International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

Bi-level multi-objective mathematical model for job-shop scheduling: the application of Theory of Constraints

Chompoonoot Kasemset [a] & Voratas Kachitvichyanukul [a]

[a] Industrial Engineering and Management, Asian Institute of Technology, Pathumthani, Thailand

Available online: 27 Nov 2009

International Journal of Production Research
Vol. 48, No. 20, 15 October 2010, 6137–6154

Bi-level multi-objective mathematical model for job-shop scheduling: the application of Theory of Constraints

Chompoonoot Kasemset* and Voratas Kachitvichyanukul

Industrial Engineering and Management, Asian Institute of Technology, Pathumthani, Thailand

(*Received 13 October 2008; final version received 9 July 2009*)

This study highlights a different systematic approach to the application of Theory of Constraints (TOC). The work describes the decisions involved in the implementation of TOC in a job-shop environment as a bi-level multi-objective mathematical model. On the first level, the decision is made by minimising idle time on the bottleneck to generate the initial schedule. The second level decision is to improve additional performance measurements by applying the multi-objective technique, while maintaining the bottleneck sequence obtained from the first level decision. Moreover, the concept of transfer lot is also adopted in this model to reduce the waiting time on each machine by allowing overlapped operations. The concept of transfer lot is applied as the constraint on earliest starting time for each job on each machine in the proposed mathematical model. Additionally, the machine set up time and product demands are also adopted to make the model practical to use in the real situation. The numerical examples for both single and multiple bottleneck cases are given to demonstrate how this approach works. The commercially available optimiser, the LINGO 10 software package, is used to solve the examples and the result shows how this approach works in practice.

Keywords: Theory of Constraints (TOC); job-shop scheduling; bi-level mathematical model; multi-objective; bottlenecks

1. Introduction

Many planning and scheduling problems can be modelled using a multi-level programming approach because of their hierarchical decision structures. The levels are interdependent and their objectives are often conflicting.

Bi-level programming is the simplest class of multi-level programming problems in which there are two interdependent decision levels that two decision-makers make decisions successively. This paper proposes a bi-level multi-objective formulation to solve the job-shop scheduling problems which operate under the Theory of Constraints (TOC) policy.

The motivation to propose this approach is two fold. Firstly, this paper presents a systematic approach to implement TOC using bi-level multi-objective model. This is different from works in published literatures that mostly employ TOC in case studies or via a strategic thinking process (Chaudhari and Mukhopadhyay 2003, Schaefers *et al.* 2004, Umble *et al.* 2006).

*Corresponding author. Email: st104182@ait.ac.th, ckjane@hotmail.com

ISSN 0020–7543 print/ISSN 1366–588X online

DOI: 10.1080/00207540903176705
http://www.informaworld.com

Secondly, the creation of a schedule based on the TOC concept for the uncomplicated system is easy to handle by using simple dispatching rules and dealing with the backward and forward scheduling process as addressed in Bolander and Taylor (2000) but when the problem is extended to multiple products and multiple processes cases, the extension of this approach is quite complex and not trivial to implement.

In this paper, the word 'bi-level' is used to correspond to the steps in the TOC procedure that tries to manage two machine types, bottleneck and non-bottleneck machines, in separate decision levels. When machines are classified into two types as mentioned above, the non-constrained machines should be managed by considering the plan of the constrained machines. To employ these concepts, the first level decisions deal with the bottleneck machine and the second level decisions deal with the non-bottleneck machines while the decisions are subordinated to the decisions made at the first level.

Furthermore, the concept of transfer lot is applied in the models at both levels in order to reduce waiting time of all machines that affect the system productivity. In addition to the concept of transfer lot, the machine set up time and product demands are also considered. These additional concepts allow the model to be applied in a real situation.

The organisation of this paper is as follows. The literature reviews about TOC and bi-level programming are addressed in Section 2. The proposed bi-level mathematical model is introduced in Section 3. In Section 4, the applications of the proposed model are illustrated with two cases; single and multiple bottlenecks. Conclusion and discussion are presented in Section 5 and Section 6.

2. Literature review

2.1 *Theory of Constraints (TOC)*

TOC practices have been extensively developed as a total solution for managing a factory to optimise on time delivery, inventory and operation cost (Jose Luis Perez 1997). In fact, every process is a link in a chain of operations, and the strength of the chain is only as strong as its weakest link, called 'constraint'. The TOC concept concentrates on how to manage the constraint of the system, the bottleneck resource or capacity constraint resource (CCR) (Goldratt and Cox 1986), to drive more income and for the company to survive in the real world of business.

Following this concept, there are only two types of machines classified in TOC; a bottleneck machine or a CCR and a non-bottleneck machine or a non-CCR. Bottleneck resource or CCR is the resource with capacity that is equal or less than the demand. There are fewer numbers of this type of resources in the factory. The other type is the non-bottleneck machine or non-CCR. It is the resource with capacity that is greater than the demand. The main point of TOC is to balance flow of product by making the output rate at the bottleneck equal to or a little bit more than the market demand (Goldratt and Cox 1986).

There are five steps in the implementation of TOC.

Step 1: Identify the system constraint

The key step to implement TOC is to identify the system bottleneck. The selection is made based on the economic scarcity of the resource. In order to meet the customer demand the constraint resource must operate at close to 100% utilisation (Jose Luis Perez 1997).

To locate bottlenecks, many researchers proposed the bottleneck identification methods that utilise static data. Pegels and Watrous (2005), Taj and Berro (2006) and

Reid (2007) used throughput rate to identify the bottleneck. Wu *et al*. (1994) considered the expected load compared with available load to find the CCR. Under dynamic situations, Gargeya (1994) applied simulation technique to collect data and used these data to calculate two factors, the number of jobs in queue for a resource and resource criticality factor, in order to identify bottlenecks under job-shop situation. Kasemset and Kachitvichyanukul (2005, 2007, 2008 and 2009) applied both static data and data collected from simulation to classify bottleneck candidates and identified the real system bottlenecks via simulation.

Step 2: Exploit the system constraint

In this step, the utilisation of the bottleneck is maximised. There are two discussed principals to deal with the bottleneck.

(1) Avoid wasting time at the bottleneck: time wasted occurs in many forms, idle time, machine downtime, poor quality parts and unneeded parts. Thus, the production schedule should be planned in order to reduce wasted time and maximise the use of bottleneck capacity at the early step.
(2) Increase utilisation of the bottleneck by setting a constraint buffer.

Drum-buffer-rope (DBR) is a popular control mechanism used in TOC. It is named after the three essential elements of the solution. Drum or constraint is used to set the pace of production for the non-bottleneck process to follow. Buffer is the stored part to keep the bottleneck busy by preventing shortages. Rope is the tool to tie the whole process to work synchronously (Goldratt and Cox 1986). The aim of the solution is to obtain a robust and dependable process that will allow the system to produce more, with fewer inventories, less reworks/defects, and better on-time delivery. The outcome is to protect the weakest link in the system, to protect against process variation and dependency and to maximise its effectiveness (Youngman 2003).

Another technique to exploit the constraint is known as 'transfer lot' or 'lot splitting'. Transfer lots are applied to move partially completed portions of a production lot to the next downstream process. Transfer lots allow overlapped operation to gain the benefit by reducing waiting time at the bottlenecks and reduce the total completion time.

Step 3: Subordinate the system constraint

For non-bottleneck machines, the level of utilisation is determined by the system constraint and activating them to the maximum level is simply unwise and will lead to excess inventory on unneeded parts. The non-bottleneck machines are subordinated to the bottleneck machine by using a shipping buffer to absorb the fluctuated total processing time.

Step 4: Elevate the system constraint

This step attempts to increase the bottleneck capacity which can be achieved by shifting some part to another process if it is possible or by adding a machine or by using a subcontract. The decisions depend on the specific production characteristic.

Step 5: If the system constraint is violated, go back to the first step but do not let the system inertia become the additional constraint

The effectiveness of TOC is addressed in Schaefers *et al*. (2004) with the case of one small- and medium-sized enterprise (SME) that applied the TOC policy to improve the planning and scheduling system. The result showed that TOC can help in shortening manufacturing lead time while improving overall company performances (i.e. increasing service level, decreasing transportation cost, WIP reduction, etc.).

The comparison of TOC with just in time (JIT) is presented in Watson and Patti (2008). This study employed simulation technique to evaluate the performance measure between DBR from TOC and Kanban from JIT. The result showed that under the TOC system, DBR helps in achieving higher throughput, shorter lead time and lower inventory.

Likewise, many research works in scheduling also concentrate on TOC application. Some recent research also reports about DBR application in job scheduling (Sirikrai and Yenradee 2006, Wu and Yeh 2006).

2.2 *Bi-level programming*

Many planning and scheduling problems contain a hierarchical decision structure, each with interdependent and often conflicting objectives. These types of problems can be modelled using a multi-level programming approach. The applications of multi-level programming for a planning and scheduling problem have been presented in many research works. Karsiti *et al.* (1992) applied simulation to study the multi-level dynamic job-shop scheduling using heuristic dispatching rules. Adam *et al.* (1993) addressed the procedure to assign due date to multi-level assembly job-shop. Haouba and Xie (1999) addressed the case of two-level assembly system in their study of multi-level assembly flow control. Yadav *et al.* (2000) applied a multi-level heuristic search algorithm to solve the production scheduling problem.

Bi-level programming is the simplest class of multilevel programming problem in which there are two interdependent decision levels. Also, bi-level programming is a tool for modelling decentralised decisions consisting of the objective(s) of the leader at its first level and that of the follower at the second level.

The general formulation of a bi-level programming problem (BLPP) is (Colson *et al.* 2005):

$$\begin{aligned} &\operatorname*{Min}_{X} : F(X, Y) \\ &\text{s.t. } G(X, Y) \leq 0, \\ \\ &\operatorname*{Min}_{Y}\ y : f(X, Y) \\ &\text{s.t. } g(X, Y) \leq 0, \end{aligned} \tag{1}$$

where $X \in R^{n1}$ and $Y \in R^{n2}$. The variables of problem are divided into two classes, namely the upper-level variables $X \in R^{n1}$ and the lower-level variables $Y \in R^{n2}$. Similarly, the functions $F\colon R^{n1} \times R^{n2} \rightarrow R$ and $f\colon R^{n1} \times R^{n2} \rightarrow R$ are the upper-level and lower-level objective functions respectively, while the vector-valued functions $G\colon R^{n1} \times R^{n2} \rightarrow R^{m1}$ and $g\colon R^{n1} \times R^{n2} \rightarrow R^{m2}$ are called the upper-level and lower-level constraints respectively. (For more recent work in this area see, for example, Emam (2005), and Roghanian *et al.* (2007)).

Many research works apply the bi-level programming to solve the scheduling and planning problems.

Logendran *et al.* (1995), dealt with the group scheduling problem that is separated into two levels. The first level is to determine the sequence of part within a part family and the second level is to find an appropriate sequence for groups of jobs.

Kalof and Wang (1996) formulated a bi-level mathematical model for the flow-shop scheduling problem and proposed an algorithm to find the solution. In their research, decisions are made at two levels. The first level decision is made by considering the objective of the shop owner while the second level decision considers the objective of the lower level planner or customers.

Lin and Shaw (1997) developed the bi-level intelligent scheduling system in which product-level and board-level scheduling were integrated to schedule jobs for PCB manufacturing systems under a daily to weekly planning horizon. Product-level scheduling is first used to determine the type and quantity of products made on each day over a week. Board-level scheduling is then used to fine-tune the production schedule to best utilise the capacity and improve the line's productivity. Their experimental results showed its competence in the problem with a tight due date, high order and product variations.

Xu and Randhawa (1998) proposed a two-phase approach to examine the impact of job-shop scheduling rules and tool selection policies for a dynamic job-shop in a tool shared under a flexible manufacturing environment. This two-phase method is used to optimise both job movement and tool movement.

Haouba and Xie (1999) addressed the case of two-level assembly system in their study of multi-level assembly flow control. Two levels represent the upper level machines and the assembly machines. The optimal policy from this case depends on the bottleneck machine, the machine with the smallest production capacity, and all machines should produce at the same rate as the bottleneck machine.

Bolander and Taylor (2000) addressed the creation of a schedule based on the TOC concept with the objective to maximise the throughput at the constraint. Once the constraint is identified, the demand information, time and quantity of jobs are translated into capacity requirements of the constraint that best meets customer due dates and maximise throughput at the constraint. Then, all other schedules upstream and downstream from the constraint are subordinated to this schedule. Upstream processes are scheduled by offsetting the constraint schedule back to the gateway operation. Scheduling of downstream processes is done simply on a first-come first-serve basis. Since the constraints already incorporated due date requirements, the movement of material to downstream processes is already in the proper priority sequence. The procedure is demonstrated using a system with a single product and a single processing line with five machines. However, when the problem is extended to multiple products and multiple processes cases, the extension of this approach is quite complex and is not trivial to implement.

Zuo *et al.* (2007) proposed bottleneck-driven decomposition procedure to deal with the job-shop scheduling problem with multiple constraint machines (JSPMC). This procedure starts by concentrating on bottleneck machines first in order to minimise maximal tardiness. Then, the non-bottleneck machines are scheduled by the earliest due date (EDD) dispatching rule. The result from this procedure is superior compared with the shifting bottleneck algorithm and TOC method in terms of the trade-off between solution quality and computational time.

Erdirik-Dogan and Grossmann (2008) proposed a bi-level decomposition algorithm in which there are two levels; an upper level planning and a lower scheduling problem to solve single-stage multi-product continuous plants with parallel units in chemical production. Their proposed algorithm is the aggregation between a planning problem as an upper level and a scheduling problem as a lower level. The production target is solved from the upper level. Then, the detailed schedule is derived from the lower level with the tight upper bound given from the upper level. They indicate that their proposed algorithm can reduce the number of total iterations and reduce infeasibilities and mismatches in predicting the production that may occur between both level decisions.

Ho and Li (2008) developed the scheduling model based on the TOC concept. In this model, they proposed the unique objective function, to minimise throughput-dollar-day (TDD) and inventory-dollar-day (IDD), and compared the result with those obtained from traditional dispatching rules. The result showed that this objective is feasible and outperforms traditional dispatching in terms of objective function value but not in terms of due date and cycle time.

A bi-level formulation is proposed in this paper. The first level aims to generate an initial schedule that minimises the idle time of the bottleneck machine. The objective function for the first level is a single objective. The second level model optimises the schedule again to satisfy multiple objectives subjected to the constraints on job sequence on bottlenecks obtained from the first level. At this level, the other resources are managed simultaneously so that the bottleneck still works at the maximum utilisation. Moreover, the concept of transfer lot, the machine set up time and product demands are also adopted in this proposed model.

3. The proposed mathematical method

In this research, a two-level mathematical model is formulated for scheduling job-shop operated under a TOC policy. The first model level aims to minimise idle time on the bottleneck machine by applying the unique constraint, called ‘transfer lot constraint’, that allows overlapped operation in order to reduce waiting time on the bottleneck machine. The result from this level is the job sequence on bottleneck that minimises idle time on the bottleneck.

The second level model is similar to the first level model with the following changes:

(1) The addition of constraints that freeze the job sequences on the bottleneck.
(2) Different objective functions are considered as multiple objectives.

Since the second level considered more than one objective, there may not be a solution that is the best with respect to all objectives because of incommensurability and conflict among objectives. A solution may be best in one objective but worst in others. Therefore, there usually exist a set of solutions for the multiple-objective cases which cannot simply be compared with one another. For such solutions, called non-dominated solutions or Pareto optimal solutions, no improvement in any objective function is possible without sacrificing at least one of the other functions (Gen and Cheng 2000).

In this study, three performance measures are used in the second level model which includes completion time, tardiness, and earliness. For each criterion k, the gap (Δ_k) can be found between the value of the k criterion F_k and the best value Z_k obtained using only the single objective for criterion k. The objective function of the second level is defined as a single weighted aggregative of all the gaps of all the criteria.

Lastly, the full job schedule is generated in the second level model by retaining the maximum use of bottlenecks capacity as the bottleneck subordination idea that aims to manage other resources to support the bottleneck.

3.1 *Model assumption*

This 'bi-level multi-objective mathematical model for job-shop scheduling' is based on the following assumptions:

(1) The location of system bottleneck(s) is known.
(2) All parameters are known and deterministic.
(3) The optimal values of each system performance, maximum completion time ($C_{\max}$), tardiness ($T_{\max}$) and earliness ($E_{\max}$), are known or can be found from solving each using single objective.
(4) No re-entrant job.
(5) Setup type is an attached setup, a detached or off-line set up is not considered in this model.
(6) Transfer lot size is applied with the constant size for each product and the same size for all machines/processes.

3.2 *Model formulation: Bi-level Multi-objective mathematical model for job-shop scheduling*

Indices: $i, i' =$ Job $\quad i = 1, 2, \ldots, n$
$j, j' =$ Machine $\quad j = 1, 2, \ldots, m$
$k =$ Performance measure factor $\quad k = 1, 2, 3$
$b =$ Bottleneck machine $\quad b = 1, 2, \ldots, B$

Decision variable: $X_{ij} =$ Start time of job i on machine j

$$Y_{ii'j} = \begin{cases} 1, & \text{if job } i \text{ is processed before } i', \\ 0, & \text{Otherwise} \end{cases}.$$

Parameter: $U_i =$ Demand unit for job i
$s_{ij} =$ Set up time of job i on machine j
$p_{ij} =$ Process time per unit of job i on machine j
$t_i =$ Transfer lot size of job i
$r_i =$ Ready time of job i
$D_i =$ Due date of job i

$$\text{Minimise: } \sum_{b=1}^{B} [\max_{i,b}\{X_{ib} + s_{ib} + U_i P_{ib}\} - \sum_{i=1}^{n} (s_{\text{ib}} + U_i P_{ib})] \tag{2}$$

Where X_{ib} solves

$$\text{Minimise: } (\Delta_1 + \Delta_2 + \Delta_3) \tag{3}$$

When

$$\Delta_{\text{k}} = \text{F}_{\text{k}} - \text{Z}_{\text{k}} \tag{4}$$

And

F_1 : Maximum completion time ($C_{\max}$) $\quad \max_{i,j}\{X_{ij} + s_{ij} + U_i P_{ij}\}$

F_2 : Maximum tardiness ($T_{\max}$) $\quad \max_{i,j}\{0, (X_{ij} + s_{ij} + U_i P_{ij}) - D_i\}$

F_3 : Maximum earliness ($E_{\max}$) $\quad \max_{i,j}\{0, D_i - (X_{ij} + s_{ij} + U_i P_{ij})\}$

Subject to

$$X_{ij} + s_{ij} + (t_i p_{ij}) \leq X_{ij'} \qquad \forall i \tag{5}$$

$$\left.\begin{array}{l} X_{ij} + s_{ij} + (U_i p_{ij}) \leq X_{i'j} + M(1 - Y_{ii'j}) \\ X_{i'j} + s_{i'j} + (U_{i'} p_{i'j}) \leq X_{ij} + M Y_{ii'j} \end{array}\right\} \qquad \forall j \tag{6}$$

$$X_{ij} \geq r_i \qquad \forall i \tag{7}$$

$$I_{ijj'} = \left\{\begin{array}{ll} 0, & \text{if } p_{ij} \leq p_{ij'} \\ (p_{ij} - p_{ij'})U_i + (s_{ij} - s_{ij'}), & \text{if } p_{ij} > p_{ij'} \end{array}\right\} \qquad \begin{array}{l} \forall i \\ \forall j, j' \end{array} \tag{8}$$

$$X_{ij'} \geq X_{ij} + s_{ij} + (p_{ij}\ t_i) + I_{ijj'} \tag{9}$$

$$X_{ij}, I_{ijj'} \geq 0 \tag{10}$$

$$Y_{ii'j} \text{ is binary variable} \tag{11}$$

Additional constraints for second level model:

$$X_{ij} \leq X_{ib} \quad \text{for } j = b \tag{12}$$

$$Y_{ii'j} = Y_{ii'b} \quad \text{for } j = b \tag{13}$$

The first level objective, Equation (2) is to minimise idle time on the bottleneck that equals the completion time of the bottleneck minus the total processing time required on the bottleneck. The subscript 'b' refers to the bottleneck machine. This objective attempts to maximise the bottleneck utilisation. The second level objective, Equation (3) is to minimise the sum of Equation (4) that is the gap between each performance measure factor and its optimal value obtained by solving each using a single objective. Both objectives are subject to Equation (5) precedence constraints, Equation (6) machine conflict constraints. Equation (7) is the job ready time constraint. Equation (8) is the total waiting time between transfer lot and it is zero when the processing time of the preceding process is less than the succeeding process. Equation (9) is the restriction on the earliest starting time when transfer lot is applied. The detail of Equations (8) and (9) is explained in the next section. Non-negativity constraints and binary constraints are shown in Equations (10) and (11), respectively. Furthermore, there are two additional constraints for the second level model. Equations (12) and (13) are the constraints that freeze the job sequence on the bottleneck obtained from the solution of the first level model.

3.3 *Explanation of transfer lot constraint*

Considering the transfer lot, the particular constraint must be added in order to allow the earliest possible starting time for each job on each machine. Figures 1(a)–(c) illustrate the transfer lot constraints.

When the processing time of job i on preceding machine j is less than or equal to its processing time on succeeding machine j', job i can be started on j' immediately or after one transfer lot is finished on j (see Figure 1(a)).

Conversely, if the processing times of job i on preceding machine j is larger than its processing time on succeeding machine j', the additional constraint must be added in order to avoid two situations shown in Figure 1(b).

In Figure 1(c), $I_{ijj'}$ is the sum of waiting time between each transfer lot for the case that p_{ij} is greater than $p_{ij'}$. $I_{ijj'}$ is only valid and bears a feasible value when $p_{ij'}$ is less than p_{ij}.

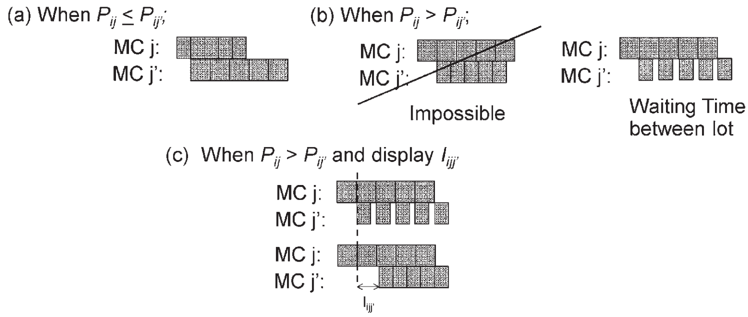

Figure 1(a)–(c). Introduction to transfer lot constraint.

Thus, the job's earliest starting time on each machine should be the ending time of one transfer lot plus its total waiting time between each transfer lot.

Following the TOC idea, if there is no information about the size of transfer lot, the minimum value of the transfer lot will be set to one.

4. Illustrative example

In this research, numerical examples are presented for both the single and multiple bottleneck cases to illustrate how the approach works. The LINGO 10 software package is used to find the optimal solution for the examples by solving the two-level mathematical model in two steps. For clarity, it is assumed that the locations of the bottleneck are known. For the situations where the location of the bottleneck is not known, bottleneck identification procedures can be applied (see, for example, Kasemset and Kachitvichyanukul 2007 and 2008).

4.1 *Single bottleneck case*

Consider a job-shop with five different machines producing three different types of products. Each job-type must be processed in a specific machine sequence. The machine sequences for products, average operation times and set up times are shown in Table 1(a)–(c).

Demands of all products are 1500 units and the due dates are the same for all products as 30,000 minutes. The job-shop operates continuously for 7 days and 24 hours per day. Given that station B is identified to be the system bottleneck.

4.1.1 *First level model application*

In this example, station B is the system bottleneck. The first level model generates the job sequence that minimises the idle time of station B. In this case, transfer lot size is set to the minimum value of one (no information given about transfer lot size).

Figure 2 shows the job schedule from this model. As a result of the inclusion of the transfer lot, the completion time and the waiting time at the bottleneck are reduced. The optimal value, bottleneck idle time (I_b) is equal to 29 minutes. The primary schedule

Table 1(a). Process steps and detail for product 1 (single bottleneck case).

Process step	Machine	Setup time (minutes)	Process time (minutes)
1	A	25	4
2	B	10	8
3	C	0	5
4	F	30	6
5	G	15	3

Table 1(b). Process steps and detail for product 2 (single bottleneck case).

Process step	Machine	Setup time (minutes)	Process time (minutes)
1	F	30	6
2	B	10	8
3	C	0	5
4	G	15	3
5	A	25	4

Table 1(c). Process steps and detail for product 3 (single bottleneck case).

Process step	Machine	Setup time (minutes)	Process time (minutes)
1	A	25	5
2	G	10	8
3	C	0	5
4	F	30	6
5	B	25	4

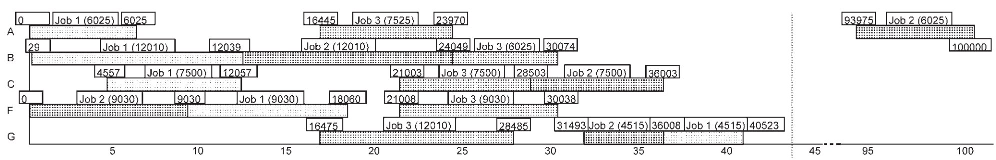

Figure 2. Schedule from the 1st level model for single bottleneck case (minutes).

generated from this level gives the job sequence at station B as job 1 (start at the 29th minute), job 2 (start at the 12,039th minute) and job 3 (start at 24,049th minute).

The job sequence for station B is now fixed and the sequence is added as the set of constraints in the second level model to retain the maximum use of the bottleneck's capacity. The formulation of this constraint set is also presented in the next step.

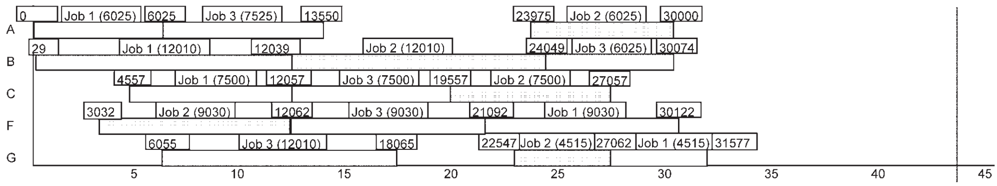

Figure 3. Schedule from the 2nd level model for single bottleneck case (minutes).

4.1.2 *Second level model application*

In this step, the previous schedule is re-optimised with multi-objective while maintaining the maximum use of the bottleneck's capacity by applying the constraint set that is the result from the previous step.

Additional constraints from the first level (station B is represented as $j=2$).

$$X_{12} \leq 29 \tag{14}$$

$$X_{22} \leq 12{,}039 \tag{15}$$

$$X_{32} \leq 24{,}049 \tag{16}$$

$$Y_{122} = 1 \tag{17}$$

$$Y_{132} = 0 \tag{18}$$

$$Y_{212} = 0 \tag{19}$$

$$Y_{232} = 0 \tag{20}$$

$$Y_{312} = 1 \tag{21}$$

$$Y_{322} = 0 \tag{22}$$

Equations (14) to (16) are used to force the start time of each job on station B to be no later than the start time that are the results from the first level. Equations (17) to (22) are the constraints to enforce the job order on station B.

Figure 3 shows the schedule generated from this step. The idle time on the bottleneck is still 29 minutes, while the maximum completion time and tardiness are reduced. A summary comparison is given in Table 2.

The result in Table 2 shows that the schedule generated from the first model level gains the minimum idle time on bottleneck machine at 29 minutes whereas the other performance measures, $C_{\max}$ (100,000 minutes) and $T_{\max}$ (70,000 minutes), are different from the optimal values which are obtained using single objective model for each performance measure. Thus, the second model level is required to improve the other performance factors. The final schedule from the second-level model is improved for both $C_{\max}$ (reduce to 31,577 minutes) and $T_{\max}$ (reduce to 1577 minutes) while the maximum use of the bottleneck is maintained.

Table 2. Summary result table for single bottleneck case.

Step	Performance measure (minutes)								
	I_b		C_{max}		T_{max}		E_{max}		Multi-objective value
	Current	Optimal	Current	Optimal	Current	Optimal	Current	Optimal	
First level (min I_b)	29	29	100,000	30,074	70,000	74	0	0	–
Second level (multi-objective)	29	29	31,577	30,074	1577	74	0	0	3006

Table 3(a). Process steps and detail for product 1 (multi-bottleneck case).

Process step	Machine	Setup time (minutes)	Process time (minutes)
1	A	25	4
2	B	10	8
3	C	0	5
4	F	30	6
5	G	15	3

Table 3(b). Process steps and detail for product 2 (multi-bottleneck case).

Process step	Machine	Setup time (minutes)	Process time (minutes)
1	F	30	6
2	B	10	8
3	C	0	5
4	G	15	3
5	A	25	4

Table 3(c). Process steps and detail for product 3 (multi-bottleneck case).

Process step	Machine	Setup time (minutes)	Process time (minutes)
1	A	25	5
2	G	10	8
3	C	0	5
4	F	30	6
5	B	25	4

4.2 *Multi-bottleneck case*

Consider again a job-shop with five different machines producing three different types of products. Each job-type requires a number of operations to be performed at specified kinds of machines in specified sequence. The visitation sequences, average operation times and set up times are shown in Tables 3(a)–3(e).

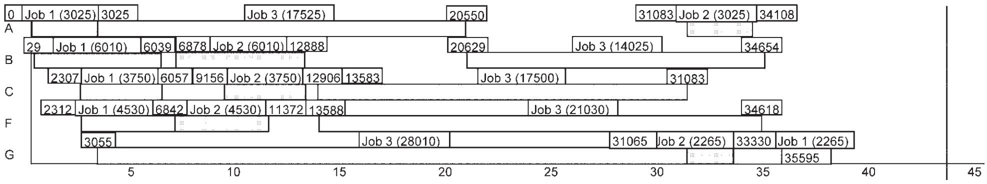

Figure 4. Schedule from the 1st level model for multi-bottleneck case (minutes).

Demands of each product are 750 units, 750 units and 3500 units respectively. Due date of products 1 and 2 are 30,000 and for product 3 is 43,200 minutes. The job-shop operates continuously for 7 days and 24 hours per day. Given that F and G are identified to be the bottlenecks.

4.2.1 *First level model application*

At this step, the schedule is generated to minimise the total idle time on both stations F and G. Transfer lot is also applied and set to the minimum value of one (no information given about transfer lot size).

Figure 4 shows the job schedule obtained by the first level model. The optimal total idle time is equal to 7583 minutes. This includes the idle times of 4528 minutes on station F and 3055 minutes on station G, respectively. The total idle time on both system bottlenecks is 7583 minutes.

The primary schedule generated from this level gives the job sequence of F as job 1 (start at the 2312th minute), job 2 (start at the 6842th minute) and job 3 (start at the 13,588th minute) and the job sequence of G as job 1 (start at the 33,330th minute), job 2 (start at the 31,065th minute) and job 3 (start at the 3055th minute)

Then, the job sequence of F and G are fixed and converted to be the set of constraints in the second level model as shown in the following section.

4.2.2 *Second level model application*

The previous schedule from the first level is re-optimised with multi-objective while maintaining the maximum use of the bottleneck's capacity by applying the constraint set that is the result from the previous step.

Additional constraints from the first level (station F and G are represented as $j=4$ and 5).

$$X_{14} \leq 2312 \tag{23}$$

$$X_{24} \leq 6842 \tag{24}$$

$$X_{34} \leq 13{,}588 \tag{25}$$

$$Y_{124} = 1 \tag{26}$$

$$Y_{134} = 0 \tag{27}$$

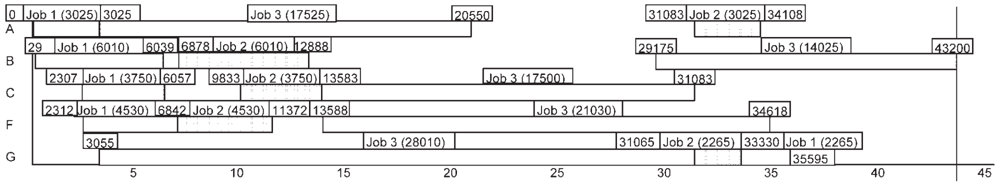

Figure 5. Schedule from the 2nd level model for multi-bottleneck case (minutes).

$$Y_{214} = 0 \tag{28}$$

$$Y_{234} = 1 \tag{29}$$

$$Y_{314} = 0 \tag{30}$$

$$Y_{324} = 0 \tag{31}$$

$$X_{15} \leq 33{,}330 \tag{32}$$

$$X_{25} \leq 31{,}065 \tag{33}$$

$$X_{35} \leq 3055 \tag{34}$$

$$Y_{125} = 0 \tag{35}$$

$$Y_{135} = 0 \tag{36}$$

$$Y_{215} = 1 \tag{37}$$

$$Y_{235} = 0 \tag{38}$$

$$Y_{315} = 0 \tag{39}$$

$$Y_{325} = 1 \tag{40}$$

Equations (23) to (25) and (32) to (34) are used to ensure that the start time of each job on station F and G to be no later than the start time that are the results from the first level. Equations (26) to (31) and (35) to (40) are the constraints to enforce the job order on station F and G correspondingly.

The schedule obtained from the second level model is further improved as shown in Figure 5. The total idle time on the system bottlenecks is still the same as the previous step, 7583 minutes, while the other performance measures are changed. The result comparison is given in Table 4.

The result in Table 4 shows that the schedule generated from the first model level gains the minimum total idle time on bottlenecks machine at 7583 minutes (on F is 4528 minutes and on G is 3055 minutes) whereas the other performance measures,

Table 4. Summary result table for multi-bottleneck case.

Step	Performance measure (minutes)								
	I_b		C_{max}		T_{max}		E_{max}		Multi-objective value
	Current	Optimal	Current	Optimal	Current	Optimal	Current	Optimal	
First level (min I_b)	7583	7583	35,595	35,595	5595	471	8546	0	–
Second level (multi-objective)	7583	7583	43,200	35,595	5595	471	0	0	12,729

C_{max} (35,595 minutes), T_{max} (5595 minutes) and E_{max} (8546 minutes), are different from the optimal values which are obtained using single objective model for each performance measure. Thus, the second model level is required to improve the other performance factors. The final schedule from the second level is different compared with the result from the prior level; C_{max} (increase to 43,200 minutes) and E_{max} (reduce to 0 minute) while the maximum use of the bottleneck is maintained. This is noticed as E_{max} is improved when C_{max} is degraded. These are the effects of multi-objective optimisation in the second level.

5. Conclusion

This 'bi-level multi-objective mathematical model for job-shop scheduling' is built based on the concept of TOC with some unique features;

(1) The first level mathematical model aims to minimise idle time on the system bottleneck.
(2) The second level mathematical model generates the complete schedule that maintains the maximum use of the bottleneck's capacity while improving the other performance measures by applying multi-objective optimisation.
(3) Transfer lot concept is adopted to reduce the waiting time of each machine. This results in the reduction of system's completion time.

Numerical examples are presented for both the single and multiple bottleneck cases to illustrate how the approach works.

The results from numerical examples indicate that,

(1) The schedule obtained from this proposed model is more realistic because it can exploit the use of the bottleneck's capability (minimum idle time at bottlenecks), and at the same time, take more advantage in other performance factors (satisfy multi-objective of each performance measure).
(2) Focus on C_{max} and T_{max}, the schedule generated from the proposed method are different in both performance measures comparing with the optimal value from the single objective approach in both cases because when the job sequence on the bottleneck is fixed, the improvement in C_{max} and T_{max} under multiple objective technique cannot be fully satisfied because (1) the fixed in job order on bottlenecks and (2) E_{max} that is negatively correlated with C_{max} and T_{max}.

(3) Focus on E_{max}, the proposed model generates the final schedule that E_{max} is improved to the optimal value from single objective technique even though it is negatively correlated with C_{max} and T_{max} as mentioned previously.
(4) If only the single objective that is minimising idle time on the system bottleneck is applied, it is possible for the final schedule to be worse in terms of other performance measures. Thus, the second level model is needed to improve the other performance measures in the final schedule.
(5) However, if the other performance measures are well thought-out, there is no reason to say that the final schedule from this proposed model is worse than the optimal schedule even though some performance factors are worse, C_{max} and T_{max}, because the solution from multiple-objective technique is a 'compromised solution' that meets 'some criteria' that is a satisfying solution in dealing with the case of multiple criteria (Tabucanon 1988).
(6) Since the multiple objectives in this study are formulated as a single weight-aggregated objective by giving the same priority for all objectives, different priority can be applied via varying weights of each performance measure that directly affects the final schedule.

The result from this study shows the advantages of this proposed method;

(1) This method provides an alternative approach to TOC implementation. The approach is more systematic than most case study approaches which are mainly an *ad hoc* procedure.
(2) The proposed bi-level multi-objective mathematical model is practical in the real situation compared with other models that deal with job-shop scheduling problems because the consideration of machine set up time and product demand.
(3) The application of transfer lot results in reducing waiting time of all machines, particularly on bottlenecks, that affects the system productivity.
(4) The way to conduct this mathematical model in the form of bi-level programming allow the schedule to maximise the use of the bottleneck's capacity; at the same time the other performance measures (C_{max} and T_{max} and E_{max}) can be improved using multi-objective techniques.

6. Discussion

There are some important observations when this procedure is applied. As the bottleneck job sequence is fixed, this may limit the improvement of the schedule in the second level. That result in the improvement of some factor may not be improved to the maximum extent because the restriction from the sequence on the bottleneck machine is the first level decision. At least, the main point to maximise the use of bottleneck capacity is maintained.

The size of the transfer lot is not directly considered in this study but in practice much research work studies how to calculate the size of the transfer lot in a scheduling problem. This kind of problem is known as 'scheduling with lot streaming or lot splitting'. Amongst this research, a recent one is from Koo *et al.* (2007). They developed the way to find the optimal batch size at the bottleneck machine to maximise the profit of the overall system. The size of the transfer lot can be considered as a decision variable in the mathematical model to find the optimal size or using this model to evaluate the size of the transfer lot and study the effect from varying the size of the transfer lot.

For a larger problem size, the two-step solution procedure may be replaced by metaheuristic methods that can provide solutions for both decision levels simultaneously. This may be an interesting area for further research. Some recent meta-heuristic for job shop scheduling can be found in Udomsakdigool and Kachitvichyanukul (2008), and Pongchairerks and Kachitvichyanukul (2009).

References

Adam, N.R., *et al.*, 1993. Due date assignment procedures with dynamically updated coefficients for multi-level assembly job shops. *European Journal of Operational Research*, 68 (2), 212–227.

Bolander, S.F. and Taylor, S.G., 2000. Scheduling techniques: a comparison of logic. *Production and Inventory Management Journal*, 41 (1), 1–5.

Colson, B., Marcotte, P., and Savard, G., 2005. Bi-level programming: a survey. *4OR: A Quarterly Journal of Operations Research*, 3 (2), 87–107.

Chaudhari, C.V. and Mukhopadhyay, S.K., 2003. Application of theory of constraints in integrated poultry industry. *International Journal of Production Research*, 41 (4), 799–817.

Emam, O.E., 2005. A fuzzy approach for bi-level integer non-linear programming problem. *Applied Mathematics and Computation*, 172 (1), 62–71.

Erdirik-Dogan, M. and Grossmann, I.E., 2008. Simultaneous planning and scheduling of single-stage multi-product continuous plant with parallel lines. *Computers and Chemical Engineering*, 32 (11), 2664–2683.

Gargeya, V.B., 1994. Resource constraint measures in a dual constrained job shop. *Omega*, 22 (6), 659–668.

Gen, M. and Cheng, R., 2000. *Genetic algorithms and engineering optimisation*. New York: John Wiley and Sons, Inc.

Goldratt, E.M. and Cox, J., 1986. *The goal: a process of ongoing improvement*. Massachusetts: North River Press.

Haouba, A. and Xie, X., 1999. Flow control of multi-level assembly systems. *International Journal of Computer Integrated Manufacturing*, 12 (1), 84–95.

Ho, T.F. and Li, R.K., 2008. Bottleneck-based heuristic dispatching rule for optimising mixed TDD/IDD performance in various factories. *International Journal of Advance Manufacturing Technology*, 36 (7–8), 773–779.

Jose' Luis Pe'rez, R., 1997. TOC for world class global supply chain management. *Computers Industrial Engineering*, 33 (1/2), 289–293.

Kalof, J.K. and Wang, W., 1996. Bi-level programming applied to the flow shop scheduling problem. *Computers and Operation Research*, 23 (5), 443–451.

Karsiti, M.N., Cruz, J.B., and Mulligan, J.H., 1992. Simulation studies of multilevel dynamic job shop scheduling using heuristic dispatching rules. *Journal of Manufacturing Systems*, 11 (5), 346–358.

Kasemset, C. and Kachitvichyanukul, V., 2005. Simulation-based tool for implementing theory of constraints. *Proceedings of the 6th APIEMS Conference*, 4–7 December 2005, Manila, Philippines. Manila: APIEMS2005, Article 4426.

Kasemset, C. and Kachitvichyanukul, V., 2007. Simulation-based procedure for bottleneck identification. *Proceedings of AsiaSim 2007 (Asia Simulation Conference 2007)*, 10–12 October 2007, Seoul, Korea. Berlin: Springer, 47–55.

Kasemset, C. and Kachitvichyanukul, V., 2008. Simulation-based procedure for implementing theory of constraints: Extension for cases with multiple bottlenecks. *Proceedings of the 9th APIEMS Conference*, 3–5 December 2008, Bali, Indonesia. Bandung: Department of Industrial Engineering, Institut Teknologi Bandung and Institut Teknologi Sepuluh Nopember, 1811–1819.

Kasemset, C. and Kachitvichyanukul, V., 2009. Simulation Tool for TOC implementation. *Proceeding of ASIMMOD 2009*, 22–23 January 2009, Bangkok, Thailand. Bangkok: ASIMMOD2009. Technical Committee, 86–97.

Koo, P.H., Bulfin, R., and Koh, S.G., 2007. Determination of batch size at a bottleneck machine in manufacturing systems. *International Journal of Production Research*, 45 (5), 1215–1231.

Lin, F.R. and Shaw, M.J., 1997. Scheduling printed circuit board production systems using the two-level scheduling approach. *Journal of Manufacturing Systems*, 16 (2), 129–149.

Logendran, R., Mai, L., and Talkington, D., 1995. Combined heuristics for bi-level group scheduling problems. *International Journal of Production Economics*, 38, 133–145.

Pegels, C.C. and Watrous, C., 2005. Application of the theory of constraints to a bottleneck operation in a manufacturing plant. *Journal of Manufacturing Technology Management*, 16 (3), 302–311.

Pongchairerks, P. and Kachitvichyanukul, V., 2009. A two-level particle swarm optimisation algorithm on job-shop scheduling problems. *International Journal of Operational Research*, 4 (4), 390–411.

Reid, R.A., 2007. Applying the TOC five-step focusing process in the service sector. *Managing Service Quality*, 17 (2), 209–234.

Roghanian, E., Sadjadi, S.J., and Aryanezhad, M.B., 2007. A probabilistic bi-level linear multi-objective programming problem to supply chain planning. *Applied Mathematics and Computation*, 188, 786–800.

Schaefers, J., Aggoune, R., Becker, F., and Fabbri, R., 2004. TOC-based planning and scheduling model. *International Journal of Production Research*, 42 (13), 2639–2649.

Sirikrai, V. and Yenradee, P., 2006. Modified drum-buffer-rope scheduling mechanism for a non-identical parallel machine flow shop with processing-time variation. *International Journal of Production Research*, 44 (7), 3509–3531.

Tabucanon, M.T., 1988. *Multiple criteria decision making in industry*. Amsterdam: Elsevier.

Taj, S. and Berro, L., 2006. Application of constrained management and lean manufacturing in developing best practices for productivity improvement in an auto-assembly plant. *International Journal of Productivity and Performance Management*, 55 (3/4), 332–345.

Udomsakdigool, A. and Kachitvichyanukul, V., 2008. Multiple colony ant algorithm for job-shop scheduling problem. *International Journal of Production Research*, 46 (15), 4155–4175.

Umble, M., Umble, E., and Murakami, S., 2006. Implementation theory of constraints in a traditional Japanese manufacturing environment: the case of Hitachi tool engineering. *International Journal of Production Research*, 44 (10), 1863–1880.

Watson, K.J. and Patti, A., 2008. A comparison of JIT and TOC buffering philosophies on system performance with unplanned machine downtime. *International Journal of Production Research*, 46 (7), 1869–1885.

Wu, H.H. and Yeh, M.L., 2006. A DBR scheduling method for manufacturing environments with bottleneck re-entrant flows. *International Journal of Production Research*, 44 (5), 883–902.

Xu, Z. and Randhawa, S., 1998. Evaluation of scheduling strategies for a dynamic job shop in a tool-shared, flexible manufacturing environment. *Production Planning & Control*, 9 (1), 74–86.

Youngman, K.J., 2003. A guide to implementing the theory of constraint (TOC) for innovative production, supply chain, service provision, distribution, marshalling, healthcare, turnaround management, cooperate finance and strategy [online]. Available from: http://www.dbrmfg.co.nz [Accessed 15 December 2004].

Zuo, Y., Gu, H., and Xi, Y., 2007. Study on constraint scheduling algorithm for job shop problems with multiple constraint machines. *International Journal of Production Research*, 46 (17), 4785–4801.