

# Optimization of Bivariate Cost Function arising in Economic Order Quantity problem using Genetic Algorithm and Bacterial Foraging Algorithm

## 1. Introduction

### 1.1] Overview of the Economic Order Quantity (EOQ)

#### Model:

The most common inventory situation faced by manufacturers, retailers, and wholesalers is that stock levels are depleted over time and then are replenished by the arrival of a batch of new units. A simple model representing this situation is the following economic order quantity model or, for short, the EOQ model. (It sometimes is also referred to as the economic lot-size model.) Units of the product under consideration are assumed to be withdrawn from inventory continuously at a known constant rate, denoted by  $a$ ; that is, the demand is  $a$  units per unit time. It is further assumed that inventory is replenished when needed by ordering (through either purchasing or producing) a batch of fixed size ( $Q$  units), where all  $Q$  units arrive simultaneously at the desired time. For the basic EOQ model to be presented first, the only costs to be considered are  $K$  setup cost for ordering one batch,  $c$  unit cost for producing or purchasing each unit,  $h$  holding cost per unit per unit of time held in inventory. The objective is to determine when and by how much to replenish inventory so as to minimize the sum of these costs per unit time. We assume continuous review, so that inventory can be replenished whenever the inventory level drops sufficiently low.

One of the banes of any inventory manager is the occurrence of an inventory shortage (sometimes referred to as a stock out)—demand that cannot be met currently because the inventory is depleted. This causes a variety of headaches, including dealing with unhappy customers and having extra record keeping to arrange for filling the demand later (backorders) when the inventory can be replenished. By assuming that planned shortages are not allowed, the basic EOQ model presented above satisfies the common desire of managers to avoid shortages as much as possible.

(Nevertheless, unplanned shortages can still occur if the demand rate and deliveries do not stay on schedule.)

However, there are situations where permitting limited planned shortages makes sense from a managerial perspective. The most important requirement is that the customers generally are able and willing to accept a reasonable delay in filling their orders if need be. If so, the costs of incurring shortages described should not be exorbitant. If the cost of holding inventory is high relative to these shortage costs, then lowering the average inventory level by permitting occasional brief shortages may be a sound business decision. The EOQ model with planned shortages addresses this kind of situation. Planned shortages now are allowed. When a shortage occurs, the affected customers will wait for the product to become available again. Their backorders are filled immediately when the order quantity arrives to replenish inventory. Under these assumptions, the pattern of inventory levels over time has the appearance shown in **Fig. 1.1**. The inventory levels extend down to negative values that reflect the number of units of the product that are backordered. Let  $p$  shortage cost per unit short per unit of time short,  $S$  inventory level just after a batch of  $Q$  units is added to inventory,  $Q - S$  shortage in inventory just before a batch of  $Q$  units is added. The total cost per unit time now is obtained from the following components. Production or ordering cost per cycle  $K + cQ$ . During each cycle, the inventory level is positive for a time  $S/Q$ . The average inventory level during this time is  $S/2$  units, and the corresponding cost is  $hS/2$  per unit time. Hence,

$$\text{Holding cost per cycle} = \frac{hS^2}{2a}$$

Similarly, shortages occur for a time  $\frac{Q-S}{a}$ . The average amount of shortages during this time is  $\frac{Q-S}{2}$  units, and the corresponding cost is  $\frac{p(Q-S)}{2}$  per unit time. Hence,

$$\text{Shortage cost per cycle} = \frac{p(Q-S)}{2} * \frac{Q-S}{a} = \frac{p(Q-S)^2}{2a}$$

Therefore,

$$\text{Total cost per cycle} = K + cQ + \frac{hS^2}{2a} + \frac{p(Q-S)^2}{2a}$$

And the total cost per unit time is

$$T = \frac{K + cQ + \frac{hS^2}{2a} + \frac{p(Q-S)^2}{2a}}{\frac{Q}{a}}$$

$$T = \frac{aK}{Q} + ac + \frac{hS^2}{2Q} + \frac{p(Q-S)^2}{2Q}$$

In this model, there are two decision variables ( $Q$  and  $S$ ), so the optimal values are found by setting the partial derivatives  $\partial T/\partial Q$  and  $\partial T/\partial S$  equal to zero. Thus,

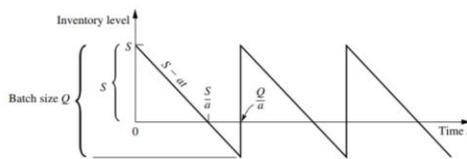
Solving these equations simultaneously leads to  $S$

$$S^* = \sqrt{\frac{2aK}{h}} \sqrt{\frac{p}{p+h}}$$

$$Q^* = \sqrt{\frac{2aK}{h}} \sqrt{\frac{p+h}{p}}$$

The maximum shortage is

$$Q^* - S^* = \sqrt{\frac{2aK}{p}} \sqrt{\frac{h}{p+h}}$$



**Fig.1.1** – Inventory level as a function of time for the EOQ model with planned shortages

**1.2] Brief Description of the Genetic Algorithm**

Genetic algorithm solves smooth or non-smooth optimization problems with any types of constraints, including integer constraints. It is a stochastic, population-based algorithm that searches randomly by mutation and crossover among population members.

The following outline summarizes how the genetic algorithm works:

1. The algorithm begins by creating a random initial population.
2. The algorithm then creates a sequence of new populations. At each step, the algorithm uses the individuals in the current generation to create the next population. To create the new population, the algorithm performs the following steps:
  - a. Scores each member of the current population by computing its fitness value. These values are called the raw fitness scores.
  - b. Scales the raw fitness scores to convert them into a more usable range of values. These scaled values are called expectation values.
  - c. Selects members, called parents, based on their expectation.
  - d. Some of the individuals in the current population that have lower fitness are chosen as *elite*. These elite individuals are passed to the next population.
  - e. Produces children from the parents. Children are produced either by making random changes to a single parent—*mutation*—or by combining the vector entries of a pair of parents—*crossover*.
  - f. Replaces the current population with the children to form the next generation.
3. The algorithm stops when one of the stopping criteria is met.

The genetic algorithm may use one of the following conditions to determine when to stop:

- **Generations** — the algorithm stops when the number of generations reaches the value specified.
- **Time limit** — the algorithm stops after running for an amount of time in seconds equal to the value specified.
- **Fitness limit** — the algorithm stops when the value of the fitness function for the best point in the current population is less than or equal to some Fitness limit.

**1.3] Brief Description of the Bacterial Foraging Algorithm**

Application of group foraging strategy of a swarm of E.coli bacteria in multi-optimal function optimization is the key idea of this new algorithm. The process, in which a bacterium

moves by taking small steps while searching for nutrients, is called chemotaxis. The key idea of BFOA is mimicking chemotactic movement of virtual bacteria in the problem search space.

$p$  : Dimension of the search space,  
 $S$  : Total number of bacteria in the population,  
 $N_c$  : The number of chemotactic steps,  
 $N_s$  : The swimming length.  
 $N_{re}$  : The number of reproduction steps,  
 $N_{de}$  : The number of elimination-dispersal events,  
 $P_{ed}$  : Elimination-dispersal probability,  
 $C(i)$  : The size of the step taken in the random direction specified by the tumble.

Foraging theory is based on the assumption that animals search for and obtain nutrients in a way that maximizes their energy intake  $E$  per unit time  $T$  spent foraging. Hence, they try to maximize a function like  $E/T$  (or they maximize their long-term average rate of energy intake). Maximization of such a function provides nutrient sources to survive and additional time for other important activities (e.g., fighting, fleeing, mating, reproducing, sleeping, or shelter building). Bacterial Foraging optimization theory is explained by following steps

- Chemotaxis
- Swarming
- Reproduction and
- Elimination-Dispersal

#### 1. Chemotaxis

This process simulates the movement of an E.coli cell through swimming and tumbling via flagella. Biologically an E.coli bacterium can move in two different ways. It can swim for a period of time in the same direction or it may tumble and alternate between these two modes of operation for the entire lifetime. Suppose  $\theta^i(j, k, l)$  represents  $i^{th}$  bacterium at  $j^{th}$  chemotactic,  $k^{th}$  reproductive and  $l^{th}$  elimination-dispersal step. Let  $C(i)$  be the size of the step taken in the random direction specified by

the tumble (run length unit). Then in computational chemotaxis the movement of the bacterium may be represented by

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta}{\sqrt{\Delta \cdot \Delta}} \quad (1.6)$$

Where  $\Delta$  indicates a vector in the random direction whose elements lie in  $[-1, 1]$ .

#### 2. Swarming

A group of E.coli cells arrange themselves in a traveling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemo-effector. The cells when stimulated by a high level of succinate, release an attractant aspartate, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density.

#### 3. Reproduction

The least healthy bacteria eventually die when each of the healthier bacteria (which yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.

#### 4. Elimination and Dispersal

Gradual or sudden changes in the local environment where a bacterium population lives may occur due to various reasons. Events can occur such that all the bacteria in a region are killed or a group is dispersed into a new part of the environment. For example, a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients. Events can take place in such a fashion that all the bacteria in a region are killed or a group is dispersed into a new location. To simulate this phenomenon in BFOA some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space. Elimination and dispersal events have the effect of

possibly destroying chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place the bacteria near good food sources. From a broad perspective, elimination and dispersal are parts of the population-level long-distance motile behavior.

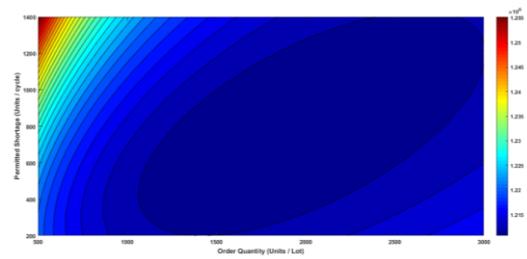


Fig.2.2. Contour Plot of the Bivariate Cost Function

**2. Problem Definition and Solution**

**2.1] Specifying Parameter Values**

In the inventory model optimization problem, the order quantity or lot size  $Q$  and the maximum permitted shortage  $S$  are treated as independent variables while all other quantities appearing in the cost function are treated as parameters. The cost function is hence a bivariate objective function and practical intuition tells us that it should have a minimum (optimal) value for some values of  $Q$  and  $S$  with arbitrarily specified values of all other parameters.

For this exercise, we have assumed the following values for the various cost function parameters.

- $D = 24000 \text{ unit/year}$
- $c = 50 \text{ Rs/unit}$
- $K = 400 \text{ Rs/setup}$
- $h = 10 \text{ Rs/(unit * year)}$
- $p = 15 \text{ Rs/(unit * year)}$

**2.2] Graphical Visualization of Cost Function**

Figures 2.1 and 2.2 show the cost function plotted versus the two independent variables  $Q$  and  $S$ . The surface and contour plots clearly indicate the presence of a global minimum of the cost function.

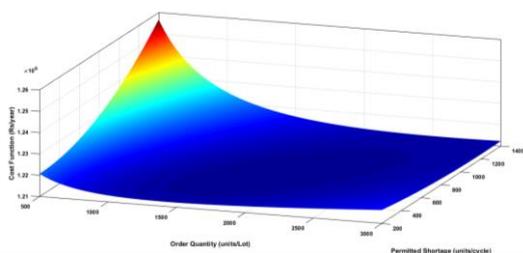


Fig.2.1. Surface Plot of the Bivariate Cost Function

**3. Results and Discussion**

**3.1] Genetic Algorithm code output**

The following table lists the results of the Genetic algorithm code section. Multiple trials were run to obtain an optimal solution since the selection process of the GA involves the use of random number generators and repeatability is not achievable as it is.

Trial	Q	
1	1788.9	
2	1791.1	
3	1818.9	
4	1788.9	
5	1801.6	

**3.2] BFO Algorithm code output**

The optimal solution obtained from the Bacterial Foraging Algorithm is

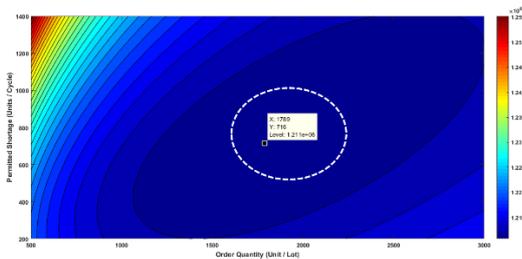
- ❖  $Q_{opt} = 1800 \text{ units/Lot}$
- ❖  $S_{opt} = 720 \text{ units/Cycle}$
- ❖  $C(Q_{opt}, S_{opt}) = 1.211e + 06 \text{ Rs/year}$

**3.3] Comparison of GA and BFOA results with Analytical Solution**

Equations (1.3) and (1.4) provide means to determine the exact values of  $Q$  and  $S$  which would yield the minimum value of the cost function. As seen from the tabulation that follows, the results obtained using GA and BFAO are in good

agreement with the exact analytical solution. On a general note, the GA is better in terms of time economy and accuracy as compared to the BFOA for this case. However this observation may not be general and the performance is evidently a strong function of the nature of the problem, parameter values and variable bounds.

The analytical results are marked in the contour plot of the cost function and shown in the figure below. It can be clearly seen that these results lie on the lowest level contour.



**Fig.2.2.** Contour Plot of cost function with optimal solution marked

### 3.4] Conclusions

An Economic Order Quantity model with planned shortages was considered wherein the objective was to determine the optimal lot size and maximum shortage values so as to minimize the Non-linear Bivariate cost function for particular values of the various function parameters. Analytical solutions to this problem are available and these solutions were compared with the numerical results obtained by using Genetic algorithm and Bacterial Foraging optimization algorithm. The result were in exceptional agreement with each other indicating the potency of GA and BFOA to solve complex multivariable optimization problems involving non-linearity.

## 4. References

1. <https://in.mathworks.com/matlabcentral/fileexchange/53835-bacterial-foraging-optimization>
2. Saraswati, D., Sari, D.K. and Johan, V., 2017, November. Using genetic algorithm to determine the optimal order quantities for multi-item multi-period under warehouse capacity constraints in kitchenware manufacturing. In *IOP Conference Series: Materials Science and Engineering* (Vol. 273, No. 1, p. 012020). IOP Publishing.
3. Eroglu, A. and Ozdemir, G., 2007. An economic order quantity model with defective items and shortages. *International journal of production economics*, 106(2), pp.544-549.
4. Esperet, E. and Piolat, A., 1991. Production: planning and control. In *Advances in Psychology* (Vol. 79, pp. 317-331). North-Holland.
5. Du, K.L. and Swamy, M.N.S., 2016. Bacterial Foraging Algorithm. In *Search and Optimization by Metaheuristics* (pp. 217-225). Birkhäuser, Cham.
6. Passino, K.M., 2010. Bacterial foraging optimization. *International Journal of Swarm Intelligence Research (IJSIR)*, 1(1), pp.1-16.