A MATHEMATICAL METHOD FOR MANAGING THE SYSTEM CONSTRAINT

S. A. Badri*

Department of Industrial Engineering, Iran University of Science and Technology
Postal Code 16846-13114, Tehran, Iran, s_a_badri@yahoo.com

M. B. Aryanezhad

Department of Industrial Engineering, Iran University of Science and Technology
Postal Code 16846-13114, Tehran, Iran, mirarya@iust.ac.ir

*Corresponding Author

(Received: October 22, 2008 – Accepted in Revised Form: March 11, 2010)

Abstract The goal of theory of constraints (TOC) is to maximize output, which is achieved by identifying and managing the critically constrained resources. To manage the constraints, Goldratt proposed five focusing steps (5FS). If we increase constrained output, the output of system will be increased. In this paper, we focus on step four of the 5FS and use the remained capacity of non-constraint to elevate the system's constraint.

Keywords Theory of constraints, five focusing steps, queueing system, threshold

چکیده هدف تئوری محدودیت‌ها (TOC) بیشینه نمودن خروجی است، که از طریق شناسایی و مدیریت منابع محدود حاصل می‌شود. به منظور مدیریت محدودیت‌ها، گلدرات پنج مرحلهٔ اجرایی (5FS) پیشنهاد نموده است. اگر ما خروجی محدودیت را افزایش دهیم، خروجی بدست آمده از سیستم افزایش خواهد یافت. در این مقاله، ما بر روی گام چهار از مراحل پنج‌گانهٔ تئوری محدودیت‌ها تمرکز نموده و از ظرفیت باقیمانده در غیر محدودیت در جهت بالابردن سطح محدودیت سیستم استفاده می‌کنیم.

1. INTRODUCTION

Theory of constraints (TOC) is a systems management philosophy and an effective approach to production planning and control developed by Goldratt [1-3]. It is based on the fact that constraints determine the performance of a system and that any system contains only a few constraints. Goldratt and Fox [4] define three important performance indicators, which are:

(a) Throughput. Defined as the rate at which the manufacturing business sells its finished products.
(b) Inventory. Inventory is the storage cost of the raw material components and finished goods which are not yet been sold.
(c) Operational expenses. It is the cost incurred in turning inventory into throughput.

If any of the aforementioned performance indicators change, it will affect the financial measurement at the strategic level. To overcome these difficulties, theory of constraints has been emerged as an efficient management philosophy. It focuses on the goal of manufacturing organizations i.e. to increase throughput with the reduction in inventory and operational cost [5].

The TOC has two major components. First, a philosophy which underpins the working principles of TOC. This is often referred to as TOC's 'logistics paradigm' and consists of five steps for on-going improvement, the drum-buffer-rope (DBR) scheduling methodology and the buffer management information system. To address the policy constraints and effectively implement the process of on-going improvement, Goldratt [3, 6] developed a generic approach called Thinking process (TP). This is the second component of TOC. The working principles of TOC and the application procedure of the TP are discussed in the following two subsections.

1.1 On-going improvement process Main focus of TOC is to determine the system constraints and effectively manage it. To manage the constraints, Goldratt [7] proposed five steps process of on going improvement also known as five focusing steps or 5FS. These steps are:

1. *Identify the system's constraint(s).* These may be physical (e.g. materials, machines, people, demand level) or managerial. Constraints determine the systems throughput and should therefore attract serious attention from managers.
2. *Decide how to exploit the system's constraint(s).* If the constraint is physical, then the objective should be to make the constraint as effective as possible. A managerial constraint should not be exploited but be eliminated and replaced with a policy which will support to increase throughput.
3. *Subordinate everything else to the above decision.* This means that every other component of the system (non-constraints) must be adjusted to support the maximum effectiveness of the constraint. Because constraints dictate a firm's throughput, resource synchronization with the constraint will lead to more effective resource utilization.
4. *Elevate the system's constraint(s).* If existing constraints are still the most critical in the system, rigorous improvement efforts on these constraints will improve their performance. As the performance of the constraints improve, the potential of non-constraint resources can be better realized, leading to improvements in overall system performance. Eventually the system will encounter a new constraint.
5. *If in any of the previous steps a constraint is broken, go back to step 1. Do not let inertia become the next constraint.* TOC is a continuous process and no policy (or solution) will be appropriate (or correct) for all time or in every situation. It is critical for the organization to recognize that as the business environment changes, business policy has to be refined to take account of those changes.

1.2. Five step thinking process According to Goldratt [3], while dealing with constraints, managers are required to make three generic decisions. These are:

(a) Decide what to change.
(b) Decide what to change to.
(c) Decide how to cause the change.

To address these questions, the TP prescribes a set of five tools in the form of cause-and-effect diagrams.

1. *Current reality tree.* Current reality tree identifies the root causes and the core problem of an organization. The effectiveness of the current reality tree depends on the experience and intuition of the involved individuals.
2. *Evaporating cloud.* Evaporation cloud try to identify the solution of the core problem generated in the previous step.
3. *Future reality tree.* This is also known as evaluation and improvement step. It deals with the implementation process of the solution identified in previous step. After the implementation, evaluation is carried out to improve the solution before they are really implemented.
4. *Prerequisite tree.* Prerequisite tree helps to surface and eliminate the obstacles in the implementation process of a chosen solution, by determining all the intermediate steps that are necessary to execute the chosen solution.
5. *Transition tree.* This step is generally carried out when the people executing the plan are not the same as one who develops it. Its function is to identify all the action needed in the current environment to achieve the intermediate objectives that were identified earlier in the prerequisite tree.

The product mix decision problem is one important application of the TOC's ongoing improvement process. Extensive studies have been carried out to identify the optimal product mix to maximize the profit [8]. Goldratt [2] proposed traditional TOC algorithm for solving the product mix problem. Although, TOC algorithm is capable to solve the problems of this nature, in cases of multi constraint resource problem it is unable to provide optimal or near optimal solution [9]. The failure of TOC algorithm to identify the solution under the existence of multi constraint resources promoted the various researchers to refine the TOC algorithm to overcome its deficiency. The Integer linear programming (ILP) has been explored with

TOC product mix heuristic by Luebbe and Finch [10]. Lee and Plenert [11] and Plenert [9] tested ILP formulation to identify a product mix and compared how it fully utilized the bottlenecks and proved its superiority over the TOC heuristic's solution. However, the computational time required to achieve the optimal solution using the ILP is high, thus required some other heuristics to surmount it. Fredenall and Lea [12] proposed a revised TOC (RTOC) heuristic that identifies the optimal product mix for the problems where the TOC product mix heuristic failed previously. RTOC provided same results as ILP in most of the cases. Aryanezhad and Komijan [13] proposed an improved algorithm, which could reach optimum solution in determining product-mix under TOC. Efficiency of their improved algorithm in reaching the optimum solution was compared with the ILP method of Fredendall and Lea [12] through an example. Tsai et al [14] developed an algorithm for optimizing a joint products further processing decision under the theory of constraints. Bhattacharya et al. [15] presented results over their fuzzy linear programming approach to the problem. Wang et al. [16] used immune-based approaches, such as self-adaptive regulation and vaccination.

All above papers and the literature has focused more on the first two steps of the TOC. In this paper, we focus on step four of the 5FS and use the remained capacity of non-constraint to elevate the system's constraint.

In the current section, a brief idea of TOC is presented. A review of literature on TOC product mix problem is also discussed. Section 2 provides the description of the problem. Section 3 is devoted to model. Section 4 contains proposed heuristic algorithm. The illustrative example is presented in section 5. Section 6 deals with computational results. The conclusion is given in section 7.

2. PROBLEM DESCRIPTION AND NOTATION

2.1 Problem description We consider simple manufacturing system that has one capacity-constrained resource (CCR). We focus on two stations in this system: CCR and one of non-constraint resource named, NC. Demands arrive to the CCR in a Poisson manner with rate λ demands/hour. Operations at CCR are done independently with exponential operation times, with mean operation time equal to μ^{-1}. The work in the NC is interruptible, allowing a worker in the NC to switch to CCR with little delay or lost productivity. The switch from NC to CCR would occur at those moments when the queue of waiting demands in the CCR becomes "too long". The reverse switch from the CCR to the NC occurs once the number of demands is sufficiently small. For the NC, we must have sufficiently many workers there (on average) so that the NC due to switching workers stays non-constraint. The goal is find thresholds for switching workers between CCR and NC to increase the output of system.

2.2 Notation The following notations are made for the remainder of this paper in order to make the analysis mathematical.

i	index for number of workers at CCR
j	index for the integer value of demand at CCR
w	total workers at CCR and NC stations (a positive integer)
w_c	number of workers permanently assigned to the CCR
$\overline{w}_{nc}$	time-average number of workers at NC
$\overline{w}_0$	minimum specified threshold for $\overline{w}_{nc}$
λ	demands arrival rate at CCR
μ	service rate for CCR worker
k	finite capacity of demand for CCR
C_{ij}	minimum value of *i* and *j*
$\overline{w}_c$	time-average number of workers at CCR
π_{ij}	steady-state probability of the system being in state (*i*, *j*)
L	lower threshold for switching worker from CCR to NC
U	upper threshold for switching worker from NC to CCR

3. THE MODEL

We assume that the CCR is a continuous time Markov process having a two-dimensional state space.

(i, j)=system state having i workers and j demands in the CCR. $i = w_c, w_c + 1, \ldots, w$; $j = 0,1,2,\ldots,k$

In the (i, j) state space, probabilistic flows tend to travel in two-dimensional loops or cycles. This behavior is distinctly different from the more familiar birth and death queues found in queueing textbooks. A typical loop may start near the beginning with the system nearly empty, i.e., index j at or near zero and index i at w_c. Then, as more demands enter the CCR, the state of the system moves positively along with j direction. Eventually, a worker from the NC comes to the CCR, perhaps twice or even three times. As these workers are added, the state of the system then moves positively along with i direction. Eventually, CCR queue subsides, moving the state toward lower values j, and ultimately the one or more workers come back to the NC. As the number of CCR workers decreases, the system state tends back toward low values of all two indices, toward system states having few demands and few workers. This cycle repeats continuously.

There are two events, which lead to change the state: a demand arrives at the CCR and the operation completion of a demand at CCR.

3.1 Switching thresholds We define two thresholds: the "upper threshold", used for determining the time when the workers move from the NC to the CCR, and "lower threshold", used for the reverse assignment. For simplicity, we consider policies in which (i) worker movements can occur only upon either entrance of a new demand at the CCR queue or operation completion of a demand at CCR (the former allows a worker to be switched from the NC to the CCR; the latter allows the reverse). (ii) only one worker may be switched at a time. We want to determine optimal values of thresholds for moving workers from one station to the other.

Define the upper threshold as

U : Whenever the ratio of j (number of demands at CCR) to the i (number of workers at CCR) is greater than or equal to U , the number of active CCR workers increases from i to i+1. where $i = w_c, w_c + 1, \ldots, w - 1$ and $j = 0,1,2,\ldots,k$.

Define the lower threshold as

L : Whenever the ratio of j (number of demands at CCR) to the i (number of workers at CCR) is less than or equal to L, the number of active CCR workers decreases from i to i-1. where $i = w_c + 1, w_c + 2, \ldots, w$ and $j = 0,1,2,\ldots,k$.

3.2 Markovian balance equations In this section, we develop the required Markovian balance equations for a specific values of thresholds. In order to clarify the balance equation, all possible transitions for (i,j) are shown in Figure 1.

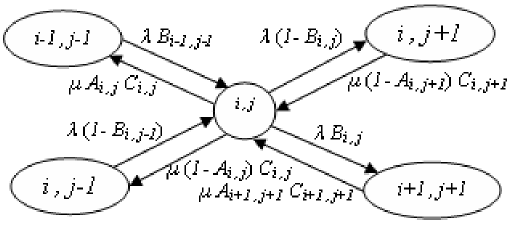

Figure 1. All possible transitions for (i,j) where $w_c < i < w$ and $0 < j < k$

The general detailed balance equation can be written as follows:

$$\begin{aligned}
&\left[\lambda\, B_{ij}\left(1-E_j\right)\left(1-D_i\right)+\lambda\left(1-B_{ij}\right)\left(1-E_j\right)\right.\\
&+\mu\, A_{ij}\, C_{ij}\left(1-G_i\right)\left(1-F_j\right)\\
&\left.+\mu\left(1-A_{ij}\right)C_{ij}\left(1-F_j\right)\right]\pi_{ij}\\
&=\mu\, A_{i+1\,j+1}\, C_{i+1\,j+1}\left(1-E_j\right)\left(1-D_i\right)\ \pi_{i+1\,j+1}\\
&+\mu\left(1-A_{i\,j+1}\right)C_{i\,j+1}\left(1-E_j\right)\pi_{i\,j+1}\\
&+\lambda\left(1-G_i\right)\left(1-F_j\right)B_{i-1\,j-1}\quad \pi_{i-1\,j-1}\\
&+\lambda\left(1-B_{i\,j-1}\right)\left(1-F_j\right)\ \pi_{i\,j-1} \qquad\qquad \forall\, i\,,\, j
\end{aligned}\tag{1}$$

This equation holds for any $i = w_c, w_c + 1, \ldots, w$ and $j = 0,1,\ldots,k$. where

$$A_{i\,j} = \begin{cases} 1 & if\ \dfrac{j}{i} \le L \\ 0 & otherwise \end{cases}\tag{2}$$

$$B_{ij} = \begin{cases} 1 & if \ \dfrac{j}{i} \geq U \\ 0 & otherwise \end{cases} \tag{3}$$

$$C_{ij} = \min(i, j) \tag{4}$$

$$D_i = \begin{cases} 1 & if \ i = w \\ 0 & otherwise \end{cases} \tag{5}$$

$$F_j = \begin{cases} 1 & if \ j = 0 \\ 0 & otherwise \end{cases} \tag{6}$$

$$E_j = \begin{cases} 1 & if \ j = k \\ 0 & otherwise \end{cases} \tag{7}$$

$$G_i = \begin{cases} 1 & if \ i = w_c \\ 0 & otherwise \end{cases} \tag{8}$$

The left hand side of equation (1) represents the average transitions from state (*i*, *j*). The first two terms in the bracket specify the transitions due to receiving demand, Specifically with switching worker from NC to CCR and not respectively. The last two terms in the bracket specify the transitions due to satisfying demand, Specifically with switching worker from CCR to NC and not respectively. Also the right hand side of equation (1) represents the average transitions into state (*i*, *j*). The first two terms denote the transitions due to satisfying demand. Specifically, with switching worker from CCR to NC and not respectively. The last two terms indicate the transitions due to receiving demand. Specifically, with switching worker from NC to CCR and not respectively.

A_{ij} and B_{ij} state whether the ratio of *j* (number of demands at CCR) to *i* (number of workers at CCR) is less than or equal to U and the ratio of *j* to *i* is greater than or equal to U, respectively. C_{ij} is minimum of i and j. D_i and G_i state whether the number of workers at CCR is equal to w and w_c, respectively. Besides, F_j and E_j state whether the number of demands at CCR is equal to zero and k, respectively. To find the steady-state probabilities for a specific value of thresholds, we can solve the corresponding system of linear equations, which contain the balance equations given in equation (1), and the normalizing constraint:

$$\sum_{i=w_c}^{w} \sum_{j=0}^{k} \pi_{ij} = 1 \tag{9}$$

3.3 NC constraint For the NC, we must have sufficiently many workers there (on average) so that the NC due to switching workers stays non-constraint and switching worker can't make this NC into a bottleneck resource. Therefore, we require that the time-average worker complement in the NC to be equal or exceeds some minimum specified threshold.

$$\overline{w}_{nc} \geq \overline{w}_0 \tag{10}$$

The time-average number of workers at NC is

$$\overline{w}_{nc} = w - \overline{w}_c \tag{11}$$

$$\overline{w}_c = \sum_{i=w_c}^{w} i\,\pi_{i\bullet} \tag{12}$$

$$\overline{w}_{nc} = w - \sum_{i=w_c}^{w} i\,\pi_{i\bullet} \tag{13}$$

where the "dot" (•) notation signifies summation over all values of the missing index.

3.4 Objective function and optimization problem Every system has at least one bottleneck which limits the system's ability to improve its goal.

The Theory of Constraints is based on the principle that the goal of any economic enterprise is to make money, now and in the future, and that the system's constraints determine its capacity to make money. If we increase CCR output, the output of system will be increased. Therefore, the main objective function conceptualized here is related to the maximization of CCR output.

$$output = \sum_{i=w_c}^{w} \sum_{j=0}^{k} \mu C_{ij} \, \pi_{ij} \tag{14}$$

s.t. (1) - (10)

3.5 Model with no cooperation

If there is no switching of workers (CCR and NC are separate non-interacting stations), then CCR can be regarded as the $M/M/w_c/k$ queueing system, which has a finite capacity of size k and w_c servers. Demands arrive to the CCR in a Poisson manner with rate λ demands/hour. Operations at CCR are done independently with exponential operation times in which mean operation time equal to μ^{-1}. steady-state probability of the system being in state *j* (number of demands at CCR) is denoted herein by π_j and written as follows [17]:

$$\pi_0 = \left[1 + \sum_{n=1}^{w_c-1} \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n + \sum_{n=w_c}^{k} \frac{1}{w_c!} \frac{1}{w_c^{\,n-w_c}} \left(\frac{\lambda}{\mu} \right)^n \right]^{-1} \tag{15}$$

$$\pi_j = \begin{cases} \left(\frac{\lambda}{\mu} \right)^j \frac{1}{j!} \pi_0 & if \; 0 < j \le w_c - 1 \\ \left(\frac{\lambda}{\mu} \right)^j \frac{1}{w_c!} \frac{1}{w_c^{\,j-w_c}} \pi_0 & if \; w_c \le j \le k \end{cases} \tag{16}$$

In addition, CCR output is written as follows:

$$\sum_{j=0}^{k} \mu C_{w_c j} \, \pi_j \tag{17}$$

4. HEURISTIC ALGORITHM

The following three conjectures are used in our heuristic to limit two thresholds that will be considered:

Conjecture 1. Increasing upper threshold decreases CCR output but also increases time-average NC manpower.

Conjecture 2. Increasing lower threshold decreases CCR output but also increases time-average NC manpower.

Conjecture 3. If for $L = L_1$ and $U = U_1$, the time-average worker is less than $\overline{w}_0$, then there exists no feasible solution for $L = L_1$ and $U = U_2$ or $L = L_2$ and $U = U_1$ where $L_2 < L_1$ and $U_2 < U_1$.

Conjecture 1 states that, a policy that pulls workers from the NC at a large threshold will result in decreased CCR output but also better time-averaged manpower in the NC. Conjecture 2 states that, allowing workers to stay shorter in the CCR before switching to the NC provides less output CCR and thus the NC will "get better" in terms of time-average NC manpower. Conjecture 3 state that, decreasing lower or upper threshold decreases time-average NC manpower.

Our proposed algorithm solves the problem as follows:

Step 1: Calculate the R_{ij} ratio as follows:

$$R_{ij} = \frac{j}{i}$$

where $i = w_c, w_c + 1, \ldots, w$ and $j = 0,1,2,\ldots,k$.

Step 2: Sort values in ascending order of R_{ij} with no repetitions and call them $R_1, R_2, \ldots, R_m$.

Step 3: Determine $CU = \{U_1, U_2, \ldots, U_n\} = \{R_t \,|\, t = 1,2,...,m\}$, $n \le m$ where CU is the set of candidate values for the upper threshold with $R_t \ge 1$.

Step 4: Determine $CL = \{L_1, L_2, \ldots, L_m\} = \{R_t \,|\, t = 1,2,...,m\}$, where CL is the set of candidate values for the lower threshold.

Step 5: Set $L = 1$ and $U = U_1 = 1$.

The heuristic starts with total cooperation that all w workers treat as workers in the CCR (meaning L=1 and U=1). In these state we have maximum output but time-average NC manpower maybe infeasible.

Step 6: Calculate steady-state probability of the system and time-average worker in NC. If the time-average worker is greater than or equal to $\overline{w}_0$ go to step 7 (current solution is a feasible solution) else set the next member of CU as U and L=U, repeat step 6.

Step 7: If L is equal to L_1 stop and the current values of thresholds provide best solution else set the pervious member of CL as L; calculate steady-state probability of the system and time-average worker in NC.

If the time-average worker is greater than or equal to $\overline{w}_0$ repeat step 7 else stop and the pervious values of thresholds provide best solution.

A flow chart of the heuristic method is included in Figure 2.

5. EXAMPLE

To illustrate the steps described in the previous section, consider the following example: $\lambda = 12$, $\mu = 4$, $\overline{w}_0 = 0.6$, $k = 5$, $w = 3$ and with two dedicated workers in the CCR. The algorithm solves the problem as follows:

Step 1: R_{ij} ratio is shown in Table 1.

Table 1. R_{ij} **ratio**

i \ j	0	1	2	3	4	5
2	0	$1/2$	1	$3/2$	2	$5/2$
3	0	$1/3$	$2/3$	1	$4/3$	$5/3$

Step 2: Sort values in ascending order of R_{ij} and call them $R_1, R_2, \ldots, R_m$.

$$\begin{array}{cccccccccc} 0 & 1/3 & 1/2 & 2/3 & 1 & 4/3 & 3/2 & 5/3 & 2 & 5/2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ R_1 & R_2 & R_3 & R_4 & R_5 & R_6 & R_7 & R_8 & R_9 & R_{10} \end{array}$$

Step 3: Determine the set of candidate values

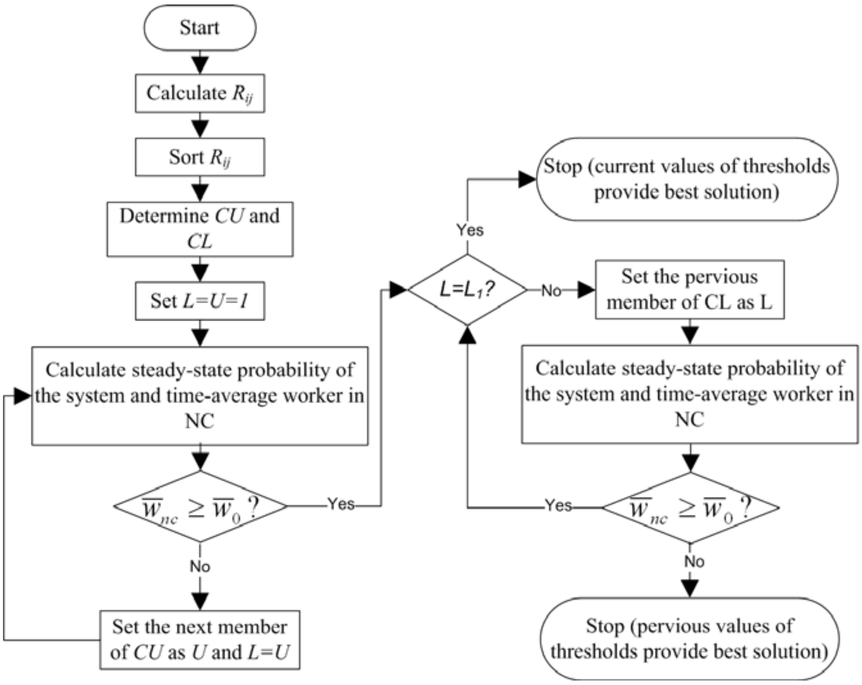
Start
Calculate R_{ij}
Sort R_{ij}
Determine CU and CL
Set $L=U=1$
Calculate steady-state probability of the system and time-average worker in NC
$\overline{w}_{nc} \geq \overline{w}_0$?
Yes
No
Set the next member of CU as U and $L=U$
$L=L_1$?
Yes
No
Stop (current values of thresholds provide best solution)
Set the pervious member of CL as L
Calculate steady-state probability of the system and time-average worker in NC
$\overline{w}_{nc} \geq \overline{w}_0$?
Yes
No
Stop (pervious values of thresholds provide best solution)

Figure 2. A flow chart of our heuristic method

for the upper threshold.

$$CU = \{U_1, U_2, U_3, U_4, U_5, U_6\}$$
$$CU = \left\{1, \tfrac{4}{3}, \tfrac{3}{2}, \tfrac{5}{3}, 2, \tfrac{5}{2}\right\}$$

Step 4: Determine the set of candidate values for the lower threshold.

$$CL = \{L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}\}$$
$$CL = \left\{0, \tfrac{1}{3}, \tfrac{1}{2}, \tfrac{2}{3}1, \tfrac{4}{3}, \tfrac{3}{2}, \tfrac{5}{3}, 2, \tfrac{5}{2}\right\}$$

Step 5: Set $L = 1$ and $U = U_1 = 1$.

Step 6: Calculate steady-state probability of the system and time-average worker in NC.

According to the section 3.2, we write balance equations for current values of thresholds as follows:

$$12\pi_{20} = 4\pi_{21}$$
$$16\pi_{21} = 12\pi_{20} + 8\pi_{22}$$
$$20\pi_{22} = 12\pi_{21} + 12\pi_{33}$$
$$24\pi_{33} = 12\pi_{22} + 12\pi_{34}$$
$$24\pi_{34} = 12\pi_{33} + 12\pi_{35}$$
$$12\pi_{35} = 12\pi_{34}$$

To find the steady-state probabilities for current values of thresholds, we solve the corresponding system of linear equations, which contain above equations and the normalizing constraint:

$$\pi_{20} + \pi_{21} + \pi_{22} + \pi_{33} + \pi_{34} + \pi_{35} = 1$$

Steady-state probabilities for current values of thresholds are:

$\pi_{20} = 0.0455$, $\pi_{21} = 0.1364$, $\pi_{22} = 0.2045$,

$\pi_{33} = 0.2045$, $\pi_{34} = 0.2045$ and $\pi_{35} = 0.2045$.

Thus, the time-average worker complement in NC is calculated as follows:

$$\overline{w}_{nc} = w - \sum_{i=w_c}^{w} i\,\pi_{i\bullet}$$

$$\overline{w}_{nc} = 3 - \sum_{i=2}^{3} i\,\pi_{i\bullet} = 0.3864$$

The time-average worker less than $\overline{w}_0 = 0.6$, therefore set the next member of CU as U ($U = U_2 = 4/3$) and $L = L_6 = 4/3$; repeat step 6. Summary of step 6 is shown in Table 2.

When $L = U = 5/3$, the time-average worker is greater than $\overline{w}_0$. Therefore, we go to step 7.

Step 7: Set $L = L_7 = 3/2$; calculate steady-state probability of the system and time-average worker in NC. The time-average worker is greater than $\overline{w}_0$, therefore repeat step 7. Summary of step 7 is shown in Table 3.

When $L = 1$ and $U = 5/3$, the time-average is less than $\overline{w}_0$. Therefore, The algorithm stops and the pervious values of thresholds ($L = 4/3$ and $U = 5/3$) provide best solution. It means whenever R_{ij} ratio is greater than or equal to $5/3$, the number of active CCR workers are increased from *i* to *i+1* (for $i = 2$; $j = 0,1,2,3,4,5$) and whenever R_{ij} ratio is less than or equal to $4/3$, the number of active CCR workers are decreased from *i* to $i-1$ (for $i = 3$; $j = 0,1,2,3,4,5$).

The optimum output is calculated as follows:

$$output = \sum_{i=w_c}^{w} \sum_{j=0}^{k} \mu C_{ij}\,\pi_{ij}$$
$$output = \sum_{i=2}^{3} \sum_{j=0}^{5} (4) C_{ij}\,\pi_{ij} =$$
$$(4)\left(0\pi_{20} + 1\pi_{21} + 2\pi_{22} + 2\pi_{23} + 2\pi_{24} + 3\pi_{34} + 3\pi_{35}\right)$$
$$output = 8.9094$$

According to the section 3.5, if there is no switching of workers, steady-state probability of the system is calculated as follows:

Table 2. Summary of step 6

L	U	steady-state probabilities	$\overline{w}_{nc}$
1	1	$\pi_{20} = 0.0455$, $\pi_{21} = 0.1364$, $\pi_{22} = 0.2045$, $\pi_{33} = 0.2045$, $\pi_{34} = 0.2045$, $\pi_{35} = 0.2045$	0.3864 (infeasible)
$4/3$	$4/3$	$\pi_{20} = 0.0348$, $\pi_{21} = 0.1043$, $\pi_{22} = 0.1565$, $\pi_{23} = 0.2348$, $\pi_{34} = 0.2348$, $\pi_{35} = 0.2348$	0.5304 (infeasible)
$3/2$	$3/2$	$\pi_{20} = 0.0348$, $\pi_{21} = 0.1043$, $\pi_{22} = 0.1565$, $\pi_{23} = 0.2348$, $\pi_{34} = 0.2348$, $\pi_{35} = 0.2348$	0.5304 (infeasible)
$5/3$	$5/3$	$\pi_{20} = 0.0282$, $\pi_{21} = 0.0845$, $\pi_{22} = 0.1268$, $\pi_{23} = 0.1901$, $\pi_{24} = 0.2852$, $\pi_{35} = 0.2852$	0.7148 (feasible)

Table 3. Summary of step 7

L	U	steady-state probabilities	$\overline{w}_{e}$
$3/2$	$5/3$	$\pi_{20} = 0.0318$, $\pi_{21} = 0.0954$, $\pi_{22} = 0.1431$, $\pi_{23} = 0.2146$, $\pi_{24} = 0.1288$, $\pi_{34} = 0.1288$, $\pi_{35} = 0.2576$	0.6137 (feasible)
$4/3$	$5/3$	$\pi_{20} = 0.0318$, $\pi_{21} = 0.0954$, $\pi_{22} = 0.1431$, $\pi_{23} = 0.2146$, $\pi_{24} = 0.1288$, $\pi_{34} = 0.1288$, $\pi_{35} = 0.2576$	0.6137 (best solution)
1	$5/3$	$\pi_{20} = 0.0371$, $\pi_{21} = 0.1112$, $\pi_{22} = 0.1668$, $\pi_{23} = 0.1317$, $\pi_{24} = 0.0790$, $\pi_{33} = 0.0790$, $\pi_{34} = 0.1580$, $\pi_{35} = 0.2371$	0.5259 (infeasible)

$$\pi_0 = \left[1 + \frac{1}{n!}\left(\frac{12}{4}\right)^n + \sum_{n=2}^{5} \frac{1}{2!}\frac{1}{2^{n-2}}\left(\frac{12}{4}\right)^n\right]^{-1}$$

$$\pi_0 = 0.0247$$

$$\pi_j = \begin{cases} \left(\frac{12}{4}\right)^j \frac{1}{j!}\pi_0 & if\ 0 < j \le 1 \\ \left(\frac{12}{4}\right)^j \frac{1}{2!}\frac{1}{2^{j-2}}\pi_0 & if\ 2 \le j \le 5 \end{cases}$$

$$\pi_1 = 0.0740$$

$$\pi_2 = 0.1109$$

$$\pi_3 = 0.1664$$

$$\pi_4 = 0.2496$$

$$\pi_5 = 0.3744$$

CCR output is calculated as follows:

$$output = \sum_{j=0}^{5} \min(2, j)(4)\ \pi_j$$

$$output =$$

$$(4)\begin{bmatrix} (0 \times 0.0247) + (1 \times 0.0740) + (2 \times 0.1109) \\ + (2 \times 0.1664) + (2 \times 0.2496) + (2 \times 0.3744) \end{bmatrix}$$

$$output = 7.5069$$

Our heuristic method identifies an output of 8.9094. The total output for this problem assuming no cooperation ($M / M / w_c / k$ queueing system) is 7.5069.

The above example shows that the heuristic method solution exceeds $M/M/w_c/k$ queueing system output by an additional 1.4025, which is a 19% increase over $M/M/w_c/k$ queueing system output.

6. COMPUTATIONAL RESULTS

For the example discussed in the previous section, consider the following sets of parameter values: $\lambda = 4,6,8,10\, and\, 12; k = 5,6,7\, and\, 8$. Table 4 contains the results of all combinations of λ and k values. The third column of the table contains the output obtained from heuristic method (O^{HM}). The fourth column contains O^{NoC}, total output assuming no cooperation ($M/M/w_c/k$ queueing system). The fifth column provides the percentage that output of heuristic method over the no-cooperation solution.

Several conclusions can be drawn from Table 4:

- Our heuristic method clearly performs better than $M/M/w_c/k$ queueing system.
- By increasing of λ and keeping other characteristics as constant, the output obtained from our heuristic method and $M/M/w_c/k$ queueing system will be increased.
- By increasing of k and keeping other characteristics as constant, the output obtained from our heuristic method and $M/M/w_c/k$ queueing system will be increased.

Table 4. Results of the example for variety of λ and k values

λ	k	O^{HM}	O^{NoC}	$\frac{O^{HM} - O^{NoC}}{O^{NoC}} \times 100$
4	5	3.9730	3.9149	1.48
4	6	3.9910	3.9579	0.84
4	7	3.9970	3.9791	0.45
4	8	3.9990	3.9896	0.24
6	5	5.8169	5.4893	5.97
6	6	5.9098	5.6400	4.78
6	7	5.9553	5.7416	3.72
6	8	5.9777	5.8123	2.85
8	5	7.3934	6.5455	12.95
8	6	7.6150	6.7692	12.49
8	7	7.7181	6.9333	11.32
8	8	7.8164	7.0588	10.73
10	5	8.3559	7.1635	16.65
10	6	8.6580	7.3824	17.28
10	7	8.7631	7.5347	16.30
10	8	8.9725	7.6443	17.38
12	5	8.9094	7.5069	18.68
12	6	9.0037	7.6843	17.17
12	7	9.0991	7.7949	16.73
12	8	9.1594	7.8656	16.45

7. CONCLUSION

This paper proposed a policy for managing the system constraint. A model was developed based on that policy and heuristic algorithm was suggested to solve this model. The proposed heuristic algorithm compared with $M/M/w_c/k$ queueing system. The computational results indicate that the proposed algorithm clearly performs better than $M/M/w_c/k$ queueing system.

8. REFERENCES

1. Goldratt, E. M. and Cox, J., "The Race", New York, North River, (1986).
2. Goldratt, E. M., "The Haystack Syndrome: Sifting Information from the Data Ocean?", New York, North River, (1990).
3. Goldratt, E. M., "Theory of Constraints: What Is This Thing Called the Theory of Constraints and How Should It Be Implemented?", New York, North River, (1990).
4. Goldratt, E. M., Fox, R. E., "The Race", New York, North River, (1986).
5. Dettmer, H. W., "Breaking the constraints to world-class performance" Milwaukee, ASQ Quality, (1998).
6. Goldratt, E. M., "It's Not Luck", New York, North River, (1994).
7. Goldratt, E. M., "The goal", New York, North River, (1984).
8. Buxey, G., "Production scheduling: Practice and theory", ***European Journal of Operational Research***, Vol. 39, (1989), 17–31.
9. Plenert, G., "Optimized theory of constraints when multiple constrained resources exist", ***European Journal of Operational Research***, Vol. 70, No. 1, (1993), 126–133.
10. Luebbe, R. and Finch, B., "Theory of constraints and linear programming: a comparison", ***International Journal of Production Research***, Voll. 30, No. 6, (1992), 1471–1478.
11. Lee, T. N. and Plenert, G., "Optimizing theory of constraints when new product alternatives exist", ***Production and Inventory Management Journal***, Vol. 34, No. 3, (1993), 51–57.
12. Fredenall, L. D. and Lea, B. R., "Improving the product mix heuristic in the theory of constraints", ***International Journal of Production Research***, Vol. 35, No. 6, (1997), 1535–1544.
13. Aryanezhad, M. B. and Komijan, A. R., "An improved algorithm for optimizing product mix under the theory of constraints", ***International Journal of Production Research***, Vol. 42, No. 20, (2004), 4221–4233.
14. Tsai, W. H., and Lai, C. W., and Chang, J. C., "An algorithm for optimizing joint products decision based on the Theory of Constraints", ***International Journal of Production Research***, Vol. 45, No. 15, (2007), 3421–3437.
15. Bhattacharya, A., and Vasant, P., and Sarkar, B., and Mukherjee, S.K., "A fully fuzzified, intelligent theory-of-constraints product-mix decision", ***International Journal of Production Research***, Vol. 46, No. 3, (2008), 789-815.
16. Wang, J. Q., and Sun, S. D., and Si, S. B., and Yang, H. A., "Theory of Constraints product mix optimization based on immune algorithm", ***International Journal of Production Research***, Vol. 47 No. 16, (2009), 4521-4543
17. Askin, R. G. and Standridge, C. R., "Modeling and Analysis of Manufacturing Systems". New York, Wiley, (1993).