A new heuristic approach to solve product mix problems in a multi-bottleneck system

Alireza Rashidi Komijan*

Ph.D., Dep. of Industrial Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran

Mir B. Aryanezhad

Professor, Dep. of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Ahmad Makui

Assistant Professor, Dep. of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Abstract

 Product mix problem (PMP) is one of the most important and complicated problems in production systems. Different approaches have been applied to solve this problem, among them, theory of constraints (TOC) has been widely considered since 1990s. This paper develops a distinguished algorithm to solve product mix problems that is efficient both in single and multi-bottleneck problems. At first, the new algorithm uses a mathematical model to aggregate different priorities assigned to products by different bottlenecks and finds an initial solution. Then tries to improve the solution by solving a set of linear inequalities. It is shown that the new approach obtains better solutions than the previous algorithms.

Keywords: Product mix problem; Theory of constraints; Group decision-making; Integer linear programming

1. Introduction

 Theory of constraints (TOC) is a production planning philosophy that was introduced in The Goal [6]. It aims to improve the system throughput by efficient use of bottleneck(s). When there is a bottleneck in the system, the demands of products cannot be fully met. In this case, the problem is to determine product mix in such a way that high level of profit is obtained. This is called the product mix problem. Although PMP can be formulated as an Integer Linear Programming (ILP) model, it cannot be solved easily when the problem is large scale. If the number of products increases, the difficulty of solving ILP model will increase exponentially. As a result, many researchers focused on heuristic algorithms.

 Some of the researchers developed algorithms to solve PMP in single bottleneck environment whereas others focused on multi-bottleneck systems. Although many algorithms have been developed to solve PMP in multi-bottleneck systems, most of them failed in reaching desirable solutions. The main reason of their failure is that all decisions are made based on the most capacitated constraint (dominant bottleneck). The common process in the existing algorithms is that the product mix is determined based on priorities of products, i.e. the product with the higher priority is produced first. These priorities are assigned by bottlenecks to products. Each bottleneck assigns a priority (weight) to each product. Most of the previous researches just used the priorities assigned by the dominant bottleneck in determining product mix. Considering only one bottleneck in decision making and ignoring the importance of other ones may distort the solution because only a part of information is used. It is similar to searching for the local optimum instead of the global one.

 In this paper, all bottlenecks are considered in product mix decision simultaneously. Each bottleneck assigns a weight to each product. So, there

will be k weight values for each product (k is the number of bottlenecks). A mathematical model is used to aggregate these values and assign an aggregated weight to each product. The initial product mix is obtained using the aggregated weights. Then, the possibility of improving solution is examined through "decrease and increase process". In this process, we try to improve the profit by decreasing the production amount of one product and increasing the other one. In the new algorithm, alternatives to decrease and increase are defined and the best one is selected easily. This causes the speed of solving a problem by the new algorithm to be independent from the dimension of the problem. So, the new algorithm can be applied in large scale problems properly. In the new algorithm, decrease and increase process is done simply by solving a set of linear inequalities. Besides the simplicity of the new algorithm, it is shown that it will reach better solution than the existing algorithms. In Sections 3 and 4 the new algorithm and its logic are described in details.

2. Literature review

Solving PMP using TOC heuristic is an interest area for many researchers. Several algorithms have been developed on this theme since 1990. The first (traditional) algorithm based on TOC was verified by several researchers such as Luebbe and Finch [10] and Patterson [11]. They showed that traditional algorithm could lead the optimum solution as ILP.

Balakrishnan and Cheng [2] criticized the traditional algorithm and showed that it could not reach optimum solution in multi-bottleneck systems. Furthermore, Lee and Plenert [9] showed that the algorithm was inefficient in handling problems in which new product alternatives were to be added to an existing production line. Also, Plenert [12] showed by an example that the traditional algorithm was inefficient in utilizing bottleneck resource properly.

In 1997, Fredendall and Lea [5] revised the traditional algorithm using the concept of dominant bottleneck. The dominant bottleneck is the most capacitated resource. In the revised algorithm, dominant bottleneck assigns a weight (priority) to each product and the initial product mix is developed based on these weight values. Then, the revised algorithm tries to improve the solution using the neighborhood search concept. Fredendall and Lea [5] claimed that the revised algorithm could reach the optimum solution in all cases.

Aryanezhad and Rashidi Komijan [1] discussed several disadvantages of the revised algorithm and showed that it could not reach the optimum solution in all cases and might be stopped in a non-optimum solution. They also showed that the revised algorithm faced burdensome computations when the number of products increase. They discussed that stop condition of the revised algorithm was not defined properly as it might cause the algorithm to reach a non-optimum solution. Then they developed their own algorithm which is called "improved algorithm" [1]. It starts with an initial solution and then finds the best path to reach the improved solution with a logical procedure. The ability of improving solution by the improved algorithm is much more than the revised one but however, it is also limited. Tsai and Lai [15] expanded the improved algorithm to systems including joint products.

Hsu and Chung [7] developed an algorithm that was similar to the algebraic concept of simplex to some extent. The main assumption of this algorithm is that the model is continuous not integer. Clearly, there is no need to heuristic algorithms when the problem is considered continuous as simplex can solve large scale problems using advanced softwares like GAMS.

Souren et al. [14] categorized conditions in which TOC heuristic can reach optimum solution and discussed problems that it fails in reaching optimum.

Rashidi Komijan and Sadjadi [13] considered throughput and late delivery cost as decision making criteria and solved PMP problem using TOPSIS technique.

Among other researches, we can mention Kee and Schmidt [8], Bhattacharya and Vasant [4] and Bhattacharya et al. [3]. Kee and Schmidt [8] developed a general model including TOC and Activity-Based Costing (ABC). Bhattacharya and Vasant [4] solved PMP in such a way that the decision maker reached higher degree of satisfaction and lesser degree of fuzziness. Bhattacharya et al. [3] discussed PMP with fuzzy cost function.

3. The new algorithm logic description

In this section, the logic of the new algorithm is discussed in details. In almost all PMP algorithms, the production plan is developed based on priorities of products. These priorities (weights) are determined by the bottlenecks. If there is one bottleneck in the system, there will be a unique weight vector and production plan can be easily obtained. As the resulted product mix is optimal in view of the bot-

tleneck, it is optimal for the whole system. In a multi-bottleneck system, each bottleneck presents its own weight vector and as a result, there may be several weight vectors. The question is that which one of the weight vectors should be considered in developing production plan. Some preliminary algorithms used the weight vector associated to the dominant bottleneck. Naturally the result may be good in view of that bottleneck not necessarily the others [1]. Considering only one bottleneck in determining product mix and ignoring the importance of other ones is a pitfall for an algorithm.

The new algorithm overcomes this problem letting all bottlenecks contribute in determining product mix. In the new algorithm, bottlenecks and products are treated as decision-makers (DMs) and alternatives respectively. Using this analogy, the product mix problem is considered as a group decision making problem. Each DM assigns a weight to each alternative (individual weights).

Also, as DMs (bottlenecks) do not have equal importance, the weights of DMs are determined in the new algorithm. Then, individual weights are aggregated through a mathematical model. The initial production plan is developed using the aggregated weights.

The rest of the algorithm is dedicated to improve the initial plan. This is done using the "decrease and increase process". Through this process, the scheduler may decrease the production amount of a product and increase the other one provided that the process leads to a higher profit. Clearly there are too many alternatives for decrease and increase even in a medium scale problem.

There are some features in the new algorithm that cut the number of alternatives considerably. In the algorithm, it is discussed that the product candidate to decrease should have higher priority that the product considered to increase. On the other hand, as the initial plan is based on the aggregated weights, it has minimum derivation from the desired plan. So the number of alternatives for decrease and increase process is limited and the time consumed in reaching the solution is decreased considerably.

Decrease and increase process is applied in previous researches [1,5] but in the algorithm, it is done in a completely different manner that leads to better solution than the previous algorithms. The new algorithm is described in Section 4 and through a numerical example, its result is compared to traditional, revised and improved algorithms. It is shown that the new algorithm is more efficient than the previous ones.

4. New algorithm

The following notations are used in the new algorithm:

i	Index of product, $i=1,...,n$,
j	Index of resource, $j=1,...,m$,
t_{ij}	Processing time of product i on resource j,
D_i	Demand of product i,
P_i	Produced units of product i,
SP_i	Selling price of product i,
RM_i	Raw material cost of product i,
CM_i	Contribution margin of product i,
AC_j	Available capacity of resource j,
RC_j	Required capacity of resource j,
PR_{ij}	Priority of product i assigned by bottleneck j,
F_{ij}	Weight of product i assigned by bottleneck j,
W_i	Aggregated weight of product i,
W'_j	Weight of bottleneck j,
L_j	Time left in bottleneck j,
H_j	Maximum number of iterations according to bottleneck j,
H	Real maximum number of iterations (according to the group of bottlenecks),
n	Number of products,
m	Number of bottlenecks.

Step 1. Identify the system bottleneck(s).

Each resource that its required capacity exceeds the available one, is considered as a bottleneck. Required capacity of resource j is calculated as follows:

$$RC_j = \sum_{i=1}^{n} D_i t_{ij} \tag{1}$$

If $RC_j > AC_j$, then j is a bottleneck.

Step 2. Determine the weight of each product assigned by each bottleneck (individual weights).

For each bottleneck, different products have different priorities. Priority of product i in view of bottleneck j is calculated as follows:

$$PR_{ij} = \frac{CM_i}{t_{ij}} \tag{2}$$

where CM_i is calculated as follows:

$$CM_i = SP_i - RM_i \tag{3}$$

Weight of product i in view of bottleneck j is calculated by normalizing PR_{ij} values:

$$F_{ij} = \frac{PR_{ij}}{\sum\limits_{i=1}^{n} PR_{ij}} \tag{4}$$

Typically, weight vector of products in view of bottleneck j is ($F_{1j},...,F_{ij},...,F_{nj}$). Each bottleneck has a weight vector expressing the weights that it assigns to different products. These weight vectors are called individual weights.

Step 3. Develop production plans regarding the weights obtained in Step 2.

In this step, m production plans are developed based on the weights calculated in Step 2. In other words, each bottleneck develops the production plan regarding its own weight vector. Clearly, these plans should be feasible, i.e. the production plan developed by one bottleneck should be feasible for others.

Step 4. Determine the weights of bottlenecks.

To determine the weights of bottlenecks, the profit related to each production plan should be calculated. Normalizing these profit values will lead to the weights of bottlenecks.

Step 5. Determine the weight of each product in view of the group of bottlenecks (aggregated weight).

In order to calculate the aggregated weight of a product based on the individual weights obtained in Step 2, a mathematical model is formulated as follows:

$$\text{Min} \sum_{j=1}^{m} \sum_{i=1}^{n} W_j' \left| W_i - F_{ij} \right|$$

Subject to:

$$\sum_{i=1}^{n} W_i = 1 \tag{5}$$

$$\min_{j} (F_{ij}) \le W_i \le \max_{j} (F_{ij}) \qquad i = 1,2,...,n$$

$$W_i \ge 0$$

The decision variable, W_i, is the weight of product i assigned by the group of bottlenecks (aggregated weight of product i). Individual weights of product i (F_{ij}) should have the least difference from the aggregated weight of product i. So, the objective function of the model is to minimize these difference values. The first constraint is to normalize aggregated weights and the second one ensures that the aggregated weight of product i lies between the least and most individual weights assigned to it.

Step 6. Develop production plan regarding the aggregated weights.

In this step, the production plan is developed by the group of bottlenecks using the aggregated weights obtained in Step 5. The profit resulted from the aggregated plan may be improved. The improvement will occur during a tradeoff, i.e. when we decrease the production amount of one product and dedicate the time left in bottlenecks to other product. In the following steps, the possibility of improving the profit is examined.

Step 7. Determine allowable tradeoffs.

Decreasing production amount of product i and increasing product l is considered as an allowable tradeoff if $W_i \ge W_l$. In other words, if $W_i < W_l$, decreasing i and increasing l is useless. The reason is simple: In the initial plan, at first we considered l

for production and allocated bottlenecks time to it and produced P_l units. If it were possible to produce additional units of l, it would be done in the initial plan automatically. Without regarding to the mathematical model, there may be $2!/(n-2)!$ alternatives for decrease and increase process. Using the mathematical model and the condition described in Step 7, the number of alternatives will be cut significantly.

Step 8. Determine how many l should be increased in turn of decreasing one unit of i.

Assume that k is the number of product l that should be increased in turn of decreasing one unit of i. k is the least integer that satisfies the following inequality:

$$k \, CM_l \geq CM_i \qquad (6)$$

So, in each iteration, it is allowable to decrease one unit i provided that k units are added to l. Now the question is that how many iterations are allowable to be done. This is answered in the following step.

Step 9. Determine the maximum number of iterations.

Assuming that decreasing one unit of i and increasing k units to l as one iteration, we want to know that how many iterations are allowable to be done. L_j is the time left in bottleneck j regarding production plan based on aggregated weights and before starting decreasing and increasing process. Decreasing one unit of i causes that t_{ij} minutes are added to the time left in bottleneck j and increasing k units to l consumes kt_{lj} minutes. If $kt_{lj} - t_{ij} \leq 0$, the iterations can be done until production units of i falls to zero or the demand of l is fully met. If $kt_{lj} - t_{ij} > 0$, the iterations can be done until production units of i falls to zero, the demand of l is fully met or bottleneck j exhausts. These can be summarized as follows:

If $kt_{lj} - t_{ij} \leq 0$

Then $H_j = Min(P_i, [\dfrac{D_l - P_l}{k}])$

Otherwise

$$H_j = Min(P_i, [\dfrac{D_l - P_l}{k}], [\dfrac{L_j}{kt_{lj} - t_{ij}}])$$

where $[\dfrac{D_l - P_l}{k}]$ and $[\dfrac{L_j}{kt_{lj} - t_{ij}}]$ are integer values of $\dfrac{D_l - P_l}{k}$ and $\dfrac{L_j}{kt_{lj} - t_{ij}}$ respectively.

As the tradeoff must be feasible for all bottlenecks, the above calculation is repeated for all bottlenecks. Each bottleneck suggests a certain number of iteration (H_j) and the real maximum number of iterations is the minimum of values suggested by the bottlenecks:

$$H = \underset{j}{Min} \, H_j \qquad (7)$$

Step 10. Calculate net profit change related to the tradeoff between i and l.

In the previous steps it was cleared that:

a) It is allowable to decrease i and increase l.

b) k units of l should be increased in turn of decreasing one unit of i.

c) Part (b) can be repeated for H times.

Clearly, net profit change is calculated as follows:

Net Profit Change = $H(k \, CM_l - CM_i)$

Steps 8 to 10 is repeated for all allowable tradeoffs determined in Step 7. The tradeoff with the highest net profit change is selected.

5. Numerical example

The new algorithm is explained through an example. This is the example of Hsu and Chung [7]. Assume that a factory produces four products: *R, S, T* and *U*.

Demand, selling price and raw material cost of the products are shown in Table 1. The contribution margin (CM) is the difference between the selling price and the raw material cost. The factory uses seven resources: A, B, C, D, E, F and G. The processing time of each product in each station is presented in Table 2.

Step 1. According to the last row of Table 2, A, B, C, D and F are bottlenecks.

Table 1. Weekly demand, selling price and raw material cost of each product.

Product	Demand (Unit)	Selling Price ($)	Raw Material Cost ($)	CM ($)
R	70	100	20	80
S	60	120	60	60
T	50	110	60	50
U	150	50	20	30

Table 2. Processing time in minute, available and required capacity.

	Resource						
Product	**A**	**B**	**C**	**D**	**E**	**F**	**G**
R	20	5	10	0	5	5	20
S	10	10	5	30	5	5	5
T	10	5	10	15	20	5	10
U	5	15	10	5	5	15	0
Available Capacity	2400	2400	2400	2400	2400	2400	2400
Required Capacity	3250	3450	3000	3300	2400	3150	2200
Difference	-850	-1050	-600	-900	0	-750	200

Table 3. Products priorities in view of each bottleneck (PR_{ij}).

	R	S	T	U
Bottleneck A	4	6	5	6
Bottleneck B	16	6	10	2
Bottleneck C	8	12	5	3
Bottleneck D	0	2	3.33	6
Bottleneck F	16	12	10	2

Step 2. Using contribution margin of each product (the last column of Table 1) and processing times in Table 2, the products priorities in view of each bottleneck are obtained (see Table 3).

Normalizing each row will lead to the weight vectors:

$$(F_{RA}, F_{SA}, F_{TA}, F_{UA}) = (0.19, 0.286, 0.238, 0.286)$$

$$(F_{RB}, F_{SB}, F_{TB}, F_{UB}) = (0.471, 0.176, 0.294, 0.059)$$

$$(F_{RC}, F_{SC}, F_{TC}, F_{UC}) = (0.286, 0.429, 0.179, 0.107)$$

$$(F_{RD}, F_{SD}, F_{TD}, F_{UD}) = (0, 0.176, 0.294, 0.53)$$

$$(F_{RF}, F_{SF}, F_{TF}, F_{UF}) = (0.4, 0.3, 0.25, 0.05)$$

Step 3. The production plan is developed based on the above weights as follows:

Production plan in view of bottleneck A: (0R, 60S, 0T, 120U; Profit: 7200)

Production plan in view of bottleneck B: (70R, 50S, 50T, 0U; Profit: 11100)

Production plan in view of bottleneck C: (70R, 60S, 40T, 0U; Profit: 11200)

Production plan in view of bottleneck D: (0R, 0S, 30T, 150U; Profit: 6000)

Production plan in view of bottleneck F: (70R, 60S, 40T, 0U; Profit: 11200)

To illustrate, consider the product mix in view of B. As R is assigned the highest weight by B, it is produced first. After meeting its demand, the next important product, T, is produced. After producing 50 T, the time left in the bottlenecks is only enough to produce 50 S. The profit is calculated by multiplying the produced amount of each product and its contribution margin:

$$\text{Pr}\,ofit = 70 \times 80 + 50 \times 60 + 50 \times 50 = 11100$$

Step 4. Weights of the bottlenecks are calculated by normalizing the profit values:

$$(W'_A, W'_B, W'_C, W'_D, W'_F) = (0.154, 0.238, 0.24, 0.128, 0.24)$$

Step 5. The following model is used to determine the weight of each product assigned by the group of bottlenecks (aggregated weight).

$$
\begin{aligned}
\text{Min} \quad & 0.154|W_R - 0.19| + 0.238|W_R - 0.471| \\
& + 0.24|W_R - 0.286| + 0.128|W_R - 0| \\
& + 0.24|W_R - 0.4| + 0.154|W_S - 0.286| \\
& + 0.238|W_S - 0.176| + 0.24|W_S - 0.429| \\
& + 0.128|W_S - 0.176| + 0.24|W_S - 0.3| \\
& + 0.154|W_T - 0.238| + 0.238|W_T - 0.294| \\
& + 0.24|W_T - 0.179| + 0.128|W_T - 0.294| \\
& + 0.24|W_T - 0.25| + 0.154|W_U - 0.286| \\
& + 0.238|W_U - 0.059| + 0.24|W_U - 0.107| \\
& + 0.128|W_U - 0.53| + 0.24|W_U - 0.05|
\end{aligned}
$$

Subject to:

$$W_R + W_S + W_T + W_U = 1 \qquad (8)$$

$$0 \leq W_R \leq 0.471$$

$$0.176 \leq W_S \leq 0.429$$

$$0.179 \leq W_T \leq 0.294$$

$$0.05 \leq W_U \leq 0.53$$

The above model is simplified to a linear one:

$$
\begin{aligned}
\text{Min} \quad & 0.154(W^+_{RA} + W^-_{RA}) + 0.238(W^+_{RB} + W^-_{RB}) \\
& + 0.24(W^+_{RC} + W^-_{RC}) + 0.128(W^+_{RD} + W^-_{RD}) \\
& + 0.24(W^+_{RF} + W^-_{RF}) + 0.154(W^+_{SA} + W^-_{SA}) \\
& + 0.238(W^+_{SB} + W^-_{SB}) + 0.24(W^+_{SC} + W^-_{SC})
\end{aligned}
$$

$$
\begin{aligned}
& + 0.128(W^+_{SD} + W^-_{SD}) + 0.24(W^+_{SF} + W^-_{SF}) \\
& + 0.154(W^+_{TA} + W^-_{TA}) + 0.238(W^+_{TB} + W^-_{TB}) \\
& + 0.24(W^+_{TC} + W^-_{TC}) + 0.128(W^+_{TD} + W^-_{TD}) \\
& + 0.24(W^+_{TF} + W^-_{TF}) + 0.154(W^+_{UA} + W^-_{UA}) \\
& + 0.238(W^+_{UB} + W^-_{UB}) + 0.24(W^+_{UC} + W^-_{UC}) \\
& + 0.128(W^+_{UD} + W^-_{UD}) + 0.24(W^+_{UF} + W^-_{UF})
\end{aligned}
$$

Subject to:

$$W^+_{RA} - W^-_{RA} = W_R - 0.19 \qquad (9)$$

$$W^+_{RB} - W^-_{RB} = W_R - 0.471$$

$$W^+_{RC} - W^-_{RC} = W_R - 0.286$$

$$W^+_{RD} - W^-_{RD} = W_R$$

$$W^+_{RF} - W^-_{RF} = W_R - 0.4$$

$$W^+_{SA} - W^-_{SA} = W_S - 0.286$$

$$W^+_{SB} - W^-_{SB} = W_S - 0.176$$

$$W^+_{SC} - W^-_{SC} = W_S - 0.429$$

$$W^+_{SD} - W^-_{SD} = W_S - 0.176$$

$$W^+_{SF} - W^-_{SF} = W_S - 0.3$$

$$W^+_{TA} - W^-_{TA} = W_T - 0.238$$

$$W^+_{TB} - W^-_{TB} = W_T - 0.294$$

$$W^+_{TC} - W^-_{TC} = W_T - 0.179$$

$$W^+_{TD} - W^-_{TD} = W_T - 0.294$$

$$W^+_{TF} - W^-_{TF} = W_T - 0.25$$

$$W^+_{UA} - W^-_{UA} = W_U - 0.286$$

$$W_{UB}^{+} - W_{UB}^{-} = W_U - 0.059$$

$$W_{UC}^{+} - W_{UC}^{-} = W_U - 0.107$$

$$W_{UD}^{+} - W_{UD}^{-} = W_U - 0.53$$

$$W_{UF}^{+} - W_{UF}^{-} = W_U - 0.05$$

$$W_R + W_S + W_T + W_U = 1$$

$$0 \leq W_R \leq 0.471$$

$$0.176 \leq W_S \leq 0.429$$

$$0.179 \leq W_T \leq 0.294$$

$$0.05 \leq W_U \leq 0.53$$

$$W_{ij}^{+}, W_{ij}^{-} \geq 0$$

The optimum solution is:

$$(W_R, W_S, W_T, W_U) = (0.343, 0.3, 0.25, 0.107)$$

Step 6. The production plan regarding the aggregated weights is 70R, 60S, 40T, 0U and the profit is 11200 dollars.

Step 7. As the priority of products are R, S, T and U, allowable tradeoffs are shown in Table 4.

Step 8. To illustrate the remaining steps, consider decreasing T and increasing U (the complete calculations are in Table 5). As contribution margins of T and U are 50 and 30 respectively, two units of U should be increased in turn of decreasing one unit of T. So $k=2$ for this tradeoff.

Table 4. Allowable tradeoffs.

Decrease from	Increase to	k value
R	S	2
R	T	2
R	U	3
S	T	2
S	U	3
T	U	2

Step 9. Now the question is that how many times we can decrease one unit of T and increase two units to U. According to the product mix developed in Step 6, the time left in bottlenecks A to F are 0, 1250, 1000, 0, 1550 minutes. Decreasing one unit of T causes that 10, 5, 10, 15 and 5 minutes are added to the time left in bottlenecks and increasing two units to U consumes 10, 30, 20, 10 and 30 minutes. According to bottleneck A, the iterations can be done 40 times:

$$kt_{UA} - t_{TA} = 2 \times 5 - 10 = 0$$

$$H_A = Min(P_T, [\frac{D_U - P_U}{k}]) = 40$$

Similarly, the following values are calculated:

$$kt_{UB} - t_{TB} = 2 \times 15 - 5 > 0$$

$$H_B = Min(P_T, [\frac{D_U - P_U}{k}], [\frac{L_B}{kt_{UB} - t_{TB}}]) = 40$$

$$kt_{UC} - t_{TC} = 2 \times 10 - 10 > 0$$

$$H_C = Min(P_T, [\frac{D_U - P_U}{k}], [\frac{L_C}{kt_{UC} - t_{TC}}]) = 40$$

$$kt_{UD} - t_{TD} = 2 \times 5 - 15 < 0$$

$$H_D = Min(P_T, [\frac{D_U - P_U}{k}]) = 40$$

$$kt_{UF} - t_{TF} = 2 \times 15 - 5 > 0$$

$$H_F = Min(P_T, [\frac{D_U - P_U}{k}], [\frac{L_F}{kt_{UF} - t_{TF}}]) = 40$$

$$H = Min(H_A, H_B, H_C, H_D, H_F) = 40$$

H values for other tradeoffs, that are calculated in a similar way, are zero.

Step 10. Decreasing 40 units from T and increasing 80 units to U will increase the profit value for 400 dollars.

As H values for other tradeoffs are aero, the net profit change according to them are zero. As a result, decreasing T and increasing U is selected and the improved plan is 70R, 60S, 0T and 80U and the profit will be 11600.

Steps 8 to 10 are repeated until no profitable tradeoff is left. The results are summarized in Table 5.

Table 5. The tradeoffs calculations.

Selected Tradeoff						
Decrease from	Increase to	K	H	Net change	Product mix	Total profit
T	U	2	40	400	70R, 60S, 0T, 80U	11600
R	T	2	6	120	64R, 60S, 12T, 80U	11720
T	U	2	8	80	64R, 60S, 4T, 96U	11800
R	T	2	2	40	62R, 60S, 8T, 96U	11840

Table 6. Comparison of the results.

	Traditional Algorithm	Revised Algorithm	Improved Algorithm	New Algorithm	ILP Model
Product mix	70R,50S,50T,0U	64R,2S,50T,120U	70R,60S,80T,0U	62R,60S,8T,96U	51R,38S,50T,100U
Profit	11100	11340	11600	11840	11860

So, the final plan is 62R, 60S, 8T and 96U and the profit is 11840. The result of the new algorithm is compared to the traditional, revised and improved algorithms as well as ILP model in Table 6. The steps of these algorithms along with the solving procedures are described in the appendixes.

6. Conclusion

A number of algorithms have been developed on determining product mix under TOC. Due to the inefficiency of these algorithms, a new algorithm based on a different and new approach is developed. This paper considers product mix problem as a group decision making problem and benefits from using a linear model to aggregate products weights and finds an initial solution. Then tries to improve the solution based on a new logic for decrease and increase process. Using the linear model ensures that the number of alternatives for decrease and increase is reduced significantly. This helps the algorithm to reach the final solution in a short time. Comparison of the new algorithm with the traditional, revised and improved ones shows that the product mix resulted by the new algorithm is more desirable than all previous algorithms.

References

[1] Aryanezhad, M. B. and Rashidi Komijan, A., 2004, An improved algorithm for optimizing product mix under the theory of constraints. *International Journal of Production Research*, 42, 4221-4233.

[2] Balakrishnan, J. and Cheng, C. H., 2000, Theory of constraints and linear programming: A re-examination. *International Journal of Production Research,* 38(6), 1459-1463.

[3] Bhattacharya, A., Vasant, P., Andreeski, C., Barsoum, N., Kolemisevska, T., Dinibutun, A. T. and Dimirovski, G. M., 2006, Decision making in TOC-product-mix selection via fuzzy cost function optimization. *International Journal of Production Economics.*

[4] Bhattacharya, A. and Vasant, P., 2007, Soft-sensing of level of satisfaction in TOC product-mix decision heuristic using robust fuzzy-LP. *European Journal of Operational Research,* 177(1), 55-70.

[5] Fredendall, L. D. and Lea, B. R., 1997, Improving the product mix heuristic in the theory of constraints. *International Journal of Production Research*, 35, 1535-1544.

[6] Goldratt, E. M., 1984, *The Goal.* Croton-on-Hudson, NY: North River Press.

[7] Hsu, T. C. and Chung, S. H., 1998, The TOC-based algorithm for solving product mix problems. *Production Planning and Control,* 9(1), 36-46.

[8] Kee, R. and Schmidt, C., 2000, A comparative analysis of utilizing activity-based costing and the theory of constraints for making product mix decisions. *International Journal of Production Economics,* 63, 1-17.

[9] Lee, T. N. and Plenert, G., 1993, Optimizing theory of constraints when new product alternatives exist. *Production and Inventory Management Journal*, 34(3), 51-57.

[10] Luebbe, R. and Finch, B., 1992, Theory of constraints and linear programming: A comparison. *International Journal of Production Research,* 30(6), 1471-1478.

[11] Patterson, M. C., 1992, The product-mix decision: A comparison of theory of constraints and labor-based management accounting. *Production and Inventory Management Journal*, 33(3), 80-85.

[12] Plenert, G., 1993, Optimized theory of constraints when multiple constrained resources exist. *European Journal of Operational Research,* 70(1), 126-133.

[13] Rashidi Komijan, A. and Sadjadi, S. J., 2005, *Optimizing Product Mix in a Multi-bottleneck Environment.* Proceedings of International Conference on Computational Science and Its Applications, Suntec, Singapore, 388-396.

[14] Souren, R., Ahn, H. and Schmitz, C., 2005, Optimal product mix decisions based on theory of constraints? Exploring rarely emphasized premises of throughput accounting. *International Journal of Production Research*, 43(2), 361-374.

[15] Tsai, W. H. and Lai, C. W., 2007, An algorithm for optimizing joint products decision based on the theory of constraints. *International Journal of Production Research,* 45(15), 3421-3437.

Appendix 1: Traditional TOC algorithm [10,11]

Step 1. Identify system bottlenecks.

As shown in Table 2, A, B, C, D and F are bottlenecks.

Step 2. Calculate the products priorities in view of dominant bottleneck.

Resource B is the most capacitated constraint and is called dominant bottleneck. As shown in Table 3, priorities of R, S, T and U are 16, 6, 10 and 2 respectively.

Step 3. Determine product mix using the priorities calculated in Step 2.

According to the priorities in view of B, product mix is 70R, 50S, 50T and 0U with the profit of 11100.

Appendix 2: Revised algorithm [5]

Step 1. Identify system bottlenecks and dominant bottleneck.

As shown in Table 2, A, B, C, D and F are bottlenecks and B is dominant bottleneck.

Step 2. Check if dominant bottleneck is properly chosen or not.

The product mix in view of B is 70R, 50S, 50T and 0U. The first bottleneck that exhausts regarding this plan is A. So, A is assumed as the real dominant bottleneck.

Step 3. Determine product mix in view of dominant bottleneck.

The product mix in view of A is 0R, 60S, 0T and 120U and the profit is 7200 dollars.

Step 4. Develop the set of products candidate to decrease and increase.

As $P_{UA}(L_A + t_{SA})/CM_S = 6(1200 + 10)/60 \geq 1$ and $P_U \leq D_U$, the set of products candidate to decrease and increase is *{S, U, T, R}*.

Step 5. The result of decrease and increase process is shown in Table 7.

Table 7. Decrease and increase process.

Product mix	Profit
0R, 60S, 0T, 120U	7200
0R, 59S, 2T, 120U	7240
0R, 58S, 4T, 120U	7280
.	.
.	.
0R, 35S, 50T, 120U	8200
2R, 34S, 50T, 120U	8300
4R, 33S, 50T, 120U	8400
.	.
.	.
62R, 4S, 50T, 120U	11300
63R, 3S, 50T, 120U	11320
64R, 2S, 50T, 120U	11340*
64R, 1S, 50T, 121U	11310
64R, 0S, 50T, 122U	11280

Appendix 3: Improved algorithm [1]

Step 1. Identify system bottlenecks.

As shown in Table 2, A, B, C, D and F are bottlenecks.

Step 2. Calculate the products priorities in view of each bottleneck.

Products priorities in view of each bottleneck are shown in Table3.

Step 3. Develop the initial plan using priorities in view of dominant bottleneck.

As B is dominant bottleneck, product mix is 70R, 50S, 50T and 0U with the profit of 11100.

Step 4. Determine feasible alternatives to improve the throughput.

Feasible alternatives to decrease and increase are defined based on four simple rules (assume that i and l are products candidate to decrease and increase respectively):

a) i should have higher priority than the l in view of dominant bottleneck,

b) The demand of l is not fully met,

c) l has higher priority than i in view of at least on bottleneck,

d) l has the higher priority than products that their demands have not been fully met in view to the bottleneck in which l is more important that i.

According to rule (a), alternatives are shown in Table 8.

As the demand of T is fully met, decreasing R and increasing T is omitted. The remaining alternatives are feasible.

Table 8. Alternative for decrease and increase.

Decrease from	Increase to
R	**T**
R	**S**
R	**U**
T	**S**
T	**U**
S	**U**

Step 5. Select the best alternative.

Consider the alternative that suggests to decrease R and increase S. According to the time left in bottlenecks, decreasing one unit of R will be enough to increase one unit to S. As shown in Table 3, the disagreement of bottlenecks against decreasing one unit of R is (4+16+8+16=44) and their satisfaction to increasing one unit to S is (6+6+12+2+12=38). So, the score of this alternative is (38-44=-6). The score of other alternatives are calculated in the similar way and alternative "T to S" is selected with the score of 4.67.

Step 6. Decreasing and increasing process.

At each iteration, one unit of T is reduced and one unit is increased to S. This will increase the profit for 10 dollars. After 10 iteration, the product mix will be 70R, 60S, 40T and 0U and the profit is 11200.

As the demand of S is fully met, we consider decreasing T and increasing U. It is possible to decrease 40 units of T and increase 80 units to U. The product mix is 70R, 60S, 0T and 80U and the profit is 11600 dollars. The improved algorithm stops at this point.

Appendix 4: Integer linear programming model

The general model is as follows:

X_i :Produced units of product i $i = 1,...,n$

$$\text{Max } \sum_{i=1}^{n} CM_i X_i$$

Subject to:

$$\sum_{i=1}^{n} t_{ij} X_i \leq AC_j \qquad j = 1,...,m$$

$$X_i \leq D_i \qquad i = 1,...,n$$

$$X_i \geq 0 \qquad i = 1,...,n$$

The numerical example is formulated as follows:

X_i :Produced units of product i $i = R, S, T, U$

$$\text{Max } 80X_R + 60X_S + 50X_T + 30X_U$$

Subject to:

$$20X_R + 10X_S + 10X_T + 5X_U \leq 2400$$

$$5X_R + 10X_S + 5X_T + 15X_U \leq 2400$$

$$10X_R + 5X_S + 10X_T + 10X_U \leq 2400$$

$$30X_S + 15X_T + 5X_U \leq 2400$$

$$5X_R + 5X_S + 20X_T + 5X_U \leq 2400$$

$$5X_R + 5X_S + 5X_T + 15X_U \leq 2400$$

$$20X_R + 5X_S + 10X_T \leq 2400$$

$$X_R \leq 70$$

$$X_S \leq 60$$

$$X_T \leq 50$$

$$X_U \leq 150$$

$$X_i \geq 0$$

The optimum solution is 51R, 38S, 50T and 100U and the profit is 11860.