Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 6315613, 15 pages
https://doi.org/10.1155/2017/6315613

Research Article

A Production Planning Model for Make-to-Order Foundry Flow Shop with Capacity Constraint

Xixing Li,[1,2] Shunsheng Guo,[1,2] Yi Liu,[3] Baigang Du,[1,2] and Lei Wang[1,2]

[1]*Hubei Digital Manufacturing Key Laboratory, Wuhan University of Technology, Wuhan 430070, China*
[2]*School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China*
[3]*School of Mechanical and Electronic Engineering, Wuhan Donghu University, Wuhan 430070, China*

Correspondence should be addressed to Shunsheng Guo; guoshunsheng@whut.edu.cn

Received 30 July 2016; Revised 12 October 2016; Accepted 7 November 2016; Published 16 January 2017

Academic Editor: Junqiang Wang

The mode of production in the modern manufacturing enterprise mainly prefers to MTO (Make-to-Order); how to reasonably arrange the production plan has become a very common and urgent problem for enterprises' managers to improve inner production reformation in the competitive market environment. In this paper, a mathematical model of production planning is proposed to maximize the profit with capacity constraint. Four kinds of cost factors (material cost, process cost, delay cost, and facility occupy cost) are considered in the proposed model. Different factors not only result in different profit but also result in different satisfaction degrees of customers. Particularly, the delay cost and facility occupy cost cannot reach the minimum at the same time; the two objectives are interactional. This paper presents a mathematical model based on the actual production process of a foundry flow shop. An improved genetic algorithm (IGA) is proposed to solve the biobjective problem of the model. Also, the gene encoding and decoding, the definition of fitness function, and genetic operators have been illustrated. In addition, the proposed algorithm is used to solve the production planning problem of a foundry flow shop in a casting enterprise. And comparisons with other recently published algorithms show the efficiency and effectiveness of the proposed algorithm.

1. Introduction

Casting enterprise is a typical enterprise of MTO production; the requirements of products' quality and delivery are higher. The demand quantity prefers single and small batch which is not conducive to stock-based production, and time limit of casting enterprise is a long-term problem. Therefore, the timely delivery rate is generally relatively low in many casting enterprises [1]. In order to improve the timely delivery rate, a research of an actual foundry flow shop has been taken; it is a medium-sized manufacturing enterprise and the practical manufacturing process is shown in Figure 1. Casting unit number is automatically created in the commissioning date of order by the single assignment program. At the stage of production preparing, all of the manufacturing tasks can be divided into single piece production by process route. At the stage of production planning, the content of the single piece production will become rich and perfect with the whole ongoing processes (from modeling to finished product storage). If there is a requirement of adding or adjusting the process of a single piece in the following manufacturing process, it should be realized in the way of modifying the corresponding process route or manufacturing processing. At the stage of product sales, the life processing of the single piece is entering end gradually, and it contains two states: one is "complete," which means that customers have confirmed the order and its manufacturing processing life is completed, and the other one is "waiting for disposal," which means that there is some repair or rework of the product and the manufacturing processing is not completed.

The production process of foundry enterprise is various and complex, as well as various physical changes and chemical changes at the same time [2]. Therefore, reasonable production planning is the key element of foundry enterprise; it is not only the main component to ensure the stable operation of production but also an effective guarantee to complete production target. The order-driven foundry sector suffers strong competitive pressure at the regional and global

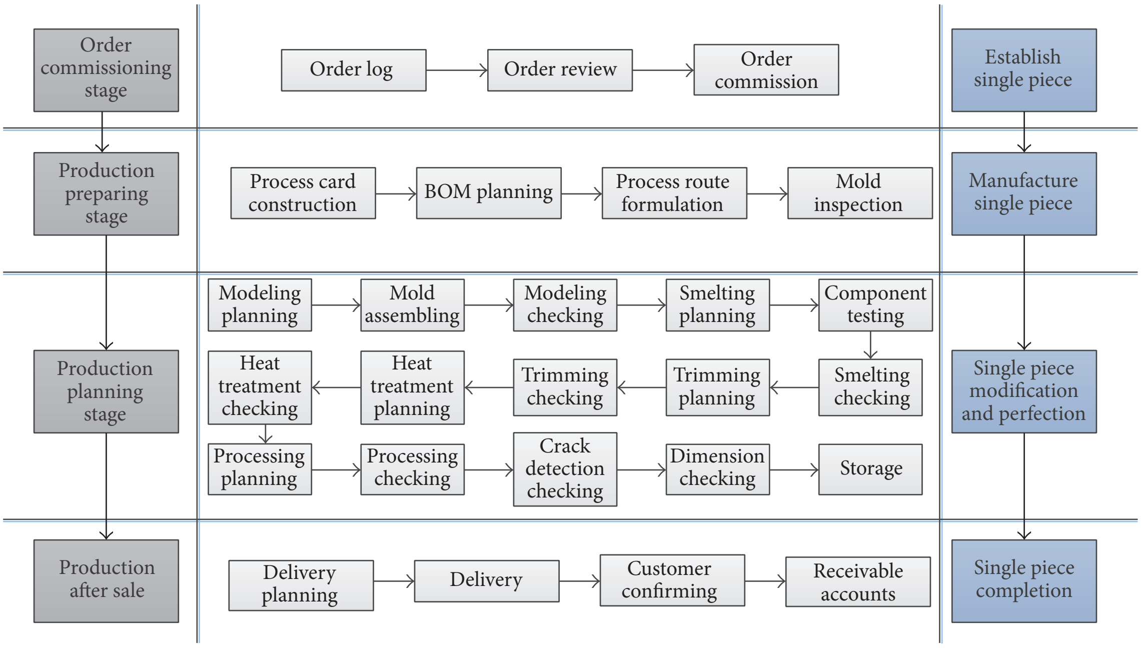


Figure 1: Actual manufacturing process.

levels as well as ecological pressure due to high energy consumption in foundry furnaces [3]. More efficient methods of elaborating production planning can directly influence sector competitiveness and energy expenses through the effective rationing of furnace loads and stock in process levels; it can effectively reduce energy expenses and direct cost. In addition, the reduction of stocking process levels can improve quality and lead time of product. However, the research of production planning in order-driven foundries has been little explored [4].

Based on the actual manufacturing process of foundry flow shop, many factors have to be considered when making the production planning, such as human resource, facility resource, technology resource, and capacity resource [5]. Because the important degrees of these factors are different from each other in different enterprises, in order to summarize the common characteristics of arranging production planning of foundry enterprise, assume that the aim of production planning is maximizing the profit in this paper, which is based on a certain amount of orders with capacity constraint of processing unit. The remainder of this paper is organized as follows. In the next section, the relevant literature related to production planning is reviewed. Problem description and optimization model are developed in Section 3. Section 4 details the proposed improved genetic algorithm. A case analysis and comparison with other algorithms are illustrated in Section 5. Conclusions are given in Section 6.

2. Literature Review

Production planning problem is a typical optimization problem, which is an important theoretical significance and economic interests in foundry industry [6–8]. The producing iron castings problem is the motivation for present work in foundry enterprise. In recent years, experts and scholars have put many researches' focus on the scheduling optimization algorithm and proposed some effective methods or models, such as multiobjective evolutionary algorithms [9] (DLP, Deterministic Linear Programming [10], ACO, Ant Colony Optimization [11], ABC, Artificial Bee Colony [12], AIA, Artificial Immune Algorithm [13], UGF, Universal Generation Function [14], and MIP, Mixed-Integer Programming [15]) and decision support optimization and simulation (SLP, System Layout Planning [16], MIND, Method for Analysis of Industrial Energy Systems [17], M&FS, Mass and Fuzzy Sets [18], SDST, Spreadsheet Decision Support Tool [19], SDSM, Scheduling Decision Support Model [20], TOFPS, Two-phase Order Fulfillment Planning Structure [21], and ERP&NN, ERP System including Neural Network [22]).

Gauri [23] demonstrated that the product-mix planning problem for batches of melt can be modeled mathematically. Weighted integer goal programming formulations were developed to determine the optimal product-mix for the immediate next heat in a small scale iron foundry, which could be useful for general iron foundries; however, the presented approach for postoptimality analysis was quite tedious, so development of a useful algorithm and necessary software was needed for easy postoptimality analysis. Teixeira Jr. et al. [24] proposed a binary integer model for production scheduling problems in market-driven foundries. The objective was to minimize the cost of manufactured products based on balancing and synchronizing the molding, pouring, and finishing steps, which aimed at eliminating high stock levels and rationing the use of production resources and foundry furnaces. Synchronization among the three production phases directly reduced production cycle time and indirectly improved the quality of products. Matičević et al. [25] developed a new mathematical model for scheduling foundry operations based on the Manufacturing Resource Planning, Just in Time, and Optimized Production Technology concepts. More recently, Camargo et al. [26] considered the production planning problem in small foundries and proposed the heuristic to solve the problem in a hierarchical way, combined the genetic algorithm with a knapsack problem algorithm to explore a larger set of alloy sequences, but did not consider the multiobjective function which could reduce both waste penalties and costs of setup, inventory, and backlogging. Gravel et al. [27] presented a genetic algorithm for the solution of an industrial scheduling problem in an Alcan aluminum foundry situated in Québec and sought the best processing sequence for n orders on m parallel machines. Gomaa [28] developed a computer aided foundry charge optimization system which could be used to minimize the charge materials and energy costs for operating a foundry melting facility during the intermediate-term and short-term planning periods (yearly, monthly, or weekly production plan). Landmann and Erdmann [29] chose a heuristic approach to find a solution to the problem, and a fuzzy logic technique was used in a higher level of abstraction originating from knowledge and experience identified in a qualitative research.

In these literatures, scholars used different algorithms or hybrid algorithms to realize production planning problems optimization for foundry enterprise, but GA is rarely used. As we know, GA is the most widely used evolutionary algorithm of production planning problem. It has good global search capability based on its self-organization, self-adaption, and self-learning, but it may converge too fast and limit the search to a local optimum prematurely if there was no effective method of population initialization. Recently, more and more papers begin to take research on this topic. Chang et al. [30] proposed a mining gene structure technique integrated with the two-phase subpopulation GA for parallel machine scheduling problem, and the mining problem of elite chromosomes was formulated as a linear assignment problem and a greedy heuristic using threshold to eliminate redundant information. Wang and Tang [31] presented an improved adaptive GA for solving the minimum makespan problem of production planning, which was inspired by hormone modulation mechanism, and then the adaptive crossover probability and adaptive mutation probability were designed. Kurdi [32] proposed a new hybrid island model GA to solve the production planning problem with the objective of makespan minimization, and a new naturally inspired self-adaptation phase strategy was proposed which was capable of striking a better balance between diversification and intensification of the search process; in the proposed self-adaptation phase strategy, the best individuals were recruited to perform a local search using Tabu Search. Huang and Süer [33] proposed a dispatching rule based genetic algorithm with fuzzy satisfaction levels to solve the multiobjective manufacturing production planning problem, and the objective was to develop a decision making platform which appropriately handles conflicts among different performance measures in a manufacturing system. Rahman et al. [34] considered a make-to-stock production system, where three related issues must be considered: the length of a production cycle, the batch size of each product, and the order of the products in each cycle. To deal with these tasks, they proposed a genetic algorithm based lot production planning approach with an objective of minimizing the sum of the setup and holding costs for permutation flow shop problems.

While because of the competition environment is becoming more and more fierce, the customers' demands are becoming more and more personalized in practical foundry flow shop, so the profit space of enterprise is becoming more and more smaller and the production structure is also changing; the traditional approaches of arranging production planning could not be effective. The foundry enterprises have to focus on two aspects: one is on-time delivery date of orders and the other is higher profit rate of orders. So the motivation of this research can be concluded as follows:

(1) Usually, the main constraint problems of making production planning for most manufacturing enterprises are human resource, outsourcing resource, manufacturing capacity, and so on. In foundry enterprise, the key constraint problem is the modeling (molding and core making) and melting capacity of melting pot, so one motivation is to propose an arranged production planning model for the order-driven foundry enterprise with modeling and melting capacity constraint of processing unit.

(2) Many useful heuristic algorithms were used to solve the multiobjective optimization problems, and different algorithms have different advantages of solving a practical problem. For solving the production planning problem, GA is the most frequently used algorithm. So the other motivation is to introduce an improved GA to solve the proposed production planning model, and the objective function is to reduce the total cost of production.

3. Problem Description and Optimization Model

3.1. Problem Description. Customers randomly submit orders with specific item type, quantity, and alloy. When the production planning manager negotiates with the customers about delivery date, usually it will obtain an unachievable date that will lead to the delay of delivery and very likely to lose future orders. While it is the most important point to achieve the best profit under the condition of satisfying the customers' demand, the product plan makers have to try their best to cut down the manufacturing cost (material cost, processing cost, delay cost, and so on) in maximum extent.

In order to make a feasible production planning which could meet the resource allocation circumstances at a lowest cost [33, 34], there are two parameters that must be considered. One is resource allocation quantity (Raq), which contains facility resource, human resource, time resource, and so on. The other one is process route (Pr), which contains modeling, melting, cleaning, casting separating, welding, and so on (the details are illustrated in Table 2). They can be expressed in the form of vector:

$$\begin{aligned} \text{Raq} &= \left(RQ_1, RQ_2, RQ_3, \ldots, RQ_p, \ldots, RQ_s\right), \\ \text{Pr} &= \left(P_1, P_2, P_3, \ldots, P_i, \ldots, P_t\right). \end{aligned} \tag{1}$$

Also, there are some special problems that exist in the actual manufacturing environment; for example, the production capacity of the processing unit will not be changed in a period of time, a process must be completed in one time, and the quality of product is usually bigger and not convenient to transport once the processing is unqualified and there is little possibility of repair. So, these problems must be taken into consideration when making the production planning.

3.2. Mathematical Model. According to the above analysis, the amount of order is fixed before making production planning, so the profit maximization is often taken as the optimization goal of making production planning, that is, the cost minimization. In the actual manufacturing environment of foundry flow shop, the capacity of processing units is unchanged in its life period, such as the melting pot; its volume is fixed, so it is different from a general manufacturing enterprise which just only considers the human resource, outsourcing resource, and transportation resource when making production planning. In this paper, an optimization model of making production planning is constructed, which is based on the modeling and melting capacity constraint of processing unit; the objective is cost minimization. Before constructing the model, some assumptions are given out as follows:

(1) Make the production planning weekly and the object is the orders which were signed last week.

(2) The important degrees of customers will not be taken into account when making the production planning, and they are just focusing on the component of product.

(3) The capacity constraint of processing unit is the key element which has been taken into consideration in this paper; the human resource constraint will not be considered.

(4) The processing units must be checked, making sure that they are good before executing the plan task, and the breakdown of processing units will not happen during the whole life of production plan.

(5) Ideally, there is no priority among the orders, and different orders have different delivery dates.

(6) The depreciation cost or consumption cost of processing facility during the whole life of production plan will not be taken into account.

Before formulating a mathematical model of the problem, the notations used throughout the remainder of this paper are defined in Notations.

3.2.1. Total Consumption Cost. The goal of resource allocation is to satisfy the acquirement of production task and decrease the total consumption cost (TCc) including material cost (Mc), process cost (Pc), delay cost (Dc), and occupy cost (Oc) of processing unit, and TCc can be illustrated as follows:

$$TCc = Mc + Pc + Oc + Dc. \tag{2}$$

In practical manufacturing environment of foundry enterprises, the demands for raw materials of different order tasks are the same; in order to reduce the mutual influence of each order task when arranging the production planning, the weight value (W^i_{gM}, $0 < W^i_{gM} < 1$, $\sum_{i=1}^{t} W^i_{gM} = 1$) of Mc_i is introduced when calculating Mc; in the same way, W^i_{gP} ($0 < W^i_{gP} < 1$, $\sum_{i=1}^{t} W^i_{gP} = 1$), W^p_{gO} ($0 < W^p_{gO} < 1$, $\sum_{p=1}^{s} W^p_{gO} = 1$), and W^i_{gD} ($0 < W^i_{gD} < 1$, $\sum_{i=1}^{t} W^i_{gD} = 1$) of Pc_i, Oc_p, and Dc_i are introduced, so (2) can be reformed into the following equation:

$$\begin{aligned} TCc = {} & \sum_{i=1}^{t} W^i_{gM} * Mc_i + \sum_{i=1}^{t} W^i_{gP} * Pc_i + \sum_{p=1}^{s} W^p_{gO} * Oc_p \\ & + \sum_{i=1}^{t} W^i_{gD} * Dc_i. \end{aligned} \tag{3}$$

As follows, (4) to (8) have illustrated the value of each variable in (3).

(1) Mc. There are different unit prices of different kinds of material, so this paper adopts (4) to summarize different material costs of different production tasks:

$$Mc_i = \sum_{j=1}^{k} \left(Mc_{ij} * \phi_i\right) \tag{4}$$

$$MC_{ij} = W_{ij} * \frac{R_j}{RT_i}, \tag{5}$$

where $\phi_i = \{0, 1\}$ and $\phi_i = 1$ represents Ptp_i containing material j; otherwise it does not, for $i = 1, 2, \ldots, t$, $j = 1, 2, \ldots, m$.

(2) Pc. There are different unit costs of different kinds of processes, and different production tasks have different processes:

$$Pc_i = \sum_{c=1}^{m} \left(Pc_{ic} * \lambda_i\right), \tag{6}$$

where $\lambda_i = \{0, 1\}$ and $\lambda_i = 1$ represents Ptp_i containing procedure c; otherwise it does not, for $i = 1, 2, \ldots, t$, $c = 1, 2, \ldots, m$.

(3) Oc. As the same batch of planning must be completed before entering into next processing unit and the production times of different plans are different from each other, there will be some processing units in the state of waiting which could result in the facility occupy cost:

$$\begin{aligned} Oc_p &= O_p * OT_p \\ OT_p &= \sum_{i=1}^{t} \left(\max\left(OTP_{ip}\right) - OTP_{ip}\right). \end{aligned} \tag{7}$$

(4) Dc. Different kinds of factors will be taken into consideration when making the production plan, so it will delay some orders' delivery date and that will also result in the delay cost:

$$Dc_i = D_i * DT_i,$$
$$D_i = \operatorname{ceil}\left(\frac{\sum_{c=1}^{m} \max\left(TP_c\right) * \lambda - \sum_{c=1}^{m} TP_{ic} * \lambda}{24}\right), \qquad (8)$$

where ceil(x) means obtaining the minimum integer which is greater than x.

3.2.2. Profit of Production Planning. Once all sales orders have been signed, the total amount (Ot) of all orders is determined which is obtained by (9). In Section 3.2.1, it is assumed that there are only four kinds of costs which have been taken into account in this mathematical model, which are material cost (Mc), process cost (Pc), delay cost (Dc), and occupy cost (Oc), so the profit of all sales orders (P_f, that is, the profit of production planning) is obtained by (10a). According to (2), (10a) can be reformed into (10b) which is shown as follows:

$$Ot = \sum_{i=1}^{t} Ot_i, \qquad (9)$$

$$P_f = Ot - TCc, \qquad (10a)$$

$$P_f = Ot - Mc - Pc - Dc - Oc. \qquad (10b)$$

3.2.3. Capacity Constraint. In order to enhance the availability of the optimization model, there are some constrains that must be considered. According to the actual situation of foundry flow shop, the production capacity is determined, so actual production and processing capacity must be considered in the process of task arrangement. The volume of the furnace is certain, so it must be taken into consideration when modeling (molding and core making) and melting. Equation (11) is the sum of modeling quantities (W_i) of production tasks i which should be less than the production capacity of the processing unit (Cpu-Modeling) when modeling, and (12) is the sum of demand quantity of material of each production task which belongs to the same batch should be less than the production capacity of the processing unit when melting (Cpu-Melting).

$$\operatorname{sum}\left(W_i\right) \le \text{Cpu-Modeling}, \qquad (11)$$

$$\operatorname{sum}\left(W_{ij}\right) \le \text{Cpu-Melting}. \qquad (12)$$

4. Solution Approach

4.1. Improved Genetic Algorithm (IGA). The genetic algorithm is a random global search algorithm which means that the process of biological evolution in nature and simulation of the survival of the fittest and also its scalability and robustness are better than other optimization algorithms, so it is often used to solve the optimization problem [35–38]. While traditional genetic algorithm is easy to converge prematurely and falls into the local optimum, one of the reasons is that it will cause many disadvantages by using the roulette method when parent solutions engender offspring solutions; the main disadvantages could be divided into two parts [39]:

(1) Some individuals with superior fitness can be evaluated rapidly which will cause the population to mature early and fall into local optimum.

(2) There is much randomness in the process of evolution, and the unelected better individuals will cause the degradation of the population.

In production planning problems of foundry industries, different scholars have proposed many different approaches by IGA. Wu et al. [40] tackled the issue by considering a two-machine flow shop problem with a truncated learning consideration, where the objective function is to minimize the makespan, and a branch-and-bound algorithm is first developed for the approximate optimal solution; then four genetic heuristic-based algorithms were proposed for the approximate optimal solution. Bandyopadhyay and Bhattacharya [41] proposed a modified genetic algorithm with a fuzzy variable crossover algorithm to minimize the value of total cost and bullwhip effect of a biobjective supplier selection problem. In comparison with a mixed-integer linear programming solver, genetic algorithm with problem-specific operators was found to provide faster (with a subquadratic computational time complexity) and more reliable solutions to very large (more than 1 million integer variables) casting sequence optimization problems [42]. Production planning problem considers the determination of machine assignment for each operation, which has been proven to a NP-hard problem, so Thammano and Teekeng [43] proposed a novel metaheuristic algorithm which was a modification of the genetic algorithm, and the proposed algorithm introduced two new concepts to the standard genetic algorithm: (1) fuzzy roulette wheel selection and (2) the mutation operation with tabu list. Worapradya and Thanakijkasem [44] developed an optimization model by integrating two main planning phases of traditional scheduling, (1) planning cast sequence and (2) scheduling of steel-making and timing of all jobs; at the same time a novel procedure was given for genetic algorithm chromosome coding which mapped Gantt chart and hierarchical chromosomes.

Production planning based on the resource allocation belongs to discrete optimization problem; in the space of discrete solutions, the approximate optimal solution is not necessary in the range from subapproximate optimal solution to better solution and maybe there is just a little step between approximate optimal solution and worst solution. Therefore, the paper illustrates the IGA from two aspects: one is adopting different evolution modes for the solutions with different fitness to ensure the diversity of population and the other one is selecting different genetic operators in the different phase of algorithm which can improve adaptive ability and better approach the approximate optimal solution in a finite number of iterations. The main differences between the IGA and the parallel genetic algorithm (PGA) are iterative process and genetic operators: (1) the iterative process of IGA is the same as the traditional GA which is serial; there is only one initial population and the evolution process is continuous from one generation to next generation, while the iterative process of PGA is parallel and there are at least two different initial populations at beginning, which can ensure the populations evolution is in parallel model; (2) the kind of genetic operators of IGA can be divided into selection operator, crossover operator, and mutation operator, while another kind of operator must be introduced into PGA which is defined as transport operator; it plays a key role in constructing the communication bridge among different initial populations; (3) the values of genetic operator in IGA are alternative in different evolution phases, while they are always the same in the iterative process of PGA. Moore [45] has proposed an accurate parallel genetic algorithm to schedule tasks on a cluster; the key element is that the parallel scheduler used M-ary encoding and included a shared communication bus constraint. Ku et al. [46] used parallel simulated annealing and genetic algorithms based on a coarse-grained model to derive solutions for solving the static facility layout problem with rectangle shape areas. The parallel implementation of simulated annealing based genetic algorithm enabled a quick search for the approximate optimal solution. Trajkovski [47] presented parallel genetic algorithm that was used for the task of evolving imperative sort programs; the individual programs with simple assembler code were illustrated and the effect of different parameters on quality of the programs and time needed for finding the solution were reported. In conclusion, the flowchart of IGA is shown in Figure 2 and the details are illustrated as follows.

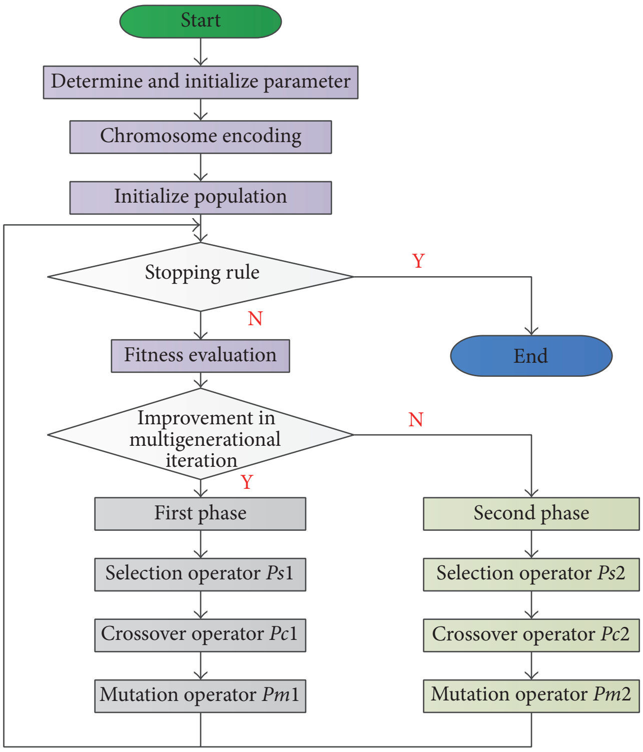


FIGURE 2: The flowchart of IGA.

4.2. Chromosome Encoding. Chromosome encoding refers to transforming the feasible solution from its solution space to search space of genetic algorithm; chromosome represents a possible solution and its coding mode usually can be divided into four types: binary coding, real coding, symbol coding, and decimal coding. Taking into consideration the actual model, this paper adopts the decimal coding and an entire chromosome coding which contains production task coding, production task process coding, and material type coding. The detailed coding scheme is shown in Table 1.

For example, the production planning with chromosome (*PtpC*) is $\{10, 3, 7, 9, 6, 2, 4, 8, 1, 5\}$, and according to the capacity constrain of processing unit, the scheduling order is {(*Ptp*-1, *Ptp*-7: *1*), (*Ptp*-3, *Ptp*-6: *2*), (*Ptp*-5, *Ptp*-9: *3*), (*Ptp*-8: *4*), (*Ptp*-2, *Ptp*-10: *5*), (*Ptp*-4: *6*)} which can be decoded into {(10, 3: *1*), (7: *2*), (9, 6, 2: *3*), (4, 8: *4*), (1: *5*), (5: *6*)} which is shown in Figure 3; the meaning of (10, 3: *1*) is that the processing order of *Ptp*-10 and *Ptp*-3 is first, and (*Ptp*-7: 2) means processing order of *Ptp*-7 is second; the remainders can be explained in the same way.

Assume that there are eleven basic processes; that is, the value of m is "11." If *Ptp-i* contains process m (P_m), the decoded corresponding value is "1"; otherwise, the value is "0"; for example, the processes of *Ptp*-10 are $P_1, P_2, P_3, P_5, P_6, P_8, P_{10}, P_{11}$, so the decoded corresponding value of $\{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}\}$ is $\{1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1\}$. The material type of *Ptp*-10 is M_1, and the decoded corresponding value of material is $\{1\}$.

4.3. Initialize Population. Initializing population of genetic algorithm has a great influence on the convergence speed and solution quality, and it is a key step in the process of genetic algorithm.

(1) Diversity of Population. The function Randperm(t) is cited to initialize population in many cases which can produce unrepeated set, which is a Matlab Function and is presented to describe the diversity as follows:

$$P = \text{Randperm}\,(t)\,. \tag{13}$$

Namely, randomly generate a set of numbers and the production planning is ranked according to these numbers' order. For instance, if $t = 10$, while $P = (10, 3, 7, 9, 6, 2, 4, 8, 1, 5)$, so the order of production task is (*Ptp-10*, *Ptp-3*, *Ptp-7*, *Ptp-9*, *Ptp-6*, *Ptp-2*, *Ptp-4*, *Ptp-8*, *Ptp-0*, *Ptp-5*).

Meanwhile, the algorithm will fall into local optimum because of unreasonable initialization, so the richness of population diversity must be taken into account during the chromosomes initialization, and, in order to ensure the diversity of population, the hamming distance, the adequacy coefficient, and the index of maximum and minimum level [48] are proposed to calculate a survival probability for the individuals; while, in this paper, the hamming distance is represented by the fitness value of population to solve the problem of diversity of population, it sets the value into 5% which is the profit rate of total orders, and the

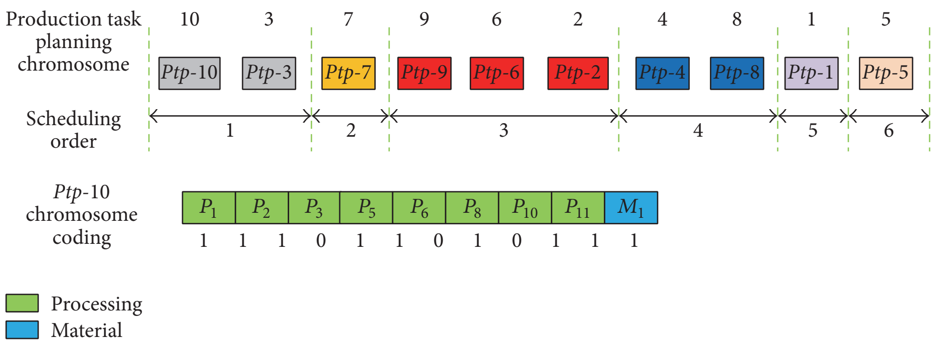


Figure 3: The scheduling order.

Table 1: Chromosome encoding.

Number	Coding name	Coding scheme
1	Production task coding	$1, 2, 3, \ldots, t$. The number is clustered according to a certain order to form an individual of the production task.
2	Production task process coding	$1, 2, 3, \ldots, m$. The number is clustered according to a certain order to form an individual of the production task process.
3	Material type coding	$1, 2, 3, \ldots, M_T$. The number is clustered according to a certain order to form an individual of the material type.
4	Decision variables	V_1, V_2. The proposed IGA is modified in two aspects: one is introducing two different evolution phases in the iterative process, and the other one is adopting different genetic operators in corresponding evolution phase. In order to divide the two different evolution phases clearly and accurately, the decision variables have been introduced and there are two conditions (V_1 and V_2) for switching two phases. V_1 represents the number of successive generations; V_2 represents the profit rate of order.

lower individual will be substituted with another random approximate optimal individual.

(2) Size of Population. In order to avoid the premature convergence and decrease the calculation amount, the population size cannot be too small or too large, and according to the actual application, the general population size is set into different value which is usually between 20 and 100.

4.4. Fitness Evaluation. Fitness function of genetic algorithm is transformed from the objective function of problems. The pros and cons of each chromosome quality in genetic algorithm are evaluated by using fitness function. The greater fitness value of chromosome is, the better quality of chromosome is. The minimum value ($\min DO$) of total cost of production planning is obtained by (14). Because the total amount of all orders (Ot) is fixed, in order to standardize calculation process of IGA, the maximum profit rate of all orders ($\max Pr$) is defined as the objective of IGA which is shown as (15):

$$\begin{aligned}\min DO &= \min(TCc) \\ &= \min\left(W_{gO} * Oc\right) + \min\left(W_{gD} * Dc\right) \\ &\quad + \min\left(W_{gM} * Mc\right) + \min\left(W_{gP} * Pc\right),\end{aligned} \tag{14}$$

$$\max Pr = \frac{Ot - \min DO}{Ot} * 100\%. \tag{15}$$

According to the proposed weight values of different factors in Section 3.1, (15) can be reformed into the following equation:

$$\max Pr = \frac{Ot - \sum_{p=1}^{s} W_{gO}^{p} * Oc_p - \sum_{i=1}^{t} W_{gD}^{i} * Dc_i - \sum_{i=1}^{t} W_{gM}^{i} * Mc_i - \sum_{i=1}^{t} W_{gP}^{i} * Pc_i}{Ot} * 100\%. \tag{16}$$

4.5. Two Different Phases. The convergence speed and efficiency of genetic algorithm are affected by the set of genetic operators (selection operator Ps, crossover operator Pc, and mutation operator Pm). The set of selection operator has a great effect on the convergence speed of algorithm, and the set of crossover operator and mutation operator will affect the ability of searching solution of algorithm. Therefore, in order to improve the adjustment ability and performance of algorithm [49–51], this paper adopts two different sets of genetic operators for the different phases of algorithm.

In the first phase, Ps is set to a larger value ($0.7 < Ps < 1$) and Pc and Pm are set to a relative smaller value. An approximate optimal solution or a subapproximate optimal solution more easily with larger Ps will be obtained. The value of Pc and Pm in the first phase of algorithm is relatively smaller, and the global search ability of algorithm is mainly affected by Pc and Pm, so when the algorithm is running in a certain phase, it will search the approximate optimal solution, repetitively, in a subspace. So the genetic operators should be adjusted and the genetic algorithm would enter into the second search phase for eliminating the premature convergence of search; namely, Ps is set into a smaller value ($0 < Ps < 0.3$), and Pc and Pm must be set into a larger value, which can make more effective solutions involved in the crossover operator and mutation operator; it will generate more new solutions.

Based on the changing of approximate optimal solution and judging rules, the two different search phases will be adopted in circulation. In the calculating process of one phase, different operators must satisfy the constraint, which is formulated in (17). It not only can ensure the diversity of population but also could avoid falling into local approximate optimal solution:

$$Ps + Pc + Pm = 1 \begin{cases} \text{Phase\_1: } 0.7 < Ps < 1, \ 0 < Pc + Pm \le 0.3, \\ \text{Phase\_2: } 0 < Ps < 0.3, \ 0.3 \le Pc + Pm \le 1. \end{cases} \quad (17)$$

4.6. Selection Operator. The stochastic tournament strategy [52–54] and elite preservation strategy [55–57] are adopted in this paper of selection operator. The stochastic tournament strategy is implemented on the whole individuals, randomly chose a series of individuals, and retained the individual which has highest fitness to next generation. The elite preservation strategy is implemented on the global approximate optimal individuals to replace the worst individual in next generation, and the process can be illustrated as follows:

(1) Selecting N individuals from a group randomly and comparing their fitness value, the individual with the highest fitness should be retained to the next generation.

(2) Repeating the above process M times, as a result, there will be M individuals from next generation.

4.7. Crossover Operator. The main purpose of crossover operator [58] is to generate better offspring by combining the genetic alleles of two selected parents from the population with probability Pc; the crossover position is generated randomly and the crossover progress can be shown in Figure 4 and it can be illustrated as follows.

Firstly, randomly generate the crossover position (here are two intersections: 7-9-6-2-4, 2-1-8-9-3) for *PtpC*-1 and *PtpC*-2 and keep the middle serial number sequence unchanged.

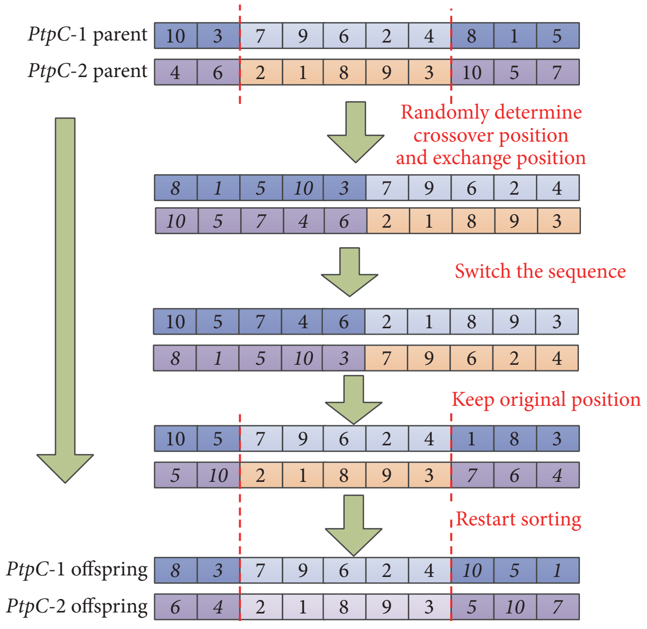

PtpC-1 parent 10 3 7 9 6 2 4 8 1 5
PtpC-2 parent 4 6 2 1 8 9 3 10 5 7
Randomly determine crossover position and exchange position
8 1 5 10 3 7 9 6 2 4
10 5 7 4 6 2 1 8 9 3
Switch the sequence
10 5 7 4 6 2 1 8 9 3
8 1 5 10 3 7 9 6 2 4
Keep original position
10 5 7 9 6 2 4 1 8 3
5 10 2 1 8 9 3 7 6 4
Restart sorting
PtpC-1 offspring 8 3 7 9 6 2 4 10 5 1
PtpC-2 offspring 6 4 2 1 8 9 3 5 10 7

FIGURE 4: The crossover operator.

Secondly, record the serial number sequence of *PtpC*-2 which is starting from the second intersection (10-5-7); when arriving at the end of the gene, return to the initial position (4-6-2-1-8-9-3) of a chromosome, and continue to record the serial number sequence of *PtpC*-2 until reaching the second intersection which is called *PtpC*-2-1 (10-5-7-4-6-2-1-8-9-3); and in the same way, we can obtain *PtpC*-1-1 (8-1-5-10-3-7-9-6-2-4).

Thirdly, switch the sequence of *PtpC*-1-1 and *PtpC*-2-1, and obtain *PtpC*-1-1 (10-5-7-4-6-2-1-8-9-3) and *PtpC*-2-1 (8-1-5-10-3-7-9-6-2-4);

Fourthly, according to the remained middle serial number sequence of *PtpC*-1 (7-9-6-2-4), remove the same serial numbers from *PtpC*-1-1 (10-5-$\underline{7}$-$\underline{4}$-$\underline{6}$-$\underline{2}$-1-8-$\underline{9}$-3) and the rest sequence of *PtpC*-1-1 is (10-5-1-8-3).

Fifthly, insert the remained middle serial number sequence of *PtpC*-1 (7-9-6-2-4) into the rest sequence (10-5-1-8-3) of *PtpC*-1-1 and the position of (7-9-6-2-4) is unchanged, and then we can obtain the *PtpC*-2 offspring (8-3-7-9-6-2-4-10-5-1).

Sixthly, in the same way, *PtpC*-2 offspring (6-4-2-1-8-9-3-5-10-7) is obtained.

4.8. Mutation Operator. While the selection operator and crossover operator are the main operators of genetic algorithm, the mutation operator can change the diversity of the population and generate some useful offspring which will improve the local search ability of population. This paper adopts the position variation of the mutation operator in probability, randomly generating the mutation genes sequence and then selecting new genes from the related gene domain [59–61]. In addition, the value of Pm is usually set value in [0.001, 0.005].

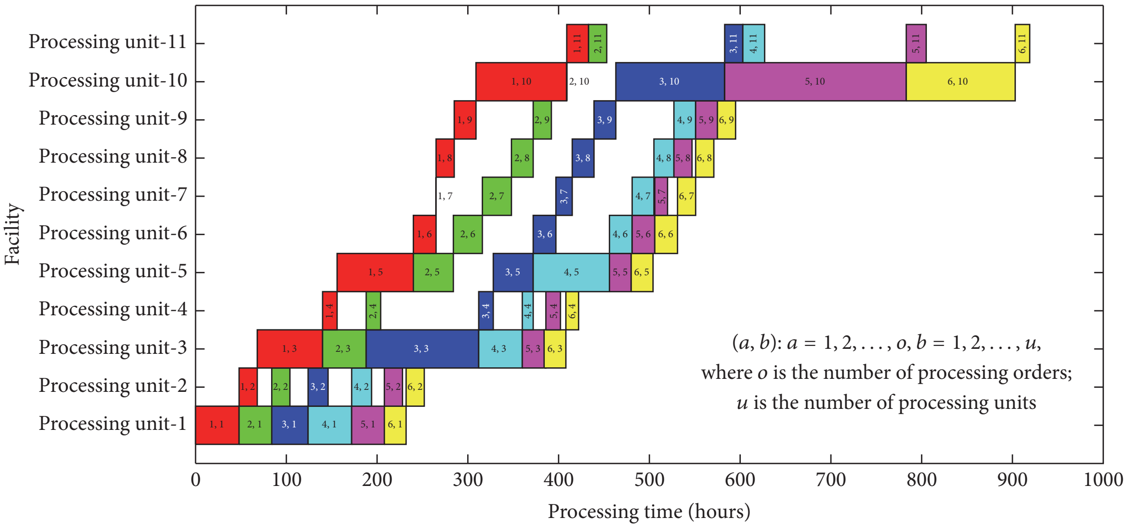


Figure 5: Gantt chart of an approximate optimal solution set of *Ptp-n* order.

4.9. Stopping Rule. The maximum value of the objective function is difficult to be determined; in order to avoid infinite loop of algorithm, the appropriate number of iterations needs to be set as the termination of algorithm, which means that when the number of iterations (*Igen*) is reaching the set value (max_*Igen*), the algorithm stops calculating.

$$Igen > \max\_Igen. \tag{18}$$

5. Illustrative Example and Performance Analysis

In order to demonstrate the application of the proposed model and algorithm, the authors investigate the actual production process of a foundry company in Anhui province of China. The products of the company are mainly casting productions according to customers' demand for the production, and it is a typical small- and medium-sized enterprise of sand casting production; there are about 400 employees; its annual output is about 10000 tons and quarterly production capacity is about 3000 tons. As usual, a completed product will undergo many different processes like modeling, melting, cleaning, dividing, and so on, and as production capacity of the processing units is different, making a flexible production plan is very important. During the process of the algorithm's implementation, the integer encoding type is adopted for gene encoding in Tables 2 and 3, and different processes of different orders are shown in Table 4; from Process-1 to Process-11 is consistent with the details of process which are illustrated in Table 2. While the molding and coring making are different processes, the processing times of them may be different, but, in the case, the molding and core making are continuous processes and are done by the same team, so in order to better achieve the approximate optimal results, we combined them as one process "modeling."

Table 2: The component of actual process parameters and processing unit.

Name	Coding	Pc_{ic} (¥)	O_k (¥)
Modeling (molding and core making)	1	40	200
Melting	2	30	150
Cleaning (cleaning and heat preservation)	3	16	140
Separating (casting separating)	4	28	150
Welding	5	28	200
Tempering (heat treatment)	6	32	180
Descaling	7	33	220
Shot blasting	8	30	160
Polishing	9	35	150
Machining	10	12	200
Delivering (packing and shipping)	11	20	120

5.1. Computational Outcomes. The parameters of proposed IGA are population size, generation number, selection ratio, crossover ratio, and mutation ratio, which are shown in Table 5 after running some pilot tests. There are two conditions of switching two phases of genetic algorithm and one of them must be satisfied:

(1) There are successive same five generation approximate optimal solutions for the genetic algorithm.

(2) The profit rate of order is less than 5%.

The IGA has been coded in Matlab R2013a and runs on a 2.6 GHz Intel(R) Core(TM) i5-3230M CPU with 4 G RAM in Windows 7 platform. After executing the Matlab codes with input parameters, an approximate optimal solution set is {(*Ptp*-1, *Ptp*-7: *1*), (*Ptp*-3, *Ptp*-6: *2*), (*Ptp*-5, *Ptp*-9: *3*), (*Ptp*-8: *4*), (*Ptp*-2, *Ptp*-10: *5*), (*Ptp*-4: *6*)} which is shown in Figure 5; the meaning of (*Ptp*-1, *Ptp*-7: *1*) is that the processing order of *Ptp*-1 and *Ptp*-7 is first, and (*Ptp*-3, *Ptp*-6: 2) means that processing order of *Ptp*-3 and *Ptp*-6 is second; the remainders can be explained in the same way. The profit rate of this

Table 3: The component production planning information of total amount of the orders, material, unit price of material, casting weight, and yield of casting.

Ptp-n	Order total (¥)	Material coding	Unit price of material (¥)	Casting weight (T)	Yield of casting	Actual input raw material weight (T)
1	104000	2	1600	20	68%	30
2	114000	1	1500	25	70%	36
3	120600	2	1600	30	68%	45
4	98000	2	1600	32	66%	49
5	120000	2	1600	28	75%	34
6	134400	2	1600	22	71%	31
7	108000	1	1500	34	72%	48
8	124200	2	1600	30	65%	47
9	128900	2	1600	26	62%	42
10	134600	2	1600	27	74%	37

Table 4: The component of production planning's process.

		Ptp-n									
		1	2	3	4	5	6	7	8	9	10
Process time (hours) of different process	Process-1	36	42	32	24	36	36	48	48	40	25
	Process-2	20	22	18	20	15	20	18	22	22	18
	Process-3	24	24	48	24	24	24	72	48	24	24
	Process-4	16	12	12	14	16	16	14	12	12	10
	Process-5	24	24	44	24	44	24	84	84	44	24
	Process-6	25	28	32	25	25	30	20	25	25	25
	Process-7	0	24	0	20	18	32	0	24	16	14
	Process-8	20	14	24	20	20	16	16	22	24	10
	Process-9	24	0	20	20	0	18	18	24	24	9
	Process-10	100	140	0	120	120	0	0	0	80	200
	Process-11	22	28	12	16	20	20	24	24	20	19

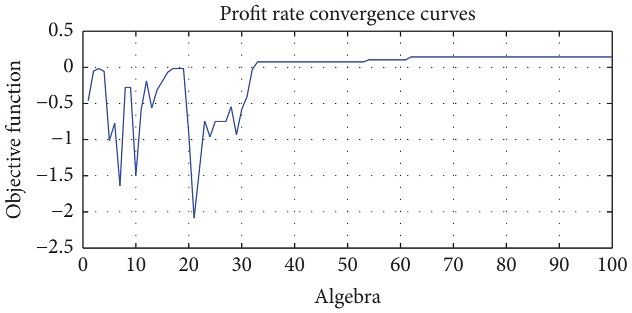


Figure 6: Convergence curves of improved GA (best profit rate is 15.05%).

approximate optimal solution is 15.05% and the convergence curve is shown in Figure 6; the comparison of signed delivery date and actual delivery date is illustrated in Table 6.

5.2. Comparison Experiment. To demonstrate the effectiveness of IGA, the comparison experiment has been carried out with comparing the obtained results with traditional genetic algorithm (Matičević et al., 2008 [25]), memetic algorithm (MA) (Gao et al. 2011 [62]), PGA (Ku et al. 2011 [46]), Iterative Backward List Scheduling Algorithm (IBLSA) (Jiang et al. 2015 [63]), and Tabu Search (TS) (Costa et al. 2015 [64]). The results of calculation based on the IGA in 30 calculation times are shown in Table 7 and the best profit rate is 15.05%. Table 8 presents the compared results which refer to the average profit rate and best profit rate. It can be observed from Table 8 that the proposed genetic algorithm is significantly better than the other four compared algorithms. The algorithm exhibiting the next best performance is TS algorithm, which clearly performs better than the other three algorithms. At the same time, the real result from company manager has been obtained and the value of best profit rate is 9.15%, which is lower than the results of five optimization algorithms.

The best result of traditional genetic algorithm is 14.45% and the convergence curve is shown in Figure 7, and the actual profit rate is 9.15%, which is offered by the company manager in actual manufacturing environment. Although the quality of the final solution obtained for both approaches is approximately equivalent for profit rate of production planning which is shown in Table 8, there is a slight difference among the approximate optimal value returned by them. In a more precise way, the proposed IGA can find a better solution

Table 5: The specific parameters.

Phase	Population size	Generation number	Cpu-Modeling (T)	Cpu-Melting (T)	*Ps*	*Pc*	*Pm*
1	40	100	80	80	0.85	0.145	0.005
2	40	100	80	80	0.25	0.745	0.005

Table 6: The comparison of delivery dates.

Ptp-n	*Ptp-1*	*Ptp-2*	*Ptp-3*	*Ptp-4*	*Ptp-5*	*Ptp-6*	*Ptp-7*	*Ptp-8*	*Ptp-9*	*Ptp-10*
Signed delivery date (day)	20	20	24	25	22	26	19	24	25	29
Actual delivery date (day)	19	34	19	38	26	19	19	27	26	34
Delay days (day)	−1	14	−5	13	4	−7	0	3	1	7

Table 7: The results of IGA.

Number	The approximate optimal solution sets of *Ptp-n*								Profit rate
1	(7, 1: *1*)	(2, 6:*2*)	(4: *3*)	(10: *4*)	(3, 5: *5*)	(8: *6*)	(9: *7*)		14.30%
2	(3, 1: *1*)	(6, 9: *2*)	(4: *3*)	(10, 5: *4*)	(8: *5*)	(2: *6*)	(7: *7*)		13.12%
3	(1, 3: *1*)	(5: *2*)	(7, 6: *3*)	(2, 10: *4*)	(4: *5*)	(9: *6*)	(8: *7*)		12.32%
4	(1, 3: *1*)	(6, 5: *2*)	(2: *3*)	(7: *4*)	(9: *5*)	(8: *6*)	(10: *7*)	(4: *8*)	14.93%
5	(3, 1: *1*)	(6, 9: *2*)	(10: *3*)	(4: *4*)	(5: *5*)	(7: *6*)	(8: *7*)	(2: *8*)	13.41%
6	(7, 1: *1*)	(2, 6: *2*)	(5, 3: *3*)	(9: *4*)	(8: *5*)	(4: *6*)	(10: *7*)		11.96%
7	(1, 7: *1*)	(3, 6: *2*)	(5, 9: *3*)	(8: *4*)	(2, 10: *5*)	(4: *6*)			15.05%
8	(1, 3: *1*)	(4: *2*)	(10, 9: *3*)	(6, 5: *4*)	(8: *5*)	(2: *6*)	(7: *7*)		13.12%
9	(1, 3: *1*)	(9, 6: *2*)	(4: *3*)	(10, 5: *4*)	(7: *5*)	(8: *6*)	(2: *7*)		13.41%
10	(1, 3: *1*)	(4: *2*)	(10, 9: *3*)	(6, 5: *4*)	(7: *5*)	(2: *6*)	(8: *7*)		13.41%
11	(3, 1: *1*)	(9: *2*)	(8, 6: *3*)	(5, 2: *4*)	(7: *5*)	(10: *6*)	(4: *7*)		14.39%
12	(3, 1: *1*)	(8, 6: *2*)	(2: *3*)	(7: *4*)	(5, 9: *5*)	(4: *6*)	(10: *7*)		14.01%
13	(3, 1: *1*)	(9, 6: *2*)	(5: *3*)	(7: *4*)	(10: *5*)	(4: *6*)	(8: *7*)	(2: *8*)	13.57%
14	(3, 1: *1*)	(9, 2: *2*)	(6, 4: *3*)	(8: *4*)	(5, 10: *5*)	(7: *6*)			8.06%
15	(3, 1: *1*)	(6, 2: *2*)	(9: *3*)	(8: *4*)	(4: *5*)	(10, 5: *6*)	(7: *7*)		12.06%
16	(3, 1: *1*)	(6, 9: *2*)	(5: *3*)	(7: *4*)	(2: *5*)	(8: *6*)	(10: *7*)	(4: *8*)	15.01%
17	(3, 1: *1*)	(2: *2*)	(8, 6: *3*)	(9: *4*)	(7: *5*)	(5, 10: *6*)	(4: *7*)		14.43%
18	(3: *1*)	(8: *2*)	(2, 1: *3*)	(6, 4: *4*)	(10, 5: *5*)	(9: *6*)	(7: *7*)		9.29%
19	(3, 1: *1*)	(7, 6: *2*)	(10: *3*)	(4: *4*)	(5: *5*)	(9, 2: *6*)	(8: *7*)		13.50%
20	(1, 3: *1*)	(6, 7: *2*)	(2, 9: *3*)	(10: *4*)	(4: *5*)	(5: *6*)	(8: *7*)		10.79%
21	(3, 1: *1*)	(8: *2*)	(9, 6: *3*)	(5, 10: *4*)	(4: *5*)	(2: *6*)	(7: *7*)		14.30%
22	(1, 7: *1*)	(9, 6: *2*)	(5, 3: *3*)	(10: *4*)	(4: *5*)	(8: *6*)	(2: *7*)		13.57%
23	(3, 1: *1*)	(8: *2*)	(2: *3*)	(4, 6: *4*)	(5, 9: *5*)	(10: *6*)	(7: *7*)		10.41%
24	(1, 3: *1*)	(10: *2*)	(4, 6: *3*)	(5, 2: *4*)	(7: *5*)	(9: *6*)	(8: *7*)		14.05%
25	(7, 1: *1*)	(3, 6: *2*)	(9, 2: *3*)	(4: *4*)	(10, 5: *5*)	(8: *6*)			10.96%
26	(3, 1: *1*)	(5, 6: *2*)	(4: *3*)	(10, 2: *4*)	(7: *5*)	(9: *6*)	(8: *7*)		15.01%
27	(7, 1: *1*)	(6, 3: *2*)	(10: *3*)	(4: *4*)	(5, 9: *5*)	(8: *6*)	(2: *7*)		13.66%
28	(1, 7: *1*)	(6, 3: *2*)	(2: *3*)	(8: *4*)	(5, 9: *5*)	(10: *6*)	(4: *7*)		14.28%
29	(7, 1: *1*)	(5, 3: *2*)	(6, 2: *3*)	(10: *4*)	(4: *5*)	(9: *6*)	(8: *7*)		11.32%
30	(7, 1: *1*)	(2, 6: *2*)	(4: *3*)	(10: *4*)	(3, 5: *5*)	(8: *6*)	(9: *7*)		11.35%

Table 8: The results comparison.

Algorithms	IGA	Traditional GA	MA	PGA	IBLSA	TS	Real result from company
Calculation times	30	30	30	30	30	30	×
Average profit rate	13.31%	12.93%	13.01%	13.17%	13.12%	13.24%	×
Best profit rate	15.05%	14.45%	14.67%	14.88%	14.76%	14.91%	9.15%

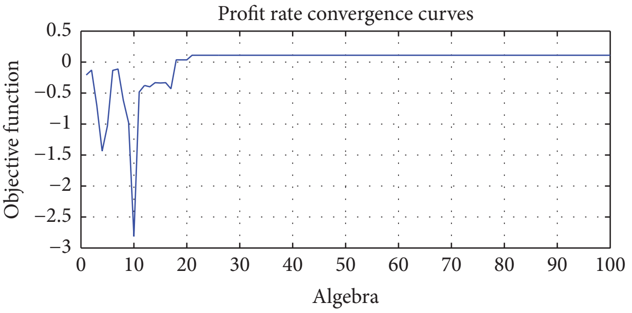


Figure 7: Convergence curves of traditional GA (best profit rate is 14.45%).

set compared to traditional genetic algorithm and the real result of company manager in terms of statistical significance.

According to the results of proposed mathematical model, the approximate optimal solutions might be varied to decrease the production cost and the enterprise manager can select it to arrange production planning in a practice situation. Furthermore, if the required data is gathered, the proposed model could be easily coded and applied by industrial/computer engineer who is familiar with operations research and optimization software like Matlab. Also, an expert system with user-friendly interface could be also designed to facilitate the proposed model by the nonexpert personnel of the enterprise managers.

6. Conclusion and Future Work

With the changing of product mode of foundry enterprise, the personalized demands of customers are more and more obvious and the status of customers is becoming higher. Almost the whole foundry enterprises are reconstructing the production structure, so how to reasonably and effectively arrange the production plan is the key problem. Combined with analysis of former researches and literatures, based on the practical manufacturing environment of foundry enterprise, an optimization mathematical model of production planning is proposed and the objective is to promote the profit rate of a certain amount of orders. In order to obtain the approximate optimal solution, a modification of traditional genetic algorithm is introduced. The IGA is mainly illustrated in two aspects: one is adopting two different phases of the process of algorithm with the different genetic operators being set separately for different phases and the other one is using two different converting rules for two different phases of algorithm; one rule is that according to the actual requirement of foundry enterprise the profit rate of orders must be higher than 5% and the other rule is that five times approximate optimal solutions are continuously appearing. Finally, an example and comparison with other algorithms are given to illustrate the proposed model and IGA. The experimental results have shown that the proposed approach is able to achieve better profit rate of orders in actual foundry industry more effectively.

In the aspect concerning the limitations of this research, it is important to notice that the constraints of this mathematical model are not the same as the actual situation of foundry enterprise, whereas there are human resource limitations, facility limitations, and transportation cost, so more factors could be taken into account in the model of production planning. A simple algorithm is often used to solve the approximate optimal solution of the mathematical problem; when more factors are introduced into the model which will increase the complexity, a hybrid algorithm should be used to solve it. Concerning the limitations of this research proposed in previous section, the suggestions for further research are listed as follows:

(1) More different factors could be introduced into the mathematical model of production planning, such as the human cost and transportation cost.

(2) A hybrid algorithm could be used to solve the production planning problem, for example, combining the advantage of Tabu Search algorithm for the higher local optimal solution with genetic algorithm, which could avoid fast convergence and promote the efficiency of algorithm.

Notations

s: The number of processing units

t: The number of production orders

m: The number of procedures

k: The number of types of materials

Ptp_i: The production task i, for $i = 1, 2, \ldots, t$

λ: $\lambda = \{0, 1\}$ where $\lambda = 1$ represents Ptp_i containing procedure c; otherwise it does not, for $i = 1, 2, \ldots, t$, $c = 1, 2, \ldots, m$

RQ_p: The resource number of processing units p

P_i: The process route of production task i, for $i = 1, 2, \ldots, t$

W_{gM}^{i}: The corresponding weight value of Mc_i

W_{gP}^{i}: The corresponding weight value of Pc_i

W_{gD}^{i}: The corresponding weight value of Dc_i

W_{gO}^{p}: The corresponding weight value of Oc_p

Mc_i: The total cost of material of Ptp_i, for $i = 1, 2, \ldots, t$

Mc_{ij}: The cost of material j of Ptp_i, for $i = 1, 2, \ldots, t$, $j = 1, 2, \ldots, k$

W_{ij}: The demand quantity of material j of Ptp_i, for $i = 1, 2, \ldots, t$, $j = 1, 2, \ldots, k$

R_j: The unit price of material j, for $j = 1, 2, \ldots, k$

RT_i: The yield of casting of Ptp_i, for $i = 1, 2, \ldots, t$

Pc_{ic}: The unit cost of procedure c in Ptp_i, for $i = 1, 2, \ldots, t$, $c = 1, 2, \ldots, m$

Oc_p: The occupy cost of processing unit p, for $p = 1, 2, \ldots, s$

O_p: The unit occupy cost of processing unit p, for $p = 1, 2, \ldots, s$

OT_p: The occupy time of processing unit p, for $p = 1, 2, \ldots, s$

OTP_{ip}: The cost time of processing unit p of Ptp_i for $i = 1, 2, \ldots, t$, $p = 1, 2, \ldots, s$

Dc_i: The delay cost of Ptp_i, for $i = 1, 2, \ldots, t$

D_i: The delay time (in days) of Ptp_i, for $i = 1, 2, \ldots, t$

DT_i: The unit cost of delay time (in days) of Ptp_i, for $i = 1, 2, \ldots, t$

TP_c: The maximum time cost of procedure c, for $c = 1, 2, \ldots, m$

TP_{ic}: The time cost of procedure c of Ptp_i, for $i = 1, 2, \ldots, t$, $c = 1, 2, \ldots, m$

Ot_i: The total amount of the order Ptp_i, for $i = 1, 2, \ldots, t$.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This research was financially supported by the Science and Technology Support Program of Hubei Province in China (2015BAA063 and 2014BAA032) and the Fundamental Research Funds for the Central Universities (2016III024 and 2016-YB-020).

References

[1] Y. K. Park and J.-M. Yang, “A scheme of supplementary production in linear programming scheduling of die casting processes,” *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 227, no. 6, pp. 855–865, 2013.

[2] F. Ballestín, F. Mallor, and P. M. Mateo, “Production scheduling in a market-driven foundry: a mathematical programming approach versus a project scheduling metaheuristic algorithm,” *Optimization and Engineering*, vol. 13, no. 4, pp. 663–687, 2012.

[3] CMC, *Cast Metal Coalition. Metalcasting Industry Technology Roadmap*, American Foundry Men’s Society-North American Die Casting Association-Steel Founder’s Society of America, Washington, DC, USA, 2003.

[4] S. A. de Araujo, M. N. Arenales, and A. R. Clark, “Lot sizing and furnace scheduling in small foundries,” *Computers and Operations Research*, vol. 35, no. 3, pp. 916–932, 2008.

[5] A.-H. Chiang, Y.-C. Chou, and W.-H. Chen, “A capacity evaluation and trading model for semiconductor foundry manufacturing,” *International Journal of Advanced Manufacturing Technology*, vol. 54, no. 1–4, pp. 1–10, 2011.

[6] F. Tao, Y. Cheng, L. Zhang, and A. Y. Nee, “Advanced manufacturing systems: socialization characteristics and trends,” *Journal of Intelligent Manufacturing*, 2015.

[7] F. Tao, L. N. Bi, Y. Zuo, and A. Y. C. Nee, “A hybrid group leader algorithm for green material selection with energy consideration in product design,” *CIRP Annals—Manufacturing Technology*, vol. 65, no. 1, pp. 9–12, 2016.

[8] F. Tao, L. Zhang, Y. K. Liu, Y. Cheng, L. Wang, and X. Xu, “Manufacturing service management in cloud manufacturing: overview and future research directions,” *Journal of Manufacturing Science and Engineering*, vol. 137, no. 4, Article ID 040912, 11 pages, 2015.

[9] J. Duda and A. Osyczka, “Multiple criteria lot-sizing in a foundry using evolutionary algorithms,” in *Evolutionary Multi-Criterion Optimization*, vol. 3410 of *Lecture Notes in Computer Science*, pp. 651–663, Springer, Berlin, Germany, 2005.

[10] M. Khakdaman, K. Y. Wong, B. Zohoori, M. K. Tiwari, and R. Merkert, “Tactical production planning in a hybrid Make-to-Stock-Make-to-Order environment under supply, process and demand uncertainties: a robust optimisation model,” *International Journal of Production Research*, vol. 53, no. 5, pp. 1358–1386, 2015.

[11] R.-H. Huang, C.-L. Yang, and W.-C. Cheng, “Flexible job shop scheduling with due window—a two-pheromone ant colony approach,” *International Journal of Production Economics*, vol. 141, no. 2, pp. 685–697, 2013.

[12] J.-Q. Li, Q.-K. Pan, and K.-Z. Gao, “Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems,” *International Journal of Advanced Manufacturing Technology*, vol. 55, no. 9-12, pp. 1159–1169, 2011.

[13] A. Bagheri, M. Zandieh, I. Mahdavi, and M. Yazdani, “An artificial immune algorithm for the flexible job-shop scheduling problem,” *Future Generation Computer Systems*, vol. 26, no. 4, pp. 533–541, 2010.

[14] V. Ebrahimipour, A. Najjarbashi, and M. Sheikhalishahi, “Multi-objective modeling for preventive maintenance scheduling in a multiple production line,” *Journal of Intelligent Manufacturing*, vol. 26, no. 1, pp. 111–122, 2013.

[15] J. Duda and A. Stawowy, “Optimization methods for lot-sizing problem in an automated foundry,” *Archives of Metallurgy and Materials*, vol. 58, no. 3, pp. 863–866, 2013.

[16] B. Li, “With SLP to plan the foundry factory of YTO group corporation,” in *Proceedings of the International Conference of Management Science and Information System*, pp. 95–99, 2009.

[17] P. Thollander, N. Mardan, and M. Karlsson, “Optimization as investment decision support in a Swedish medium-sized iron foundry—a move beyond traditional energy auditing,” *Applied Energy*, vol. 86, no. 4, pp. 433–440, 2009.

[18] T. M. Kohmann and A. C. F. Vilela, “Mathematical model for production planning of steel alloys SAE 1045 in foundries,” *Matéria*, vol. 18, no. 4, pp. 1563–1575, 2013.

[19] U. S. Sakalli and B. Birgoren, “A spreadsheet-based decision support tool for blending problems in brass casting industry,” *Computers and Industrial Engineering*, vol. 56, no. 2, pp. 724–735, 2009.

[20] K. L. Choy, Y. K. Leung, H. K. H. Chow et al., “A hybrid scheduling decision support model for minimizing job tardiness in a make-to-order based mould manufacturing environment,” *Expert Systems with Applications*, vol. 38, no. 3, pp. 1931–1941, 2011.

[21] Y.-Y. Chen, “The order fulfillment planning problem considering multi-site order allocation and single-site shop floor scheduling,” *Journal of Intelligent Manufacturing*, vol. 25, no. 3, pp. 441–458, 2014.

[22] R. Izabela and J. Mieczysaw, “Hybrid artificial intelligence system in constraint based scheduling of integrated manufacturing ERP systems,” in *Hybrid Artificial Intelligent Systems*, vol. 7209 of *Lecture Notes in Computer Science*, pp. 229–240, Springer, Berlin, Germany, 2012.

[23] S. K. Gauri, “Modeling product-mix planning for batches of melt under multiple objectives in a small scale iron foundry,” *Production Engineering*, vol. 3, no. 2, pp. 189–196, 2009.

[24] R. F. Teixeira Jr., F. C. F. Fernandes, and N. A. Pereira, “Binary integer programming formulations for scheduling in market-driven foundries,” *Computers and Industrial Engineering*, vol. 59, no. 3, pp. 425–435, 2010.

[25] G. Matičević, N. Majdandžić, and T. Lovrić, “Production scheduling model in aluminium foundry,” *Strojniski Vestnik/Journal of Mechanical Engineering*, vol. 54, no. 1, pp. 37–48, 2008.

[26] V. C. Camargo, L. Mattiolli, and F. M. Toledo, “A knapsack problem as a tool to solve the production planning problem in small foundries,” *Computers & Operations Research*, vol. 39, no. 1, pp. 86–92, 2012.

[27] M. Gravel, W. L. Price, and C. Gagné, “Scheduling jobs in an Alcan aluminium foundry using a genetic algorithm,” *International Journal of Production Research*, vol. 38, no. 13, pp. 3031–3041, 2000.

[28] A. H. Gomaa, “A systematic approach for computer-aided foundry charges optimization and planning system,” *Journal of Engineering and Applied Science*, vol. 48, no. 4, pp. 807–824, 2001.

[29] R. Landmann and R. H. Erdmann, “A heuristic approach for production scheduling in the foundry industry using fuzzy logic,” *Gestao e Producao*, vol. 18, no. 1, pp. 119–130, 2011.

[30] P.-C. Chang, S.-H. Chen, and C.-H. Liu, “Sub-population genetic algorithm with mining gene structures for multiobjective flowshop scheduling problems,” *Expert Systems with Applications*, vol. 33, no. 3, pp. 762–771, 2007.

[31] L. Wang and D.-B. Tang, “An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem,” *Expert Systems with Applications*, vol. 38, no. 6, pp. 7243–7250, 2011.

[32] M. Kurdi, “A new hybrid island model genetic algorithm for job shop scheduling problem,” *Computers & Industrial Engineering*, vol. 88, pp. 273–283, 2015.

[33] J. Huang and G. A. Süer, “A dispatching rule-based genetic algorithm for multi-objective job shop scheduling using fuzzy satisfaction levels,” *Computers & Industrial Engineering*, vol. 86, pp. 29–42, 2015.

[34] H. F. Rahman, R. Sarker, and D. Essam, “A genetic algorithm for permutation flow shop scheduling under make to stock production system,” *Computers & Industrial Engineering*, vol. 90, pp. 12–24, 2015.

[35] Q.-V. Dang, I. Nielsen, K. Steger-Jensen, and O. Madsen, “Scheduling a single mobile robot for part-feeding tasks of production lines,” *Journal of Intelligent Manufacturing*, vol. 25, no. 6, pp. 1271–1287, 2014.

[36] H. Luo, A. Zhang, and G. Q. Huang, “Active scheduling for hybrid flowshop with family setup time and inconsistent family formation,” *Journal of Intelligent Manufacturing*, vol. 26, no. 1, pp. 169–187, 2015.

[37] F. Tao, D. Zhao, Y. Hu, and Z. Zhou, “Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system,” *IEEE Transactions on Industrial Informatics*, vol. 4, no. 4, pp. 315–327, 2008.

[38] F. Tao, J. Cheng, Y. Cheng, S. Gu, T. Zheng, and H. Yang, “SDMSim: a manufacturing service supply–demand matching simulator under cloud environment,” *Robotics and Computer-Integrated Manufacturing*, vol. 45, pp. 34–46, 2017.

[39] J. Li, F. Tao, Y. Cheng, and L. Zhao, “Big data in product lifecycle management,” *International Journal of Advanced Manufacturing Technology*, vol. 81, no. 1, pp. 667–684, 2015.

[40] W.-H. Wu, W.-H. Wu, J.-C. Chen, W.-C. Lin, J. Wu, and C.-C. Wu, “A heuristic-based genetic algorithm for the two-machine flowshop scheduling with learning consideration,” *Journal of Manufacturing Systems*, vol. 35, pp. 223–233, 2015.

[41] S. Bandyopadhyay and R. Bhattacharya, “Applying modified NSGA-II for bi-objective supply chain problem,” *Journal of Intelligent Manufacturing*, vol. 24, no. 4, pp. 707–716, 2013.

[42] K. Deb, A. R. Reddy, and G. Singh, “Optimal scheduling of casting sequence using genetic algorithms,” *Materials and Manufacturing Processes*, vol. 18, no. 3, pp. 409–432, 2003.

[43] A. Thammano and W. Teekeng, “A modified genetic algorithm with fuzzy roulette wheel selection for job-shop scheduling problems,” *International Journal of General Systems*, vol. 44, no. 4, pp. 499–518, 2015.

[44] K. Worapradya and P. Thanakijkasem, “Optimising steel production schedules via a hierarchical genetic algorithm,” *South African Journal of Industrial Engineering*, vol. 25, no. 2, pp. 209–221, 2014.

[45] M. Moore, “An accurate parallel genetic algorithm to schedule tasks on a cluster,” *Parallel Computing. Theory and Applications*, vol. 30, no. 5-6, pp. 567–583, 2004.

[46] M.-Y. Ku, M. H. Hu, and M.-J. Wang, “Simulated annealing based parallel genetic algorithm for facility layout problem,” *International Journal of Production Research*, vol. 49, no. 6, pp. 1801–1812, 2011.

[47] I. Trajkovski, “Parallel genetic algorithm for creation of sort algorithms,” in *Computational Collective Intelligence: Technologies and Applications*, vol. 8083, pp. 367–376, Springer, 2013.

[48] M. Poggiolini and A. Engelbrecht, “Application of the feature-detection rule to the negative selection algorithm,” *Expert Systems with Applications*, vol. 40, no. 8, pp. 3001–3014, 2013.

[49] H. Tang, X. Li, S. Guo et al., “An optimizing model to solve the nesting problem of rectangle pieces based on genetic algorithm,” *Journal of Intelligent Manufacturing*, vol. 78, pp. 1865–1878, 2015.

[50] Y. S. Yun and M. Gen, “Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics,” *Fuzzy Optimization and Decision Making*, vol. 2, no. 2, pp. 161–175, 2003.

[51] F. Tao, C. Li, T. W. Liao, and Y. Laili, “BGM-BLA: a new algorithm for dynamic migration of virtual machines in cloud computing,” *IEEE Transactions on Services Computing*, vol. 9, no. 6, pp. 910–925, 2015.

[52] F. Tao, Y. Zuo, L. D. Xu, and L. Zhang, “IoT-Based intelligent perception and access of manufacturing resource toward cloud manufacturing,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1547–1557, 2014.

[53] C. Chamnanlor, K. Sethanan, C.-F. Chien, and M. Gen, “Re-entrant flow shop scheduling problem with time windows using hybrid genetic algorithm based on auto-tuning strategy,” *International Journal of Production Research*, vol. 52, no. 9, pp. 2612–2629, 2014.

[54] D. Lei, “Simplified multi-objective genetic algorithms for stochastic job shop scheduling,” *Applied Soft Computing*, vol. 11, no. 8, pp. 4991–4996, 2011.

[55] B. Du and S. Guo, “Production planning conflict resolution of complex product system in group manufacturing: a novel hybrid approach using ant colony optimization and Shapley value,” *Computers & Industrial Engineering*, vol. 94, pp. 158–169, 2016.

[56] T. Wang, S. Guo, and Y. Liu, "Pareto process optimization of product development project using bi-objective hybrid genetic algorithm," *Advances in Engineering Software*, vol. 65, pp. 12–22, 2013.

[57] J. A. Ventura and S.-H. Yoon, "A new genetic algorithm for lot-streaming flow shop scheduling with limited capacity buffers," *Journal of Intelligent Manufacturing*, vol. 24, no. 6, pp. 1185–1196, 2013.

[58] B. Du, S. Guo, X. Huang, Y. Li, and J. Guo, "A Pareto supplier selection algorithm for minimum the life cycle cost of complex product system," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4253–4264, 2015.

[59] Y. Cheng, F. Tao, D. Zhao, and L. Zhang, "Modeling of manufacturing service supply–demand matching hypernetwork in service-oriented manufacturing systems," *Robotics and Computer-Integrated Manufacturing*, vol. 45, pp. 59–72, 2017.

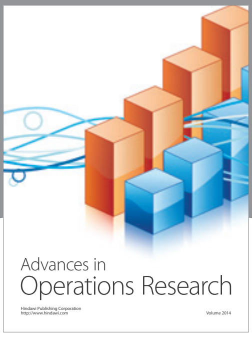
[60] F. Tao, Y. Laili, Y. Liu et al., "Concept, principle and application of dynamic configuration for intelligent algorithms," *IEEE Systems Journal*, vol. 8, no. 1, pp. 28–42, 2014.

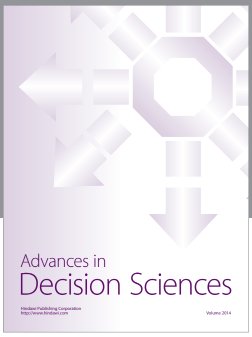
[61] L. Cheng, Q. Zhang, F. Tao, K. Ni, and Y. Cheng, "A novel search algorithm based on waterweeds reproduction principle for job shop scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1–4, pp. 405–424, 2016.

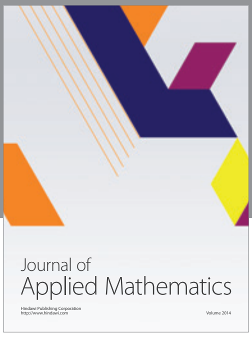
[62] L. Gao, G. Zhang, L. Zhang, and X. Li, "An efficient memetic algorithm for solving the job shop scheduling problem," *Computers & Industrial Engineering*, vol. 60, no. 4, pp. 699–705, 2011.

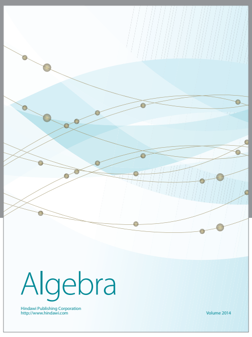
[63] S. Jiang, M. Liu, J. Hao, and W. Qian, "A bi-layer optimization approach for a hybrid flow shop scheduling problem involving controllable processing times in the steelmaking industry," *Computers & Industrial Engineering*, vol. 87, pp. 518–31, 2015.

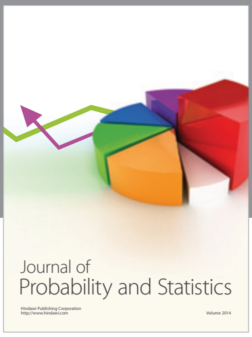
[64] A. Costa, A. Alfieri, A. Matta, and S. Fichera, "A parallel tabu search for solving the primal buffer allocation problem in serial production systems," *Computers & Operations Research*, vol. 64, pp. 97–112, 2015.


Advances in Operations Research

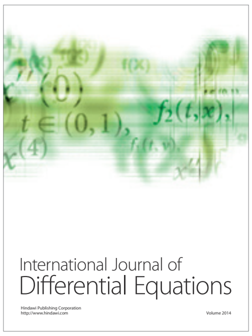

Advances in Decision Sciences


Journal of Applied Mathematics

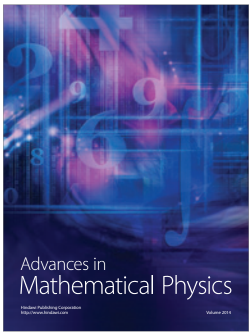

Algebra

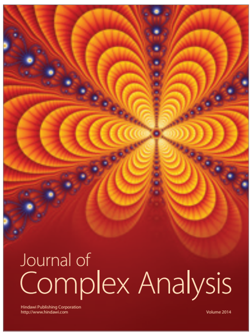

Journal of Probability and Statistics

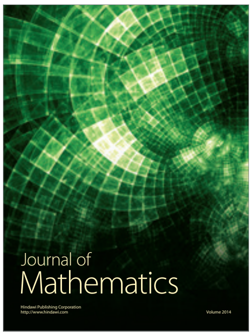
The Scientific World Journal

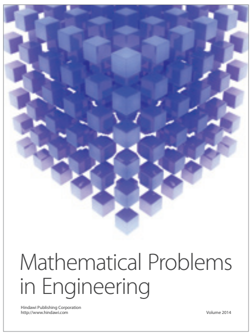

International Journal of Differential Equations

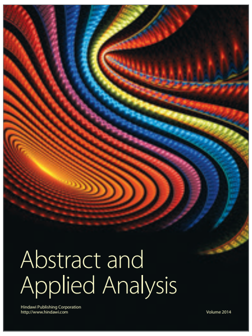
International Journal of Combinatorics

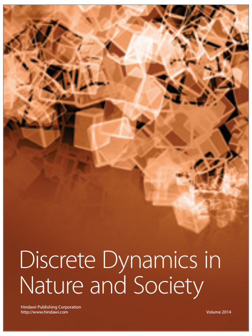

Advances in Mathematical Physics

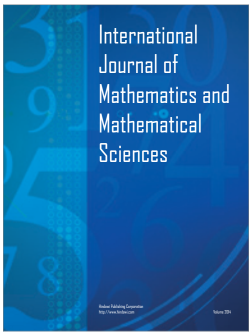

Journal of Complex Analysis


Journal of Mathematics


Mathematical Problems in Engineering

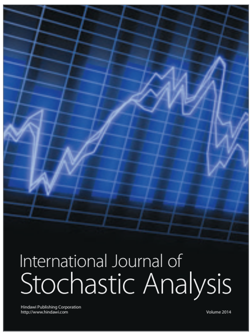

Abstract and Applied Analysis


Discrete Dynamics in Nature and Society


International Journal of Mathematics and Mathematical Sciences

Journal of Discrete Mathematics

Journal of Function Spaces


International Journal of Stochastic Analysis

Journal of Optimization