Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2015, 7(10):797-803

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Improved Tabu search algorithm to optimize the scheduling problem

Zhou Cuihong

Department of Computer Science, Hunan City University, China

ABSTRACT

Improved tabu search is proposed to solve the multiproduct continuous manufacturing facilities scheduling problem. The problem is formulated as a mixed-integer linear programming (MILP) model. The mathematical model has a linear objective function to be maximized. The problem is NP-hard and finding an optimal solution efficiently is unlikely. Therefore, heuristic techniques are more appropriate to find near-optimal solutions. The computational results are compared to manual experimental results in the plant and show that the effectiveness and applicability of the proposed methods.

Keywords: Tabu search, Scheduling, Scheduling Problem

INTRODUCTION

Despite the importance of the short-term scheduling of multi-product facilities involving continuous processes, relatively few articles have been published in this area. Sequencing and scheduling is a form of decision-making that plays a crucial role in manufacturing industries. In the current competitive environment effective sequencing and scheduling has become a necessity for survival in the market-place. Production scheduling is a common problem that occurs in multi-product manufacturing facilities where a wide range of products are produced in small quantities, resulting in frequent changeovers. The industries have to meet shipping dates that have been committed to customers, as failure to do so may result in a significant loss of goodwill. They also have to schedule activities in such a way as to use the resources available in an efficient manner.

In many manufacturing and assembly facilities each job has to undergo a series of operations. Often, these operations have to be done on all jobs in the same order implying that the jobs have to follow the same route. The machines are then assumed to be set up in series and the environment is referred to as a flowshop. Setup activities are often required when switching between jobs. Applications are common in painting and plastic industries where thorough cleaning is required between operations. Similar situations are also common in textile, glass, chemical, and paper manufacturing industries, as well as some service industries [1].

The Tabu method was partly motivated by the observation that human behavior appears to operate with a random element that leads to inconsistent behavior given similar circumstances. As Glover points out, the resulting tendency to deviate from a charted course, might be regretted as a source of error but can also prove to be source of gain.. In polyamide fiber plant orders corresponding to a specific fiber type can be assigned to a set of available spin packs. The spin beam must be equipped with specific spin packs. The amount of time required to replace spin packs depends on the fiber type previously processed in the spin beam. The changed spin packs need to be cleaned assembled and reused. Replacing the spin packs can cause the changeover costs and lead to great financial losses.

And during the replacement operation the spinning machine stays idle. Obviously, it is desirable to perform changeovers aiming to minimize the amount of time the spinning machine remain idle and avoid replacing the spin packs frequently so as to reduce the cost [2].

This paper is organized as follows: In Section 2 we give the Problem Formulation and Definitions. In Section 3 we present a Mixed Integer linear Programming. The objective function of the problem is maximization of the profit. The Tabu Search algorithm is used to solve the problem in Section 4, the efficiency of the proposed model and method are illustrated through its application to the polyamide fiber plant of a real-world manufacturing facility in China. Finally, conclusions are drawn in Section 5.

Problem Formulation and Definitions

The polyamide fiber plant is a semi-continuous multiproduct plant that includes two stages: the spinning and the post-processing processes. The spinning produces intermediate products (initial fiber) which are delivered to post-processing processes, where the intermediates are converted to final products. Each stage consists of one production line that is interconnected with a fixed topology. Transition times that arise between the processing of two successive products are in the spinning stage and sequence-dependent setup times. Each stage has a fixed production capacity for each product. Moreover, the production rates of the stages can be changed within a range. The yields of production depend on the production rates in the stages.

In many realistic problems, setup times depend on the type of job just completed as well as on the type about to be processed. In those situations, it is not valid to absorb the setup time for a job in its processing time, and explicit modifications must be made. The time interval in which job j occupies the machine is expressed s_{ij} + p_j, where i is the job that precedes s_{j0} in sequence, s_{ij} is the setup time required for job j after job i is completed, and p_j is the amount of direct processing time required to complete job j. Setup times that are sequence dependent are commonly found where a single facility produces several different kinds of items, or where a multipurpose machine carries out an assortment of tasks. The use of a single system to produce different chemical compounds may require that some amount of cleansing be carried out between process runs on different compounds, to ensure that tolerably low impurity levels are maintained. Sometimes, the extent of the cleansing depends on both the chemical most recently processed and the chemical about to be processed. Similar setup properties can be found in the production of different colors of paint, strengths of detergent, and blends of fuel. The same observations apply to certain assembly lines where retooling, inspection, or rearrangement of work stations could represent the setup activity.

In many industrial settings, the assignment of job scheduling in the previous horizon may not be suitable for the current horizon due to urgent job requirements and special setup times. In such cases, it may be necessary to move jobs from one machine to another one. This involves the possible extra sequence-dependent setup times (e.g., cleaning times) plus changeover costs. In principle, any family scheduling model can be viewed as a single-machine model with sequence-dependent setup times.

Sequence Dependent Changeover Times: In practice, there is often a changeover time $\tau_{ijm} \in$ Z≥0 between the execution of two successive activities i and j on one and the same unit of renewable stage m, e.g., for cleaning that unit. During the changeover, that resource unit is not available for processing an activity. The changeover time generally depends on the sequence of activities i, j. In the case study, if operation j executed on processing unit (pool) m produces a product whose quality is higher than that of the output product(s) of operation i , then τ_{ijm} equals the cleaning time of stage m; otherwise, τ_{ijm} = 0. Also, if i is the last operation on stage m, there is a positive cleaning time τ_{ijm} n+1.

Sequence dependent setup times (s_{ijk}) : The s_{jk} represents the sequence dependent setup time that is incurred between the processing of jobs j and k; s_{0k} denotes the setup time for job k if job k is first in the sequence and s_{j0} the clean-up time after job j if job j is last in the sequence (of course, s_{0k} and s_{j0} may be zero). If the setup time between jobs j and k depends on the machine, then the subscript i is included, i.e., s_{ijk}. If no s_{jk} appears in the β field, all setup times are assumed to be 0 or sequence independent, in which case they are simply included in the processing times.

Mathematical mode

In this classic combinatorial problem, a salesperson must visit a set of cities in a sequence that minimizes the overall cost. Analogously, in a flowshop plant, a set of products must be made in such an order that minimizes transition and other costs. The classic TSP problem formulation uses binary variables to represent the transition from one city to another. In the present formulation, the binary variables z_{ij} are used to represent a transition from product i to product j of the flowshop plant. The products sequencing is completely defined by assigning values of 0 and 1 to the z_{ij} variables [12, 13].

Scheduling Model: The criterion used is minimization costs, which includes sum of holding inventory costs and transition costs for each product. The MILP model for the scheduling problem with fixed operating conditions is as follows.

Objective: Maximization of the average profit

$$\text{Max}\sum_i P r_i W_{iM} - \sum_i \sum_j Ctr_{ij} Z_{ij} - \sum_i C_i I_i - \sum_i r_{im} P_i' W'_{iM} \tag{1}$$
$$Ts_{im} + Tp_{im} \le Ts_{i,m+1} + Tp_{im+1}, \qquad \forall i, m = 1,2$$

Subject to the following constraints:

Sequencing constraints

$$\sum_i \mathrm{Z}_{ij} = 1, \quad \forall j \tag{2}$$

$$\sum_j Z_{ij} = 1, \quad \forall i \tag{3}$$

Transition cost constraints

$$Ct \ge \sum_i \sum_j z_{ij} Ctr_{ij} \tag{4}$$

Timing constraints

$$Ts_{im} \le Ts_{i,m+1}, \qquad \forall i, m = 1,2 \tag{5}$$

$$Ts_{im} + Tp_{im} \le Ts_{i,m+1} + Tp_{im+1}, \qquad \forall i, m = 1,2 \tag{6}$$

$$Tc \ge \sum_i Tp_{im} \;\; + \sum_i \sum_j \tau_{ijm} z_{ij}, \qquad \forall m \tag{7}$$

Mass balance constraints

$$W_{im} = \gamma_{im} Tp_{im}, \quad \forall i, m \tag{8}$$

$$W_{im} = \alpha_{i,m+1} W_{i.m+1}, \; \forall i, m = 1,2 \tag{9}$$

Demand constraints

$$d_i T p_i = I_i, \quad \forall i \tag{10}$$

$$I_i > 0 \tag{11}$$

The (1) is defined as the sum of sales revenue, the transition cost, the inventory cost for the products and the average inventory cost for the final product. The first is the sales revenue, the second is transition cost. According to constraint 2, exactly one product j follows product i, and according to constraint 3, exactly one product i precedes product j. Constraint 4 represents transition costs incurred in the schedule. Constraint 5 ensures that a product cannot be processed in a stage prior to production having started in the previous stage. Similarly, constraint 6 ensures that production of any product cannot end in any stage before processing in the previous stage has ended. Constraint 7 states that the cycle time of the plant is the longest of all stage cycle times. Constraint 8 and 9 state the mass balance constraints. Constraint 10 states that the demand must be satisfied for all products in the plant and that production can be exceeded.

The Tabu Search Algorithm

A popular aspiration criterion is that the target function value be the best ever seen. If this is the case, it is obvious that this solution has never been encountered before. This is the reason for accepting the solution, although it is forbidden by the tabu list.

In this section, we describe our implementation of the tabu search approach for solving the flow shop sequencing problem for the polyamide fiber plants. The implementation of each element of the tabu search approach is now discussed.

4.1 Initial solution

To get an initial starting solution, we considered two heuristics: a method due to Nawaz, Enscore and Ham (NEH) [16].

NEH algorithm: This algorithm builds the final sequence in a constructive way, adding one product at a time, as follows:

Step 1. decreasing total transition time on the machines.
Step 2. Consider the first product and schedule
Step 3. For k = 3 to n do.
Step 4. The current partial sequence contains k product. Insert the k-th product at the position.

This algorithm is based on the assumption that a product with high total processing time should be given higher priority than a product with lower total transition time.

4.2 Neighborhood structure

In the work, a neighbor of a solution is obtained by swap and inserting into an allowed product sequence.

Neighborhood structure: In the basic Tabu search for discrete optimization the neighborhood of an iteration point is built by all the direct neighbors of this point.

Given a sequence s, we define N(s) as being the set of all sequences which can be obtained from s using one of the following schemes:

Swapping: given a sequence s, let i and j be two positions in the sequence s. A neighbor of s is obtained by interchanging the products in positions i and j. The positions i and j can be specified in one of two ways: Positions i and j are selected randomly; or they are enumerated in some systematic way such as adjacent pairwise interchange.

Insertion: given a sequence s, let i and j be two positions in the sequence s. A neighbor of s is obtained by inserting the product in position i in position j. The positions i and j can be specified in one of two ways: Positions i and j are selected randomly; or they are enumerated in some systematic way such as inserting every product in every position.

4.3 The tabu list (T)
The size of the tabu list is a very important parameter of a tabu search algorithm. T will contain the best trial sequences selected during the perturbation actions. There will be a list for each machine with a size of 9. Glover [10] who developed the generic technique of TS recommended T of 5 to 12 elements. In our experiments the size of tabu list is 3, 4, 5, 6, 7, 8 and 9 respectively.

4.4 Aspiration criterion
While the search is proceeding, several regions of the search space are classified as Tabu. In some cases, the best neighbor solution may lie in a Tabu area where its objective function value is better than the current best value. In this situation the Tabu property can be invalidated and the point will be chosen. This feature is necessary to enforce faster convergence to a good local minimum but it also might bias the search. If it happens the escape mechanism will be started after some time and the search is forced to leave this region and to explore a new one.

In order to override the Tabu list when there is a good Tabu move, the following aspiration criterion issued: the Tabu move is accepted if it produces better solution than the best obtained so far.

4.5 Stopping criteria
The algorithm stops when the number of iterations moves exceeds a specified constant 99, or the number of iterations made without improving the current solution exceeds a specified constant 16.

The Tabu Search Algorithm is generated using the following procedure:
Step 1. Choose an initial solution s $\in$ S;
Step 2. best := c(s);
Step 3. s*:= s (the current solution);
Step 4. Tabu-list:= φ;

REPEAT
Step 5. Cand(s) := {s_ $\in$ N(s) (N(S):the neighborhood of S) | the move from s to s_ is not tabu OR s_ satisfies the aspiration criterion };
Step 6. Generate a solution s $\in$ Cand(s);
Step 7. Update the tabu list;
Step 8. s_:= s;
Step 9. IF c(s) < best THEN

BEGIN
Step 10. the best-known solution s*:= s;
Step 11. best := c(s)

END
UNTIL stop criterion
Applied to the optimization problems, seven different products are processed; the Tabu search algorithm starts at some initial solution and then moves to a neighboring solution. A neighboring solution is generated by a set of admissible moves. At each iteration, the method moves to the best solution in the neighborhood of the current solution.

EXPERIMENTS SECTION

The relevant experiment data from a large-scale petrochemical plant is as Table1, Table2, Table 3 and Table 4.

Table1.The relevant data 1

product	Product Rates[kg/hr]	Demand [ton]	unit inventory cost[RMB yuan/ton]
A	560	600	0.062
B	760	667	0.674
C	700	870	0.678
D	900	240	0.480
E	500	400	0.524
F	400	450	0.660
G	300	450	0.652

Table2.The relevant data 2

Material price [104yuan/ton]	Product price [104yuan/ton]	Unit cost [kg/kg]
4.1	4.5	1.051
5.5	5.4	1.045
6.0	3.4	1. 458
7.0	6.7	1.566
7.4	5.6	1.765
6.0	5.6	1.456
4.5	6.5	1.675

Table3. Transition times

Transition time [hrs]	A	B	C	D	E	F	G
A	0	5	5	6	7	6	4
B	9	7	6	4	8	6	4
C	9	7	0	7	8	5	5
D	4	9	5	0	5	5	5
E	6	9	4	4	0	5	5
F	5	0	4	4	4	0	5
G	7	6	5	5	5	6	0

Table4. Transition costs

Transition costs [RMB yuan]	A	B	C	D	E	F	G
A	0	7974	11374	1650	7780	6910	7974
B	6664	0	11374	1650	8750	6910	3214
C	6814	7974	0	1650	6510	10510	6830
D	3214	3214	8866	0	5310	10510	6830
E	6814	6830	11374	8350	0	10510	8465
F	6814	6830	11374	8350	8846	0	8465
G	3214	7974	11374	1650	8910	10510	0

The result is obtained on a Pentium 4 PC (3.0 GHz) with VC++6.0. The optimization solution is AECGBDF, the max profit is 16307400 yuan RMB.

CONCLUSION

In this paper, a model is developed to address the scheduling of various kinds of products in a semi-continuous multiproduct plant. The model is formulated as a mixed integer linear programming problem. A mixed-integer programming model based on continuous time representation is provided for solving the production scheduling problem optimally. The model can be regarded as a semi-continuous multiproduct problem, where inventory capacities are considered and more than one product is obtained in a plant composed of two stages, which are interconnected by intermediate warehouse. A TS algorithm has been developed to obtain a better performance schedule.

REFERENCES

[1] J.Ham; D.Lee; L.Saul, Proc. 10th Int'l Workshop Artificial Intelligence and Statistics, page 120-127, **2005**.
[2] A.P.Shon; K.Grochow; A.Hertzmann; R.Rao, In Proc. NIPS, page 1233–1240, **2006**.
[3] S.Lafon; Y.Keller; R.R.Coifman, *IEEE PAMI*, 28(11): 1784-1797, **2006**.
[4] Z.Zhang; H.Zha, *SIAM J. Sci. Comput*. 26 (1): 313–338, **2004**.
[5] X.Bai; H.Yu; E.R , Proc. Int'l Conf. Pattern Recognition, page 398-401, **2004**.
[6] M.Gori; M.Maggini; L.Sarti, *IEEE PAMI*, 27(7): 1100-1111, **2005**.
[7] Y.Keselman; A.Shokoufandeh; M.F.Demirci; S.J. Dickinson, Proc. Conf. Computer Vision and Pattern Recognition, page 850-857, **2003**.
[8] J.Black; M.Gargesha; K.Kahol; P.Kuchi; S.Panchanathan, ITCOM, Internet Multimedia Systems II, Boston, July **2002**.
[9] G.Little; S.Krishna; J.Black; S.Panchanathan, ICASSP **2005**, Philadelphia, **2005**.