**Cuckoo Search Optimization for Solving Product Mix Problem**

H Al-Khazrajī¹ S Khlīl² and Z Alabacy¹

¹Control and System Engineering Department, University of Technology, Baghdad, Iraq, E-mail: 60141@uotechnology.edu.iq
²Applied Mechanics Department, Technical College-Baghdad, Middle Technical University, Baghdad, Iraq

**Abstract.** The success of industrial companies is characterized by the efficient use of its resources. Most of decisions regarding to the efficient use of resources such as man, machine and material are subject to constraints. This paper considers a class of resource allocation problem named Product Mix Problem (PMP). The PMP is considered as an important production planning decision in the field of industrial engineering and management. The objective of this paper is to find the optimal quantities of different products that are required different amount of resources to maximize the return profit of a company where resources are limited. A Cuckoo Search Optimization (CSO) is proposed in this paper to as a methodology to search optimal solution. MATLAB Software is used to perform the proposed CSO method. Then, a comparative study between the CSO solution and existing solution in the literature is given. The simulations outcomes show the superiority of CSO over other methods.

**Keywords:** Product Mix Problem; Linear Programming; meta-heuristic algorithms; Cuckoo Search Optimization;

1. **Introduction**

The Product Mix Problem (PMP) is considered an important production planning decision in the field of industrial engineering and management [1]. When the demand for products exceeds the company's capacity, companies are required to select a limited number of each product which can increase the company's profitability [2]. Therefore, the PMP aims to find the optimal quantities of different products that are required different amounts of resources to maximize the return profit subject to different constraints [3]. The PMP can be formulated as a Linear Programming Problem (LPP) [3][4] and/or Integer Linear Programming Problem (ILPP) [5]. A heuristic method based on management philosophy named Theory of Constraints (TOC) was applied to solve PMP by Goldratt in 1987 [6]. The process of TOC to solve PMP is easy and assumed that the product with high profitability should be selected first in the case of a bottleneck situation [7]. Plenert [8] showed that TOC gives an insufficient optimal solution for PMP in comparison with Linear Programming (LP) when multiple constrained resources exist. Later, Fredendall and Lea [9] improved the traditional TOC and developed a revised TOC (RTOC) to find the optimal solution for PMP under situations where the traditional TOC failed. In the same way, Hsu and Chung [10] improve the TOC to solve PMP when multiple constrained resources existed.

Many meta-heuristic algorithms such as Tabu search [5], genetic algorithms [11], hybrid Tabu search - simulated annealing [12], simulated annealing [13], immune algorithm [14], and particle swarm optimization [15] have been developed for solving PMP. In this paper, a meta-heuristic named...
Cuckoo Search Optimization (CSO) is proposed to search for an optimal solution for PMP. Then, a comparative study between the CSO solution and the existing solution in the literature is given.

2. Mathematical Model

Consider a company produces a set of $N$ products (i.e. $n = 1, 2, \ldots, N$). Each product $n$ has an upper limit of quantities $D_n$ based on the average demand for that product. Each product has a raw material cost $c_n$ and selling price $p_n$. There is a limited resource $r_j$ ($j$ is the number of resources required for each product, i.e. machine). PMP is to find the optimal quantities of each product $q_n$ that maximize the profit of the company. The PMP modelled as linear programming which is given by:

$$\begin{align*}
\text{Max} & \quad \sum_{n=1}^{N} q_n (p_n - c_n) \\
\text{s.t.} & \quad \sum_{n=1}^{N} a_{nj} q_n \leq r_j, \quad j = 1, 2, \ldots, J \\
& \quad 0 < q_n \leq D_n, \quad n = 1, 2, \ldots, N
\end{align*}$$

where

- $N$ Number of product
- $N$ Index for products
- $q_n$ Optimal quantities of each product
- $p_n$ Selling price of the product
- $c_n$ Material costs of the product
- $J$ Number of resources
- $J$ Index for resources
- $a_{nj}$ Amount of resource $j$ required to produce product $n$
- $r_j$ Maximum amount of resource $j$ available
- $D_n$ Forecast market demand for product $n$

3. Cuckoo Search Optimization

This paper presents a CSO-based approach to solve PMP. Similar to other Bio-inspired methods, CSO is a meta-heuristic population-based method developed by Yang and Deb in 2009 [15]. CSO combines two search methods (Levy flights and brood parasitism of some cuckoo species). The CSO summaries as follows [16]: the algorithm starts by initialization a population of $n_{nest}$ host nest randomly as given:

$$x_i = x_{min} + \text{Rand} \times (x_{max} - x_{min}), \quad i = 1, 2, \ldots, n_{nest}$$

where

- $n_{nest}$ Number of the host nest (population)
- $x_i$ Individual solution
- $x_{min}$ Lower bound of the search space
- $x_{max}$ Upper bound of the search space

Next, the new solution is generated based on Levy flights. Levy flights considers as one way that is used by animals to search for food. When there is no food in the surrounding area, animals move similar to Levy flight style which can be defined as a combination of shorts and long steps in different directions. The Levy flights in the CSO algorithm represents the local random walk and it is given by:

$$x_i^{new} = x_i^{best} + \alpha \otimes L(\lambda) (x_i^{best} - x^g)$$

where

- $x_i^{new}$ New solution
- $x_i^{best}$ Best solution for host $i$
- $x^g$ Best solution found among all the population (global solution)
- $\alpha$ Step size (depends on the scale of the problem), $\alpha > 0$
- $\otimes$ Element-wise multiplications
- $L(\lambda)$ Levy distribution (i.e. Levy~$u = t^{-\lambda}$), $(1 < \lambda \leq 3)$
Figure 1. Flow chart of CSO
After that, a new solution is generated based on the brood parasitism of some cuckoo birds. These Cuckoos have an aggressive reproduction strategy by laying their eggs in other bride’s nests. If the host bird does not discover that this egg is not its own, they take care of cuckoo chicks. Otherwise, they will either throw it away or simply abandon the nest and build a new one. This concept is used in the CSO algorithm as follow: the algorithm randomly selected a nest, let say \( x_i \), a new solution is a generation depends on the probability that the nest host does not discover that this egg is not its own, it can be formulated as given:

\[
x_{i}^{\text{new}} = x_{i}^{\text{best}} + H \times \text{Rand} \times (x_j - x_k)
\]

\[
H = \begin{cases} 
1, & \text{if } \theta < P_a \\
0, & \text{otherwise}
\end{cases}
\]

where

- \( H \): Coefficient related to the probability that the host bird discover the egg in the nest
- \( x_i \): A solution selected randomly form the population
- \( x_k \): A solution selected randomly form the population
- \( \theta \): a uniformly distributed random number between 0 and 1
- \( P_a \): Probability that the host bird discover the egg in the nest

The steps of CSO algorithm are shown in Figure 1.

### 4. Computational Simulation

In this section, CSO is applied to solve three examples of PMP. The three examples are taken from [7] and [8]. The three examples are indifference based on the number of products (NoP), the number of resources (NoR) and the number of bottlenecks (NoB). To verify and to see the performance of the proposed CSO approach, the CSO is compared with other existing approaches. The software MATLAB is used to run the CSO algorithm that is described in Figure 1. The CSO algorithm parameters are listed in Table 1.

| Parameters       | Values |
|------------------|--------|
| Population Size  | 100    |
| Number of iterations | 300   |
| The probability \( p_a \) | 0.7    |

![Figure 2. Chart for first example](image-url)
Table 2. Total load computations and number of bottlenecks for first example

| Resources | Products | Load calculation (min) | Total load (min) | Capacity limit (min) | Bottlenecks |
|-----------|----------|------------------------|------------------|----------------------|-------------|
| A         | P        | 15                     | 1500             | 2000                 | ✓           |
|           | Q        | 10                     |                  |                      |             |
|           |          |                        |                  |                      |             |
| B         | P        | 15                     | 1500             | 3000                 | ✓           |
|           | Q        | 30                     |                  |                      |             |
|           |          |                        |                  |                      |             |
| C         | P        | 15                     | 1500             | 1750                 | ✓           |
|           | Q        | 5                      | 250              | 1250                 |             |
|           |          |                        |                  |                      |             |
| D         | P        | 10                     | 1000             | 1700                 | ×           |
|           | Q        | 5                      | 250              | 1250                 |             |

Demand 100
Selling price ($) 90
Material price ($) 45

The chart of the first example is shown in Figure 2. The details of the first example are summarized in Table 2. Based on Table 2, this example is two products - four resources - three bottlenecks problem.

The linear programming formulation for this example is given by:

\[
\text{Max } 45P + 60Q \tag{8}
\]

s.t.
\[
15P + 10Q \leq 1700 \tag{9}
\]
\[
15P + 30Q \leq 1700 \tag{10}
\]
\[
15P + 5Q \leq 1700 \tag{11}
\]
\[
10P + 5Q \leq 1700 \tag{12}
\]
\[
0 \leq P \leq 100 \tag{13}
\]
\[
0 \leq Q \leq 50 \tag{14}
\]

The chart of the second example is shown in Figure 3. The details of the second example are summarized in Table 3. Based on Table 3, this example is four products - four resources - three bottlenecks problem.

Figure 3. Chart for second example
Table 3. Total load computations and number of bottlenecks for third example

| Resources | Products | Load calculation (min) | Total load (min) | Capacity limit (min) | Bottlenecks |
|-----------|----------|------------------------|------------------|----------------------|-------------|
| A         | P Q Pₓ Qₓ | 15 10 15 10 1500 500 750 250 | 3000            | 2400                | ✓           |
| B         | P Q Pₓ Qₓ | 15 30 15 30 1500 1500 750 750 | 4500            | 2400                | ✓           |
| C         | P Q Pₓ Qₓ | 15 5 15 5 1500 250 750 125 | 2625            | 2400                | ✓           |
| D         | P Q Pₓ Qₓ | 10 5 10 5 1000 250 500 125 | 1875            | 2400                | ✗           |

Demand 100 50 50 25
Selling price ($) 90 100 81 90
Material price ($) 45 40 45 40

The linear programming formulation for this example is given by:

\[
\text{Max } 45P + 60Q + 36P_x + 50Q_x
\]

s.t.

15P + 10Q + 15P_x + 10Q_x \leq 2400
15P + 30Q + 15P_x + 30Q_x \leq 2400
15P + 5Q + 15P_x + 5Q_x \leq 2400
10P + 5Q + 10P_x + 5Q_x \leq 2400
0 \leq P \leq 100
0 \leq Q \leq 50
0 \leq P_x \leq 50
0 \leq Q_x \leq 25

The chart of the third example is shown in Figure 4. The details of the third example are summarized in Table 4. Based on Table 4, this example is four products - four resources - two bottlenecks problem.

Figure 4. Chart for third example
Table 4. Total load computations and number of bottlenecks for third example

| Resources | Products | Load calculation (min) | Total load(min) | Capacity limit(min) | Bottlenecks |
|-----------|----------|------------------------|-----------------|--------------------|-------------|
| A         | 15       | 10                     | 1500            | 250                | 3000        | 2400        | ✓            |
| B         | 5        | 10                     | 500             | 250                | 1500        | 2400        | x            |
| C         | 15       | 5                      | 1500            | 250                | 2625        | 2400        | ✓            |
| D         | 10       | 5                      | 1000            | 250                | 1875        | 2400        | x            |

Demand: 100
Selling price ($) = 90
Material price ($) = 45

Max $45P + 60Q + 36P_x + 50Q_x$ (24)

s.t.
$15P + 10Q + 15P_x + 10Q_x \leq 2400$ (25)
$5P + 10Q + 5P_x + 10Q_x \leq 2400$ (26)
$15P + 5Q + 15P_x + 5Q_x \leq 2400$ (27)
$10P + 5Q + 10P_x + 5Q_x \leq 2400$ (28)
$0 \leq P \leq 100$ (29)
$0 \leq Q \leq 50$ (30)
$0 \leq P_x \leq 50$ (31)
$0 \leq Q_x \leq 25$ (32)

The optimal solutions for the three examples that are obtained by CSO are presented in Table (5, 6 and 7). It can be notice from Table (5) for example 1 that the best combination of product P and Q is 99 units of A and 7 units of B which is satisfy the capacity limit (all machines load less than 1770 min) and return a profit that equals to 4874$. In the same way, it can be seen from Table (6) for example 2 that the best combination of product P, Q, P_x and Q_x is 100 units of A, 5 units of B, 50 units of P_x and 0 units of Q_x which satisfies the capacity limit (all machines load less or equal than 2400 min) and return a profit that equals to 6600$.

Table 5. Optimal solution for example 1

| Number of products (units) | Total load(min) | Capacity limit(min) | Objective function ($) |
|---------------------------|-----------------|---------------------|------------------------|
| P                         | Q               |                     |                        |
| Machines                  |                 |                     |                        |
| A                         | 1485            | 70                  | 1555                   | 1700        |
| B                         | 1485            | 210                 | 1695                   | 1700        |
| C                         | 1485            | 35                  | 1520                   | 1700        |
| D                         | 990             | 35                  | 1025                   | 1700        |

Table 6. Optimal solution for example 2

| Number of products | Total load(min) | Capacity limit(min) | Objective function ($) |
|-------------------|-----------------|---------------------|------------------------|
| P                 | Q               | P_x                 | Q_x                    |              |
| Machines          |                 |                     |                        |              |
| A                 | 1500            | 50                  | 750                    | 0            | 2300       | 2400       |              | 6600        |
| B                 | 1500            | 50                  | 750                    | 0            | 2400       | 2400       |              |              |
| C                 | 1500            | 25                  | 750                    | 0            | 2275       | 2400       |              |              |
| D                 | 1000            | 25                  | 500                    | 0            | 1525       | 2400       |              |              |
Finally, it can be seen from Table (7) for example3 that the best combination of product P, Q, \( P_x \) and \( Q_x \) is 98 units of A, 50 units of B, 12 units of \( P_x \) and 25 units of \( Q_x \) which satisfies the capacity limit (all machines load less or equal than 2400 min) and return a profit that equals to 9092$.

| Number of products | Total load(min) | Capacity limit(min) | Objective function ($) |
|--------------------|-----------------|---------------------|-------------------------|
| \( P \)             | 1470            | 2400               | 2400                    |
| \( Q \)             | 500             | 1300               | 2400                    |
| \( P_x \)           | 180             | 2025               | 9092                    |
| \( Q_x \)           | 250             | 1475               |                         |

The comparison results between the CSO approach and other approaches such as integer linear programming (ILP), the theory of constraints (TOC), tabu search (TS), genetic algorithms (GA) and simulated annealing (SA) are presented in Table 8. The results of other approaches (ILP, TOC, TS, GA and SA) have been taken from [13]. In the case of the first example, ILP, TS, GA and SA methods in comparison with the CSO method show the same objective function (i.e. has the same performance). However, CSO outperforms TOC. For the second example, ILP, TOC and TS methods in comparison with the CSO method show the same objective function (i.e. has the same performance). However, CSO outperforms GA and SA. In the last case, ILP and TOC have the best objective function. However, the result of the CSO method is superior to the result of TS, GA and SA.

| Example  | NoP | NoR | NoB | ILP  | TOC  | TS   | GA   | SA   | CSO  |
|----------|-----|-----|-----|------|------|------|------|------|------|
| Example1 | 2   | 4   | 3   | 4875 | 4860 | 4875 | 4875 | 4875 | 4875 |
| Example2 | 4   | 4   | 3   | 6600 | 6600 | 6600 | 6568 | 6529 | 6600 |
| Example3 | 4   | 4   | 2   | 9110 | 9110 | 8955 | 9002 | 9074 | 9092 |

5. Conclusion

Product Mix Problem (PMP) has a direct influence on the manufacturing company’s financial performance. PMP is a well-known constrained linear-integer optimization problem. The problem seeks to optimize the use of resources (i.e. machine) with in order to increase the profit of the company. The size of the problem increases based on the number of products, the number of resources and the number of bottlenecks. Several approaches have been utilized to solve the PMP problem including integer linear programming (ILP), theory of constraints (TOC), tabu search (TS), genetic algorithms (GA) and simulated annealing (SA). In this research, a Cuckoo Search Optimization (CSO) is used to solve PMP. CSO is a meta-heuristic optimization techniques developed to solve global optimization problems. It has numerous applications in the computational intelligence problems. The results show that CSO approach is good as the LIP approach and superior to TS, GA and SA approaches in solving the PMP.

Reference

[1] Nazari-Shirkouhi S, Eivazy H, Ghodsi R, Rezaie K and Atashpaz-Gargari E 2010. Expert Systems with Applications, 37(12), pp.7615-7626.
[2] Rezaie K, Nazari-Shirkouhi S and Ghodsi R 2010. Australian Journal of Basic and Applied Sciences, 4(12), pp.6483-6491.
[3] Woubante G.W 2017. Open Science Journal, 2(2).
[4] Hodges, S.D. and Moore, P.G., 1970. The product-mix problem under stochastic seasonal demand. Management Science, 17(2), pp.B-107.
[5] Onwubolu G C 2001. International Journal of Production Research, 39(10), pp.2065-2076.
[6] Mehdizadeh E and Jalili S 2019. Journal of Optimization in Industrial Engineering, 12(1), pp.167-172.
[7] Luebbe R and Finch B 1992. International Journal of Production Research, 30(6), pp.1471-1478.
[8] Plenert G 1993. European Journal of Operational Research, 70(1), pp.126-133.
[9] Fredendall L D and Lea B R 1997. International journal of production research, 35(6), pp.1535-1544.
[10] Hsu T C and Chung S H 1998. Production Planning & Control, 9(1), pp.36-46.
[11] Onwubolu G C and Mutingi M 2001. International Journal of Production Research, 39(9), pp.1897-1910.
[12] Mishra N, Tiwari M K, Shankar R and Chan F T 2005. Expert systems with applications, 29(2), pp.446-454.
[13] Chaharsooghi S K and Jafari N 2007. Scientia Iranica, 14(3), pp.230-235.
[14] Wang J Q, Sun S D, Si S B and Yang H A 2009. International Journal of Production Research, 47(16), pp.4521-4543.
[15] Yang X S and Deb S 2009. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210-214). IEEE.
[16] Yang X S and Deb S 2014. Neural Computing and Applications, 24(1), pp.169-174.