critical in HD-DNA. In DNA algorithm, the individuals with fitness lower than the mean fitness are modified by the ligation and enzymatic action. The HA-GA algorithm is characterised by roulette wheels (selection), genetic operators like adaptive two point crossover scheme and adaptive mutation scheme. HA-GA algorithm is described in detail in Zhou *et al*. (2010). The game theoretic approach has been used in three algorithms to solve the game more effectively and efficiently. In the three algorithms, the population size at process plan level and job scheduling level is equal. The input parameters of HD-DNA are shown in Table 2.

5.1 *Case study 1 (6 × 6 problem)*

In this case study different clients have placed orders for six different products and the company has six machines which are geographically located. In this manufacturing environment, every job has multiple process plans and for each operation multiple machines are available. The details of the case study are shown in Tables 3 and 4.

Table 2. The input parameters for HD-DNA.

Parameter	Parameter value			
	GASG		GSSG	
Population size	30		50	
Probability of ligation (P_{lig})	p_{l0}	0.7	p_{l0}	0.650
	p_{l1}	0.5	p_{l1}	0.500
	p_{l2}	0.3	p_{l2}	0.250
Probability of enzymatic action (P_{enz})	p_{e0}	0.028	p_{e0}	0.024
	p_{e1}	0.020	p_{e1}	0.015
	p_{e2}	0.009	p_{e2}	0.005
Terminating value	κ	0.01	κ'	0.010
Maximum generation	300		500	

Table 3. Input data for the 6×6 problem.

Job	PP	Operations					
		O_1	O_2	O_3	O_4	O_5	O_6
J_1	$PP_{1,1}$	{1, 2, 7, 8} [6, 5, 5, 4]	{3, 4, 5} [7, 6, 6]	{6} [8]			
	$PP_{1,2}$	{1, 3} [4, 5]	{2, 4} [4, 5]	{3, 5} [5, 6]	{4, 5, 6, 7, 8} [5, 5, 4, 5, 9]		
J_2	$PP_{2,1}$	{2, 7} [4, 8]	{1, 3, 8} [2, 3, 8]	{2, 4, 6} [4, 3, 5]	{3, 5} [2, 4]	{2, 4} [3, 4]	{4, 6} [3, 5]
	$PP_{2,2}$	{1, 3, 5} [1, 5, 7]	{4, 8} [5, 4]	{4, 6} [1, 6]	{4, 7, 8} [4, 4, 7]	{4, 6} [1, 2]	{1, 6, 8} [5, 6, 4]
J_3	$PP_{3,1}$	{2, 3} [5, 6]	{1, 4, 7, 8} [6, 5, 4, 7]	{2, 5, 7} [5, 6, 5]	{3, 6} [6, 5]	{1, 6} [6, 6]	{5} [4]
	$PP_{3,2}$	{1, 8} [7, 8]	{3, 4, 7} [8, 8, 11]	{5, 8} [9, 8]			
	$PP_{3,3}$	{2, 3, 8} [7, 6, 8]	{4, 8} [7, 7]	{3, 5, 7} [7, 8, 7]	{4, 6} [7, 8]	{1, 2} [1, 4]	
J_4	$PP_{4,1}$	{1, 2, 7, 8} [7, 8, 7, 8]	{3, 4} [7, 6]	{6} [9]	{1, 7} [5, 7]		
	$PP_{4,2}$	{1, 3, 5} [4, 3, 7]	{2, 8} [4, 4]	{3, 4, 6, 7} [4, 5, 6, 7]	{5, 6, 8} [3, 5, 5]		
J_5	$PP_{5,1}$	{1} [3]	{2, 4} [4, 4]	{3, 8} [4, 4]	{5, 6, 8} [3, 3, 3]		
	$PP_{5,2}$	{2, 4} [5, 6]	{5} [7]	{3, 6} [9, 8]			
J_6	$PP_{6,1}$	{1, 2} [3, 4]	{3, 4} [4, 3]	{2, 5} [5, 3]	{3} [4]	{4, 5} [5, 6]	{3, 6} [5, 4]
	$PP_{6,2}$	{1, 3} [4, 4]	{2, 3} [5, 6]	{2, 4} [6, 7]	{6} [7]		
	$PP_{6,3}$	{1, 2, 3} [3, 5, 8]	{4, 5} [7, 10]	{3, 6} [9, 9]			

There are two types of bracket in each location of Table 3. The data in curly brackets represents ids of the machine which can perform the specific operation. The box bracket contains the time required to finish the specific operation in the listed machines. For example, the location J_2, $PP_{2,1}$, O_3 contains {2, 4, 6} and [4, 3, 5]. This implies that the O_3 operation in process plan $PP_{2,1}$ of job J_2 can be done on machine nos 2,4 and 6. The time required to

Table 4. The transportation time between machines.

Machine	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_1	0	1	3	2	3	2	3	4
M_2	1	0	2	2	3	3	5	4
M_3	3	2	0	2	3	3	5	4
M_4	2	2	2	0	2	3	2	4
M_5	3	3	3	2	0	2	2	3
M_6	2	3	3	3	2	0	1	2
M_7	3	5	5	2	2	1	0	1
M_8	4	4	4	4	3	2	1	0

Table 5. The payoff values, process plans and job schedules for different algorithms (6×6 problem).

	HD-DNA (NE point at 40th generation)			DNA (NE point at 49th generation)			HA-GA (Zhou *et al.* 2010) (NE point at 52nd generation)		
Job id	Payoff value	Process plan	Job scheduling	Payoff value	Process plan	Job scheduling	Payoff value	Process plan	Job scheduling
1	28	2	[5, 5, 6, 5]	25	2	[3, 4, 5, 5]	24	1	[2, 5, 6]
2	35	1	[4, 2, 5, 2, 4]	31	1	[2, 1, 6, 3, 2, 4]	30	1	[2,1, 4, 3, 4, 4]
3	32	2	[9, 8, 9]	31	1	[2, 1, 2, 3]	33	1	[3, 1, 2, 3]
4	32	2	[3, 4, 4, 5]	31	2	[3, 2, 3, 5]	28	2	[3, 2, 4, 5]
5	26	2	[6, 7, 8]	36	1	[1, 2, 3, 5, 4]	31	1	[1, 4, 3, 5, 6]
6	35	3	[5, 7, 9]	33	2	[1, 3, 4, 6]	41	1	[1, 4, 2, 3, 4, 3]

complete the operation is 4 min, 3 min and 5 min respectively. Table 4 contains the transportation time required to move the product from one machine to another.

Our main objective of minimising the makespan has been realised in this case study. The payoff value, strategy and job scheduling obtained from the implementation of three algorithms have been shown in Table 5. The HA-GA results for this experiment are from Seredynski (1997). The maximum payoff value is equal to makespan. The makespan of all the three algorithm are shown in Table 11. From Table 5, it can be observed that the HD-DNA outperforms the DNA and HA-GA algorithms. The HD-DNA algorithm not only has obtained lower makespan compared with others but also has achieved this lower makespan with lower generations. The Gantt chart, illustrating the allocations of operations on machines, process plan, start and finish time of each operation of the HD-DNA output, has been shown in Figure 5. Where [a, b] in the Gantt chart represents *a*th job and *b*th operation. We followed this assumption for all Gantt charts in this paper.

5.2 *Case study 2 (8 × 8 problem)*

This case study explores the networked manufacturing environment for eight different products from clients and eight available machines. The input data for this case study has been shown in Table 6 and Table 4. The output results of the three algorithms have been shown in Table 7. The Gantt chart describing the job scheduling of all the products obtained through implementation of HD-DNA algorithm has been shown in Figure 6. The HD-DNA algorithm provides lower makespan and lower generations compared to DNA and HA-GA algorithms.

5.3 *Case study 3 (6 × 8 problem)*

This case study comprises six different products and eight available machines. The detailed data of the case study is available in Tables 8 and 9. The performance and output of the three algorithms have been shown in Table 10.

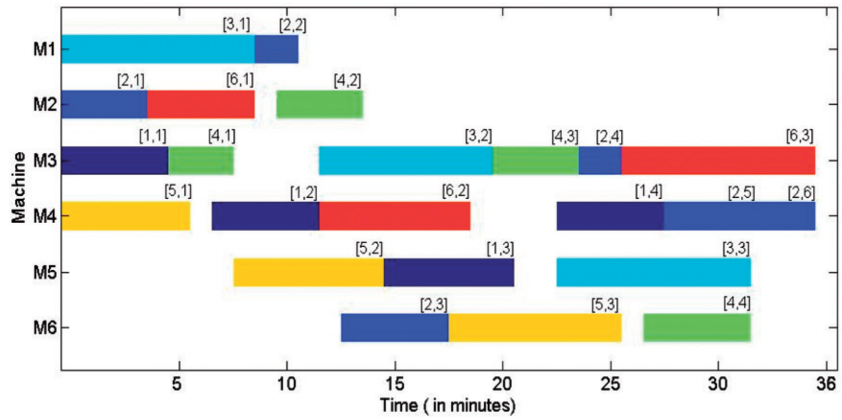

Figure 5. Gantt chart of optimal process plan decision for 6 × 6 problem.

Table 6. Input data for the 8 × 8 problem.

		Operations							
Job	PP	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
J_1	$PP_{1,1}$	{1, 2, 7, 8} [6, 5, 5, 4]	{3, 4, 5} [7, 6, 6]	{6} [8]					
	$PP_{1,2}$	{1, 3} [4, 5]	{2, 4} [4, 5]	{3, 5} [5, 6]	{4, 5, 6, 7, 8} [5, 5, 4, 5, 9]				
J_2	$PP_{2,1}$	{2, 7} [4, 8]	{1, 3, 8} [2, 3, 8]	{2, 4, 6} [4, 3, 5]	{3, 5} [2, 4]	{2, 4} [3, 4]	{4, 6} [3, 5]	{1, 3, 5} [5, 5, 8]	{4} [2]
	$PP_{2,2}$	{1, 3, 5} [1, 5, 7]	{4, 8} [5, 4]	{4, 6} [1, 6]	{4, 7, 8} [4, 4, 7]	{4, 6} [1, 2]	{1, 6, 8} [5, 6, 4]	{4}[4]	
J_3	$PP_{3,1}$	{2, 3} [5, 6]	{1, 4, 7, 8} [6, 5, 4, 7]	{2, 5, 7} [5, 6, 5]	{3, 6} [6, 5]	{1, 6} [6, 6]	{5} [4]		
	$PP_{3,2}$	{1, 8} [7, 8]	{3, 4, 7} [8, 8, 11]	{5, 8} [9, 8]					
	$PP_{3,3}$	{2, 3, 8} [7, 6, 8]	{4, 8} [7, 7]	{3, 5, 7} [7, 8, 7]	{4, 6} [7, 8]	{1, 2} [1, 4]			
J_4	$PP_{4,1}$	{1, 2, 7, 8} [7, 8, 7, 8]	{3, 4} [7, 6]	{6} [9]	{1, 7} [5, 7]				
	$PP_{4,2}$	{1, 3, 5} [4, 3, 7]	{2, 8} [4, 4]	{3, 4, 6, 7} [4, 5, 6, 7]	{5, 6, 8} [3, 5, 5]				
J_5	$PP_{5,1}$	{1} [3]	{2, 4} [4, 4]	{3, 8} [4, 4]	{5, 6, 8} [3, 3, 3]				
	$PP_{5,2}$	{2, 4, 7, 8} [5, 6, 5, 5]	{5} [7]	{3, 6} [9, 8]					
J_6	$PP_{6,1}$	{1, 2} [3, 4]	{3, 4} [4, 3]	{2, 5, 7, 8} [5, 3, 5, 4]	{3} [4]	{4, 5} [4, 6]	{3, 6} [5, 4]		
	$PP_{6,2}$	{1, 3} [4, 4]	{2, 3} [5, 6]	{2, 4, 7, 8} [6, 7, 5, 6]	{6} [7]				
	$PP_{6,3}$	{1, 2, 3} [3, 5, 6]	{4, 5} [7, 8]	{3, 6} [9, 8]					
J_7	$PP_{7,1}$	{1, 2, 3} [3, 4, 5]	{4, 5, 8} [4, 5, 5]	{6} [4]	{5} [5]	{3, 6} [4, 6]			
	$PP_{7,2}$	{3, 5, 6} [4, 5, 4]	{4, 7, 8} [4, 6, 4]	{1, 3, 4, 5} [3, 3, 4, 5]	{4, 6, 8} [4, 6, 5]	{1, 3} [3, 3]			
J_8	$PP_{8,1}$	{1, 2, 6} [3, 4, 4]	{4, 5, 8} [6, 5, 4]	{3, 7} [4, 5]	{4, 6} [4, 6]	{7, 8} [1, 2]			
	$PP_{8,2}$	{2, 6} [5, 6]	{4} [7]	{4, 6, 8} [7, 6, 8]					

Table 7. Payoff values, process plans and job schedules for different algorithms (8 × 8 problem).

	HD-DNA (NE point at 61st generation)			DNA (NE point at 77th generation)			HA-GA (Zhou *et al.* 2010) (NE point at 80th generation)		
Job id	Payoff value	Process plan	Job scheduling	Payoff value	Process plan	Job scheduling	Payoff value	Process plan	Job scheduling
1	28	1	[7, 3, 6]	30	2	[1, 4, 3, 7]	34	2	[1, 2, 5, 7]
2	31	2	[1, 4, 4, 7, 6, 1, 4]	36	2	[1, 4, 4, 7, 4, 1, 4]	38	2	[1, 4, 4, 7, 6, 8, 4]
3	35	2	[1, 7, 8]	36	2	[1, 4, 5]	37	3	[3, 8, 7, 4]
4	28	2	[5, 8, 3, 5]	37	2	[5, 8, 3, 8]	29	2	[5, 2, 3, 8]
5	31	2	[2, 5, 3]	27	2	[7, 5, 6]	40	1	[1, 2, 3, 4, 5]
6	35	2	[3, 2, 8, 6]	37	3	[2, 5, 3]	33	3	[1, 5, 6]
7	33	2	[6, 8, 1, 8, 1]	35	1	[2, 8, 6, 5, 6]	29	2	[6, 8, 4, 4, 1]
8	27	2	[6, 4, 4]	33	1	[6, 7, 3, 4, 7]	25	2	[2, 4, 6]

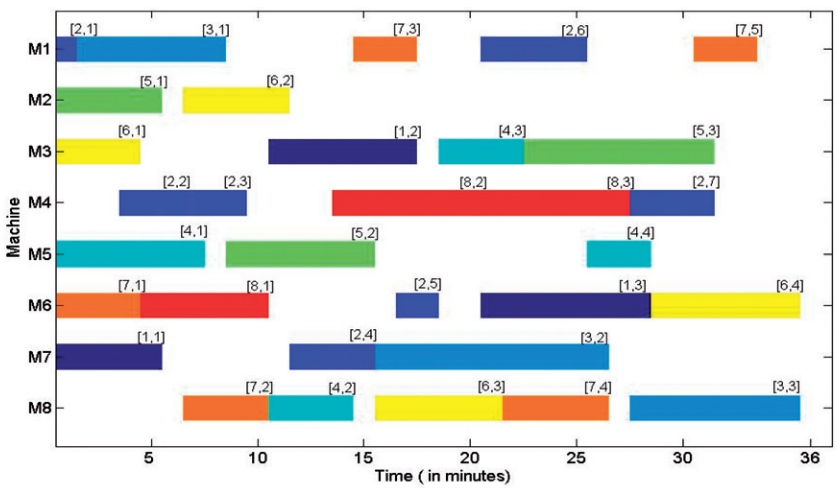

Figure 6. Gantt chart of optimal process plan decision for 8 × 8 problem.

A Gantt chart corresponding to HD-DNA results has been shown in Figure 7. HD-DNA algorithm performs better than DNA and HA-GA algorithm.

6. Results and discussion

The performance of the algorithms will be compared on the basis of three parameters which are makespan, number of generations and computational time. Makespan is the time taken to complete the processing of the products with the help of available resources. The lower the makespan the better is the performance of the algorithm and the productivity of the manufacturing resources. The maximum payoff value is equal to makespan. The makespan of all three algorithms are shown in Table 11. In order to find the process plans and scheduling plans for minimum makespan, we need to run the heuristic algorithm for multiple generations. An algorithm with lower generation will be able to converge quickly compared with other algorithms. However, lower generations will not guarantee lower computational time because additional features like hill climbing search and dynamic ligation and enzymatic action do take slightly higher computational time in each generation. Thus we have shown the computational time in Table 11. Due to lower computational time, we will be able to obtain near optimal process plans quickly which will improve the responsiveness of the manufacturing facility.

Table 8. Input data for 6×8 problem.

Job	PP	Operations				
		O_1	O_2	O_3	O_4	O_5
J_1	$PP_{1,1}$	{2, 4} [18, 22]	{7, 8} [39, 36]	{1, 2} [11, 10]	{8, 6} [31, 34]	{3, 8} [26, 24]
	$PP_{1,2}$	{2, 4} [18, 22]	{3, 5} [21, 23]	{1, 2, 4} [10, 12, 15]	{5, 6} [32, 30]	{3, 8} [26, 24]
	$PP_{1,3}$	{2, 4} [18, 22]	{3, 5} [21, 23]	{1, 7} [45, 44]	{3, 8} [26, 24]	
J_2	$PP_{2,1}$	{2, 4} [18, 22]	{7, 8} [39, 36]	{1, 2} [37, 39]	{3, 8} [26, 24]	
	$PP_{2,2}$	{2, 4} [18, 22]	{8, 6} [20, 21]	{1, 2, 4} [10, 12, 15]	{5, 6} [36, 38]	{3, 8} [26, 24]
	$PP_{2,3}$	{2, 4} [18, 22]	{8, 6} [20, 21]	{1, 7} [45, 44]	{3, 8} [26, 24]	
	$PP_{2,4}$	{2, 4} [18, 22]	{3, 5} [21, 23]	{1, 2, 4} [10, 12, 15]	{5, 6} [36, 38]	{3, 8} [26, 24]
	$PP_{2,5}$	{2, 4} [18, 22]	{3, 5} [21, 23]	{1, 7} [45, 44]	{3, 8} [26, 24]	
J_3	$PP_{3,1}$	{1, 4} [22, 25]	{6, 7} [24, 22]	{5, 8} [20, 19]	{2, 4} [22, 27]	
	$PP_{3,2}$	{3, 5} [12, 15]	{4, 6} [24, 23]	{2, 3, 5} [30, 31, 24]	{2, 4} [22, 27]	
	$PP_{3,3}$	{3, 5} [12, 15]	{6, 7} [21, 22]	{1, 8} [32, 30]	{2, 4} [22, 27]	
J_4	$PP_{4,1}$	{1, 4} [22, 25]	{6, 7} [42, 44]	{2, 4} [22, 27]		
	$PP_{4,2}$	{1, 4} [22, 25]	{5, 8} [41, 43]	{2, 4} [22, 27]		
	$PP_{4,3}$	{3, 5} [12, 15]	{4, 5} [24, 23]	{2, 3, 5} [30, 31, 29]	{2, 4} [22, 27]	
	$PP_{4,4}$	{3, 5} [12, 15]	{6, 7} [21, 22]	{1, 8} [32, 30]	{2, 4} [22, 27]	
J_5	$PP_{5,1}$	{2, 4} [18, 22]	{1, 3} [22, 25]	{6, 7} [24, 22]	{5, 8} [20, 18]	{3, 4} [22, 27]
	$PP_{5,2}$	{2, 4} [18, 22]	{3, 5} [12, 15]	{6, 8} [19, 21]	{1, 7} [32, 31]	{3, 4} [22, 27]
	$PP_{5,3}$	{2, 4} [18, 22]	{3, 5} [12, 15]	{1, 2, 6} [50, 52, 54]	{3, 4} [22, 27]	
J_6	$PP_{6,1}$	{2, 4} [18, 22]	{1, 7} [22, 24]	{6, 7} [24, 22]	{5, 8} [20, 18]	{3, 4} [22, 27]
	$PP_{6,2}$	{2, 4} [18, 22]	{1, 3} [21, 25]	{6, 7} [24, 22]	{5, 8} [20, 18]	{3, 4} [22, 27]
	$PP_{6,3}$	{2, 4} [18, 22]	{3, 5} [12, 15]	{6, 8} [53, 51]	{3, 4} [22, 27]	
	$PP_{6,4}$	{2, 4} [18, 22]	{3, 5} [12, 15]	{1, 2, 6} [50, 52, 54]	{3, 4} [22, 27]	

These three algorithms have been coded in MATLAB software and the problem is tested on Intel® Core™2 Duo CPU T7250 @2.00 GHz, 1.99 GB of RAM. Finally, from the results we have realised that the proposed algorithm HD-DNA obtains minimum makespan in lower computational time and generations compared with DNA & HA-GA algorithms.

7. Conclusion

In this paper, we develop a conceptual model for generating optimal process plans in the context of a network-based manufacturing environment. The model is general enough to be applicable to a variety of enterprises with

Table 9. Transportation time between machines.

Machine	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_1	0	3	7	10	3	5	8	12
M_2	3	0	4	7	5	3	5	8
M_3	7	4	0	3	8	5	3	5
M_4	10	7	3	0	10	8	5	3
M_5	3	5	8	10	0	3	7	10
M_6	5	3	5	8	3	0	4	7
M_7	8	5	3	5	7	4	0	3
M_8	12	8	5	3	10	7	3	0

Table 10. Payoff values, process plans and job schedules for different algorithms (6×8 problem).

	HD-DNA (NE point at 73rd generation)			DNA (NE point at 92nd generation)			HA-GA (Zhou *et al.* 2010) (NE point at 88th generation)		
Job id	Payoff value	Process plan	Job scheduling	Payoff value	Process plan	Job scheduling	Payoff value	Process plan	Job scheduling
1	128	3	[2, 3, 1, 3]	153	3	[2, 3, 7, 8]	153	3	[2, 3, 7, 8]
2	139	3	[2, 8, 7, 8]	131	5	[4, 3, 1, 3]	131	5	[4, 3, 1, 3]
3	98	2	[3, 6, 5, 2]	128	1	[4, 7, 5, 2]	128	1	[4, 7, 5, 2]
4	125	3	[5, 6, 3, 2]	104	1	[1, 6, 4]	104	1	[1, 6, 4]
5	150	3	[4, 5, 6, 2]	140	3	[2, 5, 2, 4]	140	3	[2, 5, 2, 4]
6	145	3	[4, 3, 8, 4]	153	4	[2, 3, 6, 3]	153	4	[2, 3, 6, 3]

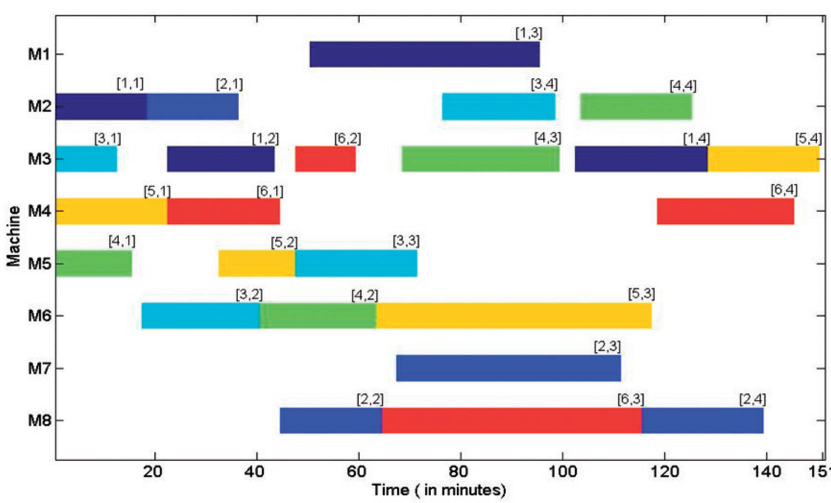

Figure 7. Gantt chart of optimal process plan decision for 6×8 problem.

networked manufacturing systems. We investigated the difference between a network manufacturing system and a traditional manufacturing system and found new characteristics and requirements of networked manufacturing systems. Consequently, we developed a game theory approach to formulate the mathematical model to represent the game and incorporated a nature-inspired evolutionary algorithm known as HD-DNA to solve the game efficiently.

Table 11. Makespan and total number of generations for all three cases.

	HD-DNA			DNA			HA-GA (Zhou *et al.* 2010)		
Exp. no.	NG	CT (in sec)	Makespan	NG	CT (in sec)	Makespan	NG	CT (in sec)	Makespan
1	40	29	35	49	34	36	52	35	41
2	61	50	35	77	54	37	80	54	40
3	73	63	150	92	74	153	88	72	153

Note: NG, number of generations; CT, computational time.

This game was further divided into two sub-games, GASG and GSSG, in which process planning and scheduling can occur. Networked manufacturing can integrate the two sub-games which co-operate with each other to achieve the objective of the model such as an optimal process plan for each job. Furthermore, owing to the problem's complexity and for effective and efficient generation of optimal process plans, we proposed a DNA and HD-DNA algorithm to find the Nash point of the game. Moreover, we tested our proposed algorithm with different experimental cases and found that for all the cases our proposed algorithm is capable of obtaining near optimal results. From this we successfully filled the gap in the literature.

The present developed model is particularly well suited for networked manufacturing problems. With this model, it would be possible to increase the efficiency of the manufacturing system. The approach developed in this work will most likely pave a new way for improving the developments in future networked manufacturing systems. As an extension of this research, we plan to apply the proposed approach for more practical network-based manufacturing systems. One further work avenue is to develop a multi-agent-based scheduling system which mainly focuses on effective information exchange and system implementation.

Acknowledgements

The authors are very grateful to the Editor and three anonymous reviewers for their valuable comments and suggestions on an earlier version of the paper that helped make the paper more informative and easily understandable.

References

Anuj, P., Chan, F.T.S., and Deshmukh, S.G., 2010. FMS scheduling with knowledge based genetic algorithm approach. *Expert Systems with Applications*, 38 (4), 3161–3171.

Anuj, P., Chan, F.T.S., and Deshmukh, S.G., 2011. Application of knowledge-based immune system (KBAIS) for computer aided process planning in CIM context. *International Journal of Production Research*, DOI:10.1080/00207543.2011.616234.

Brandimarte, P. and Calderini, M., 1995. A hierarchical bi-criterion approach to integrated process plan selection and job shop scheduling. *International Journal of Production Research*, 33 (1), 161–181.

Chan, F.T.S., Chung, S.H., and Chan, L.Y., 2008. An introduction of dominant genes in genetic algorithm for FMS. *International Journal of Production Research*, 46 (16), 4369–4389.

Chan, F.T.S., Zhang, J., and Li, P., 2001a. Modelling of integrated, distributed and cooperative process planning system using an agent-based approach. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 215 (10), 1437.

Chan, F.T.S., *et al.*, 2001b. The relevance of outsourcing and leagile strategies in performance optimisation of an integrated process planning and scheduling model. *International Journal of Production Research*, 47 (1), 119–142.

Chryssolouris, G. and Chan, S., 1985. An integrated approach to process planning and scheduling. *Annals of the CIRP*, 34 (1), 413–417.

Chryssolouris, G., Chan, S., and Cobb, W., 1984. Decision making on the factory floor: an integrated approach to process planning and scheduling. *Robotics and Computer-Integrated Manufacturing*, 1 (3–4), 413–417.

Fan, Y.S., Liu, F., and Qi, G.N., 2003. *Networked manufacturing systems and its applications*. Beijing: China Machine Press.

Henry, C.W., *et al.*, 2002. An innovative scheme for product and process design. *Journal of Materials Processing and Technology*, 123 (1), 85–92.

Huang, S.H., Zhang, H.C., and Smith, M.L., 1995. A progressive approach for the integration of process planning and scheduling. *IIE Transactions*, 27 (4), 465–464.

Jiang, B., *et al.*, 1999. An automatic process planning system for the quick generation of manufacturing process plans directly from CAD drawings. *Journal of Materials Processing and Technology*, 87 (1–3), 97–106.

Jiang, P.Y., *et al.*, 2007. e^2-MES: An e-service-driven networked manufacturing platform for extended enterprises. *International Journal of Computer Integrated Manufacturing*, 20 (2–3), 127–142.

Khoshnevis, B. and Chen, Q.M., 1991. Integration of process planning and scheduling function. *Proceedings of IIE integrated systems conference and society for integrated manufacturing conference*, 2 (3), 165–175.

Kim, Y.K., Park, K., and Ko, J., 2003. A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. *Computers and Operations Research*, 30 (8), 1151–1171.

Liu, F., Liu, J. and Lei, Q., 2002. The Connotation and research development trend of networked manufacturing. *Proceedings of China mechanical engineering annual conference*, Beijing, China, 22–27.

Li, X.Y. and Chaoyong, Z., 2010. An agent-based approach for integrated process planning and scheduling. *Expert Systems with Applications*, 37 (2), 1256–1264.

Li, W.D. and McMahon, C.A., 2007. A simulated annealing-based optimisation approach for integrated process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, 20 (1), 80–95.

Mak, K.L. and Fung, R.Y.K., 2006. Integrated process planning and scheduling/rescheduling – An agent based approach. *International Journal of Production Research*, 44 (18–19), 3627–3655.

Moon, C., Kim, J., and Hur, S., 2002. Integrated process planning and scheduling with minimising total tardiness in multi-plants supply chain. *Computers and Industrial Engineering*, 43 (1–2), 331–349.

Premaratne, S., Tritos, L., and Chan, F.T.S., 2010. Integration of manufacturing and distribution networks in a global car company – network models and numerical simulation. *International Journal of Production Research*, 49 (11), 3127–3149.

Saygin, C. and Kilic, S.E., 1999. Integrated flexible process plans with scheduling in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology*, 15 (4), 268–280.

Seredynski, F., 1997. Competitive co-evolutionary multi-agent systems: the application to mapping and scheduling problems. *Journal of Parallel and Distributed Computing*, 47 (1), 39–57.

Seredynski, F., 1998. Distributed scheduling using simple learning machines. *European Journal of Operational Research*, 107 (2), 401–413.

Seredynski, F., Koronacki, J., and Janikow, C.Z., 2001. Distributed multiprocessor scheduling with decomposed optimisation criterion. *Future Generation Computer Systems*, 17 (4), 387–396.

Shao, X.Y., *et al.*, 2009. Integration of process planning and scheduling – a modified genetic algorithm based approach. *Computers and Operations Research*, 36 (6), 2082–2096.

Shukla, S.K., Tiwari, M.K., and Son, Y.J., 2008. Bidding-based multi-agent system for integrated process planning and scheduling: a data-mining and hybrid Tabu-SA algorithm-oriented approach. *International Journal of Advanced Manufacturing Technology*, 38 (1–2), 163–175.

Usher, J.M. and Fernandes, K.J., 1996. Dynamic process planning – the static phase. *Journal of Materials Processing and Technology*, 61 (1–2), 53–58.

Wong, T.N. and Leung, C.W., 2006. Dynamic shop floor scheduling in multi-agent manufacturing system. *Expert Systems with Applications*, 31 (3), 486–494.

Zhang, H.C., 1993. IPPM – A prototype to integrated process planning and job shop scheduling functions. *Annals of the CIRP*, 42 (1), 513–517.

Zhang, J., *et al.*, 2003. A holonic architecture of the concurrent integrated process planning system. *Journal of Materials Processing and Technology*, 139 (1–3), 267–272.

Zhou, G.H., Jiang, P.Y., and Huang, G.Q., 2009. A game theory approach for job scheduling in networked manufacturing. *International Journal of Advanced Manufacturing Technology*, 41 (9–10), 972–985.

Zhou, G.H., *et al.*, 2010. A game-theoretic approach to generating optimal process plans of multiple jobs in networked manufacturing. *International Journal of Computer Integrated Manufacturing*, 23 (12), 1118–1132.