American Journal of Applied Sciences 6 (4): 601-607, 2009
ISSN 1546-9239

The Coordinated Scheduling Support System of Production and Delivery

Ming-Feng Yang and Yi Lin
Department of Information Management, Yu Da College of Business,
Taiwan, Republic of China

Abstract: Problem statement: Traditional scheduling models which only address the sequence of jobs to be processed at the production stage under some criteria are no longer suitable and should be extended to cope with the distribution stage after production. In a rapidly changing environment, competition among enterprises has a tendency to turn towards competing between supply chain systems instead of competing between individual companies. Emphasizing on the coordination and the integration among various members of a supply chain has become one of the vital strategies for the modern manufacturers to gain competitive advantages. **Approach:** This research focuses mainly on a class of two-stage scheduling problem, in which jobs need to be delivered to customers by vehicles after the completion of their respective production. It is assumed that the transportation time of a vehicle is constant and jobs to be delivered occupy different physical spaces. **Results:** The result of this research is to show the scheduling problem with the objective of minimizing total completion time is intractable and to develop a heuristic by incorporating properties inherited in an the optimal schedule. In addition, we take a Decision Support System (DSS) view to construct a Scheduling Support System (SSS) for solving the scheduling problem with delivery coordination. **Conclusion/Recommendations:** The scheduling support system with an additional problem management subsystem can provide more useful information for users when the management makes a strategic decision than traditional scheduling methods can. It can give firms a competitive advantage on the global competitive market.

Key words: Problem management, computational complexity, decision support system

INTRODUCTION

In the current competitive environment for manufacturing, the application of supply chain management is of increasing interests. In order to be competitive, companies tend to put significant emphasis on the coordination of activities along different stages of a supply chain. These stages comprise of suppliers, manufacturers, distributors and customers. In particular, the issue of coordinating the stages of production and distribution has been widely discussed.

In most manufacturing and distribution systems, semi-finished jobs are transferred from one processing facility to another by transporters such as Automated Guided Vehicles (AGVs), robots and conveyors and finished jobs are delivered to customers or warehouse by vehicles such as trucks. A Flexible Manufacturing System (FMS) is an integrated system that has been introduced to give more flexibility by overcoming the traditional hypotheses such as infinite buffer area between machines, infinite number of transporters available and the instantaneously delivery of jobs from one machine to another. This system is a computer controlled production unit, which consists of a single numerically controlled machine, a material handling device and a storage area for parts. The scheduling problem is the most important problem encountered when managing a flexible manufacturing system.

An important example of decision making that affects both a supplier and a manufacturer is the delivery process between them. The supplier processes jobs and delivers them to the manufacturer. The manufacturer may prefer to receive frequent deliveries of small batches from the supplier, because this will enable the manufacturer to achieve better resource utilization. However, the supplier may be reluctant to deliver very frequently because of the resulting high delivery cost. Importantly, the scheduling decisions must be coordinated with the related batching and delivery decisions.

This research addresses the scheduling problem in which jobs are processed on a single machine and then

Corresponding Author: Ming-Feng Yang, Department of Information Management, Yu Da College of Business, Miaoli County, 361 Taiwan

delivered to customers with the objective to minimize the total completion time of the jobs. The purpose is to demonstrate this class of scheduling problem is intractable and propose a heuristic procedure involving job sequence and batch composition for the general case to derive approximate solutions. In addition, a Scheduling Support System (SSS) is constructed to provide more useful information for users when they make a more strategic decision than traditional scheduling methods can. It also can give firms a competitive advantage on the global competitive market.

METERIALS AND METHODS

In the literature in recent years, Maggu and Das explicitly took the transportation issue into account by considering a two-machine flow shop makespan problem with unlimited buffer spaces on both machines in which a sufficiently large number of transporters are to deliver a job when its production on the first machine is completed to the other machine immediately with a job-dependent transportation time[1]. Maggu *et al.*[2] studied the same problem with the additional constraint that some jobs must be scheduled consecutively. Kise investigated a variant of the problem with only one transporter by which only one job can be carried at a time. He demonstrated that this problem is ordinarily NP-hard even with job-independent transportation times[3].

Another line of research with transportation considerations focuses on the transportation of finished jobs to customers. These models redefine the job completion time as the time when a job arrives at the customer. Potts and Hall and Shmoys studied single-machine scheduling models with unequal job arrival times and delivery times. They implicitly assumed that a sufficient number of vehicles are available at all times to deliver finished jobs to customers immediately. They proposed heuristics as well as worst-case analysis for their methods[4-5]. Woeginger studied the same problem in the parallel-machine environment with equal job arrival times and provided a heuristic with a worst-case analysis[6].

In yet another line of research, several papers by[7-10] investigated the scheduling problems in which jobs are delivered to customers in batches. Potts and Van Wassenhove advocated that jobs can be batched together due to two possible reasons; one is that jobs share the same setup on a machine and another occurs when a machine can process several jobs simultaneously[7]. As to the latter, Ahmadi *et al.*[11] considered a class of two-machine batching and scheduling problems in which the batch processor, which can process a batch of jobs simultaneously, plays an important role.

Lee and Chen investigated machine scheduling models that impose constraints on both transportation capacity and transportation times. They categorized this class of scheduling problems based on the type of transportation situations into two different types. The first type is intermediate transportation in a flow shop where jobs are transported from one machine to another for further processing. The second type is the delivery of finished jobs to customers. Jobs are delivered in batches by one or more vehicles with finite or infinite capacity. They assumed that sizes of all jobs are of consistence[12]. Chang and Lee[13] extended their work to the situation in which each job requires different physical spaces for delivery, whereas Li *et al.*[14] considered a problem involving job deliveries to multiple customers at different locations.

RESULTS

The results can be categorized into two aspects:

- Flow shop scheduling with delivery system
- Scheduling support system.

Flowshop scheduling with delivery system: We first describe the scheduling problems we are aim to tackle. Then, we give the proof of NP-hardness for the problems and examine the problems to find optimality properties. Based on it, several heuristics are proposed to solve the problems efficiently.

Strong NP-hardness: We will show that the scheduling problem we studied is strongly NP-hard by reduction from the BIN PACKING problem, which is known to be NP-hard in the strong sense by[15].

Theorem 1: The scheduling problem $F_2 \to D, k = 1 \left| v = 1, c = z \right| C_{\max}$ is NP-hard in the strong sense.

Theorem 1 provides the proof of NP-hardness for our scheduling problem. As a result, proposing a heuristic procedure for this intractable problem to derive approximate solutions is justifiable. We present some optimality properties that will be useful throughout our study. Hence, only solutions that satisfy all the optimality properties are considered further.

Optimality properties:

Theorem 2: There exists an optimal schedule for the problem $F_2 \to D, k = 1 \left| v = 1, c = z \right| C_{\max}$ that satisfies the following properties:

- Jobs are processed on the first machine without idle time
- Jobs are processed on both machines with the same sequence
- Jobs assigned to one batch are processed consecutively on either machine
- Jobs assigned to one batch are scheduled by Johnson's rule

The related proofs will be indicated in further discussion.

Scheduling support system: Scheduling Support System (SSS) is one of Decision Support System (DSS) for handling the scheduling problems. One of the differences between SSS and DSS is on their model bases. Model base of scheduling support system is composed of one to several scheduling algorithms. Kuo and Hwang established the prototype of the real-time scheduling support system. They pointed out that the success of production system is base on the efficiency of production scheduling and the control of shop floor[16]. The results showed the production system is more flexible through the friendlier interface.

Problem management system: We thought that traditional DSS emphasizes on the management of solutions and neglects the essentiality of the decision process, i.e., problem management and DSS needs the use of a problems management system to provide more efficient support for the future. The steps of problems management is as shown in Table 1.

Ecker, Gupta and Schmidt applied it in scheduling problems as in Fig. 1. In the Ecker's system, there are three mainly modules to describe the system, that is, problem analysis module, schedule construction module and evaluation module[17]. Problem analysis module includes hard constraints and soft constraints. Hard constraints are the constrains that must be satisfied during the process of solving the scheduling problems, such as precedence relations, routing conditions, resource availability, ready times and setup times. Soft constraints are the desired conditions, such as job sequences, resource utilization and objective minimization.

The schedule construction comprises of four phases. The first phase is the initial schedule generation by using algorithms where hard constraints are considered. The second phase is called basic schedule. The third one is conflict detection. In this phase, the priority of properties and initial schedule is analyzed.

Table 1: Problem management process

Steps of problems management	Cognitive activity
Problem finding	Identify environmental factors
Problem representation	Identify the type of problems
Information surveillance	Search environmental information
Solution generation	Generate several solutions
Solution evaluation	Evaluate several solutions

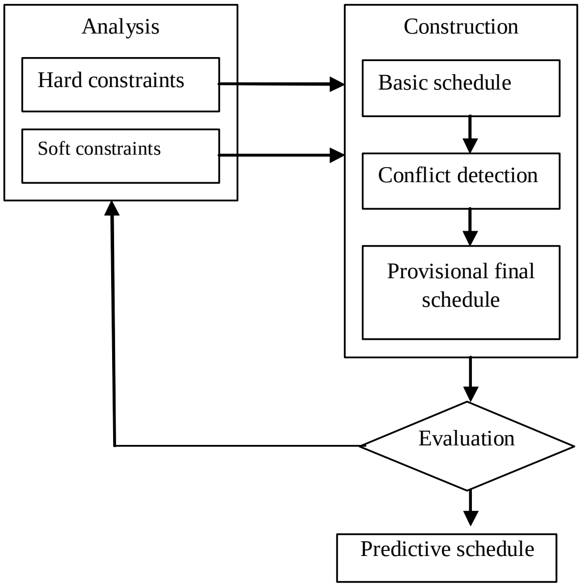

Fig. 1: Problem Management Subsystem

In the fourth phase, according to the desired objective and soft constraints, a final schedule is generated and then evaluated. The problem management subsystem is constructed based on Ecker's research. By such procedures, scheduling support system will provide better support than just managing solutions.

System framework: The schema of the scheduling support system we constructed for the scheduling problem with transportation consideration is based on the schema of decision support system. Our scheduling support system includes four modules: database management system, model base management, problem management system and user interface in Fig. 2. These four sub-systems are established according to the Ecker's model. The database management system is responsible for retrieving internal stored data or external information, such as processing data of jobs, vehicle information and other properties. The model based management system consists of efficient algorithms for the problem we studied. Problem management system is responsible for analyzing the desired scheduling problem and choosing a proportionate algorithm.

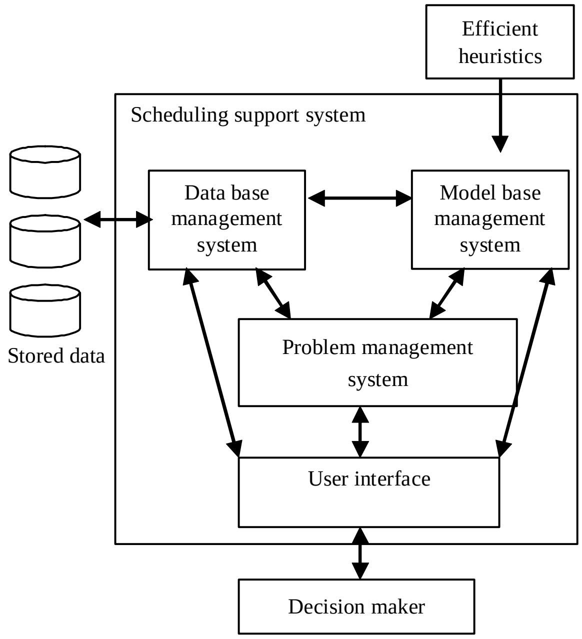
Efficient heuristics
Scheduling support system
Data base management system
Model base management system
Stored data
Problem management system
User interface
Decision maker

Fig. 2: Schema of the Scheduling Support System

DISCUSSION

Flowshop scheduling with delivery system:
Problem discussion: The scheduling problem that we are interested in is commonly arising from a flexible manufacturing environment. Specifically, we consider a two-machine flowshop problem with type-2 transportation, which is described formally as follows. There is a set of n independent jobs, N = {J_1, J_2, …, J_n}, to be processed without preemption at a manufacturing system consisting of two machines, each of which is a discrete processor, namely, a machine can only process at most one job at a time. These two machines, ordered as M_1 and M_2, are continuously available from time zero onwards. Every job comprises of two operations associated with respective processing times on both machines. Before the first operation has been completed on M_1, the second one cannot be started for processing on M_2. It is assumed that all the jobs are simultaneously available at the beginning. Moreover, each job is associated with a job size, s_j, which represents the physical space job J_j occupies when being loaded in the vehicle.

After processing in the manufacturing system, the finished jobs are delivered in batches to the respective customers by a vehicle. There is only one vehicle available initially with a capacity of z at the manufacturing facility. The capacity of the vehicle is measured by the total physical space that the vehicle provides for one delivery. Furthermore, since we only consider the case of one customer area where all customers are located with proximity to each other, the transportation time, denoted by T, is assumed to be constant for each trip. The transportation time is made up of two parts: one is the time from the manufacturing facility to the customer, denoted by t_1 and the other, denoted by t_2, is that from the customer to the manufacturing facility, where t_1 and t_2 are independent of the jobs being transported.

The problem we are mainly concerned is to determine the job processing sequence in the manufacturing system together with the delivery schedule such that the time required for processing and delivering all of the jobs is minimized.

Following the three-field notation schema, $\alpha | \beta | \gamma$, introduced by Graham *et al.* [18] and the additional notation of Lee and Chen[12], our problem of minimizing the makespan is denoted by $F_2 \rightarrow D, k = 1 \big| v = 1, c = z \big| C_{\max}$. In the α field, the notation "$F_2 \rightarrow D, k = 1$" represents the problem in which jobs are first processed on a two-machine flowshop environment and then delivered to one customer area. In the β field, v denotes the number of identical vehicles at the manufacturing facility and c is the capacity of vehicles. Here, the notation "$v = 1, c = z$" means that only one vehicle with a capacity of z is considered in our problem. In the γ field, a regular measure of performance $C_{\max}$ is used in order to reduce the turnaround time. For convenience of analysis, the makespan of a schedule σ, denoted as $C_{\max}$ (σ), is defined in this study as the time when the vehicle finishes delivering the last batch to the customer area and returns to the manufacturing facility. Unless ambiguity would result, we simplify $C_{\max}$ (σ) to $C_{\max}$.

Strong NP-hardness: There is a proof of theorem 1 to the result of Strong NP-Hardness.

Theorem 2: The scheduling problem $F_2 \rightarrow D, k = 1 \big| v = 1, c = z \big| C_{\max}$ is NP-hard in the strong sense.

Proof: Consider a special case of $F_2 \rightarrow D, k = 1 \big| v = 1, c = z \big| C_{\max}$ in which, for every job, the processing times of both operations are zero. Hence all of the jobs are ready for delivering to the customer at the beginning. In such a case, because of the constant transportation time, minimization of the number of delivery batches will achieve the optimality and thus the problem can be regarded as a bin packing problem, which is a well-known strongly NP-hard problem. Therefore, our problem is also NP-hard in the strong sense.

Optimality properties:
Theorem 3: There exists an optimal schedule for the problem $F_2 \to D, k = 1 | v = 1, c = z | C_{max}$.

Proof:

- If there exists idle time before processing of all operations is finished on the first machine, the subsequent operations can be moved earlier without increasing the makespan
- Suppose that there exists an optimal schedule in which some, at least two, jobs, say J_i and J_j, have shifts in their processing orders when traveling through the machines. If, in the schedule, J_i is processed earlier than J_j on the first machine (machine 1) and the opposite is occurred on the second machine (machine 2), then interchanging the positions of these two jobs on machine 1 and keeping the processing sequence on machine 2 will achieve the desired condition with no effect on the objective value. It is clear that, for job J_j, its position on machine 1 is moved forward. The start of processing its second operation on machine 2 will not be affected. For job J_i, though its position on machine 1 is shifted backwards, the completion time of its first operation is still no latter than the start time of the second operation of J_j and thus is undoubtedly prior to the start time of its second operation on machine 2. Consequently, by repeating the above procedure, a schedule with jobs requiring a sequence change between machines can always be modified by changing the sequence of jobs on machine 1 in accordance with that on machine 2 without increasing the makespan. In other words, an optimal schedule of our problem is always a permutation schedule
- Based on the property (ii), without loss of generality, it is sufficient only to consider the situations in which jobs maintain a specific permutation traversing the entire system. Suppose an optimal schedule S satisfying property (ii), in which there exists a batch, say B_k, whose jobs are not processed consecutively on the machines due to some “interrupted” jobs (not necessary adjacent to each other). All we have to do is to move all those interrupted jobs forwards, while keeping their sequence, until the last job of it is processed just before the first job of B_k so as to result in the consecutive processing of jobs in B_k. It can be seen that the completion time of the last job in B_k on machine 2 would not be postponed, that is, the ready time of batch B_k is unchanged. Note that the vehicle leaves the manufacturing system with some batch only after the jobs in that batch are all finished. Hence, by repeating the procedure, a new schedule with processing consecutiveness of jobs in one batch can be constructed and the makespan of it is no other than that of S and that is to say, an optimal schedule always satisfies the property
- It is known that Johnson’s rule, also known as an SPT(I)-LPT(II) schedule. It is a classical algorithm which can achieve the optimality for the $F_2 \| C_{max}$ problem. It is easy to see that, based on property (iii), to sequence the jobs which are arranged to the first batch can be considered as a $F_2 \| C_{max}$ problem and thus the jobs should be scheduled by Johnson’s rule. Note that Johnson’s rule is based on the assumption that both machines are available at the beginning of the time horizons; however, as to the job sequencing within some other batch, say B_k, there is a possibility that machine 2 is not available while machine 1 starts to process the first job. Fortunately, it is of little significance for the use of Johnson’s rule. It can be verified by only introducing an insubstantial job, J_0, with zero processing time on machine 1 and τ on machine 2, where τ is the right time that is the period of unavailable time of machine 2 (which is resulted from the processing of the preceding batch). Since the processing time of J_0 on machine 1 is zero, it is classified into Set I, which contains all the jobs with $p_{1j} < p_{2j}$ and is definitely processed as the first job according to SPT order of p_{1j}. Therefore, the sequence of jobs assigned to B_k determined by Johnson’s rule remains the same.

Scheduling support system:
Model base management system: In the model based management system, we propose some efficient heuristics. First of all, we review the FFD algorithm and the SPT(I)-LPT(II) schedule, which are both useful for establishing the heuristics.

First fit decreasing algorithm: Let batches are indexed as B_1, B_2,…, with each initial capacity of z. Sort the jobs in non-increasing order of their size (s_j). Place the first job, whose size is the largest, in B_1. If the jth largest job is considered, then place it to the lowest indexed batch whose current content does not exceed z-s_j, or generate another batch and put the job in.

The SPT(I)-LPT(II) schedule: Partition the jobs into two sets, with Set I containing all the jobs with $p_{1j} < p_{2j}$ and Set II all the jobs with $p_{1j} < p_{2j}$. The jobs with $p_{1j} < p_{2j}$ may be put in either set. The jobs in Set I go first and they go in increasing order of p_{1j} (SPT); the jobs in Set II follow in decreasing order of p_{2j} (LPT). Ties may be broken arbitrarily.

According to the optimality properties mentioned in Sec 3.4, we provide several efficient heuristics for the scheduling we studies as follows.

Heuristic procedure H1:

Step 1: Assign jobs into batches by using the FFD algorithm. Set the total number of resulting batches as b

Step 2: Sequence the jobs within each batch as an SPT(I)-LPT(II) schedule

Step 3: Sequence the batches in SPT order of P_j where batches are indexed according to their order

Step 4: Starting with B_1, assign jobs in B_k to the machines, for $k = 1, 2, \cdots, b$. Dispatch each completed but undelivered batch whenever the vehicle becomes available. If multiple batches have been completed when the vehicle becomes available, dispatch the batch with the smallest index

Heuristic procedure H2:

Step 1: Assign jobs into batches by using the FFD algorithm. Set the total number of resulting batches be b

Step 2: Sequence the jobs within each batch as an SPT(I)-LPT(II) schedule

Step 3: Sequence the batches with $P_j \phi T$ in SPT order, then those with $P_j > T$ followed by the inequality $\min(\rho_k, \phi_l) \leq \min(\rho_l, \phi_k)$

Step 4: Starting with B_1, assign jobs in B_k to the machines, for $k = 1, 2, \cdots, b$. Dispatch each completed but undelivered batch whenever the vehicle becomes available. If multiple batches have been completed when the vehicle becomes available, dispatch the batch with the smallest index

Heuristic procedure H3:

Step 1: Sequence the jobs as an SPT(I)-LPT(II) schedule

Step 2: Assign jobs into batches by using the algorithm similar to the FFD algorithm except that the sort of the jobs has been determined in Step 1. Set the total number of resulting batches be b. Index the batches according to their order

Step 3: Starting with B_1, assign jobs in B_k to the machines, for $k = 1, 2, \cdots, b$. Dispatch each completed but undelivered batch whenever the vehicle becomes available. If multiple batches have been completed when the vehicle becomes available, dispatch the batch with the smallest index

CONCLUSION

This study investigates the flow shop scheduling models that explicitly consider constraints on transportation and job sizes. The finished jobs are transferred from the processing facility and delivered to one and only one customer or warehouse by a vehicle with capacity *z*. The considered objective function is the makespan.

New complexity result is derived and several efficient heuristic are proposed for the problem. In addition, from the DSS view, we establish the prototype of SSS which provides more useful information for users when he makes a strategic decision to solving this class of coordinated scheduling problems. Moreover, problem management subsystem is introduced in the system to improve traditional DSS. In the future research, it is interesting to extend the existing model to develop solutions for problems with other objective functions.

REFERENCES

1. Maggu, P.L. and G. Das, 1980. On 2×n sequencing problem with transportation times of job. Pure Applied Math. Sci., 12: 1-6. http://140.118.33.10:4550/resserv/resserv?sid=Elsevier:Scopus&genre=unknown&issn=&volume=12&issue=&spage=1&epage=6&pages=1-6&artnum=&date=1980&title=Pure+and+Applied+Mathematika+Sciences&atitle=On+2+%c3%97+n+sequencing+problem+with+transportation+times+of+jobs&aufirst=P.L.&auinit=P.L.&auinit1=P&aulast=Maggu.
2. Maggu, P.L., G. Das and R. Kumar, 1981. On equivalent job-for-job block in 2×n sequencing problem with transportation times. J. Operat. Res. Soc. Jap., 24: 136-146. http://ci.nii.ac.jp/cinii/servlet/QuotDisp?LOCALID=ART0001515093&DB=NELS&USELANG=jp.
3. Kise, H., 1991. On an automated two-machine flowshop scheduling problem with infinite buffer. J. Operat. Res. Soc. Jap., 34: 354-361. http://ci.nii.ac.jp/cinii/servlet/QuotDisp?LOCALID=ART0001515591&DB=NELS&USELANG=jp.

4. Potts, C.N., 1980. Analysis of a heuristic for one machine sequencing with release dates and delivery times. Operat. Res., 28: 1436-1441. DOI: 10.1287/opre.28.6.1436.
5. Hall, L.A. and B. Shmoys, 1992. Jackson's rule for single-machine scheduling: Making a good heuristic better. Math. Operat. Res., 17: 22-35. DOI: 10.1287/moor.17.1.22.
6. Woeginger, G.J., 1994. Heuristics for parallel machine scheduling with delivery times. Acta Inform., 31: 503-512. DOI: 10.1007/BF01213203.
7. Potts, C.N. and L.N. Van Wassenhove, 1992. Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. J. Operat. Res. Soc., 43: 395-406. http://www.palgrave-journals.com/jors/journal/v43/n5/abs/0430502a.html.
8. Webster, S. and K.R. Baker, 1995. Scheduling groups of jobs on a single machine. Operat. Res., 43: 692-703. DOI: 10.1287/opre.43.4.692.
9. Potts, C.N. and M.Y. Kovalyov, 2000. Scheduling with batching: A review. Eur. J. Operat. Res., 120: 228-249. DOI: 10.1016/S0377-2217(99)00153-8.
10. Cheng, T.C.E., B.S. Gordon and M.Y. Kovalyov, 1996. Single machine scheduling with batch deliveries. Eur. J. Operat. Res., 94: 277-283. DOI: 10.1016/0377-2217(96)00127-0.
11. Ahmadi, J.H., R.H. Ahmadi, S. Dasu and C.S. Tang, 1992. Batching and scheduling jobs on batch and discrete processors. Operat. Res., 40: 750-763. http://portal.acm.org/citation.cfm?id=146260.146285&dl=&dl=.
12. Lee, C.Y. and Z.L. Chen, 2001. Machine scheduling with transportation considerations. J. Schedul., 4: 3-24. DOI:10.1002/1099-1425(200101/02)4:1<3::AID-JOS57>3.0.CO;2-D.
13. Chang, Y.C. and Y.L. Lee, 2004. Machine scheduling with job delivery coordination. Eur. J. Operat. Res., 158: 470-487. DOI: 10.1016/S0377-2217(03)00364-3.
14. Li, C.L., G. Vairaktarakis and C.Y. Lee, 2005. Machine scheduling with deliveries to multiple customer locations. Eur. J. Operat. Res., 164: 39-51. DOI: 10.1016/j.ejor.2003.11.022.
15. Garey, M.R. and D.S. Johnson, 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. 1st Edn., W.H. Freeman, San Francisco, CA., ISBN: 0716710447, pp: 338.
16. Kuo, W.H. and S.L. Hwang, 1998. A prototype of a real-time support system in the scheduling of production systems. Int. J. Ind. Ergonom., 21: 133-143. DOI: 10.1016/S0169-8141(96)00078-9.
17. Ecker, K., N.D. Gupta and G. Schmidt, 1997. A framework for decision support systems for scheduling problems. Eur. J. Operat. Res., 101: 452-462. DOI: 10.1016/S0377-2217(96)00186-5.
18. Graham, R.L., E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan, 1979. Optimization and approximation in deterministic sequencing and scheduling theory: A survey. Ann. Discrete Math., 5: 287-326. DOI: 10.1016/S0167-5060(08)70356-X.