International Symposium on Scheduling 2006
July 18 - 20, 2006, Tokyo, Japan
6B3

A UNIFIED MODELING AND SOLUTION PRINCIPLE FOR FINE SCHEDULING

Kenji Muramatsu
Department of Management Systems Engineering
School of Information Science and Engineering
Tokai University
1117 Kita-Kaname, Hiratsuka, Kanagawa, 259-1292, Japan
kenji@keyaki.cc.u-tokai.ac.jp

Abstract

This paper presents a unified method for formulating a multi-item multi-process dynamic lot size scheduling problem and its extensions into fine mathematical models. Then, the paper refers to its global optimization oriented solution principle, which is based on Lagrangian decomposition coordination method together with heuristics.

In modeling, first, we derive a dynamic equation of processing of an item and the accompanied work-in-process stock transition. It is described by use of "echelon inventory" so as to ensure additively separable property of the model and enable its decomposition.

Then, we guarantee feasibility of processing on a machine. Placing the inequality constraint to interdict machine interference attains it. Last, we integrate all of the processing over the whole processes. Placing the one to interdict work-in-process stock shortage also attains it.

Further, in the extended problem, our finding is that besides the constraints stated above there exists some additional restrictions unique to the problem, which specify the operation, whether it is real processing or set up, and yet define relative states among multiple system elements. Then, instead of formulating those directly, we introduce imaginary items and their work-in-process once and then place the constraints to interdict excess and shortage of them under some additional assumptions. This gives a means for solution.

Keywords: scheduling, unified approach, global optimization, dynamic lot sizing, Lagrangian decomposition coordination method, object oriented optimization technology.

1. INTRODUCTION

1.1 Objective of the Study

The paper presents a unified method for formulating a multi-item multi-process dynamic lot size scheduling problem (MIMPDLSSP) and its extensions into fine mathematical models. The fine scheduling in this paper means that scheduling technology enables the potential of high resolution, time-variant or real time nature, and global optimization orientation.

Obviously, production is a collection of various processing on machines and processes, and yet it is accompanied by work-in-process stock transition. Additionally, most of the production problems belong to the MIMPDLSSP and its extensions.

Therefore, the fundamental of fine scheduling is to enable us to deal with the dynamic equation of the processing and its work-in-process stock transition in explicit in a model, and yet do it over the whole system. Accordingly, in order to guarantee feasibility of processing and synthesis of all processing over it, some other modeling phases follow this.

Further, in each extended problem, there exist some restrictions unique to the problem besides the constraints stated above. Usually, they are placed on the operations whether it is real processing or set up. Furthermore, they are described as the complicated relations among system elements such as machines, machine units, or subsidiary resources. Accordingly, even if it is able to formulate them

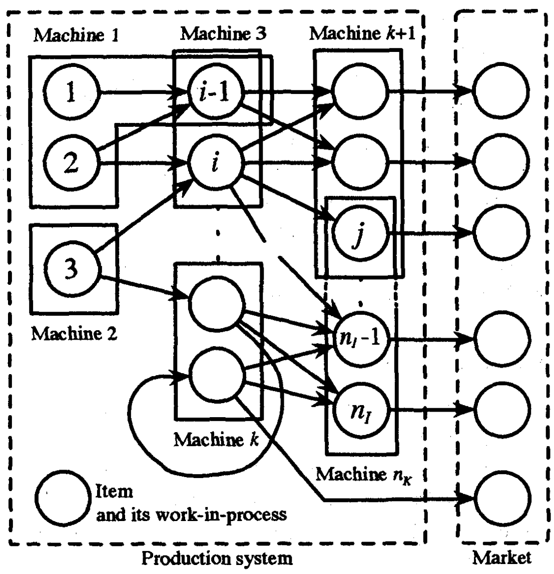

Fig. 1 System diagram of Multi-Item Multi-Process Dynamic Lot Size Scheduling Problem. Arc denotes move of item.

directly, the means for satisfying them has not been known. Therefore, instead of direct formulation, we are obliged to devise any effective means.

In solution, the scale of the problem becomes so large that problem decomposition is inevitable. Then, we refer to a global optimization oriented solution principle that is based on Lagrangian decomposition coordination method (abr. LDC method) together with heuristics.

1.2 Advantage and Benefit of the Presented Method

In a real scheduling problem, there exist various heterogeneous decision features in a single problem. By use of the conventional terminology, these are so-called lot sizing, lot sequencing, dispatching, and loading, and so on. However, they are actually encrypted in a problem and hence cannot be divided into each.

Accordingly, it would be the ideal to make simultaneous decision of which item should be produced and at what time, on which machine, and how much, over the whole system from the viewpoint of global optimization. In addition, high resolution, real time nature, and global optimization orientation have been required so as to adapt to various advanced requirements and business process innovation today. However, it seems that the conventional scheduling method is unable to adapt to those business requirements today.

Indeed, it has been seen the tendency that in academism the concern is focused not on modeling but on optimization technique, and in industry also, there has not been the move to innovate the principle of modeling in scheduling probably because of so called stereotype.

Although the conventional scheduling method is really full of variety, it has been seen the tendency that a certain specific decision feature is focused on and dealt with apart from the other (Błażewicz *et al.*, 2002). It is unable to do this sort of overdriven abstraction unless we put any artificial constraint on a real problem. The very constraint placed for the purpose of solution is apt to become rather the obstacle to our unified method for fine scheduling. For this reason, we are obliged to innovate on modeling and solution principle for scheduling.

The presented method enables us to treat the dynamic equation of processing of each item and the accompanied work-in-process stock transition in explicit in the model. Accordingly, if necessary, it is able to reduce work-in-process stock down to the limit by contracting processing interval on consecutive two processes. Simultaneously, it leads to reduction of production lead time and advance of operation rate.

1.3 Literature Review

The paper that intends simultaneous decision stated in sec. 1.2 has not been known except a few papers (Warman and Muramatsu, 2002; Muramatsu *et al.*, 2003; Muramatsu and Warman, 2002; Serizawa *et al.*, to appear 2006; Kobayashi and Muramatsu, 2005; Kobayashi and Muramatsu, to appear 2006).

2. PROBLEM DESCRIPTION

2.1 MIMPDLSSP

Suppose an ordinary system that consists of multiple processes, multiple machines, and multiple items (involving parts, semi-products, and final products.). Refer to Fig.1. A machine processes items on each process, but it is unable to process multiple items at a time. Switching of an item to the other incurs set up cost and time.

In addition, a process may have the machines whose processing time differs and it depends on the machine used and the item processed. Set up time and cost also depends on both. As for shop type, any move of items is allowed. In other wards there is no restriction, whether it is flow shop or job shop, involving circulation, Product structure is also general. Furthermore, in a system there exist many items that are made of many items and at the next process are built in many items (that have both of immediate predecessors and immediate successors in the terminology of graph theory.) For any item, holding cost is incurred and it is proportional to the quantity and duration.

We would like to find a good schedule for all machines and all items in the sense of minimizing the whole cost consisting setup and holding cost over the planning horizon without allowing shortage of an item and delay.

2.2 Extended Problems

There are extensions toward various directions. However, in this paper we will illustrate two cases. As stated later, mathematically any extended problem can be dealt with in the same manner, and hence just one or two examples are enough to illustrate the outline of the method for dealing with extensions.

2.2.1 The problem with family set up

Suppose that a set of items is partitioned into several families of items and that family set up also occurs besides usual set up in switching from one family to the other and it depends only on the family that is processed anew. Ref. to

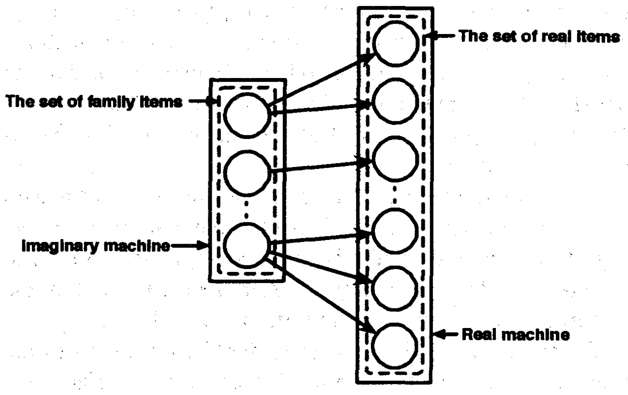

Fig. 2 Introduction of imaginary machine and imaginary (family) items

Fig. 2. For instance, in some die-cast process there is the case in which several items share one identical die. In this case, set up is decomposed into family set up and the usual one that depends on the operation unique to each item. The other feature of the problem is the same as the one of MIMPDLSSP.

2.2.2 A problem with sequence dependent set up

Suppose that in MIMPDLSSP there is a process in which set up operation assumes the sequence dependent aspect for the following reasons. Each item belongs to two distinct attribute families such as "color" and "shape" simultaneously. In addition, for each attribute, set up operation occurs in switching of attribute families and it depends only on the attribute family processed anew.

Consequently, set up is decomposed into the three parts: set up operation of attribute 1; the one of attribute 2; and the one unique to each item. Obviously, if switching of items occurs within the same attribute family for any attribute, family set up does not occur.

3. MODELING

In this section, we will outline our unified modeling approach to the MIMPDLSSP and its extensions.

3.1 Key Idea for a Unified Modeling and Minimum Unit of Modeling

The principal concept in our modeling is to present a miniature model that is able to describe all of the production activities as it is. The key idea is to note processing and the accompanied work-in-process transition and yet to comprehend the work-in-process as an effective medium for deriving a feasible solution unique to the problem.

For that sake, we note the state of whether system element k is at work at timeslot t for item i or not. Then, for any trinity of item i, system element k, and timeslot t, we call this unit "primitive object." Therefore, the primitive object becomes the minimum unit of configuring the model and hence its resolution.

In this definition, system element means a machine, machine unit, or any other subsidiary resource such as die. Operation, whether it is related to real processing or set up, does not necessarily work by a single system element alone. Especially, in a case of extended problem, more than or equal to two elements specify almost always the operation unique to the problem as illustrated in sec. 4. This fact is one of our finding.

Then, for the trinity of item i, machine k, and timeslot t, we let decision variable δ_{it}^k denote the state of primitive object, which takes 1 if it is at the state of operation, else then takes 0. Fundamentally, we formulate almost all of the features ruling production activities by use of those δ_{it}^k.

Consequently, variable δ_{it}^k specifies the resolution of the model. Furthermore, we note that using this variable enables modeling without making conscious of each decision feature encrypted in a problem. Then, we call the presented concept, modeling methodology, and solution principle the object oriented optimization technology (abr. O2O technology) collectively.

3.2 Symbols

The problem that we discuss is so large and complicated that it is unable to avoid the ambiguity of modeling without mathematical notation. So, we will cite a collection of the symbols used in this section and the next one.

Problem data and variables

for item $i, i \in I$, timeslot $t, t \in T$, and machine (or system element) $k, k \in K$.

x_{it}: denotes work-in-process stock in the sense of echelon inventory of item i at the terminal of timeslot t. Let the initial inventory at the terminal of timeslot 0, x_{i0} be given.

r_{it}: denotes shipment requirement of item i at timeslot t. This is given from part explosion using order quantity and product structure data.

p_i^k: denotes the rate of processing per one timeslot of item i at machine k.

h_i: denotes holding cost per timeslot of item i per timeslot.

c_i^k: denotes the set up cost incurred when item i is processed on machine k.

$s_{i\max}^k$: denotes the set up time incurred when item i is processed on machine k.

δ_{it}^k: stated in sec. 3.1.

s_{it}^k: denotes the timeslot number of remaining set up time, where as for the state of "at waiting for processing", let s_{it}^k takes -1. Note that s_{it}^k is not necessarily state variable but also decision variable partially together with δ_{it}^k.

Sets

$K(i)$: the set of machines available to process item i

$M(k)$: the set of items allowed to process by machine k

$S(i)$: the set of items succeeding to item i. i.e. the set of items that item i is build in.

L: denotes the set of the machines accompanied by family set up, $l \in L$

F_l: the set of family (imaginary) items related to machine l.

F_l^a: the set of items of attribute family a on machine l.

G_l: $G_l = F_l \bigcup_{j \in F_l} S(j)$. i.e. the union of the set of family items on machine l and the set of the entire accompanied real items.

Index function

$I_0(\cdot)$: the index function that takes 1 if the value of the inside of ( ) is 0, else then 0.

$I_+(\cdot)$: the index function that takes 1 if the value of the inside of ( ) is positive, else then 0.

3.3 Modeling of MIMPDLSSP

Obviously, it is the constraint but not the objective function to characterize MIMPDLSSP. This is further divided into the following four phases:

3.3.1 Dynamic equation of processing and the accompanied work-in-process stock transition

As having been stated already, if an item is processed, its work-in-process stock increases and if used, decreases.

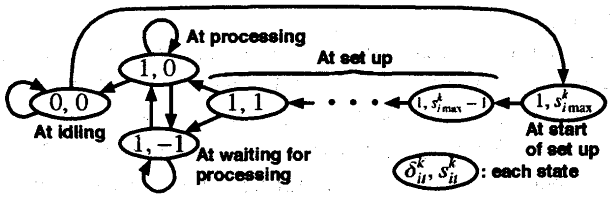

Fig. 3 Diagram of state transition

Then, we formulate it into the dynamic equation of work-in-process transition. This modeling enables high resolution, dynamism, and real time or time-variant nature of the model. This is formulated by use of "echelon inventory" in order to guarantee additively separable property and hence problem decomposition as stated in sec. 3.4.

Mathematically, it is formulated as:

Transition equation of work-in-process stock in the sense of echelon inventory

$$x_{it} = x_{it-1} - r_{it} + \sum_{k \in K(i)} p_i^k \delta_{it}^k I_0\left(s_{it}^k\right), \quad \forall i \in I, t \in T \quad (1)$$

3.3.2 Admissible set of actions and state transition diagram

In the states of machine, there are four of "at idling", "at waiting for processing", "at processing", and "at set up". The action that is admitted to take at a time for an item and a system element depends on the state at that time. For instance, if a machine state is on the way of set up, it is not allowed to disrupt set up, but yet if it is at processing, it is able to stop processing or to continue.

Precisely, Fig. 3 describes the admissible set of actions and state transition. The state is stated by use of δ_{it}^k together with s_{it}^k, where we note that s_{it}^k denotes the remaining time of set up or the state of waiting for processing. As for δ_{it}^k, we defined it in sec. 3.1. Especially, we describe the state of "waiting for processing" by use of $(\delta_{it}^k, s_{it}^k) = (1, -1)$, apart from the state of "idling". Accordingly, s_{it}^k is also a decision variable partiallly.

The content stated in Fig. 3 is indispensable to both processes of modeling and solution. Then, in order to refer to it, let operator T and T^{-1} denote as:

> For any given state at $t-1$ timeslot, $(\delta_{it-1}^k, s_{it-1}^k)$, the operator T denotes the set of admissible states at timeslot t defined in Fig. 3, $(\delta_{it}^k, s_{it}^k)$. Inversely, for any given state at timeslot t, $(\delta_{it}^k, s_{it}^k)$, the operator T^{-1} denotes the sets of admissible states at timeslot $t-1$.

State transition specified by the operator T and T^{-1} (2)

We note the fact that since there is a degree of freedom (or room for selecting an action) in the T and T^{-1}, optimization problem arizes.

3.3.3 Constraint to ensure feasibility of processing on each machine

Since multiple items share one machine, we need to place some mechanism to avoid confusion of processing on a model. It is rather simple. Under the constraint stated in sec. 3.3.4, just putting of the constraint to interdict machine interference for each machine attains the feasibility of processing. It is given as:

The constraint to interdict machine interference for each machine

$$\sum_{i \in M(k)} \delta_{it}^k - 1 \le 0, \quad \forall k \in K, t \in T \quad (3)$$

3.3.4 Constraint to integrate all the processing over the whole processes

Unless work-in-process falls short for any item, any trouble does not happen to occur in production activities. That is, all of the processing advances in order over the whole system. Then, just putting the inequality constraint to interdict work-in-process shortage for any work-in-process item also enables the synthesis of the entire processing in the whole system. This is given as:

The constraint to interdict the shortage of work-in-process of each item

$$\sum_{j \in S(i)} \left(x_{jt} - x_{j0}\right) - x_{it} \le 0, \quad \forall i \in I, t \in T \quad (4)$$

where x_{it} is work-in-process stock level in the sense of echelon inventory. As for the devivation of this expression, see (Muramatsu *et al.*, 2003).

After all, these four phases are the fundamentals common to MIMPDLSSP and hence common to almost all of the scheduling problems. In other wards, these are the necessary and sufficient condition to find a feasible solution for MIMPDLSSP and for a case of extended problem they are a necessary condition.

3.4 Additively Separable Property

Scheduling problem belongs to the category of dynamic optimization problem or optimal control one fundamentally. Since each work-in-process stock transition uses one dimension of the axis of the state space, the model becomes so large that it is unable to solve the problem without decomposition. The requisite for ensuring the decomposition is to derive additively separable property in the model. The definition is to be able to divide the problem into sub-problems so that the same decision variable may not appear in more than or equal to two sub-problems.

Then, in order to achieve this property, we formulate work-in-process stock transition into the expression by using the concept of "echelon inventory". As for the idea of echelon inventory, refer to (Clark and Scarf, 1960). For that sake, we deal with stock as follows. Once each work-in-process item is processed, it is stocked not only until it is built into the other item but also until the final product into which it has been built is shipped. For any other item also, just the part corresponding to the added value is stocked in the same manner. Even if manipulating like this, there is no difference as for the fact that work-in-process items are held in the system as the whole. For this reason, we derive eq.(1) and ineq.(4) in sec. 3.3 by use of "echelon inventory."

3.5 Objective function and the model

Objective function

$$f(\delta,x,s) \stackrel{\mathrm{d}}{=} \sum_{i\in I}\sum_{t\in T}\left(\sum_{k\in K(i)} c_i^k\left(1-\delta_{it-1}^k\right)\delta_{it}^k + h_i x_{it}\right) \quad (5)$$

$$= \sum_{i\in I} f_i(\delta_i,x_i,s_i) \quad (6)$$

The model of the MIMPDLSSP

$$\min_{\delta} \quad f(\delta,x,s)$$
$$\text{subject to} \quad (1)-(4)$$

4. MODELING OF EXTENDED PROBLEMS

Further, in a case of extended problem, some other feature unique to the problem is observed. First of our finding is that the indispensable feature always appears as a restriction and yet it is placed on the operation, whether it is real processing or set up. Furthermore, the restriction is placed in addition to the constraints of fundamental models mentioned in sec. 3. Accordingly, both do not conflict with each other.

Precisely, the restriction is divided into a few types by the way of its modeling. However, some of them finish just by simple and partial correction of the fundamental model, for instance, confine admissible set or state space. The other is the one that defines relative states among more than or equal to two system elements such as machine, machine unit, or auxiliary resource like die. Accordingly, what we do in modeling of an extended problem anew is to formulate that feature into an additional constraint. However, even if it has been formulated directly, there is no guarantee that it is able to solve it. The situation is the same as in MIMPDLSSP.

Then, in the case of extend problems also, we will pay attention to the work-in-process, in line with the method presented in sec. 3.

We introduce imaginary items called family items or attribute family items, the accompanied work-in-process, and imaginary machine as a means for deriving a feasible solution. In other words, by allowing excess or shortage of work-in-process, we will admit an infeasible solution once and then drive it into a feasible one through the process of coordinating the work-in-process. As for the method, we present LDC method in sec. 5.

4.1 Additional constraints to the problem accompanied by family set up

Once family items and their work-in-process are introduced, what charanterizes family set up is the following constraint.

Processing of real item and the one of the corresponding family item must be always synchronized. Concretely,

1) When any real item is at "processing," then the corresponding family item has to be also at "processing."
2) When any real item is at "set up," then the family item has to be at the state of "waiting for processing."
3) When any family item is at "set up," then any real item has to be neither at "processing" nor at "set up."

It is difficult to realize all of the phasing or time factor simultaneously even if it is able to formulate these into a model in explicit. Then, instead of this direct means, we present a method for placing the condition 4), 5) under the following assumptions.

A1. Introduce an imaginary machine for producing family items and the accompanied work-in-process stock.

A2. For any family item, let the rate of processing per timeslot by imaginary machine be one unit, and also for any real item work-in-process, let it be consumed at the rate of one unit per timeslot if a real item is processed.

4) Simultaneous set up of a family item and the corresponding real items is not allowed.
5) Neither excess nor shortage of work-in-process of any family item is allowed.

Because if 4) and 5) are satisfied, then processing of the real items that are accompanied by family set up and the one of the family item is always syncronized, clearly, constraints 1)-3) are equivalent to place 4) and 5) under the assumptions A1 and A2.

Here, we note that all of constraints 1)-3) are the ones that specify phasing. On the other hand, in the new constraint 5) work-in-process of family item is introduced as a medium for deriving a feasible solution.

For the machine accompnaied by family set up also, the machine interference interdiction constraint in (3) in sec. 3 holds already. Accordingly, it is able to extend the set of items that the above constraint 4) covers up to the union of the set of the entire family items and the set of the entire real items accompanied by familly set up.

Thus, we get

the constraint to interdict simultaneous set up over family items and the corresponding real items

$$\sum_{j\in G_l}\sum_{k\in K(j)} \delta_{jt}^k I_+\left(s_{jt}^k\right) - 1 \le 0, \forall l \in L, t\in T \quad (7)$$

As for the constraint 5), modeling is direct.

The constraint to interdict excess and shortage of a family item

$$\sum_{t'=1}^{t}\left(\sum_{j\in S(i)} \delta_{jt'}^k I_0\left(s_{jt'}^k\right) - \delta_{it'}^{k'} I_0\left(s_{it'}^{k'}\right)\right) = 0,$$
$$\forall i\in F_l, l\in L, k\in K(j), k'\in K(i), t\in T \quad (8)$$

4.2 Additional constraints to the problem accompanied by sequence dependent set up

This problem differs from the one accompanied by family set up in appearance. However, mathematically both are close. Just replacing the word of "family" in the former by "attribute family 1" and "attribute family 2" and then placing the constraints 4) and 5) respectively for both of them, we get the model of this problem.

The constraint to interdict simultaneous set up over family items and the corresponding real items

$$\sum_{i \in G} \sum_{k \in K(i)} \delta_{it}^{k} I_{+}\left(s_{it}^{k}\right) - 1 \leq 0, \forall t \in T \qquad (9)$$

The constraint to interdict excess and shortage of a family item

$$\sum_{t'=1}^{t}\left(\sum_{j \in S(i)} \delta_{jt'}^{k} I_0\left(s_{jt'}^{k}\right) - \delta_{it'}^{k'} I_0\left(s_{it'}^{k'}\right)\right) = 0,$$
$$\forall i \in F_l^a, a = \{1,2\}, l \in L, k \in K(j), k' \in K(i), t \in T \quad (10)$$

As illustrated by those two cases, the extended problems are specified by the additional constraints on the operation. It is common in the point that they appear as the phasing of operations among multiple system elements. Accordingly, the same modeling approach can be applied to the other extensions too. As for the verification of the assertion, there is no means except for making sure of it on many extended problems. e.g. (Kobayashi and Muramatsu, 2005; Kobayashi and Muramatsu, to appear 2006).

5. LDC METHOD AND HEURISTICS

The unified modeling would enable a unified approach to solution. As mentioned in sec. 3.2, problem decomposition is inevitable. As far as the problem is formulated by use of echelon inventory and additively separable property holds in the model, it is able to decompose it in the same way.

Then, we present LDC oriented method together with heuristics. As for decomposition, item based decomposition is the simplest and direct. In addition, for each sub-problem, dynamic programming approach is also the simplest and direct. We recommend them respectively unless there is any other specific reason.

The last problem is how to coordinate sub-problems. The way of resolving all of the constraint violations has precedence over the one of finding good solution. However, no mathematically beautiful characteristic holds in our problems, and hence it is difficult to resolve the constraint violation completely for any item at any timeslot and for any system element for any timeslot also by a mathematically rigid method alone. Its tendency increases as operation rate of a process or processes increase and the number of similar constraints increases. However, it does not mean that LDC method is ineffective to decrease the violation number down to some level but means that there has not been known a theory of guaranteeing any convergence.

Then, we present a method that relies on any heuristics after decreasing the violation number to a predetermined threshold by use of LDC method. LDC method is one of Lgarangian relaxation method. In this method, each Lagrangian multiplier value is coordinated according to the perturbation value of the accompanied constraint. Accordingly, it is done independently of the other parts. However, a heuristics enables us to deal with neighborhood simultaneously according to its state. Hence, it is more effective than rigid LDC method in finding a feasible good solution unless it is sensitive to the objective function value of the problem. Therefore, from the viewpoint of practice, use of any heuristics together with LDC method is recommended.

6. CONCLUSION

The paper presented a unified method for formulating a multi-item multi-process dynamic lot size scheduling problem and its extensions into fine mathematical models. Then, the paper referred to its global optimization orientated solution principle, which is based on Lagrangian decomposition coordination method together with heuristics.

References

Błażewicz, J. *et al.* (2002). *Scheduling Computer and Manufacturing Processes*, second edn, Springer, Berlin.

Clark, A. J. and H. Scarf. (1960). Optimal Policies for a Multi-Echelon Inventory Problem, *Manage. Sci.*, Vol. 6, pp. 457–490.

Kobayashi, M. and K. Muramatsu. (2005). An Extension of Job Shop Scheduling Problem, *J. Jpn. Ind. Manage. Assoc.*, Vol. 56, No. 4, pp. 246–255.

Kobayashi, M. and K. Muramatsu. (2006). A Dynamic Lot Size Scheduling Model with Sequence Dependent Set Up and Its Solution Method, *J. Jpn. Ind. Manage. Assoc.*, Vol. 57, No. 4 (to appear). (in Japanese).

Muramatsu, K. and A. Warman. (2002). Business Model for Manufacturing District Promotion by Autonomy and Coordination, *Trans. of the Japan Society for Production Management*, Vol. 9, No. 2, pp. 9–20. (in Japanese).

Muramatsu, K., A. Warman, and M. Kobayashi. (2003). A Near-Optimal Solution Method of Multi-Item Multi-Process Dynamic Lot Size Scheduling Problem, *JSME Int. J. Ser. C*, Vol. 46, No. 1, pp. 46–53.

Serizawa, T., H. Karasawa, and K. Muramatsu. (2006). Push Pull Mixed Type of Optimal Compound Production Planning Method, *Trans. of the Japan Society for Production Management*, Vol. 13, No. 1 (to appear). (in Japanese).

Warman, A. and K. Muramatsu. (2002). Multi-Item Multi-Stage Dynamic Lot Size Scheduling with Setup Time: Lagrangean Decomposition Coordination Methods, *J. Jpn. Ind. Manage. Assoc.*, Vol. 53, No. 5, pp. 385–396. (in Japanese).