Modeling and Analysis of Single Machine Scheduling Based on Noncooperative Game Theory[1)]

WANG Chang-Jun XI Yu-Geng

(*Institute of Automation, Shanghai Jiaotong University, Shanghai* 200030)

(E-mail: cjwang@sjtu.edu.cn, ygxi@sjtu.edu.cn)

Abstract Considering the independent optimization requirement for each demander of modern manufacture, we explore the application of noncooperative game in production scheduling research, and model scheduling problem as competition of machine resources among a group of selfish jobs. Each job has its own performance objective. For the single machine, multi-jobs and non-preemptive scheduling problem, a noncooperative game model is established. Based on the model, many problems about Nash equilibrium solution, such as the existence, quantity, properties of solution space, performance of solution and algorithm are discussed. The results are tested by numerical example.

Key words Single machine scheduling, game theory, Nash equilibrium, job′s performance objective

1 Introduction

With the globalization of production and economics, modern manufacturers face much more competitive pressure. In a competitive market, more attention should be paid to different objectives from various demanders, sometimes other than the benefits of the manufacturer. The idea of serving demanders, especially the heterogeneous objectives of different demanders, should be emphasized in production scheduling. Therefore, production scheduling problem can be modeled as a multi-objective optimization problem, where jobs with independent objectives compete for scarce machine resource.

In traditional scheduling research, including single objective and multi-objective, the global objective combining all the jobs′ homogeneous objectives is studied[1,2]. Such simple coordinating method ignores the independent objectives of different jobs. Moreover, only few global objectives can be considered at the same time. Hence, some important problems about calculation and pertinence in traditional scheduling need to be solved.

Recently, by referring to the idea in economics, some researchers[3,4] decomposed the original problem according to the characters of scheduling. Then, the resource assignments could be implemented by the individuals′ autonomous behaviors, where each individual had its own performance objective. The market-like and negotiation method based on the market-like behaviors of jobs and machines was adapted in [5] to organize the scheduling. The collision and bottleneck were commendably solved by the introduction of price to negotiation. However, the necessary mathematic model and performance analysis were lacking. The auction method was introduced in [6] and [7] to study the competition for machine resource among jobs in single machine and multi-machine environments, respectively. The above researches mainly focused on solving the complexity of scheduling problem by simulative market mechanism, but failed to consider the individual performance objectives of demanders, and thus, could not accommodate to the flexibility and variety of market′s requirements. Hence, it is necessary to investigate new methodology to describe and solve the production scheduling problem.

A lot of successful applications of noncooperative game(NG) theory[8] have been reported in many complex optimization and decision problems. NG emphasizes individual rational in which the equilibrium solutions of competition satisfying individual rational are defined by Nash equilibrium(NE) solution. By designing a rational competitive mechanism according to some global objective, a satisfactory solution concurrently guaranteeing the individual and collective requirements can be induced. [9] modeled the routing control problem as the NG problem in which users competed for the scarce communication route according to their independent objectives. [10] studied the application of NG in motion planning problem of multi-robot, where each robot would compete for the scarce time and path resource in order to optimize its own independent performance objective. It can be concluded that NG is a useful tool for studying the global optimization based on individual competition. It also

1) Supported by National Natural Science Foundation of P. R. China (60274013, 60474002) and Shanghai Development Found for Science and Technology (04DZ11008)

Received July 28, 2003; in revised form July 30, 2004

provides a new and interesting idea for production scheduling. Works studying scheduling problem from the viewpoint of NG have been seldom reported. [11] introduced the idea of "coopetition" to study multi-machine scheduling problem, and built a theoretic game model, in which each job and machine had its own objective. However, no theoretic analysis and instance simulation were reported. The approach used in [12] was based on evolutionary game theory and NE concept, where operations of job were modeled as decision makers and the machines as strategies. NE schedule could be found through searching in payoff matrix. With the increasing of problem scale, the construction of payoff matrix became much more difficult, which resulted in the complexity in theoretic analysis and solving. Although the NG concept was adopted in above papers, they still failed to give strict mathematical description and necessary theoretic analysis. And the jobs′ heterogeneous objectives were never considered. All these are difficulties in the research area. Altogether, the production scheduling researches based on NG, even the basic model, theoretic analysis and algorithm, are lacking.

Influenced by the work of [10] in some variable definitions in NG description for multi-robot motion planning, the paper develops the NG model for single machine, multi-job and non-preemptive scheduling, where each job, as an independent decision maker, competes for different interval machine resource according to its own performance objective. Jobs′ strategies are mutually influenced. The relationship between the scheduling based on NG and traditional scheduling is studied. Then, the theoretic and algorithm problems of NE solution are investigated. Instance simulations are given at last.

2 Problem formulation

2.1 Decision variables and strategy set of jobs

In Fig. 1, the basic parameters for job i include release date r_i, execution time p_i, due date d_i, waiting time w_i and completion time C_i, where $C_i - r_i - w_i = p_i$. Given r_i, p_i and d_i, w_i is selected as decision variable and the feasible strategy of job i is

$$w_i^{valid} \in [0, +\infty] \tag{1}$$

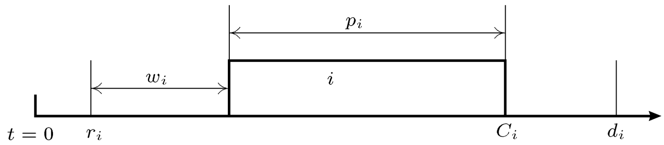

Fig. 1 The basic variables and strategy of job i's

The feasible strategic choices of all the jobs forms an n dimensional space:

$$W = w_1^{valid} \times w_2^{valid} \times \cdots \times w_n^{valid} \tag{2}$$

Obviously, W is a non-empty, closed and convex subset in Euclidean space. Given each job′s strategic choice, a schedule is formed by

$$\boldsymbol{w} = (w_1, w_2, \cdots, w_n), \quad \boldsymbol{w} \in W \tag{3}$$

As shown in Fig. 2, since machine resources are scarce, there is collision between the strategies of job i and job j while $\Delta w_{ij} = |(w_j + r_j + \frac{1}{2}p_j) - (w_i + r_i + \frac{1}{2}p_i)| < \frac{1}{2}(p_i + p_j)$. Define

$$w_{ij}^{coll} = \{\boldsymbol{w} \in W | \Delta w_{ij} < \frac{1}{2}(p_i + p_j)\} \tag{4}$$

Considering the collision between two arbitrary jobs, we can give the infeasible subset in W as

$$W^{coll} = \bigcup_{i \neq j} w_{ij}^{coll} \tag{5}$$

Then, the feasible strategic space of all jobs is given by

$$W^{valid} = W - W^{coll} \tag{6}$$

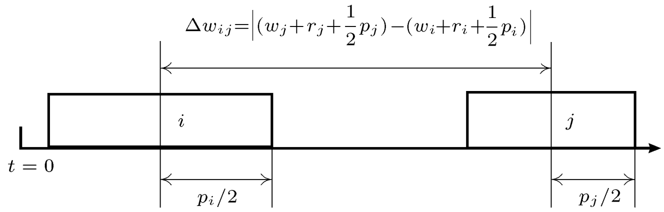

Fig. 2 Collision and feasible set between two job′s strategies

The above definitions are based on job′s strategy, *i.e.*, the waiting time. We can also build the job′s sequencing space, denoted by $S(W^{valid})$, which is a mapping of W^{valid}. The element in $S(W^{valid})$ is

$$\boldsymbol{s}(\boldsymbol{w}) = (s_1(\boldsymbol{w}), \cdots, s_n(\boldsymbol{w})) \tag{7}$$

where $\boldsymbol{w} \in W^{valid}$, $s_i(\boldsymbol{w}) \in (1, \cdots, n)$ and $s_i(\boldsymbol{w}) \neq s_j(\boldsymbol{w})$. Given a feasible scheduling strategy w, there exists a fixed feasible process sequence $s_1(\boldsymbol{w}), \cdots, s_n(\boldsymbol{w})$.

Obviously, $S(W^{valid})$ is a discrete space. For the single machine scheduling problem with n jobs, there are $n!$ elements in $S(W^{valid})$. However, there are infinite elements in the continuous space W^{valid}. Hence, $W^{valid} \to S$ is a multi-valued mapping.

2.2 Job′s performance objective

The performance objective in traditional scheduling is the combination of homogenous job′s objectives. Here, considering the practical requirements of production scheduling problem, each job has its independent optimization requirement. For a non-preemptive scheduling problem with n jobs, the performance objective of job $i (1 \leqslant i \leqslant n)$ can be defined as

$$J_i(\boldsymbol{r}, \boldsymbol{p}, \boldsymbol{d}, \boldsymbol{w}) = f_i(r_i, p_i, d_i, w_i) + \sum_{j \neq i} W_{ij}(\boldsymbol{w}) \tag{8}$$

where $f_i(w_i)$ is continuous and monotonically increasing over w_i,

$$W_{ij}(\boldsymbol{w}) = \begin{cases} 0, & \boldsymbol{w} \in W^{valid} \\ +\infty, & otherwise \end{cases} \tag{9}$$

From (8), the performance objective of job i has two terms. The former is the function of job i′s own variables r_i, p_i, d_i and strategy w_i. The latter reflects the mutual influence among the jobs′ strategy choices. Any job′s strategy choice influences the others′ feasible strategy sets, and then influences the others′ optimal strategy choices. Focusing on the multi-person multi-objective optimization problem (8) and (9), we want to find a strategy $\boldsymbol{w}^* = (w_1^*, w_2^*, \cdots, w_n^*)$ ($\boldsymbol{w}^* \in W$), such that the following holds for each i:

$$J_i(w_1^*, \cdots, w_{i-1}^*, w_i^*, w_{i+1}^*, \cdots, w_n^*) \leqslant J_i(w_1^*, \cdots, w_{i-1}^*, w_i, w_{i+1}^*, \cdots, w_n^*) \tag{10}$$

It means that there is no other strategy for job i which can further improve its own performance objective. $\boldsymbol{w}^*$ is called the Nash equilibrium (NE) schedule of (8) and (9). NE is the equilibrium result that satisfies each job′s independent objective. Denote the set of all $\boldsymbol{w}^*$ by W^*. Since $\boldsymbol{r}, \boldsymbol{p}$ and $\boldsymbol{d}$ are all given, they will be omitted in the following.

$J_i(\boldsymbol{w}^*)$ must be finite, or equation (10) is meaningless. Combining (8) and (9), we get $\boldsymbol{w} \in W^{valid}$. Hence, NE schedule must be feasible, and $W^* \subset W^{valid}$.

2.3 The Relationship between scheduling based NG and traditional scheduling

The global performance objective in traditional scheduling research usually consists of the former terms of all the jobs′ objectives (8). Since NE schedules are always feasible, (9) equals zero for NE schedule. Therefore, the global performance objective in traditional scheduling research can be written as the combination of the homogenous (8), *i.e.*,

$$L(\boldsymbol{w}, \boldsymbol{\omega}) = \sum_{i=1}^{n} \omega_i J_i(\boldsymbol{w}) \tag{11}$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2, \cdots, \omega_n) \in R_+^n$. Or,

$$L(\boldsymbol{w}, \boldsymbol{\omega}) = \max_{1 \leqslant i \leqslant n} J_i(W) \tag{12}$$

Formulas (11) and (12) can be expressed as $L = \sum_{i=1}^{n} \omega_i C_i$ and $L = \max_i C_i$ which just correspond to the traditional problems $1|r_j|\sum \omega_j C_j$ and $1|r_j|C_{\max}$, respectively. Most traditional scheduling problems can be described in the similar way. Therefore, our work enlarges the implication and the scope of traditional scheduling research. Furthermore, the NE schedule described by (10) emphasizes the satisfaction of jobs′ independent performance objectives, which agrees with the new scheduling idea of serving demanders.

3 Analysis of NE schedule′s characters

3.1 Existence of NE schedule

The existence of NE schedule is an important issue of NG application research. It is not easy to discuss the problem in scheduling problem[7]. Most researches studied it based on Kakutani fixed point theorem[8], *etc.* However, from the game problem (8) and (9), the existence of penalty term in (8) makes it difficult to guarantee the function continuous and concave-convex. Here, we prove it by utilizing the inherent character of scheduling problem as follows.

Theorem 1. The global optimal scheduling $\boldsymbol{w}^* = (w_1^*, w_2^*, \cdots, w_i^*, \cdots, w_n^*)$ of $1|r_j|\sum_{j=1}^{n} \omega_j f_j(w_j)$ is an NE schedule of NG problem (8) and (9).

Proof. If $\boldsymbol{w}^* = (w_1^*, w_2^*, \cdots, w_i^*, \cdots, w_n^*)$ is not the NE schedule of NG problem (8) and (9), there must exist a job i whose strategy w_i^* does not satisfy condition (10). Constitute a new strategy $\boldsymbol{w} = (w_1^*, \cdots, w_{i-1}^*, \bar{w}_i, w_{i+1}^*, \cdots, w_n^*)$, where $\bar{w}_i$ satisfies $J_i(w_i^*, \cdots, w_{i-1}^*, \bar{w}_i, w_{i+1}^*, \cdots, w_n^*) \leqslant J_i(w_1^*, \cdots, w_{i-1}^*, w_i^*, w_{i+1}^*, \cdots, w_n^*)$.

Obviously, $\sum_{j \neq i} W_{ij}(\boldsymbol{w}) = 0$. Since all the strategies of other jobs are unchanged, $\sum_{j \neq k} W_{kj}(\boldsymbol{w}) = 0,\ j, k \neq i$. Hence, $J_i(\boldsymbol{w}) \leqslant J_i(\boldsymbol{w}^*)$ and $J_k(\boldsymbol{w}) = J_k(\boldsymbol{w}^*)$, $k \neq i$. It follows that $\sum_{i=1}^{n} \omega_i f_i(\boldsymbol{w}) \leqslant \sum_{i=1}^{n} \omega_i f_i(\boldsymbol{w}^*)$, which indicates w is the optimal schedule for $1|r_j|\sum_{j=1}^{n} \omega_j f_j(w_j)$ problem. It is contrary to the above proposition. The theorem is thus proved. □

For the discrete optimization problem $1|r_j|\sum_{j=1}^{n} \omega_j f_j(w_j)$, the global optimal schedule always exists, so there must be NE schedule of NG problem (8) and (9).

3.2 The Characters and number of NE schedules

Consider a single machine scheduling example with two jobs: $r_1 = 0$, $p_1 = 8$; $r_2 = 5$, $p_2 = 12$. The objective of each job is to minimize C_1 and C_2^2, respectively. From (10), $\boldsymbol{w} = (0, 3)$ (*i.e.*, $\boldsymbol{s}(\boldsymbol{w}) = (1, 2)$) and $w = (17, 0)$ (*i.e.*, $\boldsymbol{s}(\boldsymbol{w}) = (2, 1)$) are all NE schedules. However, when r_2 equals 8, $\boldsymbol{w} = (0, 0)$ (*i.e.*, $\boldsymbol{s}(\boldsymbol{w}) = (1, 2)$) is still an NE schedule, while $\boldsymbol{w} = (20, 0)$ (*i.e.*, $\boldsymbol{s}(\boldsymbol{w}) = (2, 1)$) is not. From the above example, we know that NE schedule is not always unique. The reason resulting in this phenomenon is the collision among jobs′ objectives, which is influenced by r and p of jobs. In order to investigate the inward relationship between the number of NE schedules and the basic variables of jobs, we at first give Theorem 2.

Theorem 2. For each $\boldsymbol{w} \in W^*$, there exists a unique $\boldsymbol{s}(\boldsymbol{w}) \in S(W^{valid})$; for each $\boldsymbol{s}(\boldsymbol{w}) \in S(W^{valid})$, there exists at most one $\boldsymbol{w} \in W^*$.

Proof. Given an arbitrary feasible schedule, there is only one process sequence. Hence, for each $\boldsymbol{w} \in W^*$, there exists a unique $\boldsymbol{s}(\boldsymbol{w}) \in S(W^{valid})$.

Given an arbitrary $\boldsymbol{s}(\boldsymbol{w}) \in S(W^{valid})$, *i.e.*, a process sequence, the corresponding feasible scheduling strategy $\boldsymbol{w}$ may be NE, or not (see above example). If $\boldsymbol{w}$ is an NE schedule, *i.e.*, $\boldsymbol{w} = (w_1, \cdots, w_n) \in W^*$ since $f_i(w_i)$ is monotonically increasing over the waiting time w_i, for each $i (1 \leqslant i \leqslant n)$, $w_i = w_i^* = \arg\min_{w_i \in w_i^{valid}} K_i(w_1, \cdots, w_n) = \arg\min_{w_i \in w_i^{valid}} f_i(w_i)$ is unique. So $\boldsymbol{w}$ is a unique NE corresponding to $\boldsymbol{s}(\boldsymbol{w})$. □

Theorem 2 builds the relationship between NE set and the sequencing space. Under the assumption that $f_i(w_i)$ is monotonically increasing over w_i, the machine resource that each job competes for is the interval beginning from its release date till it could be completed without interruption. Define $\boldsymbol{w}_0 = (0, 0, \cdots, 0)$ as the zero waiting strategy. Under the choice of zero waiting strategy, along with the increasing direction of time slot, we can find the first job set I_1 from the whole job set I, where two arbitrary jobs in I_1 collides mutually. Furthermore, we denote the set of the jobs which may be the i-th processed job in W^* by $S_i(W^*)$. The following theorem gives the relationship between set I_1 and $S_1(W^*)$.

Theorem 3. $I_1 = S_1(W^*)$.

Proof. Obviously, $I_1 \subseteq I$, $S_1(W^*) \subseteq I$.

For any $i \in I$, $i \notin I_1$, there must exist a job j in I_1 that does not collide with job i. It follows that $r_j + p_j \leqslant r_i$. Since $f_i(w_i)$ is continuous and monotonically increasing over w_i and $r_j + p_j \leqslant r_i$, all the schedules with job i being firstly processed are not NE schedules, because job j could reduce its waiting time to occupy the machine resources before job i arrives without collision. Hence, $i \notin S_1(W^*)$, *i.e.*, for any $i \in I$, if $i \in S_1(W^*)$, then $i \in I$.

Similarly, for any $i \in I$, if $i \notin S_1(W^*)$, then $i \notin I_1$, *i.e.*, for any $i \in I$, if $i \in I_1$, then $i \in S_1(W^*)$.

Thus, the proof is completed. □

Denote the number of jobs in I_1 by $|I_1|$. Theorem 2 indicates that NE schedules of model (8) and (9) are no more than $n!$. Theorem 3 further explains that the job firstly processed in NE schedules could and only could be selected from I_1. So, we know that the number of NE schedules is no more than $|I_1|(n-1)!$. However, $|I_1|(n-1)!$ is just an upper bound of the number of NE schedules. It is difficult to obtain the exact number of NE schedules.

3.3 Performance analysis of NE schedule

The above NE schedule satisfies the independent performance requirements of jobs. Assume the global performance objective is given by

$$L(\boldsymbol{w}) = \sum_{i=1}^{n} C_i(\boldsymbol{w}) \tag{13}$$

which is corresponding to the strongly NP-hard problem $1|r_j|\sum C_j$. We evaluate the performance of NE schedules $\boldsymbol{w}^*(\boldsymbol{w}^* \in W^*)$ of problem (8) and (9) by the optimal schedule of (13). Assume the global optimal schedule of $1|r_j|\sum C_j$ is $\boldsymbol{w}^{opt}$, and in $\boldsymbol{w}^{opt}$, jobs are processed in the order of $1, 2, \cdots, n$. Job k is the last release job, *i.e.*, $r_k = \max\limits_{1\leqslant i\leqslant n} r_i$. At first, we have

$$L(\boldsymbol{w}^{opt}) \leqslant L(\boldsymbol{w}^*) \tag{14}$$

Furthermore,

$$C_1 \geqslant r_1 + p_1, C_2 \geqslant r_1 + p_1 + p_2, \cdots, C_n \geqslant r_1 + p_1 + p_2 + \cdots + p_n \tag{15}$$

Hence,

$$L(\boldsymbol{w}^{opt}) = C_1 + C_2 + \cdots + C_n \geqslant nr_1 + np_1 + (n-1)p_2 + \cdots + p_n \tag{16}$$

By the definition of NE schedule and the monotony of $f_i(w_i)$, if other jobs$'$ strategies are given, each job would reduce its waiting time as minimum as possible to optimize its performance objective. Since job k arrives finally, the performance objective of each job i satisfies

$$J_i(\boldsymbol{w}^*) \leqslant (p_1 + p_2 + \cdots + p_n) + r_k \tag{17}$$

in which, if all the jobs$'$ release dates are r_k and job i is finally processed, equality holds. We have

$$L(\boldsymbol{w}^*) = \sum_{i=1}^{n} J_i(\boldsymbol{w}^*) < n(p_1 + p_2 + \cdots + p_n) + nr_k \tag{18}$$

Combining (14), (16), and (18), we build the relationship of performance ratio between NE schedules of single machine scheduling with independent job$'$s objective and optimal schedule of traditional scheduling in the followings.

Theorem 4. The performance ratio between NE schedule $\boldsymbol{w}^* \in W^*$ of problem (8) and (9) and the optimal schedule $\boldsymbol{w}^{opt}$ of global performance objective (13) satisfies

$$1 \leqslant \frac{L(\boldsymbol{w}^*)}{L(w^{opt})} < \frac{nr_k + n(p_1 + p_2 + p_3 \cdots + p_n)}{nr_1 + np_1 + (n-1)p_2 + \cdots + p_n}$$

For $C_k \geqslant r_k + p_k$, formula (16) could be redescribed as

$$L(\boldsymbol{w}^{opt}) = C_1 + C_2 + \cdots + C_n > p_1 + p_2 + \cdots + p_n + r_k \tag{19}$$

Combining (18), we get

Corollary 1. The performance ratio between NE schedule $\boldsymbol{w}^* \in W^*$ of problem (8) and (9) and the optimal schedule $\boldsymbol{w}^{opt}$ of global performance objective (13) satisfies $1 \leqslant \dfrac{L(\boldsymbol{w}^*)}{L(\boldsymbol{w}^{opt})} < n$.

Table 1 provides an instance, which obtains the upper bound of Corollary 1, where M is an arbitrarily large positive number, and ε is an arbitrarily small positive number.

Table 1 A worst-case example for NE

Job	1	2	3	$\cdots$	n
r_i	0	ε	ε	$\cdots$	ε
p_i	M	0	0	$\cdots$	0

In optimal schedule of (13), the machine would wait ε time units until $2, 3, \cdots, n$ arrive, then executes them followed by 1. Hence,

$$L(\boldsymbol{w}^{opt}) = \sum_{i=1}^{n} C_i = \varepsilon + \cdots + \varepsilon + (\varepsilon + M) = n\varepsilon + M \tag{20}$$

In NE schedule, the machine can execute $1, 2, \cdots, n$ without idle. By the definition of NE, this processing order is NE. Therefore,

$$L(\boldsymbol{w}^*) = \sum_{i=1}^{n} C_i = M + (M + \varepsilon) \cdots + (M + (n-1)\varepsilon) = nM + \frac{n(n-1)}{2}\varepsilon \tag{21}$$

So, we have

$$\frac{L(\boldsymbol{w}^*)}{L(\boldsymbol{w}^{opt})} = \lim_{M\to\infty} \frac{nM + \dfrac{n(n-1)}{2}\varepsilon}{n\varepsilon + M} = n \tag{22}$$

The above result indicates that the NE schedules that satisfy the jobs$'$ independent performance objectives may have poor global performance. This conclusion means that individual rational and collective rational in NG are usually inconsistent. Therefore, based on the above model and results, the design of game mechanism to induce an NE schedule close to some global objective, such as (13) *etc.* is an important research issue.

4 Algorithm and simulation analysis

Theorem 2 indicates that each sequence is corresponding to no more than one NE schedule. Theorem 3 further explains that the job firstly processed in NE schedules could and only could be selected from collision job set I_1. So, a simple algorithm to solve NE schedules space W^* of (8) and (9) is given as follows:

Step 0. All the jobs choose zero waiting strategy, then find I_1;

Step 1. Find a new sequence $\boldsymbol{s}(\boldsymbol{w}) \in S(W^{valid})$, where $s_1(\boldsymbol{w}) \in I$;

Step 2. $\boldsymbol{s}(\boldsymbol{w}) \rightarrow \boldsymbol{w}$;

Step 3. Judge whether the $\boldsymbol{w}$ is an NE schedule or not according to (10). If yes, store it in W^*; otherwise, go on;

Step 4. Judge whether all the eligible elements in $S(W^{valid})$ are searched or not. If yes, go on; otherwise, go to Step 1;

Step 5. End.

Find I_1 requires $O(n)$; the eligible $\boldsymbol{s}(\boldsymbol{w})$ has $|I_1|(n-1)!$ at most; the operation judging whether a $\boldsymbol{w}$ is an NE schedule or not costs $\frac{1}{2}n(n-1)$. Hence, the complexity of above algorithm in the worst case is $O(|I_1| \times (n-1) \times n!)$. For its high computational cost, the algorithm has poor practical value. Here, we merely use it to calculate an instance to explain the theoretic analysis results.

Consider a single machine scheduling problem with 5 jobs. Table 2 gives the jobs$'$ basic variables. Each job$'$s performance objective is to minimize (8). Obviously, jobs 1, 2 and 3 compose the collision set I_1 (*i.e.*, $S_1(W^*)$). From the analysis about the upper bound of NE number in Section 3.2, we know that NE schedules are no more than $|I_1|(n-1)! = 3 \times 4! = 72$. The above algorithm can generate 52 NE schedules. Measured by the optimal schedule of (13), the globally optimal NE sequence is $(2,3,4,5,1)$ with $L = 54$, and the globally worst sequences are $(3,1,5,2,4)$ and $(3,1,2,5,4)$ with $L = 76$. The results accord with Theorem 4 and Corollary 1.

Table 2 Jobs′ release date and processing time

Job	1	2	3	4	5
r_i	0	1	3	5	10
p_i	6	3	4	2	3

5 Conclusion

With the globalization of economics and the increasing competition of modern market, the characters implicated in production scheduling that different demanders with heterogeneous requirements compete for scarce production resource become much more clear. For such a multi-person multi-objective optimization problem, it is necessary to study production scheduling problem from the viewpoint of NG. Therefore, single machine, multi-job and non-preemptive scheduling model based on NG is formulated. The paper proves the existence of NE schedule, analyzes the relationship between NE schedule space and sequencing space, and gives an upper bound of the number of NE schedules. Then, the global performance of NE schedules is quantificationally studied under the measurement of global optimal solution in traditional scheduling research. At last, an algorithm solving the whole NE schedules space is developed. Computational simulation validates the above conclusion. The work in this paper establishes an important foundation for the future research that objective of machine, *i.e.*, the global objective would be included.

References

1 Fry Td, Armstrong Rd, Lewis H. A framework for single machine multiple objective sequencing research. *Omega*, 1989, **17**(6): 595∼607

2 Karl Kempf, Reha Uzsoy, Stephen Smith, Kevin Gary. Evaluation and comparison of production schedules. *Computers in Industry*, 2000, **42**(2∼3): 203∼220

3 Shen Weiming. Distributed manufacturing scheduling using intelligent agents. *IEEE Transactions on Intelligent Systems*, 2002, **17**(1): 88∼94

4 Zhao Bo, Fan Yu-Shun. Applications of MAS to production scheduling research. *Control and Decision*, 2003, **18**(1): 1∼6

5 Grace Yuh-jiun Lin, James J Solberg. Integrated shop floor control using autonomous agents. *IIE Transactions*, 1992, **24**(3): 57∼71

6 Walsh W E, Wellman M P, Wurman P R, MacKie-Mason J K. Some economics of market-based distributed scheduling. Proceedings of the 18$^{\text{th}}$ International Conference on Distributed Computing Systems, IEEE, Amsterdam, The Netherlands: IEEE Computer Society Press, 1998. 612∼621

7 Erhan Kutanoglu, David Wu S. On combinatorial auction and lagrangean relaxation for distributed resource scheduling. *IIE Transactions*, 1999, **31**(9): 813∼826

8 Zhang Wei-Ying. Game Theory and Information Economics. Shanghai: Shanghai People′s Press, 1996. 112∼120

9 Orda A, Rom R, Shimkin N, Competitive routing in multiuser communication networks, *IEEE Transactions on Networking*, 1993, **1**(5): 510∼521

10 LaValle S M, Hutchinson S A. Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 1998, **14**(6): 912∼925

11 Teredesai T, Ramesh VC. A multi-agent mixed initiative system for real-time scheduling. Proceedings of the IEEE System, Man and Cybernetics Conference, San Diego, USA: IEEE Publisher, 1998. 439∼444

12 David Ben-Arieh, Manoj Chopra. Evolutionary game-theoretic approach for shop floor control. Proceedings of the IEEE System, Man and Cybernetics Conference, San Diego, USA: IEEE Publisher, 1998. 463∼468

WANG Chang-Jun Ph. D. candidate in Shanghai Jiaotong University. His research interests include scheduling and control of production plan.

XI Yu-Geng Professor of Shanghai Jiaotong University. Received his Ph. D. from Technical University of Munich, Germany in 1984. His research interests include predictive control, large-scale system, and intelligent robotics.