

A Heuristic Approach for Assembly Scheduling and Transportation Problems with Parallel Machines

Peng-Sheng You¹, Yi-Chih Hsieh², Ta-Cheng Chen³, Yung-Cheng Lee⁴

¹Graduate Institute of Marketing and Logistics/Transportation, National Chiayi University, ChiaYi 600; ²Department of Industrial Management, National Formosa University, Yunlin 632; ³Department of Information Management, National Formosa University, Yunlin 632; ⁴Department of Security Technology and Management, WuFeng University, Chiayi 621.
Email: psyuu@mail.ncyu.edu.tw, yhsieh@nfu.edu.tw, tchen@nfu.edu.tw, ycllee@wfu.edu.tw

Received 2013

ABSTRACT

Many firms have to deal with the problems of scheduling and transportation allocation. The problems of assembly scheduling mainly focus on how to arrange orders in proper sequence on the assembly line with the purpose of minimizing the maximum completion time before they are flown to their destinations. Transportation allocation problems arise in how to assign processed orders to transport modes in order to minimize penalties such as earliness and tardiness. The two problems are usually separately discussed due to their complexity. This paper simultaneously deals with these two problems for firms with multiple identical parallel machines. We formulate this problem as a mixed integer programming model. The problem belongs to the class of NP-complete combinatorial optimization problems. This paper develops a hybrid genetic algorithm to obtain a compromised solution within a reasonable CPU time. We evaluate the performance of the presented heuristic with the well-known GAMS/CPLEX software. The presented approach is shown to perform well compared with well-known commercial software.

Keywords: Heuristic Approach; Scheduling; Transportation

1. Introduction

Assembly scheduling and transportation allocation are two practical problems that industries usually face. Li et al. [1,2] indicated that components in the PC industry are usually stored in warehouses. When receiving orders from customers, PC companies must arrange their orders' processing sequence on the assembly line to assemble the components for products. Then, an order is ready to be shipped to the customer after assembly process is completed. Since each order has a due date, orders' processing sequence and transportation allocation decisions have significant impact on whether the orders can be shipped to their customers on time so as to reduce inventory holding cost, and earliness and tardiness penalties. In other words, in order to minimize overall costs, the manufacturer must consider the following two problems simultaneously: (1) production scheduling, which describes the orders' processing sequence, and (2) transportation allocation, which describes the allocation of orders to various transport modes in the scheduling period. A number of production scheduling and transportation allocation works have been separately studied [3-5]. Regarding the production scheduling problem, the pur-

pose of this research is similar to previous ones. On the other hand, regarding the transportation allocation problem, this paper focuses on the issue of how to allocate processed orders to transport modes. Researches on the integration of production and distribution have received much attention. Sarmiento and Nagi [6] provided a literature review on this area. Moreover, synchronization of production and road transportation with emphasis on vehicle routing problem also has studied by many researchers (Blumenfeld et al. [7], Chen and Vairaktaraki [8], Fumero and Vercellis [9], Lee and Chen [10]). However, limited works have studied the problem of synchronizing production and transportation allocation. Li et al. [1] investigated a single destination's air- transportation assignment problem with a single machine production floor shop. Later, this work was extended by Li et al. [2] to a cases with multiple destinations and by Li et al. [11] to a case with parallel machines in assembly. Since the problem has been shown to be NP-hard [2], Li et al. [11] dealt with the problems by decomposing it into two subproblems. In the first subproblem, they solved the transportation allocation problem by determining the transportation departure time for each order and meeting it with the assembly. In the second subproblem, they dealt with the

assembly scheduling problem, where they scheduled the orders in the assembly to meet the delivery requirements of transportation. You and Hsieh [3] dealt with an assembly and transportation allocation problem with a single machine. The objective of this paper is to simultaneously develop the decisions for production scheduling and transportation allocation under multiple parallel machines so as to minimize overall costs.

2. Model Description

A make-to-order manufacturer receives N orders from distinct customers. Among these orders, order- i 's shipping destination, order due date and order size are denoted by G_i , d_i and Q_i , respectively. Before being transported to their destinations, these orders need to be processed by assembling components on one of G_i parallel machines. The assembly sequence and release time of these N orders have a direct impact on the inventory holding cost and the selections of transportation. Inventory holding cost incurs when there is waiting time before the completed order can be shipped to its destination. We use the symbol h_i to denote the holding cost of order i per unit time. For each completed order, an order can be split and allocated to more than one carriage and delivered separately to its destination. Let F be the total number of carriages and D_f be the departure time of carriage f . The transportation cost per unit product, when allocated to carriage f is denoted by c_f . In addition to the transportation cost, an delivery earliness penalty cost exists if the arrival time of an order reaches its destination airport before its order due date, and a delivery tardiness penalty cost if the order reaches its destination airport beyond its order due date. The delivery earliness penalty cost per unit product per unit time for order- i is denoted by α_i . The purpose is to minimize total cost, which consists of inventory holding cost, the transportation cost of orders allocated into carriages, delivery earliness penalty costs, and delivery tardiness penalty. In addition to the previous assumptions, we further make the following assumptions: (1) The time and cost taken of transporting a completed order by local transportation from the manufacturer to an airport, together with the assembly setup time and cost of each order, are included in the assembly time and cost. (2) The total assembly processing time of an order is directly proportional to its quantity. (3) Business processing time and cost, together with load time and load cost of each carriage, are included in the transportation time and transportation cost. (4) Order fulfillment is considered to be achieved when the order reaches its destination on time.

The combined model aims to minimize the overall cost by determining the orders' assembly sequence and allocating orders to existing carriages. The factors which are taken into account in the model include: (a) the number

of available carriages for the planning horizon, (b) the departure and arrival time of the carriages, (c) the designated capacity and its corresponding transportation cost, and (d) the possible capacity in each carriage with its corresponding carriage cost. Prior to formulating the mathematical model, we introduce some additional notation as follows.

Notation:

k : the destination index, $k = 1, 2, \dots, K$.

B_f : the capacity of carriage f ,

G_f : the destination of carriage f ,

P_i : the processing time to assemble order i ,

A_f : the arrival time of carriage f ,

x_{if} : the quantity of the portion of order- i allocated to carriage f ,

y_{im} : 1 if order- i is assembled on machine m , 0 otherwise,

R_i : the release time to assemble order i ,

C_i : the assembly completion time of order i ,

P_{ij}^{\max} : maximum production capacity of manufacturer j for order- i ,

Z_{if} : 1 if order- i is shipped by flight f , 0 otherwise,

PI_{ijm} : 1 if the processing sequence of order i precedes processing sequence of order j on machine m , immediately, 0 otherwise.

Min

$$\sum_{i=1}^N \sum_{f=1}^{F_i} x_{if} (c_f + \alpha_i \max\{0, d_i - A_f\} + h_i (D_f - C_i)) \quad (1)$$

Subject to

$$Z_{0f} = 0, \quad \forall f \quad (2)$$

$$Z_{N+1,f} = 0, \quad \forall f \quad (3)$$

$$Z_{if} - (G_i - G_f) \leq 0, \quad \forall i, f \quad (4)$$

$$\sum_{f=1}^{F_i} Z_{if} = 1, 1 \leq i \leq N \quad (5)$$

$$(1 - Z_{if})M \geq A_f - U_i, \quad \forall i, f \quad (6)$$

$$(1 - Z_{if})M \geq C_i - D_f, \quad \forall i, f \quad (7)$$

$$x_{if} = Z_{if} Q_i, \quad \forall i, f \quad (8)$$

$$\sum_{i=1}^N x_{if} \leq Cap_f, \quad \forall f \quad (9)$$

$$C_i = R_i + P_i, \quad 0 \leq i \leq N+1 \quad (10)$$

$$R_1 = 0 \quad (11)$$

$$R_{N+1} \geq \sum_{i=1}^N P_i \quad (12)$$

$$\sum_{m=1}^M y_{im} = 1, 1 \leq i \leq N \quad (13)$$

$$PI_{iim} = 0, \quad 0 \leq i \leq N+1, \quad 0 \leq m \leq M \quad (14)$$

$$\sum_{j=1}^{N+1} PI_{ijm} = y_{im}, \quad 1 \leq i \leq N, \quad 1 \leq m \leq M \quad (15)$$

$$\sum_{j=1}^N PI_{ijm} = y_{jm}, 1 \leq j \leq N+1, 1 \leq m \leq M \quad (16)$$

$$PI_{i0m}=0, 0 \leq i \leq N+1, 0 \leq m \leq M \quad (17)$$

$$PI_{N+1jm}=0, 0 \leq j \leq N+1, 0 \leq m \leq M \quad (18)$$

$$C_j - C_i - MPI_{ij} \geq P_j - M, 0 \leq i \leq N, 0 \leq j \leq N \quad (19)$$

$$Z_{ij} \leq 1, PI_{ijm} \in \{0,1\} \quad (20)$$

3. Research Methodology

The proposed problem is a mix integer program. Due to the computational complexity of the model, no approach is guaranteed for solving the problem optimally within a polynomial time. To overcome this difficulty, we will develop a solution approach to obtain a compromised solution within a reasonable CPU time. The approach is an iteration method in which the concept of genetic algorithm (GA) and mathematical programming are considered. We refer to it as HGA. GAs are a particular class of evolutionary algorithms, which generate exact or approximate solutions to optimization and search problems using techniques inspired by evolutionary biology. For details, readers are referred to the textbook by Michalewicz [12]. The heuristic approach of HGA is addressed as follows:

First, let s_{mn} be a n -dimensional vector with n orders processed on machine m and element $s_{mn}(j)$ at j -th entry denoting the order corresponding to the j -th job processed on machine m . Note that the processing release time of a job does not necessarily happen immediately after the completion time of its previous job. Let w be a N -dimensional vector with element w_j at j -th entry denoting the idle time between order j 's release time and its proceeding job's completion time. For convenience, let $s_0=0$. Then, the relationship between $CS_{mn(j-1)}$ and $RS_{mn(j)}$ can be expressed as

$$RS_{mn(j)} CS_{mn(j-1)} + wS_{mn(j)} \quad (21)$$

At each iteration of HGA, the values of the assembly sequence s_{mn} and the idle times w are determined by GA. The values of s_{mn} are used to derived the values of PI_{ijm} , and the values of w are used to derived the values of c_i and R_i . The derived values satisfy constraints (10) to (19).

Solution procedure:

(a) Initial population:

The processing sequence is codified with N distinct integer numbers within the range of $[1, N]$. Idle time w is codified as a binary string. The first N elements of a chromosome are used to generate assembly sequence on each machine and the remaining represents the idle times.

(b) Genetic operators:

Four standard genetic operators, namely the Cloning operator, Parent selection, Crossover operator and Mutation operator repeatedly performed until the maximum

number of iterations K_{max} is reached. An elitism strategy and the Roulette-wheel selection are used in Cloning operator and parent selection, respectively.

(c) Fitness:

Note the value of the first N entries of a chromosome, (u_j, \dots, u_N) , is used to generate assembly sequence on each machine where u_j is the order appearing at the j -th position in the number sequence. The assignment of orders to machines is according to the following rule. Let

$$CP_{mj} = \sum_{j=1}^j P_j \quad (22)$$

be the minimum processed time needed to complete jobs 1 to j on machine m . In addition, let η_{mj} denote the number of orders assigned to machine m before order u_j is assigned. Then, we sequentially assign orders to the machine with minimum value of CP . The sequence of the orders on each machine also determines the values of PI_{ijm} . Moreover, according to equation (21) and idle time, we can further determines C_i and R_i . Let us refer to the model obtained by substituting the known variables of PI_{ijm} , C_i , R_i , y_{im} into the original model as model-1. Then, solving model-1 by linear programming, we can obtain x_{ij} , z_{ij} and the objective value. The objective value is viewed as fitness.

(d) Solution update:

Updating solution once the result obtained in step 3 is smaller than the objective value found up to now.

(e) Stopping criteria:

Steps (2-4) are executed repeatedly until the stop criterion is matched.

4. Experiments

The experiments was designed in terms of (K, F, M, N, T) . The combination of $(3, 10, 2, 10, 24)$ is set. Ten test cases were generated. The destination for each order and each carriage was generated from uniform distribution over $[1, F]$. The total assembly processing time of order i was set at $P_i = \gamma Q_i$. The due date of each order was generated over $[P_i + T_i, 6(P_i + T_i)]$.

The commercial software GAMS/CPLEX modeling language was adopted for the purpose of feasible solution comparison with the proposed HGA for all problem types. The proposed HGA was coded in Visual C++ 6.0 programming language. Both HGA and the GAMS/CPLEX model were implemented on an Intel Core 2 Duo personal computer equipped with a speed of 2.4 GHz and 2GB of memory. For identifying the gaps between the results obtained from CPLEX and HGA for optimal solution, the Lingo global solver was also used to solve problem type 1. For practical concerns, all algorithms were terminated if the execution time exceeded 5 hours.

Table 1. Computational result for problem type 1.

| No | Lingo | | CPLEX | | HGA | | | Sol. Gap | | |
|---------|------------|----------|----------|------|------|-------|------|----------|--------|-------|
| | GlobalTime | Sol Time | Sol Time | max | min | sigma | Time | GC | GH | |
| 1 | 1157 | 3656 | 1166 | 11.0 | 1158 | 1157 | 0.2 | 37 | 0.00% | 0.83% |
| 2 | 682 | 758 | 708 | 9.5 | 685 | 682 | 0.2 | 34 | -0.01% | 3.72% |
| 3 | 515 | 613 | 532 | 6.1 | 516 | 515 | 0.1 | 36 | -0.04% | 3.21% |
| 4 | 883 | 283 | 895 | 6.7 | 889 | 883 | 0.4 | 37 | -0.04% | 1.35% |
| 5 | 1224 | 419 | 1231 | 3.6 | 1228 | 1224 | 0.4 | 33 | -0.02% | 0.60% |
| 6 | 593 | 292 | 612 | 17.9 | 595 | 593 | 0.2 | 37 | 0.00% | 3.06% |
| 7 | 1181 | 291 | 1194 | 7.4 | 1184 | 1181 | 0.3 | 36 | 0.00% | 1.03% |
| 8 | 809 | 292 | 815 | 9.7 | 811 | 809 | 0.1 | 34 | 0.00% | 0.68% |
| 9 | 1169 | 424 | 1189 | 6.1 | 1171 | 1170 | 0.2 | 37 | -0.01% | 1.62% |
| 10 | 1258 | 280 | 1293 | 5.4 | 1263 | 1259 | 0.4 | 35 | -0.06% | 2.70% |
| Average | | | | | | | | | -0.02% | 1.88% |

The parameters for the HGA were set as follows. The process stopped when the maximum number of iteration of $K_{max}=5NM$ is reached. Test case 1 in problem type 1 was run to determine the optimal combinations of population size, maximum number of generations, the crossover rate, mutation rate and maximum number of iterations. The population size was set to be 14; the maximum number of iterations was set to be $2(J+K)$; the cloning parameter was set at 1; the crossover rate was set as 100%; the mutation rate was set as 3%; the penalty values of each constraint were set as 100.

The criteria of performances considered were the quality of the total cost and the amount of CPU time (second). The solution percentage gap, defined as $100 \times (\text{HGA or CPLEX solution} - \text{global solution}) / (\text{global solution})$ percentage points, was used to evaluate the solution quality of the HGA or CPLEX for problem type 1. The symbols of GC and GH are used to represent the solution percentage gaps between the exact solutions obtained by Lingo global solver and CPLEX's feasible solutions, and between the exact solutions and the HGA's feasible solutions, respectively. The exact solution approach using Lingo global solver was only able to solve problem type 1. However, it failed to produce global solutions for problem type 2 after 5 hours of CPU time.

For problem 1, we can see from Table 1 that HGA solutions are very close to global solutions (the gaps are no more than 0.07 percent), HGA solutions are superior to the CPLEX solutions for all instances and the deviations of the HGA solutions are very small (no more than 0.5). This reflects that HGA can be considered as a stable approach for small-scale problems. In terms of running time, we also find that HGA outperforms CPLEX.

5. Conclusions

This paper deals with the combinatorial problem of assembly scheduling and transportation allocation under multiple identical machines. A non-linear mixed integer programming model was established and a hybrid genetic

heuristic approach was developed to solve this problem in reasonable computational time. The computational experiences show that the proposed approach is comprised and efficient.

6. Acknowledgements

The authors thank the anonymous referees for their helpful comments. This research is supported by grant NSC 101-2221-E-415 -006 -MY2 (Taiwan).

REFERENCES

- [1] K.P. Li, A.I. Sivakumar, M. Mathirajan and V.K. Ganesan "Solution methodology for synchronizing assembly manufacturing and air transportation of consumer electronics supply chain," *International Journal of Business*, Vol. 9, No. 4, 2004, pp. 361-380.
- [2] K.P. Li, A.I. Sivakumar and V.K. Ganesan, "Synchronized scheduling of assembly and multi-destination air transportation in a consumer electronics supply chain," *International Journal of Production Research*, Vol. 43, No. 13, 2005, pp. 2671-2685.
- [3] P.S. You, Y.C. Hsieh and H.H. Chen, "A hybrid heuristic to a dynamic reverse logistics network with multi-commodities and components," *RAIRO-Operations Research*, Vol. 45, 2011, pp. 153-178.
- [4] M. Zuo, W. Kuo and K.L. McRoberts, "Application of mathematical programming to a large-scale agricultural production and distribution system," *J. Operational Res. Soc.*, Vol. 42. 1991, pp. 639-648.
- [5] J.M. Garcia, S. Lozano and D. Canca, "Coordinated scheduling of production and delivery from multiple plants," *Robotics and Computer-Integrated Manufacturing*, Vol. 20, No. 3, 2004, pp. 191-198.
- [6] A.M. Sarmiento and R. Nagi, "review of integrated analysis of production-distribution systems," *IIE Transactions*, Vol. 31, 1999, pp. 1061-1074.
- [7] D.E. Blumenfeld, L.D. Burns and C.F. Daganzo, "Synchronizing production and transportation schedules," *Transportation Research B*, Vol. 25, 1991, pp. 23-27.
- [8] Z.L. Chen and G.L. Vairaktarakis, "Integrated scheduling of production and distribution operations," *Management Science*, Vol. 51, No. 4, 2005, pp. 614-628.
- [9] F. Fumero and C. Vercellis, "Synchronized development of production, inventory and distribution schedules," *Transportation Science*, Vol. 33, 1999, pp. 330-340.
- [10] C.Y. Lee and Z.L. Chen, "Machine scheduling with transportation considerations," *Journal of Scheduling*, Vol. 4, 2001, pp. 3-24.
- [11] K.P. Li, A.I. Sivakumar and V.K. Ganesan, "Complexities and algorithms for synchronized scheduling of parallel machine assembly and air transportation in consumer electronics supply chain," *European Journal of Operational Research*, Vol. 187, 2008, pp. 442-455.
- [12] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd Ed, Springer-Verlag, London, UK, 1996.