Journal of Industrial and Systems Engineering
Vol. 9, No. 4, pp 26-46
Autumn (November) 2016

Agent-based approach for cooperative scheduling

J. Behnamian[1]*

[1]*Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran*
Behnamian@basu.ac.ir

Abstract

This paper studies the multi-factory production (MFP) network scheduling problem where a number of different individual factories join together to form a MFP network, in which these factories can operate more economically than operating individually. However, in such network that is known as virtual production network with self-interested factories with transportation times, each individual factory usually focuses on self-benefits and does not care much about the others within the network. We first describe the realistic features which incorporate in problem definition. Then two different variants of the problem are considered. In the first case, we propose approximation algorithms with the best achievable theoretical guarantee in three cases: (*i*) all factories are interested in the makespan, (*ii*) all factories are interested in the sum of completion times, and (*iii*) the case in which among all factories, some factories are interested in the sum of completion times and the others are interested in the makespan. Furthermore, with considering the transportation, we model the problem as a mixed integer linear programming to minimize the makespan and total completion time and solve it by CPLEX solver to obtain Pareto solutions by applying modified ϵ -constraint approach. Experiments show that this procedure is capable of producing good results to approximate the efficient set.

Keywords: Distributed scheduling, approximation algorithms, □-constraint approach, pareto solutions, virtual corporation

1- Introduction

Scheduling problems are applied in various systems, since it is necessary to distribute the work among several entities. Formally, the scheduling problem as quoted in Pinedo (2008) is defined as "Scheduling concerns the allocation of limited resources to tasks over time. It is a decision-making process that has as a goal the optimization of one or more objectives." Since the first scheduling paper appeared in 1954, many variants of the basic scheduling problem have been formulated by differentiating between machine environments, side-constraints, and objective functions. Figure 1 shows the comparison between the single factory and virtual network production’s system with respect to three important factors to examine the production system’s capabilities.

*Corresponding author.

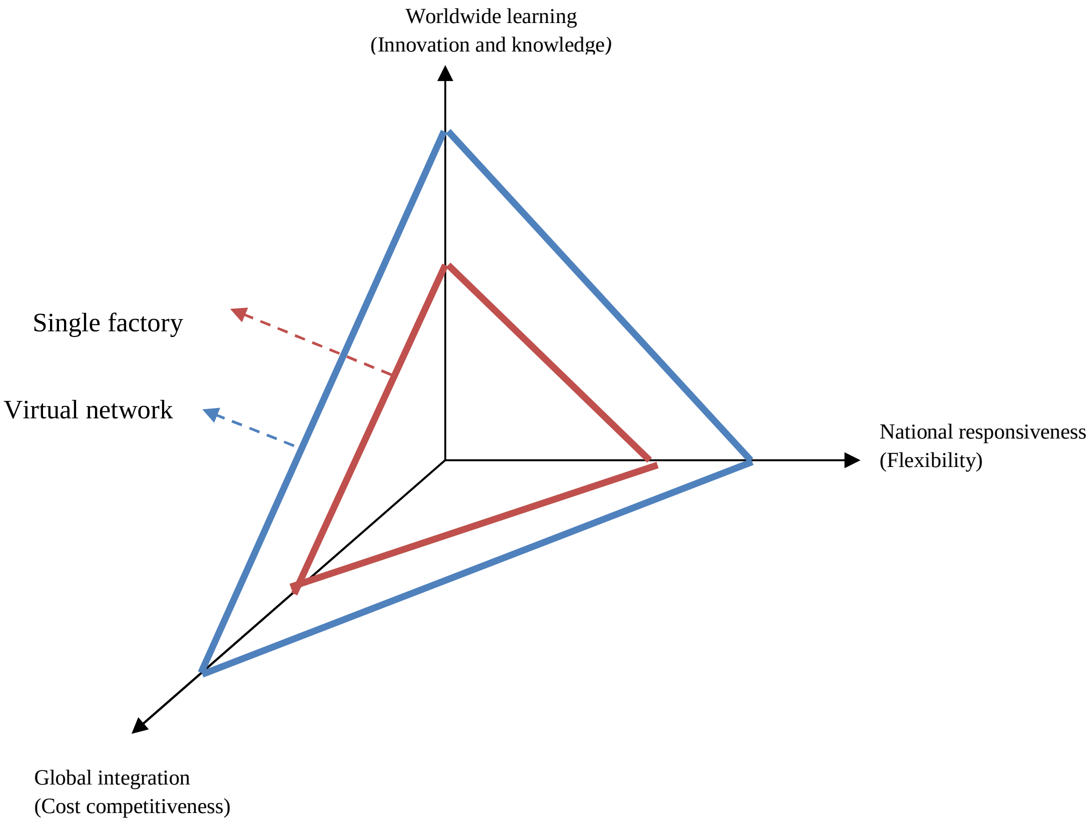


Fig. 1. Comparison between the single factory and virtual network production capabilities (Bartlett and Ghoshal (1998))

Most of the researches in production scheduling are only concerned with the optimization of a single factory. However, in the real world factories are interested to reduce their production costs and improve their production quality via participation in production activities with other professional factories distributed in different geographical places. As we see in Figure 1, many variations of collaboration are possible and factories will choose the specific type of partnership according to their individual needs.

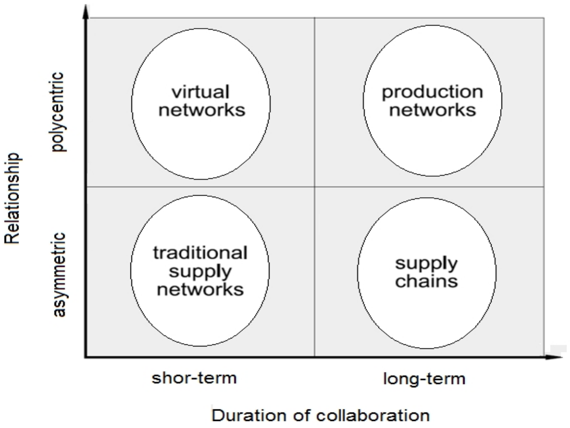


Fig. 2. Classification of cooperation concepts (Wiendahl et al. 1999)

Generally there are two ways to structure a collaborative manufacturing network which are defined as follows (Johansen et al. 2005):

- Vertical relations (or series relations), where the factory has a relationship with its suppliers or customers.
- Horizontal relations (or parallel relations), where the factory as a competitor produces similar goods on the same level.

Nowadays, many factories move away from vertical integration and are interested to perform horizontal cooperation with other factories. A further development in the continuing cooperative trend amongst manufacturing factories is the formation of production networks (Wiendahl et al. 1999). Lin et al. (2004) stated that in the today's industries, manufacturing generally involves the cooperation between several companies either formed as a virtual corporation or extended company. In fact, these networks establish a new type of relations between independent factories to manage and share the network production resources in parallel structure. Virtual production network (VPN) is a kind of the distributed system in which the smaller parts of the scheduling problem are solved by local decision makers who possibly have different objectives. This network establishes a new type of the horizontal collaboration and relations between independent factories and even competitors that establish occasional collaborations on the production order they cannot take on individually. After the order has been finished, factories as partners may go to their own ways again. VPN has four advantages (Vilana and Monroy 2009):
(*i*) improvement in operations, (*ii*) access to new markets in different geographical places,
(*iii*) diversification of the financial risks and (*iv*) access to new technologies.

Generally, one can say, the distributed activities, if managed properly, can reduce costs, increase flexibility and efficiency (Shi and Gregory 1998). Accordingly, the result is a highly flexible system characterized by low barriers to entry and exit, geographic flexibility, low costs, rapid technological diffusion, high diversification through contract manufacturers and exceptional economies of scale.

In such networks, some issues are important e.g. where to assign production so as to be must responsive, how to reduce transportation costs and how to reduce the complexity of network. As in Figure 1, in a production network, partners cooperate over a longer period since their integration is supposed to be more intensive but in VPNs, the relations are short-term corporations (Tuma 1998). It is worthwhile to notice that another characteristic distinguishing production network from virtual network is the availability of competence amongst partners in VPNs. Several industries such as aeronautics (Shi and Gregory 2003), electronics (Shi et al. 2005) or the automotive industry (Sturgeon and Florida 1999) encounter virtual networks.

The recent remarkable attention in distributed manufacturing management in both academia and the industry has demonstrated the significance of such system. During last years, many scholars have considered VPNs phenomenon from different perspectives (Monroy and Vilana Arto, 2010). Since in this respect the simultaneous scheduling of the factory in such production network has not been considered in any researches, the current paper considers the distributed scheduling (DS) for a virtual network in which several heterogeneous factories disperse geographically in different places and each of the network's factories has a parallel machines environment. In other words, in this problem, production processes performed in different factories are similar but the machines in different factories may have different speeds and different objective functions.

For this complicated production network, proposing an efficient scheduling technique is critical to react quickly to market changes. The difficulties faced in this problem are not only scheduling the jobs in a most favorable way, but also assigning the best factory to the jobs. Therefore, for generating the schedule in the distributed manufacturing environment, these two sub-problems must be simultaneously considered. Due to these reasons, scheduling in a multi-factory environment has become much more complex than traditional one. In this paper, we assumed that each factory in the production network has a parallel machine environment. Since the DS reduces to the regular scheduling problem if the number of factory (*F*) is one, and this latter problem as a classic parallel machine with makespan objective even with two machines is a NP-hard problem, we can easily conclude that the DS is also a NP-hard problem and consequently, no optimal polynomial time algorithms can be designed unless P=NP. Although there are many approximation algorithms which solve the parallel machine scheduling problem on single factory (Diedrich et al. 2010), there is not any approximation algorithm for multi-factory scheduling with parallel machines. In this paper, following Saule and Trystram (2009), we are interested to design the approximation algorithms with the best achievable approximation ratios. Furthermore, in this paper we propose a mathematical modeling. To the best of our knowledge, this paper presents for the first time a mathematical modeling and approximation algorithms for the multiple-factory production scheduling with self-interested heterogeneous factories. Because different interests often exist among self-interested factories, a multi-objective approach for the distributed scheduling is clearly requested. Therefore, to obtain Pareto solutions for the problem, we also make benefit of the ϵ-constraint technique.

The paper is organized as follows. Section 2 gives the literature review of some previous results. In Section 3, we present properties of the scheduling problem. Section 4 presents the framework of polynomial approximation algorithms in details. Mathematical modeling and its numerical results is given in Section 5. Finally, Section 6 concludes the paper with some remarks and recommends some future research areas.

2- Literature review

During the late 1970s' and the early1980s', the most of empirical researches reveal that not only the single factory needs to manage, but multi-factory scheduling should also be considered. However, the literature review indicates that the research during this period mainly was concerned with single factory scheduling.

In this respect and based on empirical research in microelectronic industry's production networks, Flaherty (1989) concluded that production network geographically has dispersed facilities and have shared common infrastructure mechanism which led it to a synergy advantage in the production. Shi and Gregory (1998) considered a relation of entities in multi-factory network as a matrix connection, where each node (i.e. factory) affects the other nodes and hence cannot be managed in isolation. Khurana and Talbot (1999) investigated that how each factory in the production network could influence each other in a complex structure. In order to increase key capability, Loeser (1999) proposed an approach for expansion of factories via cooperation with other factories. Li et al. (2000) were the first to name global manufacturing virtual networks by proposing a strategic positioning model for these factories based on three vectors: globalization, strategic alliances and value and supply chains. Fleury and Fleury (2003) give reasons that individual efficiency in today's new competitive market is not enough and it is necessary to be connected to the groups of firms which have efficient performance.

First studies about multi-plant scheduling started by the early 1980s. In the first study on joint scheduling of production and distribution in a complex network, Williams (1981) compared a dynamic programming based algorithm with several existing heuristics. The objective of the problem was to minimize average production and distribution costs. In that paper, the author assumes a constant demand rate and this limits the applicability of paper. Vercellis (1999) considered a capacitated master production planning and capacity allocation problem in a multi-plant chain in his/her mixed integer linear programming (MILP) model. Neiman and Lesser (1996) proposed a search algorithm namely repair process and the set of heuristics for job shop scheduling problem with cooperative distributed agents. In that research, the authors suppose that the agent has not a complete view of the resources availability, or of the jobs to be scheduled. The solving algorithm proposed in this paper considered all these issues in search. Sandholm (2000) studied the distributed scheduling problem and proposed a levelled commitment contracting protocol that helped to self-interested agents to adapt themselves with possible future events. Karatza (2001) presented a method which included a simulation-based modeling to examine machine performance under a variety of workloads in distributed scheduling. In that paper, the author scheduled a multi-factory production network in which half of the total machines had double the speed of others.

Jia et al. (2007) focused on distributed scheduling problems with the job shop environment. They used genetic algorithm (GA) to optimize different criteria. However, they solved several problems with single 6-job, 3-factory instances in which each factory has only four machines, corrected it and extended the modeling concept to the problem which delivery times were considered for jobs. More recently, Jia et al. (2002) refined and improved the previous GA in the same problem. Chan et al. (2007) proposed GA with dominant genes approach for distributed scheduling with the job shop environment in factories with respect to the criterion minimizing the deviation of solution obtained. By simulation runs, they also explored the significance of considering maintenance in that scheduling environment.

Based on the intelligent and agile characteristics of a multi-agents system and the optimistically capacity, Zhang et al. (2008) presented a multi-agents based model for distributed job shop scheduling problem. In that research, the authors proposed a distributed scheduling mechanism integrating multi-agents GA and dynamic scheduling strategies based on multi-agent negotiation. You and Grossmann (2008) formulated the multi-plant industrial networks scheduling as a mathematical modeling. Due to the difficulty of solving the mathematical model directly, they developed a new optimization method namely factory wide optimization to solve the combined production/distribution scheduling problem. Chung et al. (2009) studied the job shop distributed production networks scheduling problem for the case where perfect and imperfect maintenances are considered during the process of distributed scheduling. This study assumes that maintaining the reliability in an acceptable level is required for each entity in the network. They adapted the well-known GA approach to minimize the makespan of jobs. To minimize the maximum completion time, Naderi and Ruiz (2010) proposed six different mixed integer programming models for the distributed permutation flowshop scheduling problem. They also proposed several heuristics and a variable neighborhood descent for the problem. Zhang and Gen (2010) proposed a multi-objective GA approach to solve process planning and scheduling problems. The authors showed the efficacy of their proposed algorithm was better than non-dominated sorting GA II (NSGA-II).

3- Problem description

In this paper we are interested in classical settings, that is, all data in all problems used are known deterministically, when scheduling is undertaken; at any time, each job can be processed by at most one machine; each machine can process only one job; there are n independent jobs that are available at time 0; machines are available at all times if are not busy, with no breakdown; and a job once started on the machine must be completed on it without interruption. By $f = \{1, \dots F\}$ we denote the set of independent factories formed the production network, which have a specific local job cluster. Each factory f owns a cluster CL^f. Each factory has m^f identical parallel machine that all of them have speed v^f. The set of all jobs produced by factory f is denoted by n^f that can be own cluster or migrated job from other factories' cluster, with elements j_i^{fq}. By j_i^{fq} we denote the job belong to set cluster factory f that *executed* on factory q. If $f = q$, the job is executed *locally*, otherwise it is migrated and transportation time between factory f and q must be considered. If the factory f executed all local job, then $CL^f = n^f$. With this definition $n = \sum_{f=1}^{F} n^f = \sum_{f=1}^{F} CL^f$ is the number of jobs that must be scheduled on F factories. As mentioned above, factories in virtual networks plan to maximize their own profit; so we need to define two objective functions, i.e., makespan and total completion time that each factory's job cluster belongs to one of them. For this we define two symbols R1 and R2 that R1 contains all factories and their corresponding jobs which their objective function is total completion time and in similar definition, R2 indicates the factories' jobs which are interested in the makespan.

In this paper, the following basic factors must be taken into account when a production network is scheduled.

Heterogeneous factories. In a distributed manufacturing, the factories have two types of structure: parallel structure and series structure. In a parallel structure each factory may produce the finished products, but in series structure some intermediate factories may be supplied with other factories (Chung et al. 2009).

In the parallel structure, the factories can be homogeneous or non-homogeneous. In multi factory production network, each factory can be considered as an individual entity which has different efficiency and is subject to different constraints, for example, machine advances, worker cost, tax, close to suppliers, and transportation facilities, etc. In our problem, we assume that each factory has identical machines and different factories have different speeds (Sturgeon and Florida 1999). Systems of such multiple-factory manufacturing cells have many real-life applications, for example, it is common in the semiconductor manufacturing that the newer and modern machines have faster processing speeds in comparison with existing machines (Dessouky 1998). The majority of literature only has been dedicated to homogeneous factories. Since normally the real-world distributed systems are heterogeneous, we consider the case in which factories are non-homogeneous and each of them has specific machine's speed.

Self interested factories. Basically, there are two types of the distributed scheduling (Chan and Chung 2007). In the first type, to optimize the same objective function all factories which belong to the same factory, operate cooperatively. They may lose the individual's benefit to improve the global objective of network as a whole. In second one which is named a virtual network, some set of orders cannot be met, because there is not a factory of sufficient production potential, so some factories joint together and create a virtual production network in order to satisfy these production orders. Therefore, in the virtual network a number of different individual factories join together to form a multi-factory production network, in which these factories can operate more economically than operating individually. This network has a larger production capacity and many different resources. Therefore, it can meet many more production orders together than each of factories separately. In such a network, each individual factory usually focuses on self-benefits. In other words, it wants to maximize its own profit while not caring about the other factories' profits within the virtual network. In this paper, the second type of self-interested factories will be considered. In such cases, a factory is willing to accommodate other factories' jobs only for compensation.

Next two sections are devoted to two different cases. In the first case, we considered self-interested heterogonous factories and proposed several scheduling algorithms for different variants of problem with respect to the variety of self-interested factories' objective functions. In the last one, two groups of factories without caring about the individual objective of each factory in the same group are considered. In this case, for modeling and solving the new problem we assume that it is possible the jobs migrate from their original factory to other factories but a transportation time is incurred.

4- Case 1: Approximation results for production network without transportation

Assume the value of Z achieved in a given schedule σ with respect of criterion A is denoted by $Z_A(\sigma)$. Consider the problem of simultaneous minimizing the set $(Z^1, \dots, Z^F)$. Each solution σ has a corresponding value in the objective space $(Z_A^1(\sigma), \dots, Z_A^F(\sigma))$ with respect of criterion A. We say that a schedule σ is nondominated with criteria A and B, if there is no schedule $\bar{\sigma}$ such that $Z_A(\bar{\sigma}) \leq Z_A(\sigma)$, $Z_B(\bar{\sigma}) \leq Z_B(\sigma)$ and at least one of the two inequalities is strict (Agnetis et al. 2004). As mentioned earlier, all local single factory scheduling problems in production network are NP-hard and no optimal polynomial time algorithms can be designed. Thus, we are interested in polynomial approximation algorithms. In this section, we are interested to design the less achievable approximation ratios. Formally, a scheduling algorithm $algo$ has a performance ratio $(\rho^1, \dots, \rho^F)$ if for all valid schedules σ and for all factories F; $Z^f(algo) \leq \rho\, Z^f(\sigma)$ where Z^f is the objective of factory f.

Our main goal in this section is to propose algorithms with the best achievable theoretical guarantee in three cases: (i) all factories are interested in the makespan, (ii) all factories are interested in the sum of completion times, and (iii) the case in which F' factories are interested in the sum of completion times and $F'' = F - F'$ factories are interested in the makespan.

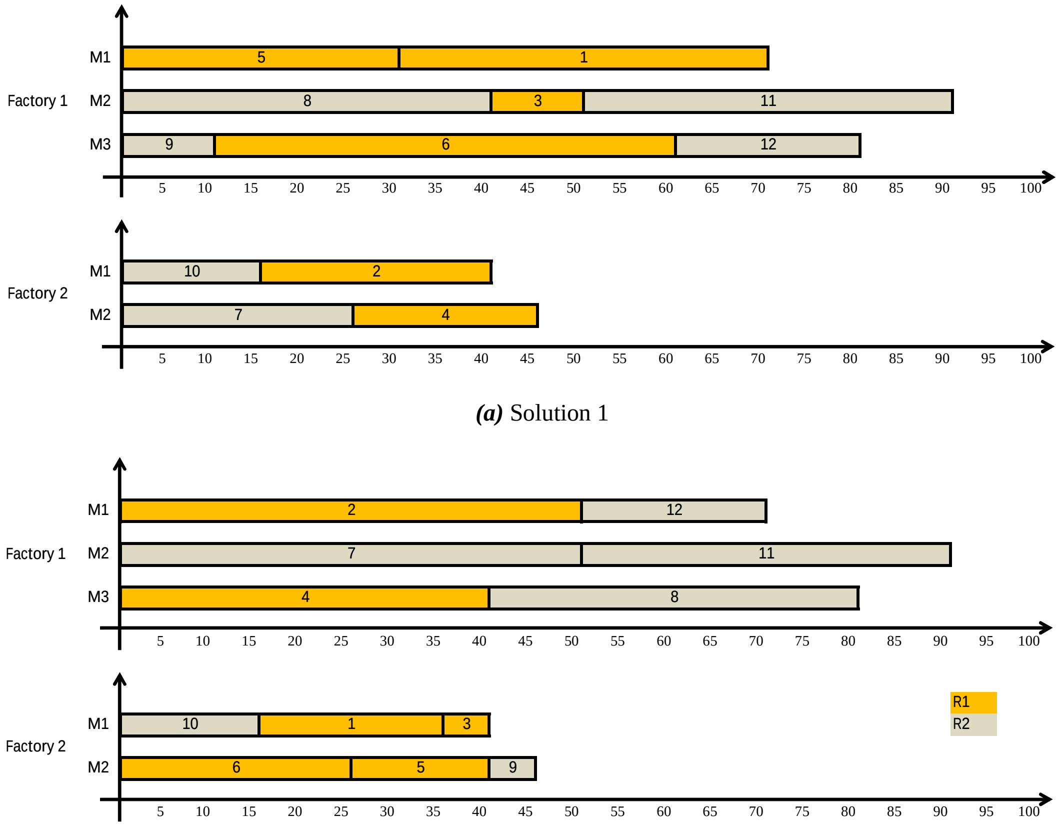
4-1- Numerical example

This sub-section considers two random generated examples with self-interested factories. With this example, we are going to show how, even if only one objective is concerned, a non-dominated solution will be obtained. Table 1 shows the processing time of 12 jobs on 2 factories in two different regions R1 and R2. Assume that the machine's speeds in factories F1 and F2 are 1 and 2, respectively.

Table 1. Processing time of jobs in factories 1 and 2

Processing times	Jobs of R1						Jobs of R2					
	1	2	3	4	5	6	7	8	9	10	11	12
p_i^1	40	50	10	40	30	50	50	40	10	30	40	20
p_i^2	20	25	5	20	15	25	25	20	5	15	20	10

Figure 3 shows the resultant Gantt chart for two examples.



(a) Solution 1

(b) Solution 2

Fig. 3. The Gantt chart of two numerical examples with two selfish factories

The objective value for each *R* of two solutions is shown in Table 2.

Table 2. Numerical example with two competing factories

Solution	*R*=1 ($C_{\max}$)	*R*=2 ($C_{\max}$)	*R*=1 ($\sum C$)	*R*=2 ($\sum C$)	*R*=1 ($\sum C$)	*R*=2 ($C_{\max}$)
1	70	90	235	260	235	90
2	50	90	230	350	230	90

As can be seen in this table, in the case which all factories have makespan objective, the solution 2 dominated the solution 1. We also see when two factories have total completion time (TCT) objective, both solutions are nondominated schedules and finally in case 3 which factory 1 is interested to TCT but factory 2 is interested to minimize makespan, solution 2 is nondominated solution.

4-2- Approximation algorithm

Before the approximation bounds are introduced, two useful propositions are proposed on the single machine problem. They will serve to compute approximability bounds and basically construct the concept that all known bounds on the best approximation ratios for the single factory case are valid for multi factory production network.

Let M denote the sum of machines in all factories and p_i denote the processing time of job *i*. We also show the completion time of job *i* in factory *f* with C_i^f and consequently, we have $C_{\max} = \max_{\substack{i\in\{1,\dots,n\},\\ f\in\{1,\dots,F\}}} C_i^f$.

Proposition 1. The distributed scheduling of the self-interested entities with makespan objective function cannot be approximated with a performance ratio less than $(1, 2, \dots, F)$ on $M = 1$ with speed v.

Proof. In this case, all factories are interested in the makespan. Consider the instance that each factory has only one job and $M = 1$ with speed v is available for scheduling $n = F$ jobs. All jobs have $p_i = 1$. Obviously, for all jobs, the best makespan of each factory that can be achieved is $\frac{1}{v}$. For any permutation with any efficient schedule, one factory will have a makespan of $\frac{1}{v}$, another one will have a makespan of $\frac{2}{v}$, and so on. Thus, it is impossible to obtain an algorithm that guarantees a ratio better than $(1, 2, \dots, F)$.

Now consider the instance where at least one factory (factory q) has a job with $p_i > 1$. In such an instance, if factory q is the first factory in permutation which gets a performance ratio of 1, all other factories will get a performance ratio worse than F. ¨

Proposition 2. The distributed scheduling of the self-interested entities with sum of completion times of unitary jobs cannot be approximated with a ratio less than $(F, \dots, F)$ on a $M = 1$ with speed v.

Proof. Let us consider that each factory owns $n^f = r$ jobs with $p_i = 1$ that must be scheduled on $M = 1$ with speed v. The optimal value for each factory f is $\sum C_i^{f^*} = \sum_{i=1}^{r} i/v = \frac{r(r+1)}{2v}$. The sum of completion times of all jobs is the sum of objective values of all factories which can be calculated by the following equation.

$$\sum C_i^* = \frac{Fr(Fr+1)}{2v}$$

All factories have the same priority for their scheduling, thus no factory should obtain a better objective value than the others. Let us consider that factory f obtains the Fth of the sum of completion times over all jobs, so we have $\sum C_i^f = \frac{\sum C_i^*}{F} = \frac{\frac{Fr(Fr+1)}{2v}}{F} = \frac{r(Fr+1)}{2v}$. We can now compute the minimum approximation ratio obtained by ensuring equity between the factories: $\frac{\sum C_i^f}{\sum C_i^{f^*}} = \frac{r(Fr+1)/2v}{r(r+1)/2v}$ which is equal to $= F - \frac{F-1}{r+1}$. Now if $r \to \infty$ then ratio tends to F. ¨

4-2-1- Competitive factories with $C_{\max}$ objective

As can be seen in the previous sub-section, for the case (i) in which all factories are interested in the makespan, no algorithm can guarantee a constant performance ratio better than $(1, 2, \dots, F)$ for scheduling jobs on the single machine. The following algorithm is proposed for multi-factory scheduling on $M \geq 1$ machines in which all factories are self-interested and have C_{max} objective. In Algorithm 1 we use ρ-approximation algorithm such as LPT (largest processing time) rule with $\rho = \frac{4}{3}$ for the single factory case.

Algorithm 1: *VDN*(C_{max}) algorithm

```
Given a $\rho$-approximation algorithm for the single factory on $M$ machines,
For each factory $f$, compute schedule in factory $f$ $(\sigma^{f})$ such as $C_{max}^{f}(\sigma^{f})$,
Group jobs of each factory $f$ according to $\sigma^{f}$,
Schedule jobs of factory $f$ according to $\sigma^{f}$ between $\left[\sum_{f'<f} C_{max}^{f'}\left(\sigma^{f'}\right), \sum_{f'\leq f} C_{max}^{f'}\left(\sigma^{f'}\right)\right]$.
```

Theorem 1. Algorithm *VDN*(C_{max}) is a $(\rho, 2\rho, \dots, F\rho)$-approximation algorithm of the distributed scheduling with self-interested factories with an unknown order that are interested in the makespan.

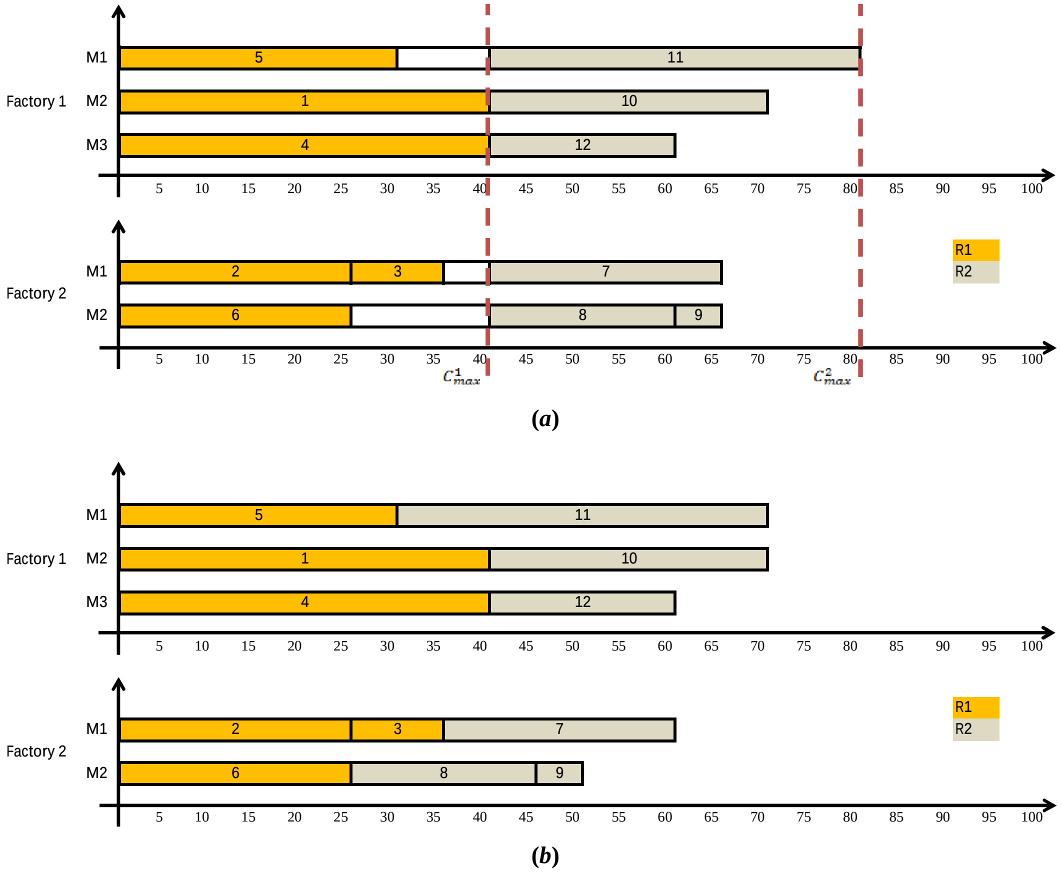
Proof. For this at first the validity of the schedule must be checked. In other words, it must be verified that the jobs of each factory f are scheduled in the disjoint intervals of length $C_{max}^{f}\left(\sigma^{f}\right)$ according to σ^{f}.

Now the theorem can proved as follows.

$$C_{max}^{f} = \sum_{f'\leq f} C_{max}^{f'}\left(\sigma^{f'}\right).$$

If the factories are ordered by increasing values of $C_{max}^{f}\left(\sigma^{f}\right)$, inequality $C_{max}^{f} \leq f\ C_{max}^{f}(\sigma^{f})$ holds true. Moreover, if we assume that σ^{f} is generated by a ρ-approximation algorithm, we have $C_{max}^{f} \leq f\rho$ and Theorem is proved. ¨

Note that if LPT rule is used, the performance ratio of *VDN*(C_{max}) algorithm is $\left(\frac{4}{3}, \frac{8}{3}, \dots, \frac{4F}{3}\right)$. Following Gantt chart shows the application of *VDN*(C_{max}) algorithm with some modification for the numerical example introduced in sub-section 4.1. In modified LPT, in each stage we assign the job with the largest processing time among the unscheduled to the fastest available factory in which the machine's speed in factories F1, F2 are 1 and 2, respectively (see Figure 4).



(a)

(b)

Fig. 4. The Gantt chart of the solution produced by applying *VDN*(C_{max}) algorithm

4-2-2- Competitive factories with TCT objective

A single factory problem with $M = 1$ can be solved in polynomial time by SPT rule. However, for more than one factory, this problem is NP-hard. As we see in *Proposition 2*, no algorithm can guarantee a constant ratio better than $(F, \dots, F)$ for the single machine case. For general form with M machines with TCT objective, the basic proposed steps are designed as shown in Algorithm 2.

Algorithm 2: *VDN* $(\sum C)$ algorithm

1: Let $\sigma^1, \sigma^2, \dots, \sigma^F$ be F modified SPT schedules on M machines. In this modification, we assumed that in each assigning, among unscheduled jobs the job with the smallest processing time is assigned to the available machine with the smallest speed.

2: Each schedule σ^f can be seen as M independent schedules denoted $\sigma_1^f, \sigma_2^f, \dots, \sigma_M^f$.

3: For each machine j, schedules σ_j^f of each factory f are mixed by the algorithm *greedy algorithm* into σ_j. In this step, we construct the schedule σ namely the *greedy algorithm* (GR) through sorting jobs in the non-decreasing order of $C_i^f(\sigma^f)$.

4: A global schedule on M machines is constructed by executing σ_j on the machine j $(1 \le j \le M)$.

In the modified SPT algorithm, in each factory we check the assignment the job to the machine which is available and compare the obtained results for all factories and the solution with smallest objective value is chosen.

Theorem 2. *VDN* ($\sum C$) Algorithm is a $(F, \dots, F)$-approximation algorithm for the case that all factories are interested in the sum of completion times.

Proof. At first, we analyze the single machine case before the general case with M machines is proved. In *single machine* case initially we compute the final schedule of all factories' jobs from the schedules of each of factory's jobs. To do this, we use $\sigma^1, \sigma^2, \dots, \sigma^F$ notations to show the independent schedules for the jobs of each factory. We construct the schedule σ namely *greedy algorithm* (GR) through sorting jobs in nondecreasing order of $C_i^f(\sigma^f)$.

In schedule σ by applying the GR algorithm on single machine, inequality $C_i^f(\sigma) \leq FC_i^f(\sigma^f)$ holds true for all f and $i \leq n^f$. Let us consider job J_i^f. On single machine, inequality $C_i^f(\sigma) \leq C_{i'}^{f'}(\sigma)$ implies $C_i^f(\sigma^f) \leq C_{i'}^{f'}(\sigma^{f'})$. Thus, for each factory f', the set of job $J^{f'}$ is scheduled before $C_i^f(\sigma)$ in σ such that $\sum_{J_{i'}^{f'} \in J^{f'}} p_{i'}^{f'} \leq C_i^f(\sigma^f)$. For each factory, there is one such set. J_i^f is completed when all jobs of $\cup_{f'} J^{f'}$ are completed. Thus, $C_i^f(\sigma) = \sum_{f'} \sum_{J_{i'}^{f'} \in J^{f'}} p_{i'}^{f'} \leq FC_i^f(\sigma^f)$.

Using *GR*, it is possible to mix F schedules and ensure that the completion time of each job in each schedule is not degraded by a factor greater than F. In particular, if a schedule is a ρ-approximation on the sum of completion times, then by mixing it with the F - 1 other schedules, the final schedule is a $(F\rho)$-approximation for this factory. In the case that all factories are interested in the sum of completion times with $M = 1$, algorithm GR($SPT^1 \dots, SPT^F$) is a $(F, \dots, F)$-approximation algorithm.

With this reasoning for the single machine case and with paying attention to the independency of machines in M parallel machine case, the Theorem 2 is proved. ¨

Figure 5 shows Gantt chart of applying *VDN* ($\sum C$) algorithm for the numerical example introduced in sub-section 4.1.

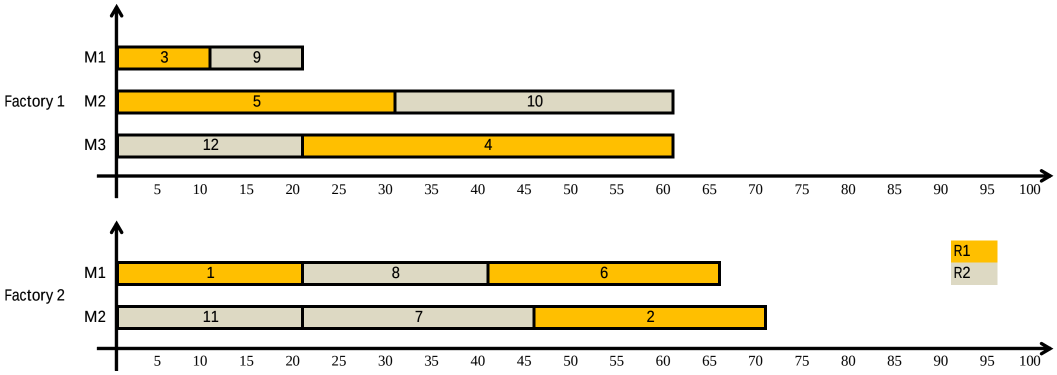


Fig. 5. The Gantt chart of the solution produced by applying *VDN* ($\sum C$) algorithm

4-2-3- Competitive factories with C$_{max}$ or TCT objective

In this case among F factories, F' factories are interested in the sum of completion times and $F'' = F - F'$ factories are interested in the makespan. The idea in this sub-section is to incorporate techniques presented two cases (*i*) and (*ii*) into a single scheduling algorithm called *VDN* ($\sum C$:C_{max}). Consider the sub-instance of F'' factories interested in the makespan. Let σ^{cmax} be the schedule generated by *VDN* (C_{max}) on this sub-instance. Algorithm 3 shows *VDN* ($\sum C$:C_{max}) procedure.

Algorithm 3: *VDN* (*∑C:C_{max}*) algorithm

1: Build a schedule $\sigma^{C_{max}}$ for all factories' makespan applying the only *VDN*(C_{max}) algorithm.
2: For each factory f interested in the sum of completion times, let σ^f be a modified SPT schedule.
3: We modify GR algorithm in which each factory is assumed to have a same priority. We can get another algorithm which the factories do not have the same priority assumption. Let $\pi^1, \pi^2, \ldots, \pi^F$ be real positive values such that $\sum_f \pi^f = 1$. We build schedule σ^π using the modified greedy algorithm (GRπ) in which the jobs are scheduled in nondecreasing order of $\frac{C_i^f(\sigma^f)}{\pi^f}$ and whenever $\forall f, \pi^f = \frac{1}{F}$ then $\sigma = \sigma^\pi$. Now in this step apply *GRπ* for the factories which are interested in the sum of completion times, for which $\pi_{cmax} = \frac{F^{"}}{F}$ and $\pi^f = \frac{1}{F}$.

Theorem 3. *VDN* (*∑C:C_{max}*) algorithm is a $(F, \ldots F, \frac{F}{F^{"}}\rho, \frac{2F}{F^{"}}\rho, \ldots., F\rho)$-approximation algorithm for the case in which F' factories are interested in the sum of completion times and $F'' = F - F'$ factories are interested in the makespan.

Proof. For the proof, we assumed that all factories interested in the sum of completion times get a performance ratio of F. The corresponding proof is similar to the one of *Theorem 2* and it is omitted.

Consider now factory f interested in the makespan. Without loss of generality, we have $C_{max}^f(\sigma^{cmax}) \leq f\rho C_{max}^{f\;*}$ (from Theorem 1) which means that $\forall i \leq n^f$, $C_i^f(\sigma^{cmax}) \leq f\rho C_{max}^{f\;*}$.

In GRπ algorithm the completion times in σ^π of jobs belonging to factory f are not degraded by a factor greater than $\frac{1}{\pi^f}$. Applying the reasoning similar to which proposed in Theorem 2, it is straightforward to prove that using GRπ algorithm, in schedule σ^π with $M = 1$, inequality $C_i^f(\sigma) \leq \frac{C_i^f(\sigma^f)}{\pi^f}$ for all f and $i \leq n^f$ holds true. Due to this statement, all jobs of the factories interested in $C_{\max}$ are not reduced by a factor greater than $\forall i \leq n^f$, $C_i^f(\sigma^{cmax}) \leq \frac{F}{F^{"}} f\rho C_{max}^{f\;*}$ which concludes the proof. ¨

5- Case 2: Mathematical modeling for production network with transportation

In this section, we add another realistic assumption to the problem. As mentioned earlier, in the classic scheduling problems set of orders must be scheduled on machines in a single factory. However, the production processes within many industrial factories are distributed over the several manufacturing factories. In non-cooperative solution, all factories are responsible for the production of their local cluster. In such system a schedule should give enough flexibility to a local scheduler. This can be attained by transporting the jobs from the overloaded machine to the machine which has fewer workloads. Nevertheless, a certain price must be paid in order to produce solutions in which all factories have an incentive to participate. Therefore, considering the transportation time from one factory to another is important in the scheduling process.

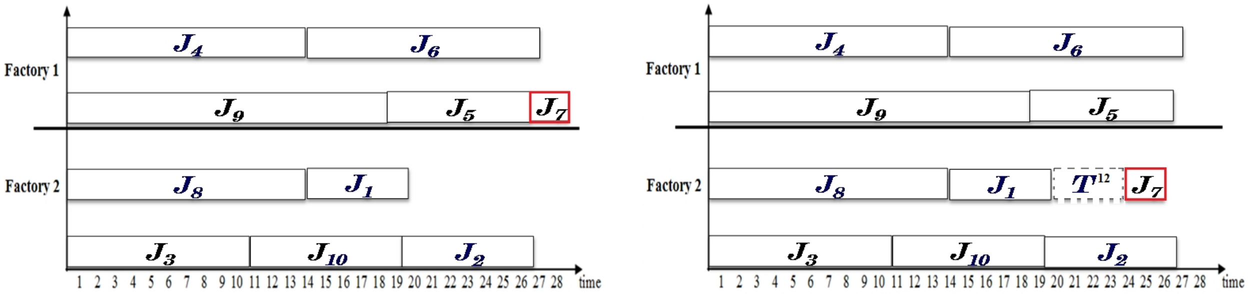
In considering the transportation time in modeling the scheduling problem some assumptions must be regarded which are as follows:

- It is assumed that there are transportation routes among the factories and number of the transporters is an infinitive with the same quantity, capacity, time and cost of the drive.
- Due to the distribution of factories and the need for coordinated scheduling, communication between the entities of network is necessary,
- The main target activity of these transporters is transferring a job form the initial factory which is ordered on it to process on another factory with the objective of solution quality improvement.

The sum of the transportation times to carry the job i form the factory f to factory q and returning it into the primary factory's warehouse (to delivered to local customers) is denoted by $T^{\,fq}$, which is assumed to be independent of the jobs being transported. We also suppose that loading and unloading times are not considered separately and included in transportation times.

It is important to notice that for considering the transportation among network's entities, an information exchange infrastructure between members is required to coordinate between global scheduling and local scheduling. Obviously, the partners' flexibility can be improved if each factory is informed about the loading and availability of manufacturing resources at various partners. Albeit, this information should be exchanged as early as possible in order to help the network's factories to plan more accurately and to adjust their capacities more effectively (Karatza 2001).

In Figure 6 with the simple numerical example, we show that how jobs' transportation can be considered in the main body of scheduling and reduce the global objective function.



(***a***) Without transportation

(***b***) After transportation

Fig. 6. The Gantt chart of the solution produced by considering transportation time

In the above figure, we can see that after the jobs are locally scheduled, at the time t=19 the first machine in Factory 2 is idle while the second machine in Factory 1 is overloaded. Absolutely, in ideal distributed structure, the system is preferable that a balanced use of machines can be achieved and the overloading is avoided. After detailed information is exchanged between two factories, the workload is uniformly distributed over all of the machines by transporting J_7 from Factory 1 to Factory 2. As seen in Figure 6, the global objective function is improved in comparison with the first scheduling.

This simple example makes sense that transportation is an integrated feature in multi-factory scheduling systems and considering this assumption makes the scheduling problems more practical.

5-1- MILP model

Mathematical models are the natural starting point for detailed problem characterization (Granot et al. 1997). Before the model is presented, we introduce the parameters and indices employed. They are defined as follows.

i, j, k index of jobs; $i, j, k \in \{1, 2, \ ..., \ n\}$

f, q index of factories; $f, q \in \{1, 2, ..., F\}$

p_i^{fq} modified processing time of job i in factory q which orginally is ordered to factory f

t^{fq} the transportation time to carry the jobs from factory f to factory q where $t^{fq} = T^{fq}/2$

L large positive number

$$x_{ij}^f = \begin{cases} 1 & \text{if job } i \text{ is scheduled immediately before job } j \text{ in factory } f, \\ 0 & \text{otherwise.} \end{cases}$$

$$y_i^f = \begin{cases} 1 & \text{if job } i \text{ is assigned to factory } f, \\ 0 & \text{otherwise.} \end{cases}$$

$$w_i^f = \begin{cases} 1 & \text{if job } i \text{ is orginally ordered to factory } f, \\ 0 & \text{otherwise.} \end{cases}$$

We introduced the dummy jobs 0 and n + 1. The processing time of these jobs is set at 0 and for all f, $C_0^f = 0$.

The problem can now be formulated as:

$$Z = \text{Min } (\sum_{j \in \text{R1}} C_j^f ,\ C_{\max}) \tag{1}$$

$$s.t. \quad \sum_{f=1}^{F} y_i^f = 1, \quad i = 1, 2, \ldots, n, \tag{2}$$

$$\sum_{\substack{i=0, \\ i \neq j}}^{n} \sum_{f=1}^{F} x_{ij}^f = 1, \quad j = 1, 2, \ldots, n, \tag{3}$$

$$\sum_{j=1}^{n} x_{0j}^f = m^f , \quad f = 1, 2, \ldots, F, \tag{4}$$

$$\sum_{f=1}^{F} x_{0j}^f \leq 1, \quad j = 1, 2, \ldots, n, \tag{5}$$

$$\sum_{\substack{i=0, \\ i \neq j}}^{n} x_{ij}^f + \sum_{\substack{k=1, \\ k \neq j}}^{n+1} x_{jk}^f = 2.y_j^f , \quad j = 1, 2, \ldots, n,\ f = 1, 2, \ldots, F, \tag{6}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} \sum_{f=1}^{F} x_{ij}^f \leq 1, \quad i = 1, 2, \ldots, n, \tag{7}$$

$$\sum_{f=1}^{F} (x_{ij}^f + x_{ji}^f) \leq 1, \quad i = 1, 2, \ldots, n-1, \quad j > i, \tag{8}$$

$$p_i^{fq} = w_i^f \, ((\frac{p_i}{v^q}) + 2t^{fq}) , \quad i = 1, 2, \ldots, n, \quad f, q = 1, 2, \ldots, F, \tag{9}$$

$$C_j^f - C_i^f \geq p_j^{fq} - L(1 - x_{ij}^f), \quad i, j = 0, 1, \ldots, n,\ i \neq j,\ f = 1, 2, \ldots, F, \tag{10}$$

$$C_{\max}^f \geq C_j^f ,\ j \in R2,\ j = 1, 2, \ldots, n,\ f = 1, 2, \ldots, F, \tag{11}$$

$$C_{\max} \geq C_{\max}^f ,\ f = 1, 2, \ldots, F, \tag{12}$$

$$x_{ij}^f , y_i^f \in \{0,1\}, \quad i, j = 0, 1, \ldots, n+1, \quad i \neq j, \quad f = 1, 2, \ldots, F, \tag{13}$$

$$C_i^f \geq 0,\ i = 1, 2, \ldots, n, \quad f = 1, 2, \ldots, F. \tag{14}$$

In the proposed mathematical model, Relation (1) is the objective function. Relation (2) indicates that job i requires only one factory. Constraint (3) states that job i in the factory f requires only one machine. Relation (4) ensures that dummy job 0 has exactly one successor on each machine in the factory. Constraint (5) guarantees that job j can be assigned at most to the first position of each factory. Constraint (6) controls that every job can be either a successor or predecessor on each machine in the factory to which it is assigned. Constraint (7) shows that every job has at most one succeeding job. Constraint set (8) states that a job cannot be at the same time both a predecessor and a successor of another job. Relation (9) modifies processing time for the job i according to the distance between original factory of job which belongs to it and factory that job finally processed on it and speed of machines in this factory. Constraint (10) establishes the relationship between the completion times of jobs i and j that are assigned to the same machine in the specific factory. Constraint sets (11) and (12) define the makespan of job which belong to R2. Constraints (13) and (14) define zero-one variables and nonnegative variables, respectively. Note that the value of C_i^f is zero when the job i is not assigned to factory f.

5-2- Solving MILP via ϵ -constraint

5-2-1- Improved approach

We point out that, to the best knowledge of the authors, there does not exist any methodology in the literature to obtain Pareto solutions for the problem modeled in previous sub-section. In order to obtain Pareto solutions we make use of ϵ -constraint approach**.** This approach, as shows in the following, chooses one objective as a main objective function of the model and all other objective functions put in constraints with their upper bounds.

$$\begin{aligned} &\min \quad f_r(x) \\ &\text{s.t.} \quad X \in S \\ &\qquad f_t(x) \le ò_t \quad \text{for all } \; t \neq r \end{aligned} \qquad (15)$$

The following results can be found in Miettinen (1999).

Theorem 4. A solution of the $\epsilon -$constraint problem is weakly Pareto optimal for multi-objective optimization (MOP).

Theorem 5. A feasible point x^* of the $\epsilon -$constraint problem is Pareto optimal if and only if it is a solution of MOP for every objective function, where $\epsilon_i = f_i(x^*)$ for $i = 1, 2, \ldots, k$ and $i \neq j$.

According to the Mavrotas (2009) with some modifications the results of this approach can improve. This modification in the problem with p objective function is as follows.

$$\begin{aligned} &\min \quad f_r(x) + eps(\sum_{\substack{t=1, \\ t\neq r}}^{p} s_t) \\ &\text{s.t.} \quad X \in S \\ &\qquad f_t(x) + s_t = ò_t \quad \text{for all } \; t \neq r \end{aligned} \qquad (16)$$

where *eps* is a small number between 10^{-3} and 10^{-6} (Mavrotas 2009).

Theorem 6. A solution of the improved ϵ -constraint is only efficient solutions and the generation of weakly efficient solutions is avoided.

Proof. See Mavrotas (2009).

5-2-2- Enumerating the solution

Given the job sets J_A and J_B of the two factories, the two objective functions $Z_A(.)$ and $Z_B(.)$, and an integer Q, find a schedule σ^*such that $Z_B(\sigma^*) \le Q$, and $Z_A(\sigma^*)$ is minimum. This methodology is known as ϵ -constraint approach in bicriteria optimization (T'Kindt and Billaut 2006).

Applying this new mathematical modeling, in a new formulation, the objective function calculates the maximum completion time of only the jobs in R2. The optimal solution to the formulation minimizes the $\sum C_j$ for jobs in R2 but simultaneously ensures that the jobs in R1 do not exceed the pre-fixed makespan value of $C_{\max}$ (Q).

In mathematical modeling, the objective function can now be formulated as:

$$\text{Min } Z = \sum_{j \in \text{R1}} C_j^f , \qquad (17)$$

And constraint $C_{\max} \le Q,$ is added to the constraint sets.

Now if Q is too small, an instance of problem may not have feasible solutions. If there is at least one feasible solution, we say that the instance is *feasible*. Note that the problem of finding, among optimal schedules, one scheduling which is also nondominated can be always addressed by *binary search* (Agnetis et al. 2004). Let σ^*be the optimal solution to $(Z_A\colon Z_B \leq \bar{Q})$. We can solve $(Z_A\colon Z_B \leq \bar{Q}/2)$, obtaining a solution σ'. If $Z_A(\sigma') > Z_A(\sigma^*)$, $\bar{Q}/2$ is indeed too small, so we try next $3\bar{Q}/4$, or else we decrease Q again to $\bar{Q}/4$. This goes on until we individuate the smallest value Q^* of Q such that the value of the optimal solution to $(Z_A\colon Z_B \leq Q)$ is still equal to $Z_A(\sigma^*)$. Following Agnetis et al. (2004) Algorithm 4 shows the scheme of algorithm for enumerating the nondominated schedules via ϵ -constraint approach.

```
Algorithm 4: Enumerating the nondominated schedules via $\epsilon$-constraint approach
  Start
    Step 1: Initialize: $S:=\emptyset$; $Q:=+\infty$; $i:=0$;
    Step 2: generate the set of non-dominated points
    {
    While the schedule is feasible do
    $i:=i+1$;
    $\sigma^i$=Solve the scheduling problem for single objective with $Z_B(\sigma^*) \leq Q$ by CPLEX;
    $S:=S \cup \sigma^i$;
    $Q':=Z_B(\sigma^i)$;
    $Q:=Q'-\epsilon$;
    End while
    }
    Step 3. Output optimization results.
  End
```

This is, however, a computationally intensive procedure and is feasible only for small-sized problem instances. We test the MILP models using CPLEX solver and consider two instances: 10-job with 2-factory problem and 25-job with 4-factory. The processing times for each instance are generated using a discrete uniform distribution from 30 to 60. Each instance is run several times with considering differnt Q in constraint $C_{\max} \leq Q$, according to *Algorithm 4* and the results of two objectives are reported in Table 3, separately.

Table 3. Solutions obtained using ϵ-constraint approach for two instances

Solution's number	*Instance 1*			*Instance 2*		
	n=10, F=2, m^f=(2,3) *R1=F1 & R2=F2*			*n=25, F=4, m^f=(2,2,2,2)* *R1=F1,F2 & R2=F3,F4*		
	C_{max}	*TCT*	**RT (s)**	C_{max}	*TCT*	**RT (s)**
1	75	inf	50	170	ins	100
2	80	427.273	300	180	2396.950	1000
3	85	427.273	300	190	2325.430	1000
4	90	427.273	300	200	2322.213	1000
5	91	427.273	300	250	2232.387	1000
6	93	407.273	300	300	2226.886	1000
7	94	360.000	300	350	2135.380	1000
8	95	358.182	300	400	2033.489	1000
9	100	357.273	300	450	2018.700	1000
10	105	357.273	300	500	1911.443	1000
11	110	354.000	300	550	1906.997	1000
12	115	352.727	300	600	1898.041	1000
13	116	347.273	300	650	1869.073	1000
14	119	339.818	300	700	1816.609	1000
15	120	335.636	300	750	1815.421	1000
16	121	331.091	300	800	1814.316	1000
17	125	331.091	300	850	1805.183	1000
18	126	331.091	300	900	1803.323	1000
19	127	331.091	300	950	1793.102	1000
20	128	331.091	300	1000	1752.509	1000
21	130	331.091	300	1050	1712.509	1000
22	135	331.091	300	1100	1712.509	1000
23	140	311.152	300	1150	1712.509	1000
24	145	310.424	300	1200	1712.509	1000
25	170	310.424	300	1500	1712.509	1000

***RT (s):** running time (second)

Figures 7 and 8 illustrate the results.

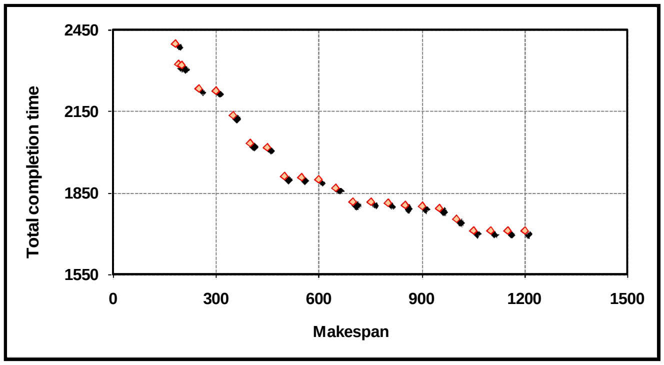


Fig. 7. The non-dominated solutions obtained by ϵ -constraint method for 10-job with 2-factory instance

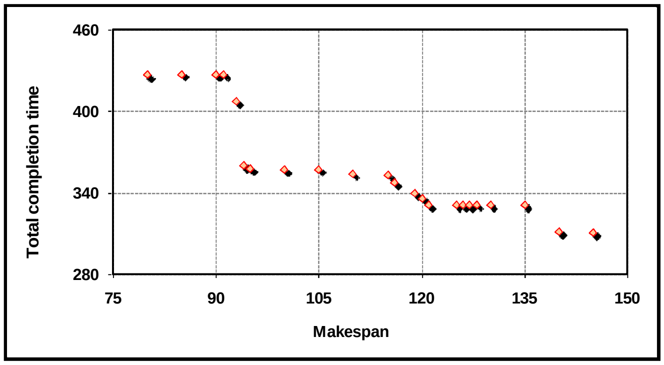


Fig. 8. The non-dominated solutions obtained by ϵ -constraint method for 25-job with 4-factory instance

The preliminary results indicate considerable potential to obtain good solutions by implementation of ϵ -constraint approach. From the results shown in Figs. 7 and 8, it can be observed that the model can find near-optimal solutions efficiently. This method is, however, a computationally intensive procedure and is feasible only for the small-size instances.

6- Conclusion and recommendation for future study

Since a multi-factory scheduling is NP-hard, no optimal polynomial time algorithms can be designed, so this paper proposed the polynomial approximation algorithms for the problem in which factories' production network can have an own criterion to optimize. Using a deeper analysis for the case that all factories are interested in the makespan, all factories are interested in the total completion time and the case that some factories are interested in the sum of completion times and remaining factories are interested in the makespan, we designed the best achievable approximation ratios by proposing the approximation algorithms for each case. In this paper, we also proposed a mixed integer linear programming for modeling scheduling problem with transportation times among network's factories. In this problem, we consider the problem where the jobs belong to one of two disjoint sets; the makespan criterion needs to be minimized for one of the sets, while the total completion time needs to be minimized for the other. In the modeling of problem, we use the multi-objective method namely augmented ϵ -constraint approach in which the total completion time must be minimized while another criterion i.e. the makespan as a constraint must be less than the predetermined value Q. By systematic changing in Q's values, we solve it in small-sized instances by solver CPLEX and generate the set of non-dominated solutions. Since the problem of scheduling known as virtual production network with self-interested factories has received surprisingly little attention in the literature, despite its ability to model several real-world situations, a worthy direction of future research is to develop heuristics and metaheuristics for large-sized instances.

References

Agnetis, A., Mirchandani, P., Pacciarelli, D., Pacifici, A.: Scheduling problems with two competing agents. Operations Research 52, 229–242 (2004).

Bartlett, C.A., Ghoshal, S.: Managing across borders: the transnational solution. Second Edition, HBS Press, Boston, MA (1998).

Chan, F.T.S., Chung, S.H.: Distributed Scheduling in Multiple-factory Production with Machine Maintenance, In: Wang, L. Shen, W. (eds.), Process Planning and Scheduling for Distributed Manufacturing, pp. 243-267. London, Springer Series in Advanced Manufacturing (2007).

Chung, S.H., Chan, F.T.S., Chan, H.K.: A modified genetic algorithm approach for scheduling of perfect maintenance in distributed production scheduling, Engineering Applications of Artificial Intelligence 22, 1005–1014 (2009).

Dessouky, M.M.: Scheduling identical jobs with unequal ready times on uniform parallel machines to minimize the maximum lateness, Computers & Industrial Engineering 34, 793–806 (1998).

Diedrich, F., Jansen, K., Pascual, F., Trystram, D.: Approximation algorithms for scheduling with reservations, Algorithmica 58, 391–404 (2010).

Flaherty, M.T.: Coordinating international manufacturing and technology. In: Porter, M.E. (ed.), Competition in Global Industries. Harvard Business School Press (1989).

Fleury, A., Fleury, M.T.: Competitive strategies and core competencies: perspectives for the internationalization of industry in Brazil. Integrated Manufacturing Systems 14 (1), 16–25 (2003).

Granot, F., Skorin-Kapov, J., Tamir, A.: Using quadratic programming to solve high multiplicity scheduling problems on parallel machines, Algorithmica 17, 100-110 (1997).

Jia, H.Z., Fuh, J.Y.H., Nee, A.Y.C., Zhang, Y.F. Integration of genetic algorithm and Gantt chart for job shop scheduling in distributed manufacturing systems, Computers & Industrial Engineering 53(2), 313–20 (2007).

Jia, H.Z., Fuh, J.Y.H., Nee, A.Y.C., Zhang, Y.F.: Web-based multi-functional scheduling system for a distributed manufacturing environment, Concurrent Engineering-Research and Applications 10(1), 27–39 (2002).

Johansen, K., Comstock, M., Winroth, M.: Coordination in collaborative manufacturing mega-networks: A case study, Journal of Engineering and Technology Management 22 (3), 226-244 (2005).

Karatza, H.D.: Job scheduling in heterogeneous distributed systems, The Journal of Systems and Software 56, 203-212 (2001).

Khurana, A., Talbot, B.: Plant missions in global manufacturing networks: A resource-based view with evidence from the global color picture tube industry. Working Paper 99–0005. University of Michigan Business School (1999).

Li, X., Shi, Y., Gregory, M.J.: Global manufacturing virtual network (GMVN) and its position in the spectrum of strategic alliance, EurOMA 7th International Annual Conference, pp. 330-337. Ghent, Belgium (2000).

Lin, H.K., Harding, J.A., Shahbaz, M.: Manufacturing system engineering ontology for semantic interoperability across extended project teams. International Journal of Production Research 42(24), 5099-5118 (2004).

Loeser, B.O.: How to set up a cooperation network in the production industry-example of the Huber + Suhner AG. Industrial Marketing Management 28, 453–465 (1999).

Mavrotas, G.: Effective implementation of the e-constraint method in multi-objective mathematical programming problems, Applied Mathematics and Computation 213, 455–465 (2009).

Miettinen, K.M.: Nonlinear Multiobjective Optimization. First Edition, Kluwer Academic, Boston(1999).

Monroy, C.R., Vilana Arto, J.R.: Analysis of global manufacturing virtual networks in the aeronautical industry. International Journal of Production Economics 126, 314–323 (2010).

Naderi, B., Ruiz, R.: The distributed permutation flowshop scheduling problem, Computers & Operations Research 37, 754-768 (2010).

Neiman, D.E., Lesser, V.R.: A Cooperative Repair Method for a Distributed Scheduling System, in: Drabble, B. (eds.), International Conference on Artificial Intelligence Planning Systems (AIPS), pp. 166-173. Edinburg, Scotland, Menlo Park (1996).

Pinedo, M.L.: Scheduling Theory, Algorithms, and Systems. Third Edition, Springer, NY (2008).

Sandholm, T.W.: Automated contracting in distributed manufacturing among independent companies, Journal of Intelligent Manufacturing 11, 271–283 (2000).

Saule, E., Trystram, D.: Multi-users scheduling in parallel systems, Parallel & Distributed Processing, 2009. IPDPS 2009, IEEE International Symposium on, 23-29 May 2009, pp. 1 – 9 (2009).

Shakhlevich, N.V., Strusevich, V.A.: Preemptive scheduling on uniform parallel machines with controllable job processing times, Algorithmica 51(4), 451-473 (2008).

Shi, Y., Fleet, D., Gregory, M.: Global manufacturing virtual network and its position in manufacturing systems. The 7th Annual International Manufacturing Symposium, Institute for Manufacturing, Department of Engineering, University of Cambridge (2005).

Shi, Y., Gregory, M.: From original equipment manufacturers to total solution providers: An emergence of global manufacturing virtual network in electronics industry. International Journal of Service Technology and Management 4 (4-6), 331-346 (2003).

Shi, Y., Gregory, M.: International manufacturing networks—to develop global competitive capabilities. Journal of Operations Management 16, 195–214 (1998).

Sturgeon, T., Florida, R.: Globalization and jobs in the automotive Industry. Center for Technology Policy, and Industrial Development, Massachusetts Institute of Technology, Cambridge (1999).

T'Kindt, V., Billaut, J.: Multicriteria Scheduling: Theory, Models and. Algorithms, Second Edition, Springer, Berlin (2006).

Tuma, A.: Configuration and coordination of virtual production networks, International Journal of Production Economics 5657, 641-648 (1998) .

Vercellis, C.: Multi-plant production planning in capacitated self-configuring two-stage serial systems, European Journal of Operational Research 119, 451-460 (1999).

Vilana, J.R., Monroy, C.R.: Structure and relationships within global manufacturing virtual networks. Intangible Capital 5(2), 152-168 (2009).

Wiendahl, H.P., Helms, K., Lutz, S.: Supply Net–New Methods for Capacity Flexibilisation in Production Networks, in: Maropoulos, P. G., McGeough, J. A. (eds.), Proceedings of the 15th International Conference on Computer-Aided Production Engineering, pp. 609 – 614. UK, Durham (1999) .

Williams, J.F.: Heuristic techniques for simultaneous scheduling of production and distribution in multi-echelon structures: Theory and empirical comparisons, Management Science 27, 336-352 (1981).

You, F., Grossmann, I.E.: Design of responsive supply chains under demand uncertainty, Computers and Chemical Engineering 32(12), 3090-3111 (2008).

Zhang, W., Gen, M.: Process planning and scheduling in distributed manufacturing system using multiobjective genetic algorithm, IEEJ Transactions on Electrical and Electronic Engineering 5, 62–72 (2010).

Zhang, Y., Li, L., Wang, H., Zhao, Y., Guo, X., Meng, C.: Approach to the distributed job shop scheduling based on multi-agent, In: IEEE International Conference on Automation and Logistics, Qingdao, pp. 2031 – 2034 (2008).