중소 제조기업을 위한 엑셀기반 스케줄링 시스템

이창수* · 최경일**† · 송영효***

* 동부익스프레스 전략연구소
** 한국외국어대학교 산업경영공학부
*** 홍익대학교 상경대학

An Excel-Based Scheduling System for a Small and Medium Sized Manufacturing Factory

Changsu Lee* · Kyungil Choe**† · Younghyo Song***

* Strategy Institute, Dongbu Express, Dongbu Financial Center, 891-10, Daechi-Dong, Gangnam-Gu, Seoul, 135-523, Korea
** School of Industrial and Management Engineering, Hankuk University of Foreign Studies, Yongin, Kyunggi, 449-791, Korea
*** School of Business, Hongik University, Chochiwon, Chungnam 339-806, Korea

Key Words : Parallel Machine Scheduling, Excel-Based Scheduling, Ant Colony Optimization

Abstract

This study deals with an Excel-based scheduling system for a small and medium sized manufacturing factory without sufficient capability for managing full-scale information systems. The factory has the bottleneck with identical machines and unique batching characteristics. The scheduling problem is formulated as a variation of the parallel-machine scheduling system. It can be solved by a two-phase method: the first phase with an ant colony optimization (ACO) heuristic for order grouping and the second phase with a mixed integer programming (MIP) algorithm for scheduling groups on machines.

1. 서 론

Small and medium sized companies usually experience difficulties in implementing information systems for several reasons: weak information infrastructure, unstable manufacturing processes, insufficient exper- tise in managing production, information systems, and so on. An effective alternative for such companies is to implement an information system based on Excel. For example, Park and Yosida (2007) explain a Manufacturing Execution System (MES) implementation method for small and medium sized companies, and give an example with an Excel-based system. Similarly, we consider an Excel-based scheduling system for a small and medium sized manufacturing factory without sufficient capability for managing full-scale information systems.

The ABC factory, a typical small company, manufactures polyurethane panels for cold storages by a make-to-order (MTO) basis. One of its competitive advantages is flexible and short delivery. As business grows, however, the scheduler has experienced severe difficulties in meeting due dates. The factory does not have enough resources to implement an MES with a scheduling system. Hence we need to develop an Excel-based scheduling system which does not require other

† 교신저자 kichoe@hufs.ac.kr

information systems but still can develop a reasonably 'good' production schedule from a manually-managed order list.

The simple process of the factory allows us to focus on scheduling the bottleneck rather than the whole process. Once the schedule of the bottleneck is determined, then those of other operations can be done in a straightforward manner. The bottleneck workstation has several 'identical' machines, and so can be regarded as a Parallel Machine Scheduling Problem (PMSP). The unique batching property of the operation, however, requires a customized algorithm.

A lot of studies for various PMSP can be found in the literature such as Pinedo (2002). Only a few similar to our problem will be mentioned here. First, Suër et al. (1997) consider a PMSP with lot splitting and setup time. They assume that all of the jobs are sorted by the early due date rule, and give several mathematical models. Cao et al. (2005) consider similar problems which need to select working machines from K available machines. They develop a combinatorial optimization model and a tabu-search heuristic to minimize the total weighted tardiness penalty.

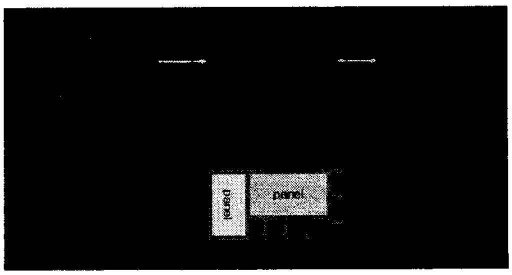
We suggest a different approach based on Ant Colony Optimization (ACO), which was introduced by Dorigo et al. (1999). It mimics the foraging behavior of ants. They usually seek out food in the surrounding area of the nest in a random manner. If it finds a food source, an ant carries some of food back to the nest. During the return trip, it deposits a pheromone on the ground, which will attract others to the food source. We discuss how to solve our problem by adjusting pheromone level.

Section 2 explains the details of our scheduling problem. Two mathematical formulations are presented in section 3: one without order mixing and the other one with order mixing. In section 4, we propose a two-phase method: the first phase with several heuristics including ACO, and the second phase with a branch and bound algorithm of Mixed Integer Programming (MIP). In section 5, we show computational results of an Excel-based scheduling system. Concluding remarks are included in the last section.

2. Problem Descriptions

The ABC factory processes customer orders of cold storages. It cannot carry inventory, because each order has unique dimensions. The factory determines the specifications of a cold storage depicted in <Figure 1>, as an order arrives. Each face of a cold storage usually consists of several frames. Each frame is made of 6 panels with heat-insulating materials inside. They initially cut panels with proper dimensions, and then assemble them into frames. A frame needs to be cured for a short period. After curing is over, the frames are shipped to a customer site so that they are built as a cold storage.

Its manufacturing process used in a typical small factory is quite simple, and is dominated by a single bottleneck, which is the assembly/curing operation. The operation with 2 'identical' assembly /curing machines processes frames by batches or groups. Frames with various sizes are fixed on a table for the operation. A table may contain more than one frame depending on its size and shape. Workers usually determine which frame is set on which table. In average, 80 percent of the table space is covered by frames and fixtures, and the remainder becomes idle.



<Figure 1> Components for a cold storage.

We will not consider this detailed 2-dimensional packing problem. In fact, once the specifications of an order are given, the scheduler groups frames so that they can be fit with tables, and determines the number of tables required. Hence, without the loss of generality, we can assume that a table can contain only a single frame. An assembly/curing machine processes a group of maximum 6 frames (or tables), and its processing time is independent of the number of frames. There is no setup time between groups. Machine breakdown is not assumed. Therefore, maintenance is not necessary.

Since the assembly/curing operation determines the manufacturing sequence and throughput of the whole factory, it is enough to schedule the bottle-neck operation. The scheduler manually maintains an order list and tries to make a production schedule with a frozen period of 2 days. Due dates for order are the most important factor for scheduling. We assume that workers are equal, and that there is no machine failure.

3. Mathematical Models

Our problem is a variation of the PMSP. We propose two mathematical models as shown in <Figure 2>: the Basic Mathematical Model (BMM) and the Order Mix Model (OMM). In the BMM, order mix is not allowed: each group consists of frames belonging to the same order. In other words, even if an order cannot fill a group with 6 frames, the unfilled group shall be processed. This assumption is so simple that the machines are not utilized well.

The OMM allows order mixing: a group may consist of frames belonging to different orders so that each group contains 6 frames whenever possible. The concepts of the models are depicted in <Figure 2>. The BMM makes the problem too simple, but will be helpful to solve the OMM.

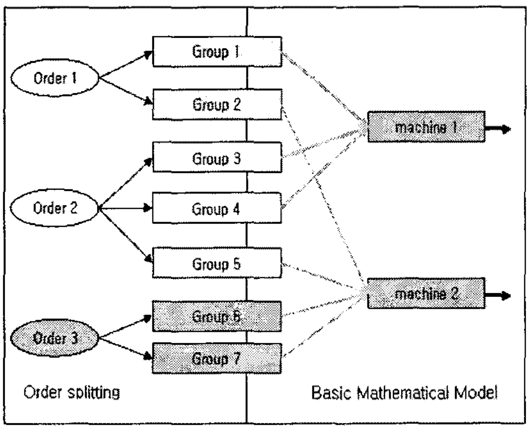
The set of customer orders and the set of resulting groups are denoted by O and G, respectively. G_l is the groups belonging to the order l. The due date for order l is denoted by d_l. The time period required for processing a group is fixed as p, and independent of groups and orders. L, N, and M are the total number of orders, groups, and machines, respectively. Decision variables in the BMM are given as follows:

X_{ijk} = 1, if group j is processed immediately after group i on machine k (group 0 is a dummy group processed always first)

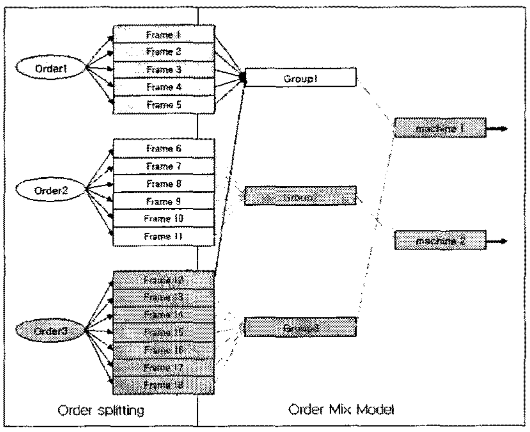
Y_{jk} = 1, if group j is assigned to machine k

C_{jk} = the completion time of group j on machine k

H_j = the final completion time of group j is as-



(a) the BMM



(b) the OMM

<Figure 2> Concepts of the mathematical models

signed to machine k

F_l = the completion time of order l, i.e., the maximum of H_j's belonging to order l

U_l = 1, if the completion time of order l is later than its due date

The BMM can be written as follow:

The BMM tries first to schedule groups and then determines how many jobs are tardy. The objective function Equation (1) minimizes the number of tardy jobs. Constraints from Equation (2) to Equation (5) determine the precedence relationships and allocations of groups. Constraint Equation (6) guarantees that machines are assigned to the similar workload. Equation (7) and Equation (8) state that an operation cannot be started before its preceding operation is completed. Equation (9) and Equation (10) indicate the completion times of a group and an order, respectively. Equation (11) defines a tardy order. The BMM itself does not yield an effective solution to our problem, because no order mix makes the bottleneck inflexible and inefficient.

$$\min \sum_{l=1}^{L} U_l \qquad (1)$$

subject to

$$\sum_{k=1}^{M} Y_{jk} = 1 \qquad \forall j = 1..N \qquad (2)$$

$$\sum_{i=0, i \neq j}^{N} X_{ijk} = Y_{jk} \qquad \forall j = 1..N,\ \forall k = 1..M \qquad (3)$$

$$\sum_{i=1}^{N} X_{ijk} \leq Y_{jk} \qquad \forall j = 1..N,\ \forall k = 1..M \qquad (4)$$

$$\sum_{j=1}^{N} X_{0jk} = 1 \qquad \forall k = 1..M \qquad (5)$$

$$\sum_{j=1}^{N} Y_{jk} \leq \lceil N/M \rceil \qquad \forall k = 1..M \qquad (6)$$

$$C_{ik} + p \leq C_{jk} + BigM * (1 - X_{ijk}) \qquad \forall i, j = 1..N, i \neq j,\ \forall k = 1..M \qquad (7)$$

$$C_{jk} \leq BigM * Y_{jk} \qquad \forall j = 1..N,\ \forall k = 1..M \qquad (8)$$

$$H_j = \max\{C_{j1}, .., C_{jk}\} \qquad \forall j = 1..N,\ \forall k = 1..M \qquad (9)$$

$$F_l = \max_{j \in G_l}\{H_j\} \qquad \forall l = 1..L \qquad (10)$$

$$BigM * U_l \geq \max\{0, F_l - d_l\} \qquad \forall l = 1..L \qquad (11)$$

$$X_{ijk}, Y_{jk} \in \{0,1\} \qquad \forall i, j = 1..N, i \neq j,\ \forall k = 1..M \qquad (12)$$

$$U_l \in \{0,1\} \qquad \forall l = 1..L \qquad (13)$$

Now, we extend the BMM to the OMM so that frames are assigned into the minimum number of groups. The set of frames is written as S. S_l denotes a set of frames belonging to order l. And the following decision variables are added to those of the BMM.

A_{js} = 1, if frame s is assigned to group j
H_{js} = the completion time of frame s belonging to group j

The following constraints for order mixing are added to the BMM: other constraints are needed to be changed accordingly.

Constraints Equation (14) and Equation (15) ensure that each frame is grouped exactly once and that each group has at most six frames. Equation (16) and Equation (17) represent the completion time of a frame. Finally, Equation (10) should be replaced with Equation (18).

4. A Two-Phase Heuristic

Our computational results imply that the BMM can be easily solved, but its results are too simple to be applicable. The OMM cannot be solved within a reasonable time limit by an exact branch-and-bound MIP algorithm, mostly because of too many binary variables for frame assignment, A_{js}. Hence, we devise a two-phase method for separating frame assignment from group allocation. In the first phase, frames are assigned into groups by considering due dates. Then in the second phase groups are scheduled to minimize tardy jobs.

In the first phase, an order is divided into frames, and then the frames are assigned to a group. We try three kinds of heuristic algorithms: a simple greedy heuristic, a genetic algorithm, and an ACO heuristic. The first two methods are relatively well-known in the literature, e.g., Guo et al. (2006). We use a general logic for the genetic algorithm. The pheromone update rules for ACO are shown in <Figure 3>.

τ_{ps} is a pheromone level which frame s is assigned immediately after frame p. Equation (20) is a local pheromone update rule where τ_0 is the initial pheromone value and ρ ($0 < \rho < 1$) is the local pheromone evaporating parameter. Equation (21) is a global update rule where S^* is the best solution and S_0 is the new best solution.

Once frames are assigned to groups, then we can use the BMM for machine allocation. If the

```
Input: A problem instance of frame assignment
  Initialize pheromone value and parameter values
  while (termination condition not met) do
  for ant k = 1 to K do
    for frame p = 1 to P do
       Frame assignment
      τ_ps = (1- ρ)* τ_ps + τ_0                          (20)
    End
 If S* > S_0 then
    for frame p = 1 to P do
      τ_ps = (1- ρ)* τ_ps + Δτ_1                         (21)
      Δτ_1 = 1/(S_0 - S*)
    End
  end while
Output: Best solution
```

<Figure 3> An ACO heuristic for frame assignment

first phase finishes frame assignment, then the second phase schedules the resulting groups by using the BMM. Now, the due date of a group is the same as the shortest one of the frames belonging to the group. A general branch-and-bound type algorithm may solve our problem reasonably well.

5. Computational results and an Excel-based scheduling system

For testing the computational efficiency of our method, we use real data from March to April in 2007. Our models are implemented in the Xpress-MP environment which is a well-known commercial package: see Guéret et al. (2002) for details. The branch-and-bound routines by Xpress-MP are used and other heuristics are coded in C. Tests are run on a Pentium M (Sonoma) Windows platform with 1GB RAM. The average computing time of the OMM with a branch-and-bound MIP algorithm is longer than one hour, which is certainly unacceptable in a real situation. Accordingly, we try the two-phase method with 3 different heuristics. The results are summarized in <Table 1>, which implies that our two-phase heuristic works relatively well. For the real data, the values of the objective function by both a genetic algorithm and an ACO are the same as true optimization ones, which is no tardy jobs.

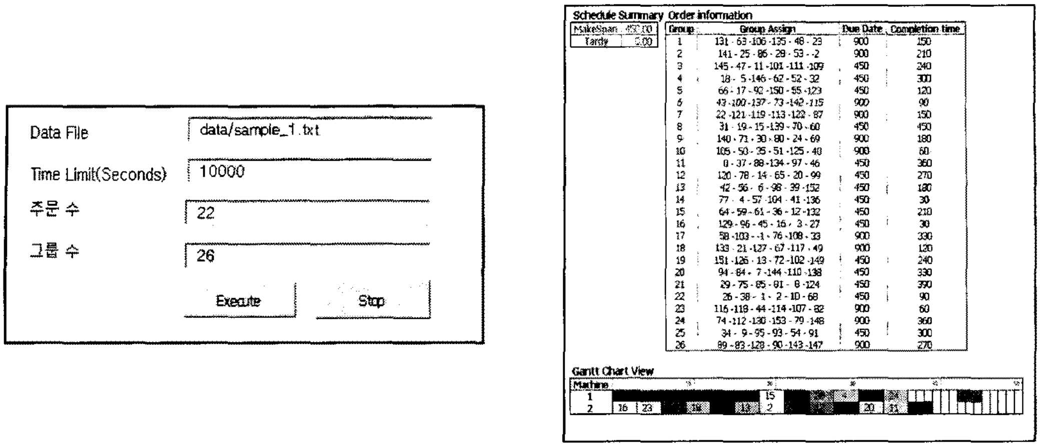
As mentioned before, the ABC factory is not capable of implementing a full-scale information system. Thus, we decide to implement our algorithms as a 'stand-alone' Excel-based scheduling system. Some examples are shown in <Figure 4>. First, the scheduler of the factory maintains the order list as an Excel file. When the scheduler needs a schedule, he/she converts the order file to a list of frames which can be done on Excel. After confirming the frame list, the scheduler types a data path of the file, a number of orders, and so on, as shown in <Figure 4(a)>. Then, the output schedule is given as a sheet like <Figure 4(b)>. The sheet includes a summary of production schedule and group assignment, and a Gantt chart. The summary contains the makespan and the number of tardy orders too. Either a genetic algorithm or an ACO heuristic can be used in the first phase. We prefer ACO to genetic algorithms, because its code is much simpler and yet as fast as genetic algorithms.

<Table 1> Computing times of the two-phase heuristic

Data Set	Heuristic used in the second phase		
	GREEDY	GENETIC	ACO
1	8.5 sec.	9.9 sec.	7.8 sec.
2	19.6	6.7	9.7
3	20.1	11.2	11.2
4	28.7	14.5	18.5

6. Concluding Remarks

We consider a Make-to-Order (MTO) factory which is not capable of full-scale information systems. The factory yet needs a scheduling system to meet due dates for customer. Hence we propose a 'stand-alone' Excel-based scheduling system for the bottleneck operation. First, we develop two mathematical models as a variation of the parallel machine scheduling problem (PMSP). Our PMSP is found hardly to be optimized by a branch-and-bound algorithm of mixed-integer-programming (MIP). Thus we devise a two-phase method to find a schedule within a reasonable time limit. The first phase uses either a genetic algorithm or an Ant Colony Optimization (ACO) heuristic to find frame assignment to groups. The second phase uses a branch-and-bound MIP routine (supported by Xpress-MP) to allocate groups into machines. The result of two-phase method shows relatively good schedules. Moreover, it is easy to handle and is very flexible for what-if simulation. Hence it can be a quick alternative for a small and



(a) an input screen (b) an output schedule

<Figure 4> Examples of the Excel-based scheduling system

medium sized company which looks for a scheduling system.

It should be noted that our system cannot consider the two-dimensional bin packing problem required for assigning frames onto a table. Certainly, it is very hard to implement such the system on Excel. Moreover, we should consider quality aspects of cold storages and statistical properties of processing times for group. For that purpose, we should consider a PERT-CPM type scheduling and a statistical quality control for storages. Hence, we may need other approaches to solve the extended problem.

Acknowledgements

The authors thank Prof. Ji-ung Sun and Mr. Seung-yong Cho for their valuable comments and help to the ACO heuristic.

References

[1] Cao, D., Chen, M., and Wan, G. (2005), "Parallel machine selection and job scheduling to minimize machine cost and job tardiness," *Computers & Operations Research*, Vol. 32, pp. 1995-2012.

[2] Chen, Z. L. and Lee, C. Y. (2002), "Parallel machine scheduling with a common due window," *European Journal of Operational Research*, Vol. 136, pp. 512-527.

[3] Chen, Z. L. and Powell, W. B. (1999), "Solving Parallel Machine Scheduling Problems by column generation," *Informs Journal on Computing*, Vol. 11, No. 1, pp. 78-94.

[4] Dorigo, M., Caro, G. D. And Gambardella, L. M. (1999), "Ant algorithms for Discrete Optimizations," *Artificial Life*, Vol. 5, pp. 137-172.

[5] Guéret, C., Prins, C., and Sevaux, M. (2002), *Applications of Optimization with Xpress-MP*, Eyrolles, Paris

[6] Guo, Z. X., Wong, W. K., Leung, S. Y. S., Fan, J. T., and Chan, S. F. (2006), "Mathematical and genetic optimization for the job shop scheduling problem in a mixed and multi product assembly environment: A case study based on the apparel industry," *Computers Industrial Engineering*, Vol. 50, pp. 202-219.

[7] Lauff, V. and Werner, F. (2004), "Scheduling with common due date, earliness and tardiness penalties for multimachine problems: A survey," *Mathematical and Computer Modeling*, Vol. 40, pp. 637-655.

[8] Lawler, E. L., Lenstra, J. K., Rinnooy, K. A. H. G., and Shmoys, D. B. (1993), "Sequencing and scheduling: algorithms and complexity," *Logistics of Production and Inventory* (eds: Garve, S. Rinnooy, A. H. G., and Zipkin, P.), North-Holland, Amsterdam, pp. 445-522.

[9] Lee, C. Y., Lei, L., and Pinedo, M. (1997), "Current

trends in deterministic scheduling," *Annals of Operations Research*, Vol. 70, pp. 1-14.

[10] Park., J.H. and Yosida, A. (2007), "A simplified MES implementation for small-sized manufacturing companies with Excel VBA," *Proceedings of the Joint Conference of the KORMS and the KIIE*, pp.82~89.

[11] Pinedo, M. (2002), *Scheduling: theory, algorithms, and systems*, 2nd ed., Prentice-Hall, New Jersey

[12] Suër, G. A., Pico, F., and Santiago, A. (1997), "Identical machine scheduling to minimize the number of tardy jobs when lot-splitting is allowed," *Computers & Industrial Engineering*, Vol. 33, pp. 277-280.