IEMS Vol. 10, No. 1, pp. 1-6, March 2011.

Batch Scheduling Problem with Multiple Due-dates Constraints

Shintaro Mohri†
Facluty of Economics, Kobegakuin University Kobe 651-2180, Japan
Tel: +81-78-974-5361, E-mail: mohri@eb.kobegakuin.ac.jp

Teruo Masuda
Faculty of Business Administration
Tezukayama University Nara, 631-8501, Japan

HiroakiIshii
Faculty of Business Administration
KwanseiGakuin University Sanda, 662-8501, Japan

Received, January 17, 2011; Revised, February 18, 2011; Accepted, February 21, 2011

Abstract. This paper describes the issue of batch scheduling.In food production, the lead-time from produc-tion to sale should be decreased becausefreshness of the product is important. Products are shipped at diverse times depending on a demand of sellers, because the types of sellers has become diversified such as super-markets, convenience stores and etc. production of quantity demanded must be completed by time to ship it then. The authors consider a problem with due-dates constraints and construct the algorithm to find the opti-mal schedule that satisfy the due-dates constraint, batch size constraint, inventory time constraint and mini-mize total flow time.

Keywords: Batch Scheduling, Dynamic Programming, Food Production

1. INTRODUCTION

A batch scheduling problem arises in the various types of manufacture. It is necessary to manage process of manufacture of food and the distribution channel by food manufacturing industry. The other hand, Japanese lifestyle changed, and an action and a demand of a consumer diversified. For example, at production of breads, there is diverse demand of the time to receive products at the retail store, so that taste of bread becomes deterioration when pass through long time from production. But retail stores include diverse type from supermarkets to convenience stores, and their business hours are different, and the time when products are sold by each store is different. In a bread factory, the products ship several times depending on demand for retail store on a day. They must produce quantity demanded by shipment on the appointed time of delivery. On the other hand, the production of breads is handled as batch process. Beginning of this process is mixing materials such as wheat or sugar. The materials are different by a kind of bread. After this process, this mixed material is treated as a unit of batch. We think about the issue of scheduling to minimize total flow time under such situation.

The batch scheduling problem on single machine was studied for various objective function by Monma and Potts (1989). Herrmann and Lee (1995) proposed genetic algorithm for single machine batch scheduling problem with due-dates. There are machine-independent family setup times. The other batch scheduling problem is surveyed by Potts *et al.* (1992, 2000). In this paper, we consider single type jobs batching problem with multiple due-dates constraint. Let s be a setup-time of batch and n be a number of the jobs in batch. The processing time of n jobs batch is defined as the function $p(n) = s + pn$ for n. We consider the batch problem with due-date constraints such as producing the demanded quantity of product should be completed before each due-date. And we propose the algorithm to construct the schedule minimizing total flow time from Dynamic Programming and introducing graph representation.

† : Corresponding Author

2. PROBLEM FORMULATION

In this section, we define some symbols and formulate our problem.

n : Number of jobs.

s : Setuptime of batch.

$p(m) = s + pm$: Processing time of m jobs.

b_j : jth batch.

$|b_j|$: Number of jobs included in jth batch.

C_j : Completion time of batch b_j.

The schedule S including n jobs are defined as a vector $(b_1, \cdots, b_k)$. is the number of batches, and all batches are processed in order of index. The total flow time of schedule $f(S)$ is shown as follows:

$$f(S) = \sum_{j=1}^{k} C_j |b_j| = \sum_{j=1}^{k} \left\{ \sum_{l=1}^{j} p(|b_l|) \right\} |b_j| \qquad (1)$$

The purpose of this problem is to find the schedule that is minimizing total flow time $f(S)$.

If there is no limitation for the maximum number of jobs included one batch, or maximum number of batches, the following equation holds.

$$f(S_n^*) = \min_{|b_1|=1,\cdots,n} \left\{ p(b_1)n + f(S_{n-|b_1|}^*) \right\} \qquad (2)$$

$$b_j = b'_{j-1} \quad (j = 2, \cdots, k) \qquad (3)$$

$$f(S_0^*) = 0 \qquad (4)$$

We can decide the all size of batches using dynamic program approach. For considering additional constraints, a definition of the graph is introduced. The following definition of the graph $G(V, E)$ is for constructing the algorithm for this problem. Let i be the number of jobs and B be the number of batches. Let $v(i, B)$ be the vertex, where B is less or equal to i and $e(v(i_2, b_2), \ v(i_1, b_1))$ be the directed edge from $v(i_2, b_2)$ to $v(i_1, b_1)$. E, V is defined as follows.

$$V = \{ v(i, B) \, | i = 1, \cdots, n, \, B = 1, \cdots, n \} \cup v(0, 0) \qquad (5)$$

$$E = \{ e(v(i_2, b_2), v(i_2, b_2) \, | \, b_2 = b_1 + 1, i_1 < i_2 \} \qquad (6)$$

The function $f^*(v(i, B))$ and $C(v(i, B))$ are defined for a vertex $v(i, B)$ as follows:

$f^*(v(i, B))$ is the minimum total flow time of the schedule that consist of the i jobs B batches.

$C(v(i, B))$ is the completion time of the schedule that consist of the i jobs B batches i.e. $Bs + ip$.

The schedule $S(B, n)$ is defined as a directed path from $v(n, B)$ to $v(0, 0)$ on the graph $G(V, E)$ A batch b_k is a directed edge $e(v(b_1 + 1, i_2), v(b_1, i_1))$ and the size of batch $|b_k|$ is $|b_k| = i_2 - i_1$. A schedule is shown as Figure 2 on this graph. We showed the algorithm to construct the optimal schedule by this graph representation (Mohri, 2007). Let n be the number of jobs. Graph G (V, E) is constructed by this definition. If there are no additional constraints, Algorithm Icalculate the optimal paths from vertexes $v(n, k)$, where $k = 1, \cdots, n$ to vertex $v(0, 0)$. The optimal schedule is obtained as the optimal path from these paths.

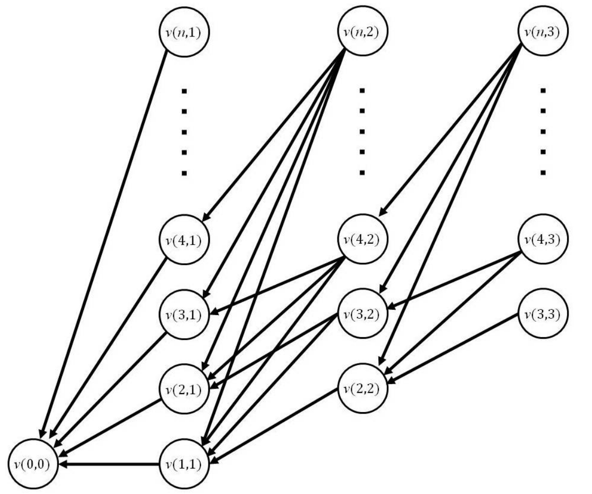
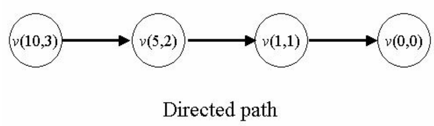


Figure 1. Graph $G(V, E)$.



10 jobs, 3 batches schedule			
	1st batch b_1	2nd batch b_2	3rd batch b_3
size of batch	1	4	5

Figure 2. Example of the Schedule.

Algorithm I

input: Graph $G(V, E)$ which is correspond to n jobs problem

output: Optimal batch sizes

f(S$_i$) = 0

For i = 1 **to** n **begin**
  for j = 1 **to** i **begin**
Calculate $f(v(i, j))$.
  Select the vertex $v(i_{j-1}, j-1)$ that minimize $f(v(i, j))$ for vertex $v(i, j)$.
  Select the vertex $v(n, B)$ in vertex set $v(i, j)$ $(j = 1, 2, \cdots, n)$ that minimize total flow time $f(v(i, j))$.
  Calculate each batch size from the path from $v(n, B)$. to $v(0, 0)$.
 end
end

By this algorithm, the optimal schedule $S_{B,n}$ is the directed path from $v(n, B)$ to $v(0, 0)$. batch b_k is directed edge $e(v(b_{1+1}, i_2), v(b_1, i_1))$. The batch size $|b_k|$ is $|b_k| = i_2 - i_1$..

3. PLURAL DUE-DATES CONSTRAINTS CASE

In this section, the following situation is considered. There are the plural due-dates, and constraints to complete the work of quantity demanded. Let $D = \{d_1, d_2, \cdots, d_m\}$ be the due-dates set. And demanded quantity $|d_k|$ is assigned for each due-date d_k. The graph described in section 2 is formed from this problem. The vertex is classified for each due-dates d_i into four subsets by following rule:

(1) $V_1 = \{v |$ for any due-date d_k, $f(v)$ less than or equal d_k and i greater than or equal $|d_k|\}$, i.e. for due-date d_k, the vertex v satisfy both time constraints and quantity constraint.
(2) $V_2 = \{v |$ for any due-date d_k, $f(v) > d_k$ and $i < |d_k|\}$, i.e. For due-date d_k, the vertex v satisfies neither time constraints nor quantity constraint.
(3) $V_3 = \{v |$ for any due-date d_k, $f(v)$ less than or equal d_k and i less than $|d_k|\}$, i.e. for due-date d_k, the vertex v satisfy time con-straints, but does not satisfy quantity constraint.
(4) $V_4 = \{v |$ for any due-date d_k, $f(v)$ and $i < |d_k|\}$, i.e. for due-date d_k, the vertex v satisfy quantity constraint but does not satisfy time constraint.

Let S be a schedule and G_S be the graph for S. We consider following cases.

(A) The graph G_S includes the vertex v that is classified into V_1 for due-date d_k. Schedule S satisfy the constraint for due-date d_k.
(B) The graph G_S includes the vertex v that is classified into V_2 for due-date d_k. Schedule S does not satisfy the constraint for due-date d_k, because function $f(v)$ is non-decreasing function.
(C) There is the path that includes the vertex v that is classified into V_3 for due-date d_k. Because function $f(v)$ is nondecreasing function, the vertex that satisfy the due-date constraint does not exist after d_k.
(D) There is the path that includes the vertex v that is classified into V_4 for due-date d_k. Because function $f(v)$ is non decreasing function, the vertex that satisfies the due-date constraint does not exist before d_k.
From these cases, the following theorem is hold.

Theorem 1: *Schedule **S** is not satisfy due-date constraint if one of the following cases is hold:*
(1) The path does not contain vertex v that belongs to V_1.
(2) The path contains vertex v that belongs to V_2.
(3) The path contains the edge that is from any vertex v_4 *to* v_3. *Here* v_4 *belongs to* V_4 *and to* v_3 *belongs to* V_3.

Proof: It is clear from above cases. From this theorem, to construct a feasible schedule, Graph $G(V, E)$ modify by following processes:
(1) Delete all vertexes v that belongs to vertex set V_2, and edge which initial vertex or terminal vertex belongs to vertex set V_2.
(2) Delete all edges that initial vertex belong to vertex set V_4 and terminal vertex belongs to vertex set V_3.

From above argument, we can construct the algorithm to obtain optimal schedule.□

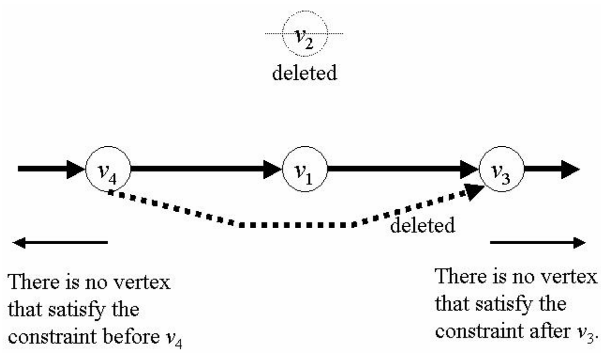


Figure 3. Vertexes and Feasible Path.

Algorithm II constructs the optimal schedule under due-date constraint. Let the number of jobs n, setup time s, processing time *pare* are given. And set of due-date and quantities that correspond to each due-date are given.Algorithm II deletes the vertexes and edges based on Theorem 1, and construct the optimal path P that satisfy the due-date constraint.

Algorithm II
input: Graph $G(V, E)$ which is correspond to n jobs problem, due-date constraint which including due-dates and quantities.
output: Optimal batch sizes from optimal path

Calculate the completion time $G(v)$ for all vertex v.

for i = 0 **to** m **begin**
For each due-date d_k, delete the edge
Delete all vertexes v that belongs to vertex set V_2, and edge which initial vertex or terminal vertex belongs to vertex set V_2.
Delete all edges that initial vertex belongs to vertex set V_4 and terminal vertex belongs to vertex set V_3.
end
for i = 0 **to n begin**
if (i = 0) **then**
$f(S_i) = 0$

else begin for j = 1 **to** *i*
Calculate $f(v(i, j))$.
Select the vertex $v(i_{j-1}, j-1)$ that $e(v(i, j), v(i_{j-1}, j-1)$ exists and minimize $f(v(i, j))$ for vertex $v(i, j)$.
end
end

Let B_s be the number of batches that satisfy the due-date constraint. Let $V(B_s)$ be the vertex set that is the collection of vertex that the number of batches is less than or equal to B_s. Select vertex $v(n, B)$ that minimize total flow time in the set $V(B_s)$. Calculate batch size from directed path from $v(n, B)$ to $v(0, 0)$.

We can construct the optimal schedule $S_{B,n}$ that satisfy all due-date constraints and minimize total flow time by applying **AlgorithmII**.

4. BATCH SIZE CONSTRAINT

In this section, we consider multi-criteria problem that introduce an objective functions for batch size. If batch size is too large or too small, such batch is infeasible because equipment restriction or cost matter. Before considering multi-criteria model, we consider the problem that batch size is treated as restriction. Let B_H be the maximum size of batch size and B_L be the minimum size of batch size. The sizes of all batches are restricted as $B_L \le |B_j| \le B_H$ for all *j*. For this problem, let $G(V, E)$ be the representation graph. At the graph *G*, the each batch sizes are show as each edge $e \in E$ with connected vertexes. The schedule of problem is representation as the path on the graph *G*. Let *P* be such representation path. And if the path *P* includes the edges that violate the constraint for batch size, the schedule of *P* violates the constraint for about batch size. Therefore to construct the feasible schedule, the path is constructed on the graph that is not including the edges which violate the batch sizes.

Let *e* be the edge that connected vertexes $v_1(i_1, B_1)$ and v_2 (i_2, B_1+1) such that $i_1 < i_2$. The batch size of *e* is $i_2 - i_1$. To construct the feasible schedule, delete the all edges that violate the batch size restriction and connected vertexes that are not able to configure the feasible path. Deleting vertexes are that all edges of same direction are removed. If a vertex is removed, all connected edges are also removed. The feasible optimal schedule is constructed on this graph.

Now we consider multi-criteria problem such that constriction of batch size is not strictly, i.e. the violation of the batch size restriction is undesirable for production cost or quality of product, however it is not unfeasible. Then we consider total flow time and batch size as objective function. To consider this problem, all edges are ordered by suitability. Edges and concerned vertexes are deleted in this order, and the feasible schedule is constructed at each steps. Algorithm III constructs the pareto optimal schedules. To construct schedule, we define the desirability of the number of jobs in batches. And let all candidates of batch size be listed sequentially in order to desirability. And all batch size.

AlgorithmIII
input: Graph $G(V, E)$ which is correspond to *n* jobs problem, Evaluation of batch size desirability
output: Optimal batch sizes

Order the all edges by desirability.
Delete all edges that not acceptable for batch size. Delete the vertexes that concern deleting edges in the same way as batch size restriction case.
For each edge *e* in order **begin**
Delete edge *e* and vertexes that concern deleting edges.
Construct the feasible schedule by Algorithm II
end

5. INVENTORY TIME CONSTRAINT

In the field of food production is unable to maintain the freshness of the product when there is a long time difference from the complete production and shipment. It is important to reduce shipping time to time from the completion time of manufacture to maintain product freshness. This section therefore introduces a new constraint about difference between completion time of production and due-date.

For this problem, we defined the due-date subset D_{b_k} which corresponding to a batch b_k. For D_{b_k}, the demand quantity of D_{b_k} is satisfied by completing processing a batch b_k. Also let $d_{b_k}^{\min}$ and g_{b_k}, $d_S^{\min}$ are defined as follows:

$$D_{b_k} = \left\{ d_i \left| \sum_{l=1}^{k-1} |b_l| < |d_l| < \sum_{l=1}^{k} |b_l| \right. \right\} \tag{7}$$

$$d_{b_k}^{\min} = \min_{d_l \in D_{b_k}} \{d_l\} \tag{8}$$

$$g_{b_k} = C_k - d_{b_k}^{\min} \tag{9}$$

$$d_S^{\min} = \min_{all\, k} \left\{ C_k - d_{b_k}^{\min} \right\} \tag{10}$$

The constraint of this problem is subject to

$$d_S^{\min} \le g_{\max} \tag{11}$$

where $g_{\max}$ is maximum difference time between completion time of production and due-date.

At the graph $G(V, E)$ of this problem, all batches are assigned to edges. If a path for a schedule includes the edge that violate this constraint. Let edge be the edge that connected vertexes $v_1(i_1, B_1 - 1)$ and $v_2(i_2, B_1)$.

If $g_{B_1} > g_{max}$ then the batch b_1 violate the constraint. To construct the feasible schedule, delete the all edges that violate the restriction and connected vertexes that are not able to compose the feasible path. Deleting vertexes are all edges of same direction are removed. If a vertex is removed, all connected edges are also removed. The feasible optimal schedule is constructed on this graph.

6. NUMERICAL EXAMPLES

In order to show the applicability of the proposal method, we solve simple case problem for example.

6.1 Due-dates constraint case

Problem definition (Example 1.)
setup time: $s = 2$
processing time: $p = 1$
Due-dates and quantities are defined as Table 1.

Table 1. due-dates and quantities.

	d_1	d_2	d_3
due date	6	14	15
quantity	4	2	2

For this problem, the vertexes are deleted based on Algorithm II and we can obtain the optimal path. Figure 4 shows the Graph and the optimal path. Table 2 shows the feasible paths and objective values.

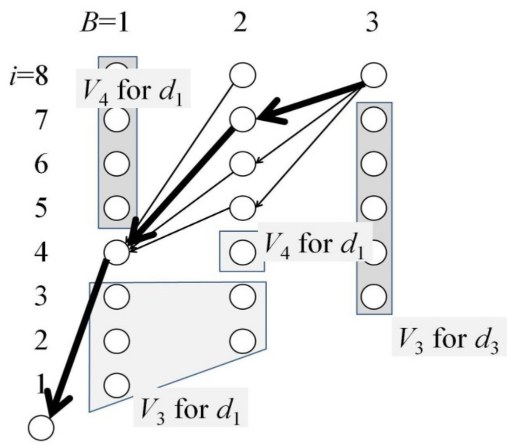


Figure 4. Graph $G(V, E)$.

Table 2. Paths and objective values.

Path	Objective value
$v(8, 3) - v(7, 2) - v(4, 1) - v(0, 0)$	47
$v(8, 3) - v(6, 2) - v(4, 1) - v(0, 0)$	48
$v(8, 3) - v(5, 2) - v(4, 1) - v(0, 0)$	51
$v(8, 2) - v(4, 1) - v(0, 0)$	48

The optimal path is $v(8, 3) - v(7, 2) - v(4, 1) - v(0, 0)$ and the optimal batch sizes are $(|b_1|, |b_2|, |b_3|) = (4, 3, 1)$.

6.2 Batch size constraint case

We suppose batch size constraint case in addition to previous problem. The constraint is defined as that any batch size must be more than 1. In Figure 4, edge $e(v(8, 3), v(7, 2))$ and $e(v(5, 2), v(4, 1))$ violate the constraint. Therefore by deleting the edges and connected vertexes, we obtain following path.

Table 3. Paths and objective value.

path	objective value
$v(8, 3) - v(6, 2) - v(4, 1) - v(0, 0)$	48
$v(8, 2) - v(4, 1) - v(0, 0)$	48

In this case both paths are optimal. the optimal batch sizes are $(|b_1|, |b_2|, |b_3|) = (4, 2, 2)$ or $(|b_1|, |b_2|) =$ (4, 4).

6.3 Inventory timeconstraint case

We suppose Inventory timeconstraint case in addition to Example 1. The constraint is defined as any inventory time must be less than or equal to 3. We can calculate the completion time and the inventory time for each vertex (Table 4).

Table 4. Inventory time for vertexes.

	Completion time	Inventory time	
$v(4, 1)$	6	0 for d_1	
$v(5, 2)$	9	5 for d_2	not allowable
$v(6, 2)$	10	4 for d_2	not allowable
$v(7, 2)$	11	3 for d_2 , 4 for d_3	not allowable
$v(8, 2)$	12	2 for d_2 , 3 for d_3	
$v(8, 3)$	14	0 for d_2 , 1 for d_3	

In Table 4, Vertexes $v(5, 2)$, $v(5, 2)$ and $v(7, 2)$ are not allowable for including path. Therefore there is no path from vertex $v(8, 3)$ and only path $v(8, 2) - v(4, 1) - v(0, 0)$ is feasible and optimal.

7. CONCLUSION

This paper considers the batch scheduling problem with due-date constraints such as producing the demanded quantity of product should be completed before each due-date and construct the algorithm to solve this problem and construct the schedule minimizing total flow time from Dynamic Programming and introducing graph representation. Furthermore we introduce batch size restriction and show the algorithm that constructs the feasible pareto optimal schedule.

Acknowledgement

One of authors, Teruo Masuda, was supported by the Tezukayama Gakuen Education and Research Fund.

REFERENCES

Herrmann, J. W. and Lee, C.-Y. (1995), Solving a class scheduling problem with a genetic algorithm, *ORSA Journal on Computing*, **7**, 443-452.

Mohri, S., (2007), Batch Scheduling Problem with due-dates Constraints, *Asia Pacific Management Review*, **13**, 129-132.

Monma, C. L. and Potts C. N. (1989), On the complexity of scheduling with batch setup times, *Operations Research*, **37**, 798-804.

Potts, C. N. and Kovalyov, M. Y. (2000), Scheduling with batching: A review, *European Journal of Operational Research*, **120**, 228-249.

Potts, C. N. and Van Wassenhove, L. N. (1992), Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity, *Journal of Operational Research Society*, **51**, 199-209.