Int J Adv Manuf Technol (2009) 40:1024–1032
DOI 10.1007/s00170-008-1388-6

ORIGINAL ARTICLE

Multi-objective optimization of collaborative manufacturing chain with time-sequence constraints

Fangqi Cheng · Feifan Ye · Jianguo Yang

Received: 26 September 2007 / Accepted: 4 January 2008 / Published online: 31 January 2008

Abstract To realize the sharing and optimization deployment of manufacturing resources, a concept of collaborative manufacturing chain (CMC) is proposed for the manufacturing of complex products in a networked manufacturing environment. To acquire the optimal CMC, a multi-objective optimization model is developed to minimize the comprehensive cost and the whole production load with time-sequence constraints. Non-dominated sorting genetic algorithm (NSGA-II) is applied to solve optimization functions. The optimal solution set of Pareto is obtained. The technique for order preference by similarity to ideal solution (TOPSIS) approach is then used to identify the optimal compromise solution from the optimal solution set of Pareto. Simulation results obtained in this study indicate that the proposed model and algorithm are able to obtain satisfactory solutions.

Keywords CMC · Multi-objective optimization · NSGA-II · Time-sequence constraint

1 Introduction

The manufacturing industry today is faced with a rapidly changing market demands and global competition. More and more manufacturers recognize that it is difficult to depend on their own manufacturing resources to grasp the ever-changing market opportunities. Especially for the task to fabricate complex products, it is very difficult for single enterprise to undertake. As manufacturing enterprises becomes more specialized, they need to communicate and work with concurrently other geographically dispersed companies. The collaboration and cooperation between enterprises is changing from a supply chain to a collaborative product manufacturing chain based on material flow and task flow, respectively. Altogether, intense competition frequently makes co-location of manufacturing activities impossible and thereby drive the need for new types of collaboration that integrate dispersed enterprises and external partners into product manufacturing networks [1, 2]. At the same time, computer network technologies, especially the Internet, have injected velocity into product manufacturing activities and enabled companies to shift their traditionally centralized product manufacturing philosophy to networked product manufacturing philosophy.

The whole value chain from the raw material to the final product is rarely carried through by a single company. In reality, the manufacturing of a final product takes place on a variety of manufacturing levels in several enterprises. The high technical standard and the high quality of the complex products nowadays would not be realizable if there were no division of work among companies. Because of the increasing cooperation between small and medium sized enterprises (SMEs), the complete optimization of the value chain makes it possible to attain a global optimum concerning processing time, costs, quality, and other factors. Based on the concept of non-hierarchical regional production networks, Fischer et al. [3] developed a virtual enterprise model to improve the competitiveness of SMEs. The model is based on the very small performance unit—

F. Cheng · J. Yang
School of Mechanical Engineering,
Shanghai JiaoTong University,
800 Dongchuan Street,
Shanghai, People's Republic of China

F. Ye (✉)
Faculty of Engineering, Ningbo University,
818 Fenghua Street,
NingBo, People's Republic of China
e-mail: yefeifan@nbu.edu.cn

the competence cells. Because of the order-specific selection of those performance units of manufacturing enterprises, it becomes possible to quickly and flexibly react in case of changing market demands. The authors suggested that the ant colony algorithm was a good tool to find an optimal manufacturing process, but did not perform a numerical example to validate the proposed algorithm. Yao et al. [4] proposed the concepts of logical manufacturing unit and physical manufacturing unit to decompose and model the networked manufacturing task and the manufacturing resources. To obtain the optimal manufacturing process, the genetic algorithm (GA) is applied to solve the single objective optimization problem. Ip et al. [5] believed that dynamic alliance and virtual enterprise are essential components of global manufacturing. Then a partner selection problem is described and modeled and the GA is used to solve the optimization problem. The aforementioned literatures model the optimization problem as a comprehensive single objective optimization function, and the classical processing methods (such as linear weighting method) transform the multiple objective functions into a comprehensive single objective function. This method is generally to assign weights to multiple objectives according to the subjective preferences of decision makers. At the same time, the optimization objectives may conflict each other. It needs to gradually modify and adjust weights to obtain the satisfying solution.

Recently many researchers have started to apply multi-objective optimization algorithms to several fields such as job shop scheduling, production planning, and enterprise production networks. Gutjahr et al. [6] proposed a multi-objective combinatorial optimization formulation for the location-routing problem and the vector evaluated GA (VEGA) [7] is used to solve the problem. Li et al. [8] described a multiobjective optimization problem of product configuration and a multiobjective genetic algorithm is designed for finding near Pareto or Pareto optimal set for the problem. The niched Pareto GA (NPGA) is used for comparing with the proposed genetic algorithm. Kuriakose and Shunmugam [9] proposed a multi-objective optimization method based on a non-dominated sorting GA (NSGA) to optimize Wire-electro discharge machining process. Ding et al. [10] provided an integrated toolbox based on NSGA-II for a holistic assessment and optimization of enterprise networks. Altiparmak et al. [11] proposed a new solution procedure based on genetic algorithms to find the set of Pareto-optimal solutions for multi-objective supply chain network design problem. With regard to the multi-objective optimization algorithms, the final result usually is not a single solution, but a whole Pareto-optimal set. For the multiobjective problem, it often involves simultaneous optimization of several incommensurable and often competing objectives. And the multi-objective optimization algorithms are suited for the problems. In view of the fact that none of the solutions in the non-dominated set is absolutely better then any other, any one of them is an acceptable solution. According to the obtained Pareto optimal solution set, the relationship among goals could be analyzed further, which benefit rational and effective decision-making.

In this paper, a multi-objective optimization model for optimizing collaborative manufacturing chain (CMC) is developed. To solve the optimization problem, our choice finally turned to a multiobjective GA framework called NSGA-II [12]. The following are the reasons: (a) its modular and flexible structure, (b) the possibility of upgrading a single-objective GA to NSGA-II, and (c) its successful applications to a wide range of optimization problems. Therefore, the NSGA-II approach is applied to obtain the Pareto optimal solutions.

The reminder of the paper is organized as follows. In Sect. 2, the concept of collaborative manufacturing chain is proposed and a multi-objective optimization model is developed. The general framework of the fast non-dominated sorting genetic algorithm (NSGA-II) is described and a subtask scheduling procedure is given in Sect. 3. Numerical experiment is conducted in Sect. 4. Section 5 brings some concluding remarks.

2 Problem formulation

2.1 Collaborative manufacturing chain

To describe the collaborative manufacturing chain (CMC), several definitions are firstly given.

Definition 1: *Collaborative manufacturing unit (CMU)*: A CMU is composed of all physical equipments located at the same place, including machining equipment, fixture accessories, and soft resources. A CMU has definite core manufacturing competences and is highly specialized and cooperation-independent, but autonomous in terms of law and economics. The CMUs are formed from existing enterprises according to their core competences. Therefore, an enterprise might unite one or more CMUs. A CMU necessarily belongs to an enterprise. All CMUs form a pool of potential core competences, out of which CMUs are activated corresponding to special requirements of manufacturing order. It is useful for the task of manufacturing complex products to select the most capable CMUs.

Definition 2: *Manufacturing task unit* (MTU): For the task of manufacturing complex parts, an enterprise cannot finish the whole production procedure by itself because of the limitations of manufacturing resources. The enterprise which received the order decomposes the whole manufacturing task into several subtasks based on the manufacturing characteristics and process planning of the complex parts. The subtask is here defined as an MTU. An MTU can contain one or more manufacturing procedures, and is the basic unit of collaboration. Because of the diversity of manufacturing processes, a MTU can be undertaken by one of the same or different types of CMUs. A CMU can finish one or more MTUs.

Suppose that a dominant CMU wins a manufacturing task which consists of several MTUs. In view of the special requirements of manufacturing complex parts, there may be execution sequence between arbitrary two MTUs. If MTU k can only begin after the completion of MTU i, i.e., MTU i precedes MTU k, then we define the connected MTU pair by (i, k) □ H. H is the set of connected MTU pairs. For the concrete MTU, different CMUs to undertake have different execution times. Based on the above analysis, it is obvious that there are time-sequence constraints among MTUs. The structure of the whole manufacturing task is described as a directed network. An example of a manufacturing task consisting of eight MTUs is shown in Fig. 1. The nodes in the figure denote MTUs, and the arrow lines denote the sequence between two MTUs. For the convenience of description, we label these MTUs such that i<k. The final MTU is labeled as MTU n. If the final MTU cannot be determined, a virtual final MTU can be created. For example, in Fig. 1 without MTU 8, it would be a puzzle to determine which is the final MTU between MTUs 6 and 7. Therefore, we have to create a virtual final MTU to follow MTUs 6 and 7, and label it as MTU 8. MTU S is the first beginning virtual MTU and MTU E is the last complete virtual MUT, and they all do not consume resources and the execution times are zeros. Thus, we can define that the completion time of final MTU d_n is the completion time of the whole manufacturing task.

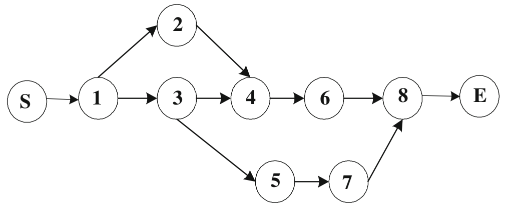

Fig. 1 Directed network of MTUs

Definition 3: *Collaborative manufacturing chain* (CMC): For the task of manufacturing complex parts, it can be decomposed into several MTUs. Every MTU can select only one CMU to undertake from a pool of CMUs. All of the manufacturing service provided by the corresponding CMU form a manufacturing service chain according to time-sequence constraints between MTUs. We define the manufacturing service chain as collaborative manufacturing chain.

For MTU *i* (*i*=1, 2,..., *n*), suppose that there are m_i CMUs responding to the tender invitation. To describe CMC distinctly, we illustrate the problem as a directed graph (Fig. 2). Each node denotes a CMU. The pairs of nodes (*i*, *i*+1) attached to an edge are arranged. Thereby, *i* is the first node—the initial node—and *i*+1 is the second node —final node—of the pairs (*i*, *i*+1).Therefore, it is necessary to insert an initial node, a so-called *source*, for all nodes which are in the beginning of the CMC. Starting from that, all CMUs are integrated in the graph according to the sequence. After the last processing step, the CMUs meet in a common final node of the directed graph, called *drain*. Thus, each CMC firstly starts from a source and finally ends in a drain. There are many CMCs from the source to the drain. The strongly emphasized route in Fig. 2 represents a concrete, realizable CMC.

2.2 Multi-objective optimization model

In order to describe the optimization model of CMC, we need the notation as follows:

N	Number of MTUs
D	Due date
β	Penalty parameter for tardiness of a period
(i, k)	Connected pairs of MTUs
H	Set of connected pairs of MTUs
m_I	Number of candidate CMUs bidding for MTU I
C_{ij}	Processing cost of CMU j for MTU i

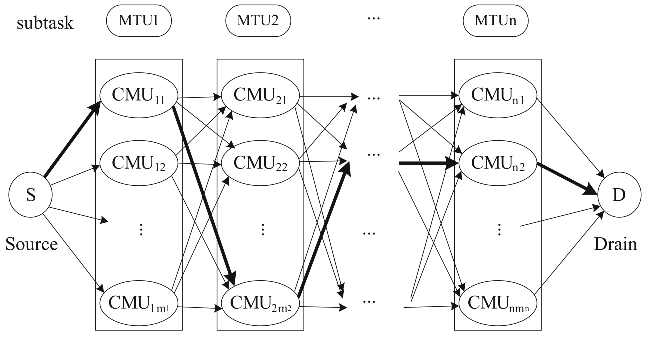

Fig. 2 Illustration of CMC

b_I Starting time of CMU j for MTU i
d_I Completion time of CMU j for MTU i
d_n Completion time of MTU n
ρ_{ij} Production load rate of CMU j for MTU I
p_{ij} Processing time of CMU j for MTU i
$LC_{ij,(I+1)q}$ Linked cost between CMU j for MTU i and CMU q for MTU i+1

The CMC optimization problem can be described as follows. Assuming a dominant CMU win, one can decompose a bid of manufacturing complex parts and the whole task into several MTUs. The CMU is not able to complete the whole task by its own capacity and resources. Therefore, it needs other CMUs to collaboratively complete the task.

D is the due date of the whole task. If the whole task is tardy, the dominant CMU will receive a tardiness penalty of β for per tardy period.

The objective is to find an optimal CMC by minimizing the total costs and the holistic production load rate.

Define the variables

$$x_{ij}(t) = \begin{cases} 1 & \text{MTU } i \text{ is contracted to CMU } j \text{ at period } t \\ 0 & \text{otherwise} \end{cases}$$

Then, the optimization problem can be described as following models.

Cost – The summation of processing cost , linked cost and possible penalty cost should be minimized.

$$\begin{aligned} \min f_1(x) = & \sum_{i=1}^{n}\sum_{j=1}^{m_i} C_{ij}\sum_{t=0}^{d_n} x_{ij}(t) \\ & + \sum_{i=1}^{n-1}\sum_{j=1}^{m_i}\sum_{q=1}^{m_{i+1}} LC_{ij,(i+1)q}\sum_{t=0}^{d_n} x_{ij}(t) + \beta[d_n - D]^+ \end{aligned} \tag{1}$$

Load – The product of production load rates of selected CMUs should be minimized.

$$\min f_2(x) = \prod_{i=1}^{n}\sum_{j=1}^{m_i}\rho_{ij}\sum_{t=0}^{d_n} x_{ij}(t) \tag{2}$$

Constraints:

$$\sum_{j=1}^{m_i}\sum_{t=0}^{d_n} x_{ij}(t) = 1 \quad i = 1, 2, \cdots, n \tag{3}$$

$$(t + p_{ij})\sum_{j=1}^{m_i}\sum_{t=0}^{d_n} x_{ij}(t) \le t\sum_{j=1}^{m_k}\sum_{t=0}^{d_n} x_{kj}(t), \forall (i,k) \in H \tag{4}$$

$$\sum_{j=1}^{m_n}\sum_{t=0}^{d_n} (t + p_{nj}) x_{nj}(t) = d_n \tag{5}$$

where $t = 0, 1, \cdots, d_n$, $[v]^+$ stands for max $\{0, v\}$.

Equation (1) is the first optimization function and the function consists of three parts. $\sum_{i=1}^{n}\sum_{j=1}^{m_i} C_{ij}\sum_{t=0}^{d_n} x_{ij}(t)$, $\sum_{i=1}^{n-1}\sum_{j=1}^{m_i}\sum_{q=1}^{m_{i+1}} LC_{ij,(i+1)q}\sum_{t=0}^{d_n} x_{ij}(t)$, and $\beta\,[d_n - D]^+$ represent minimizing the processing cost, minimizing the linked cost between CMUs, and minimizing the penalty cost for tardiness of a period respectively.

Equation (2) is the second optimization function and its objective is to minimize the holistic production load rate of a CMC.

Constraint (3) indicates that only one CMU is selected for each MTU.

Constraint (4) shows that the starting time of MTU k is always less than or equal to the summation of the starting time and the processing time of MTU i ((i, k) ∈ H).

Constraint (5) indicates that the completion time of final MTU n is equal to the summation of its starting time and the processing time. At the same time, it is the completion time of the whole manufacturing task as well.

3 Optimization algorithm embedded MTU scheduling

3.1 Pareto optimality

Multi-objective optimization problems consist of simultaneously optimizing several objective functions and the principles of multi-objective optimization are different from that in a single-objective optimization. The concept of Pareto dominance and optimality can be expressed as follows for a multi-objective minimization problem:

$$\text{Minimize} \quad f(x) = (f_1(x), f_2(x), \cdots, f_n(x))$$

$$\text{subject to} \quad g(x) = (g_1(x), g_2(x), \cdots, g_n(x)) \le 0$$

where $f(x)$ is the vector-valued function, x is the decision vector, and $g(x)$ is a vector of constraints. Considering two decision vectors a and b, a is said to dominate b:

$$\mathit{iff}\ \forall i \in \{1, 2, \cdots, n\} : f_i(a) \le f_i(b) \text{ and } \exists i \in \{1, 2, \cdots, n\} : f_i(a) < f_i(b)$$

The decision vectors that are non-dominated within the entire search space are denoted as Pareto-optimal and constitute the Pareto-optimal set or Pareto-optimal front.

3.2 NSGA-II approach

Evolutionary algorithms (EA) have been recognized to be particularly suitable to solve multi-objective optimization problems, because they deal simultaneously with a set of possible solutions which allows an entire set of Pareto-

Table 1 Procedure of *NSGA_2*

```
NSGA_2( )
first_pop(pop,ns)
non_dominated_sort( )
get_margins( )
repeat
    add_children( )
    non_dominated_sort( )
    get_margins( )
    newpop←φ and i← 1
    while |newpop|+|front(i)| ≤ ns do
        newpop= newpop ∪ front(i)
        i← i+1
    end while
    missing← ns- |newpop|
    if missing ≠ 0 then
        margin_sort(front,i,margin)
        for j← 1 to missing do
            newpop= newpop ∪ the jth
        end for
    end if
    pop← newpop
until stopping_criterion
```

optimal solutions to be evolved in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of traditional mathematical programming techniques. Moreover, EAs are less susceptible to the shape or continuity of the Pareto front [13]. This has led to the development of many successful evolutionary multi-objective optimization algorithms over the past decade.

The notion of NSGA was first suggested by Goldberg [14] and then implemented by Srinivas and Deb [15]. The main idea behind the non-dominated sorting procedure is that a ranking selection method is used to emphasize good points and a niching method is used to maintain a stable subpopulation of good points. NSGA differs from a simple genetic algorithm only in the way to select operator works. The crossover and mutation operators remain as usual. The efficiency of NSGA lies in the way of multiple objectives is reduced to a single fitness measure by the creation of number of fronts, sorted according to nondomination.

Although the NSGA approach has been successfully applied on a number of multi-objective optimization problems, the main criticism of the NSGA approach has been (i) its high computational complexity of non-dominated sorting, $O(MN^3)$ where M is the number of objectives and N is the population size, (ii) the lack of elitism, and (iii) the need for specifying the tunable sharing parameter. Recently, Deb et al. [12] reported an improved version of NSGA called NSGA-II to address all of these issues. Specifically, NSGA-II alleviates all the above difficulties by introducing a fast non-dominated sorting procedure with $O(MN^2)$ computational complexity, an elitist-preserving approach, and a parameterless niching operator for diversity preservation.

The general structure of NSGA-II is given in Table 1. An initial population *pop* of *ns* random solutions is built by the *first_pop* procedure and sorted by non-domination. The procedure *get_margins* computes for each solution *pop(i)* its crowding distance *margin(i)*. Then, each iteration of the main loop starts by calling *add_children*, to create *ns* children which are added at the end of *pop*. Finally, the resulting population with 2·*ns* solutions is reduced to a new population *newpop* by keeping the *ns* best solutions. To do this, fronts and margins must be updated using *non_dominated_sort* and *get_margins*. Starting from the front of level 1, complete fronts are then transferred to *newpop* as long as possible. The first front *front(i)* which could not be

Table 2 Parameters of CMUs

MTU	CMU	C_{ij}	ρ_{ij}	p_{ij}
1	A1	50	0.86	5
	A2	55	0.75	6
	A3	56	0.53	8
2	B1	18	0.56	6
	B2	21	0.80	8
	B3	32	0.45	5
	B4	27	0.35	4
	B5	34	0.68	6
3	C1	65	0.78	8
	C2	45	0.86	9
	C3	25	0.48	10
4	D1	54	0.57	12
	D2	48	0.69	10
5	E1	76	0.69	8
	E2	113	0.75	9
	E3	97	0.56	7
	E4	86	0.83	10
6	F1	23	0.75	6
	F2	35	0.86	8
	F3	31	0.91	10
	F4	20	0.53	9
	F5	18	0.43	7
7	G1	45	0.34	9
	G2	40	0.56	10
	G3	51	0.86	8
	G4	36	0.64	7
8	H1	37	0.53	8
	H2	42	0.76	7
	H3	40	0.68	9

Table 3 Linked cost between CMUs

	A1	A2	A3	C1	C2	C3	E1	E2	E3	E4	G1	G2	G3	G4
B1	34	73	45	25	54	76								
B2	56	87	102	34	67	89								
B3	23	97	46	97	45	23								
B4	23	28	101	34	55	30								
B5	56	47	89	46	45	65								
D1				26	31	35	67	78	70	32				
D2				45	56	37	56	34	27	34				
F1							50	45	39	60	70	34	25	66
F2							35	32	27	64	46	77	85	50
F3							78	68	32	67	56	78	45	43
F4							45	42	35	30	56	36	29	59
F5							35	64	23	64	60	76	35	61
H1											27	31	15	9
H2											67	104	12	46
H3											64	25	34	65

accommodated fully is truncated by keeping its most widely spread solutions. This is achieved by arranging its solutions in descending order of the crowding distance values, thanks to the *margin_sort* procedure, and by copying the best solutions until *newpop* contains exactly *ns* solutions. Finally, *pop* receives the contents of *newpop* for the next iteration of the GA.

3.3 NSGA-II approach for the CMC optimization

Our choice for the CMC optimization problem finally turns to NSGA-II approach because (i) its modular and flexible structure, (ii) the possibility of updating a single objective GA to multi-objective GA, and (iii) its successful applications to a wide range of problems.

The natural number encoding-based is adopted in the NSGA-II. A chromosome is an ordered list of CMUs, i.e., A chromosome is a CMC. Let $w=[w_1, w_2, ..., w_i, ..., w_n\ ]$ (i=1, 2,..., n), w_i is a gene of the chromosome, its value is between 1 and m_i (for MTU i, there are m_i CMUs to response).

Once a selection w fixes the CMUs for all MTUs, MTU scheduling can be done by the following procedure. The basic idea is to schedule all MTUs as late as possible, subject to time-sequence constraints. The steps are described as follows:

MTU scheduling procedure:

Step1: From MTU k=1 to n, calculate the initial starting time b_k and the completion time d_k:

$$b_k = \begin{cases} \max\{d_i, \forall(i,k) \in H\}, & \mathit{if}\,(i,k) \in H \\ 0 & \mathit{others} \end{cases}$$

and $d_k=b_k+p_k$; where p_k is the processing time of MTU k. For example, there are two MTU pairs (6, 8) and (7, 8) $\in H$ (See Fig. 1, Tables 2 and 5). The completion times of MTUs 6 and 7 all are 34, i.e., d6=d7=34. Since max{d6, d7} = d_6 = d_7 = b8, d8=b8+p8=34+7=41.

Step2: Calculate the tardy penalty cost from period t=0 to d_n by $P(t)=\beta[d_n - D]^+$, then return to NSGA-II algorithm.

4 Case study

In order to validate the proposed multi-objective optimization model and the NSGA-II approach for CMC, a task of manufacturing a complex part of plastic injector is introduced. The task consists of 8 MTUs, and the time-sequence relationship is shown in Fig. 1. After the dominant CMU won the task, it calls tenders for eight

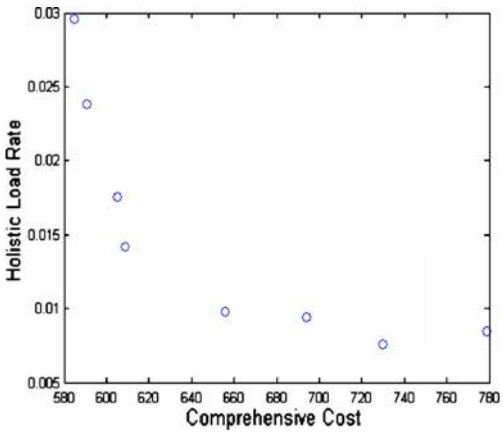

Fig. 3 Pareto solutions

Table 4 Corresponding results of Pareto optimal solutions

Pareto Solutions	Gene value of a chromosome								Objective functions		Completion time of d_8
	w 1	*w* 2	*w* 3	*w* 4	*w* 5	*w* 6	*w* 7	*w* 8	*Obj*_1	*Obj*_2	
1	1	4	2	1	4	4	3	1	585	0.02958	
ST	0	5	5	14	14	26	24	35			
CT	5	9	14	26	24	35	32	43			43
2	1	4	3	2	1	4	3	2	591	0.02382	
ST	0	5	5	15	15	25	23	34			
CT	5	9	15	25	23	34	31	41			41
3	3	1	1	2	3	5	3	1	605	0.01753	
ST	0	8	8	16	16	26	23	41			
CT	8	14	16	26	23	33	41	49			49
4	2	4	3	1	1	1	2	3	609	0.01415	
ST	0	6	6	16	16	28	24	34			
CT	6	10	16	28	24	34	34	43			43
5	1	3	3	1	4	5	1	2	656	0.00976	
ST	0	5	5	15	15	27	25	34			
CT	5	10	15	27	25	34	34	41			41
6	2	3	3	1	4	4	1	3	694	0.00939	
ST	0	6	6	16	16	28	26	37			
CT	6	11	16	28	26	37	35	46			46
7	2	4	3	1	3	3	1	2	694	0.00945	
ST	0	6	6	16	15	28	23	38			
CT	6	10	16	28	23	38	32	45			45
8	2	3	3	1	4	5	1	3	730	0.00762	
ST	0	6	6	16	16	28	26	35			
CT	6	11	16	28	26	35	35	44			44
9	2	3	3	1	2	4	1	3	779	0.00848	
ST	0	6	6	16	16	28	25	37			
CT	6	11	16	28	25	37	34	46			46

Note: *ST*: starting time; *CT*: completion time

MTUs. The numbers of qualified CMUs for all MTUs with the processing cost, production load rate, and processing time are shown in Table 2, and the linked cost among CMUs is shown in Table 3, its due date is 49. The tardiness penalty parameter is 0.85. In Table 2, the capital letters stand for the serial numbers of the MTU and each number means the serial number of the candidates responded to the MTU. For example, "E3" means the third candidate CMU bidding for MTU 5.

The algorithm was coded by MATLAB 7.0 and run at a Pentium 4, 2.99 GHz clock pulse with 512 MB memory. Appropriate parameter values were determined on the base of preliminary computational experiments. The final parameter settings were determined: population size=30; crossover rate=0.8, mutation rate=0.2, and number of generation=500. The multi-objective EA was executed 30 times for the problem with the same initial population. The results of each execution were stored in an auxiliary vector and at the end resulting Pareto optimal set that was taken as outcome illustrated in Fig. 3. The obtained solution is formed in Pareto frontier. The corresponding results are shown in Table 4. Figure 4 illustrated the comparisons of the normalized values of two objective functions and the final completion time of Pareto solutions.

The performance of the algorithm applied in this paper is compared with that of VEGA, NSGA, and NPGA. Each of the four algorithms was executed 30 times for the problem with the same initial population and the average is reported. In consideration of the size of data (there are tens of result set),

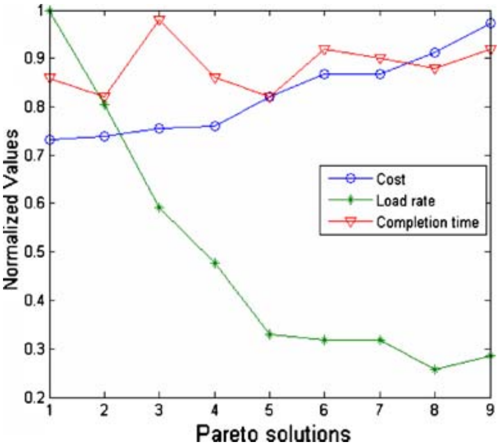

Fig. 4 Comparisons of Pareto solutions

Table 5 Number of Pareto optimal solutions that GA obtained (30 times)

Algorithms	Number of Pareto optimal solutions
Applied NSGA_II	210(87.5)
NSGA	197(82.1)
VEGA	153(63.75)
NPGA	189(78.8)

only the final statistical results are listed here. Table 5 illustrates the number of Pareto optimal solutions that each algorithm obtains. The numbers in parentheses are the percentages of Pareto optimal solutions that the corresponding algorithm has obtained. To summarize, the genetic algorithm applied in this paper, provides the best outcome.

Here the TOPSIS method proposed by Yoon and Hwang [16] is used in order to determine the best compromise solution among Pareto solutions. TOPSIS stands for technique for order preference by similarity to ideal solution, which is based upon the concept that the chosen alternative should have the shortest distance from the positive ideal solution (PIS) and the farthest from the negative ideal solution (NIS).

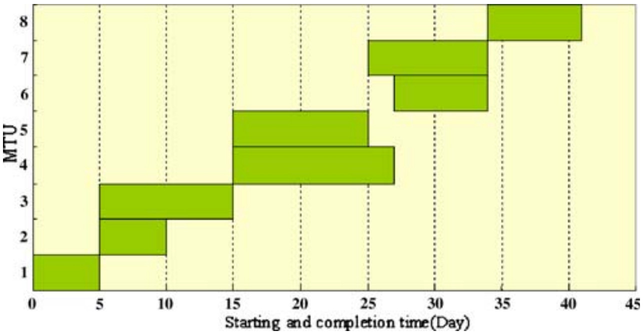

Fig. 5 Gantt chart of the best compromise solution

The decision matrix is A, and the normalized decision matrix is B. Then the PIS, NIS and the closeness coefficient C are obtained. Thus, the fifth solution is the best compromise solution among Pareto solutions (See Table 4). The chromosome of the best compromise solution is [1, 3, 3, 1, 4, 5, 1, 2] with 656 comprehensive cost and 0.00976 load rate. The corresponding starting and completion time are listed in Table 6. The completion time of the whole manufacturing task is 41. Figure 5 is the Gantt chart of the best compromise solution for completing the whole manufacturing task.

$$A = \begin{bmatrix} 585 & 0.02958 & 43 \\ 591 & 0.02382 & 41 \\ 605 & 0.01753 & 49 \\ 609 & 0.01415 & 43 \\ 656 & 0.00976 & 41 \\ 694 & 0.00939 & 46 \\ 694 & 0.00945 & 45 \\ 730 & 0.00762 & 44 \\ 779 & 0.00848 & 46 \end{bmatrix} \quad \text{Normalized decision matrix} \quad B = \begin{bmatrix} 0.0984 & 0.2279 & 0.1080 \\ 0.0994 & 0.1835 & 0.1030 \\ 0.1018 & 0.1350 & 0.1231 \\ 0.1025 & 0.1090 & 0.1080 \\ 0.1104 & 0.0752 & 0.1030 \\ 0.1167 & 0.0723 & 0.1156 \\ 0.1167 & 0.0728 & 0.1130 \\ 0.1228 & 0.0587 & 0.1106 \\ 0.1310 & 0.0653 & 0.1156 \end{bmatrix}$$

$$\mathrm{PIS} = (0.0984, 0.0587, 0.1030);\ \mathrm{NIS} = (0.1310, 0.2279, 0.1231);$$
$$\mathrm{C} = (0.2150, 0.4331, 0.5223, 0.7323, 0.8716, 0.7995, 0.8089, 0.8558, 0.7666).$$

Table 6 Best compromise solution

MTU i	CMU	ρ_{ij}	p_{ij}	b_i	d_i
1	A1	0.86	5	0	5
2	B3	0.45	5	5	10
3	C3	0.48	10	5	15
4	D1	0.57	12	15	27
5	E4	0.83	10	15	25
6	F5	0.43	7	27	34
7	G1	0.34	9	25	34
8	H2	0.76	7	34	41

5 Conclusions

The application of multi-objective optimization, which is based on non-dominated sorting genetic algorithm, increases the flexibility to select the optimal CMC for manufacturing complex parts in an networked manufacturing environment. Simultaneously considering the comprehensive cost and the whole production load, as optimization objectives, decision makers can choose the most adequate solution under the condition that the weights of multi-objectives are unknown.

The NSGA-II approach used was able to obtain a set of the Pareto optimal solutions. And TOPSIS approach is applied to identify the best compromise solution from the Pareto optimal solutions set. Performance of the NSGA-II approach is compared with that of three other genetic algorithms, and the results reveal that the genetic algorithm outperforms the others in this problem.

The results of this research are helpful for the manufacturers to quickly realize the manufacturing of complex parts by finding an appropriate CMC.

Acknowledgements The research reported in this article was supported by the Natural Science Foundation of Zhejiang Province, China (Grant No. Z604342). Also we would like to thank Professor X. F. Fan for critical reading of the manuscript.

References

1. Yang Y, Zhang J, Wan L, Chen L (2006) Internet-based collaborative product development chain for networked product development. Int J Adv Manuf Tech 28:845–853
2. Zhan HF, Lee WB, Cheung CF, Kwok SK, Gu XJ (2003) A web-based collaborative product design platform for dispersed network manufacturing. J Mater Process Tech 138:600–604
3. Fischer M, Jahn H, Teich T (2004) Optimizing the selection of partners in production networks. Robot Com-Int Manuf 20:593–601
4. Yao CF, Zhang DH, Peng WL, Bai K (2006) Research on resources optimisation deployment model and algorithm for collaborative manufacturing process. Int J Prod Res 44(16):3279–3301
5. Ip WH, Huang M, Yung KL, Wang D (2003) Genetic algorithm solution for a risk-based partner selection problem in a virtual enterprise. Comput Oper Res 30:213–231
6. Doerner K, Focke A, Gutjahr WJ (2007) Multicriteria tour planning for mobile healthcare facilities in a developing country. Eur J Oper Res 179:1078–1096
7. Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. in: Genetic Algorithms and their Applications, Proceedings of 1st International Conference on Genetic Algorithms, Lawrence Erlbaum, Hillsdale, NJ, 93–100
8. Li B, Chen L, Huang Z, Zhong Y (2006) Product configuration optimization using a multiobjective genetic algorithm. Int J Adv Manuf Technol 30:20–29
9. Kuriakose S, Shunmugam MS (2005) Multi-objective optimization of wire-electro discharge machining process by non-dominated sorting genetic algorithm. J Mater Process Tech 170:133–141
10. Ding H, Benyoucef L, Xie X (2006) A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization. Eng Appl Art if Intel 19:609–623
11. Altiparmak F, Gen M, Lin L, Paksoy T (2006) A genetic algorithm approach for multi-objective optimization of supply chain networks. Comput Ind Eng 51:197–216
12. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE T Evolut Comput 6 (2):182–197
13. Coello CAC (1999) A comprehensive survey of evolutionary-based multi-objective optimization techniques. Knowl Inform Syst: An International Journal 1(3):269–308
14. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, New York
15. Srinivas N, Deb K (1995) Multiobjective optimization using nondominated sorting in genetic algorithm. Evolut Comput 2(3): 221–248
16. Hwang CL, Yoon K (1981) Multiple attribute decision making methods and applications. Springer, Berlin Heidelberg