INVENTORY OPTIMISATION IN SERIAL SUPPLY CHAIN USING ALGORITHMS

A Project Report

Submitted by

ABHIRAM R KRISHNA
AUGIN M MENACHERIEL
MICHAEL JOSE
REUBEN MATHEW THOMAS

in partial fulfillment for the award of degree of

BACHELOR OF TECHNOLOGY

IN

MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM, KERALA, INDIA 686666

Affiliated to

APJ Abdul Kalam Technological University
Kerala, India
April 2020

DEPARTMENT OF MECHANICAL ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM 686 666
KERALA, INDIA.

CERTIFICATE

This is to certify that the report entitled

" **INVENTORY OPTIMISATION IN SERIAL SUPPLY CHAIN USING ALGORITHMS** " *submitted by Mr.* **ABHIRAM R KRISHNA,** *Mr.* **AUGIN M MENACHERIEL,** *Mr.* **MICHAEL JOSE,** *Mr.* **REUBEN MATHEW THOMAS** *to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Mechanical Engineering is a bonafide record of the project work carried out by them under my guidance and supervision.*

Dr. Kurian John	**Prof. Biju B**	**Dr.Shajan Kuriakose**
Project Guide	**Project Supervisor**	**Head of the Department**

Place:
Date: 06.05.2020 ***(Office Seal)***

ACKNOWLEDGEMENT

It is a great pleasure to acknowledge all those who have assisted and supported us for successfully completing our project.

First of all, we thank God Almighty for his blessings as it is only through his grace that we were able to complete our project successfully.

We would like to take this opportunity to extend our sincere thanks to our project guide Prof Kurian John, Assistant Professor, Department of Mechanical Engineering for his constant support and immense contribution for the success of our project.

We are also grateful to Dr Shajan Kuriakose, Head of Mechanical Engineering Department and our project co-coordinator Prof Biju B for the valuable guidance as well as timely advice which helped us a lot during the project development.

We are deeply indebted to Dr. Mathew K, Principal, Mar Athanasius College of Engineering for his encouragement and support.

We whole - heartedly thank all our classmates, for their valuable suggestions and for the spirit of healthy competition that existed between us.

ABSTRACT

Keywords : Inventory Management, Base stock policy, Genetic Algorithm, Crow Search Algorithm, Adaptive gbest - Gravitational Search Algorithm.

Important aspects of Supply chain management is inventory management because the cost of inventories in a supply chain accounts for about 30 percentage of the value of the product. The main focus of this work is to study the performance of a single-product serial supply chain operating with a base-stock policy and to optimize the inventory levels in the supply chain so as to minimize the total supply chain cost, comprising holding and shortage costs at all the installations in the supply chain In this project we study the performance of a single-product serial supply chain and decision making with the aid of Metaheuristic algorithms like Genetic Algorithm, Crow Search Algorithm and Adaptive gbest - Gravitational Search Algorithm , to compare the results obtained from these three algorithms and to conduct proper assessments.

CONTENT

CONTENTS	Page No..

LIST OF TABLES

LIST OF FIGURES

ABBRIEVIATIONS

INTRODUCTION

A supply chain is dynamic and involves the constant flow of information, product, and funds between different stages. A supply chain consists of all parties involved, directly or indirectly, in fulfilling a customer request. The supply chain includes not only the manufacturer and suppliers, but also transporters, warehouses, retailers, and even customers themselves. Within each organization, such as a manufacturer, the supply chain includes all functions involved in receiving and filling a customer request. These functions include, but are not limited to, new product development, marketing, operations, distribution, finance, and customer service. The objective of every supply chain should be to maximize the overall value generated. The value (also known as supply chain surplus) a supply chain generates is the difference between what the value of the final product is to the customer and the costs the supply chain incurs in filling the customer's request.

A typical supply chain may involve a variety of stages, including the following:

- Customers
- Retailers
- Wholesalers/distributors
- Manufacturers
- Component/raw material suppliers

TYPES OF SUPPLY CHAIN

The key goals for supply chain management should be to achieve efficient fulfillment of demand, drive outstanding customer value, enhance organizational responsiveness, build network resiliency, and facilitate financial success. In order to fulfill that, the supply chain is characterized into different models such as;

- The continuous flow models
- The fast chain models
- The efficient chain models
- The custom configured model
- The agile model
- The flexible model

The continuous flow model for supply offers stability in high demand situations that vary very little. Manufacturers that produce the same goods repeatedly with very little fluctuation can benefit from the continuous flow model. It is ideal for commodity manufacturing and is one of the most traditional supply chain models.

The fast chain model is ideal for manufacturers that manufacture products that are trendy with short life cycles. It works well with a business that must change their products frequently and that needs to get them out fast before the trend ends. It is a flexible model. The efficient chain model is a model that is best for businesses that are in very competitive markets and where end to end efficiency is the premium goal.

The custom configured models focus on providing custom configurations especially during assembly and production. It is a combination of the agile model and the continuous flow model, a hybrid of sorts.

The agile model is primarily a method of supply chain management that is ideal for businesses that deal in specialty order items. It is a model that focuses on the ability of the supply chain to amp up in some cases but also be solid when there is not much movement happening.

The flexible model gives businesses the freedom to meet high demand peaks and manage long periods of low volume movement. It can be switched on and off easily.

SUPPLY CHAIN SIMULATION

A supply chain simulation shows the behavior of a logistics network over time. The logical rules of a supply chain are represented in a model and then executed over time, making the simulation dynamic. For example, production is started when orders deplete inventory below a threshold. Such rules can be combined, and their relationships investigated as well as tested against disruptive events, like strikes and natural disasters.

Supply chain simulation is different to the analytical methods that have gone before. It provides dynamic details and opportunities for greater insight by expanding the design, analysis, and optimization toolset for supply chain managers. Furthermore, supply chain simulation is becoming a popular choice as the systems under study become increasingly complex.

Dynamic simulation models differ from analytical models in several important ways.

An analytical model of a supply chain uses linear equations to describe operations. The benefit of this is that solutions, once found, are optimal. This is also the downside: describing a supply chain in such a way requires simplification and, consequently, the difficulty of modeling increases with complexity. Furthermore, if a solution cannot be found, an analytical model cannot be used. It should also be noted that analytical models deal only with the values they are specified to capture and are of limited use outside their scope.

Dynamic simulation models capture the rules of operation and enable you to reflect all dimensions of your supply chain. The output of a dynamic simulation model shows system behavior over time and any descriptive statistics of the workings of your supply chain can be collected. Simulation models cope well with complexity and can consider real-world randomness. If they have downside, it is that they must be verified to make sure they function as expected.

HEURISTICS AND METAHEURISTICS

Heuristics are a problem-solving method that uses shortcuts to produce well-enough solutions given a limited time frame or deadline. Heuristics are a flexibility technique for quick decisions, particularly when working with complex data. Decisions made using a heuristic approach may not necessarily be optimal. The modeling of logistics systems is performed to seek the best possible system configuration to minimize costs or maximize operational performance, in order to meet or exceed customer expectations. Classically, analytic system analysis of this type has been performed using optimization, simulation, or heuristics. However, in the past two decades, a newer class of techniques, metaheuristics, has emerged as a capable method for quickly providing near-optimal solutions for problems that exact optimization cannot solve. This article outlines recent advances in metaheuristics development, and considers the ability of these advanced techniques to resolve various logistics and supply chain problem types. Specifically, the article discusses the ant colony optimization, genetic algorithm, simulated annealing, and tabu search metaheuristics. The capabilities of these metaheuristic techniques to examine supply chain risk and disruptions, intermodal operations, customer service trade-offs, backhaul strategies, and simultaneous facility location and vehicle route problems are proposed. The article concludes by describing how faculty can bring these techniques into the classroom to ensure their students enter the logistics and supply chain field with a current and relevant understanding of the state of the art in supply chain design techniques.

ORDER POLICIES

Order policies specify how one should calculate the amount of stock that needs to order. We can configure these policies in the settings for forecasting. In the simplest case everything has a universal replenishment policy that is used by all of the products that we sell.

Reordering or replenishment process needs to define review period for reordering, and an ordering quantity. Then it needs the inventory parameters to determine whether an order for replenishment should be placed at the time of review or not. Based on how the review period and order quantities are defined, there are a few options to drive the reordering.

Continuous Review and Periodic Review refer to such a frequency of review to determine when orders must be placed for replenishment.

In the continuous review process, the inventory levels are continuously reviewed, and as soon as the stocks fall below a pre-determined level, replenishment order is placed.

Under periodic review, the inventory levels are reviewed at a set frequency. At the time of review, if the stock levels are below the pre-determined level, then an order for replenishment is placed, otherwise it is ignored till the next cycle.
Order quantity and order up to level refer to the process that is used to determine how much is ordered when a replenishment order is placed.

In the first process, the "order quantity" is fixed. If the review determines that an order should be placed, then the order for a pre-defined quantity for that item-location combination is placed for replenishment. The order quantity for all replenishment orders is fixed in this method, though order day may vary or may be fixed depending on the review method.

The second process defines a pre-determined "order up-to level" instead. The actual order quantity is determined as the difference between the on-hand stock on the review day, and the pre-determined "order up-to level". The order quantity in this process will differ from one order to another depending on the on-hand quantity on the day of the review.

Between these two sets of parameters, four basic reordering process options become available.

APPLICATIONS

A supply chain is comprised of all the businesses and individual contributors involved in creating a product, from raw materials to finished merchandise. ... Examples of supply chain activities include farming, refining, design, manufacturing, packaging, and transportation.

AMAZON

Amazon is a US electronic commerce and cloud computing company. Their headquarters are based in Seattle, Washington and they are the largest internet-based retailer in the United States. Amazon was one of the first companies that started selling book online. Currently their range of products doesn’t stop there; they also sell music, videogames, shoes, clothing, luggage and many other accessories. Amazon offers about everything you can think of and their variety in offers and products along with their customer driven shopping and recommendations is a hit with customers. One of the reasons why Amazon can have such a wide spectrum of products is the fact that they are not limited by physical spaces, since they don’t have actual stores. Their supply chain goes from the lowest levels of inventory, through the logistics of the order itself all the way up to an outstanding distribution chain of their products in an international scale. Amazon can currently ship close to 10 million different products. This diversity gives it an edge against competitors and makes it a perfect example of what efficient supply chain management can accomplish.

THE COCA-COLA COMPANY

Main makers, marketers and distributors of drink concentrates and non-alcoholic syrups. The main office is located in Atlanta, GA but their products are distributed to virtually every country in the world. Their preparation, distribution and transportation logistics are in line with a segmentation strategy for their customers when it comes to the size and presentation of their products. Aside from having an extremely successful supply chain, Coca-Cola participates in sponsorships, partnerships, and alliances; thus creating a great management and marketing of their products.

COLGATE

The main toothpaste brand made by Colgate-Palmolive, dedicated to producing, distributing and selling oral hygiene and home cleaning products since the last part of the 19th century. Colgate keeps present all aspects of product diversity, effectiveness, optimization and customer support and it uses an effective distribution channel that encompasses all aspects of care and maintenance**.** Their products are sold in many venues such as pharmacies, supermarkets, convenience stores and small wholesalers, thus creating an excellent impact within their distribution channels and management.

In this modern era, supply chain management has become a vital element for industries and other manufacturing sectors. Here we are formulating an algorithm to obtain the maximum optimization so that we can run it at the lowest cost possible. Our work is solely concentrated on this objective.

LITERATURE REVIEW

1) The compact genetic algorithm - George R Harik

His paper clearly illustrates the mapping of the simple GA's parameters into those of an equivalent compact GA. Computer simulations compare both algorithms in terms of solution quality and speed. This work raises important questions about the use of information in a genetic algorithm, and its ramifications show us a direction that can lead to the design of more efficient GAs.

2) Genetic algorithm based clustering technique - Ujjwal Maulik

In his paper demonstrated the effectiveness of the GA- based clustering algorithm in providing optimal clusters, several artificial and real-life data sets with the number of dimensions ranging from two to ten and the number of clusters ranging from two to nine have been considered. The results show that the GA-clustering algorithm provides a performance that is significantly superior to that of the K-means algorithm for these data sets.

3) An improved genetic algorithm for optimizing total supply chain cost in inventory location routing problem - Ahmad Sayed

Here the author formulated a mathematical formula to minimize the total supply chain cost. Being NP-hard, an Improved Genetic Algorithm (IGA) is designed and used to solve the problem. Two instances (10 and 30 customers) are solved; to study the effect of the total vehicles capacity (number of available vehicles per depot and vehicle capacity), on the total supply chain cost. The results show that, the IGA outperforms the GA in reaching lower cost, especially for high number of customers. The superiority of the obtained solution performance is basically achieved on the expense of computational time. For the considered problem, the total cost decreases with the increase of vehicle capacity due to the usage of fewer depots.

4) A New Parallel Genetic Algorithm for Reducing the Bullwhip Effect in an Automotive Supply Chain – Umut Tosun

The main objective of this paper is to form a parallel genetic algorithm to reduce the bullwhip effect and cost in an automotive supply chain.

5) A genetic algorithm for the two-stage supply chain distribution problem associated with a fixed charge - N.Jawahar

Here the author shows that GA generates better solution than the approximation method and is capable of providing solution either equal or closer to the lower bound solution of the problem.

6) A simulation-based genetic algorithm for inventory optimization in a serial supply chain - J. Sudhir Ryan Daniel

This study makes an attempt to optimize the inventory (i.e. base-stock) levels at different members in a serial supply chain with the objective of minimizing the TSCC. A GA is proposed to optimize the base-stock levels. The proposed GA is found to perform very well in terms of yielding solutions that are not significantly different from the optimal solutions (obtained through complete enumeration of solution space) but with significantly less computing effort.

7) A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm - Alireza Askarzadeh.

This journal is based on the idea that crows store their excess food in hiding places and retrieve it when the food is needed. The results obtained by CSA are compared with the results of various algorithms. Simulation results reveal that using CSA may lead to finding promising results compared to the other algorithms.

8) Island-based Crow Search Algorithm for solving optimal control problems - Mert SinanTurgut

This literature includes different applications of this algorithm on engineering design problems. This study embeds the fundamentals of the island model concepts into the Crow Search Algorithm to improve its probing capabilities of the search domain, by means of the periodically interacting subpopulations on the course of iterations.

9) A modified crow search algorithm (MCSA) for solving economic load dispatch problem – Farid Mohammadi

This paper presents a novel evolutionary optimization algorithm namely the modified crow search algorithm (MCSA) for solving the non-convex economic load dispatch (ELD) problem which improves the crow search algorithm (CSA) by an innovative selection of the crows and adaptive adjustment of the flight length

10) Development and applications of an intelligent crow search algorithm based on opposition based learning – Shalini Shekhawat

The author introduces the intelligent crow search algorithm (ICSA). They author says that the results reveal that ICSA exhibits competitive performance on benchmarks and real applications when compared with some contemporary optimizers.

11) CCSA: Conscious Neighbourhood-based Crow Search Algorithm for Solving Global Optimization Problems – Hoda Zamani

The author introduces us to a conscious neighbourhood-based crow search algorithm (CCSA) which is proposed for solving global optimization and engineering design problems. It is a successful improvement to tackle the imbalance search strategy and premature convergence problems of the crow search algorithm.

12) An overview of Gravitational Search Algorithm utilization in optimization problems-Norlina Mohd Sabri, Mazidah Puteh, Mohamad Rusop Mahmood, 2013 IEEE 3rd International Conference

The aim of the paper is to investigate GSA utilization in various optimization problems. Based on the literature, GSA is capable of providing more accurate, effective and robust high-quality solution for most of the optimization problems. The algorithm has been applied in various applications and has solved various optimization problems such as in power system, controller design, network routing, sensor networks, software design and many more. GSA has showed better performance in solving the optimization problems compared to other previous algorithms such as PSO, ACO and ABC.

13) GSA: A Gravitational Search Algorithm-Esmat Rashedi, Hossein Nezamabadi-pour, Saeid Saryazdi, Information Sciences- 8 March 2009

In this paper, a new optimization algorithm based on the law of gravity and mass interactions is introduced. The proposed method has been compared with some well-known heuristic search methods. The obtained results confirm the high performance of the proposed method in solving various nonlinear functions.

14) SCGSA: A sine chaotic gravitational search algorithm for continuous optimization problems-Jianhua Jiang, Keqin Li, Expert Systems with Applications, Volume 144, 15 April 2020

In this paper, with insightful utilization of sine cosine algorithm, puts forward a sine chaotic gravitational search algorithm (SCGSA) as a further step of CGSA to escape from its local optima stagnation. The experiments show remarkable results in both the speed of convergence and the ability of finding global optima in 30 benchmark functions.

15) A novel hybrid particle swarm optimization and gravitational search algorithm for multi-objective optimization of text mining-Mohamed Atef Mos, Applied Soft Computing, Volume 90, May 2020

In this study, mining big social media data is re-formulated into a multi-objective optimization (MOO) task for an extractive summary. A Gravitational Search Algorithm (GSA) is utilized for optimizing several expressive objectives for generating a concise summary of SM. Moreover, particle swarm optimization (PSO) is mixed with GSA in a new shape to strengthen a local search ability and slow convergence speed in standard GSA.

16) Performance analysis and optimization for maximum exergy efficiency of a geothermal power plant using gravitational search algorithm- Osman Özkaraca, Ali Keçebaş, Energy Conversion and Management, Volume 185, 1 April 2019

It used the gravitational search algorithm to maximize its exergy efficiency. System exergy efficiency can be maximized from 14% to 31% using gravitational search algorithm. Thus, a more compatible operating strategy between system components is ensured.

17) The Whale Optimization Algorithm-Seyedali Mirjalili, Andrew Lewis Advances in Engineering Software, Volume 95, May 2016

The Whale Optimization Algorithm inspired by humpback whales is proposed. The results on the unimodal functions show the superior exploitation of WOA. The exploration ability of WOA is confirmed by the results on multimodal functions. The results on structural design problems confirm the performance of WOA in practice.

18) Particle swarm optimization: An improved particle swarm optimization algorithm-Yan Jiang, Xianing Wu, Applied Mathematics and Computation, Volume 193, Issue 1, 1 October 2007

In the new algorithm, a population of points sampled randomly from the feasible space. Then the population is partitioned into several sub-swarms, each of which is made to evolve based on particle swarm optimization (PSO) algorithm. At periodic stages in the evolution, the entire population is shuffled, and then points are reassigned to sub-swarms to ensure information sharing. This method greatly elevates the ability of exploration and exploitation. Simulations for three benchmark test functions show that IPSO possesses better ability to find the global optimum than that of the standard PSO algorithm.

19) Ant colony optimization: Introduction and recent trends-Christian Blum, Physics of Life Reviews, Volume 2, Issue 4, December 2005

The inspiring source of ant colony optimization is the foraging behavior of real ant colonies. This behavior is exploited in artificial ant colonies for the search of approximate solutions to discrete optimization problems, to continuous optimization problems, and to important problems in telecommunications, such as routing and load balancing.

20) Application of flower pollination algorithm for optimal placement and sizing of distributed generation in Distribution systems- P. Dinakar, Prasad Reddy, T.Gowri Manohar, Journal of Electrical Systems and Information Technology, Volume 3, Issue 1, May 2016

The main contribution of the paper is to find the optimal locations of Distributed generator (DG) units and sizes. Index vector method is used for optimal DG locations. In this paper new optimization algorithm i.e. flower pollination algorithm is proposed to determine the optimal DG size.

21) A global best artificial bee colony algorithm for global optimization-Weifeng Gao, Lingling Huang, Journal of Computational and Applied Mathematics, Volume 236, Issue 11, May 2012

The artificial bee colony (ABC) algorithm is a relatively new optimization technique which has been shown to be competitive to other population-based algorithms. Inspired by differential evolution (DE).

22) Schuffled complex evolution algorithm:Optimal use of the SCE-UA global optimization method for calibrating watershed models-Qingyun Duan, Vijai K.Gupta, Journal of Hydrology, Volume 158, Issues 3–4, 15 June 1994

A global optimization method known as the SCE-UA (shuffled complex evolution method developed at The University of Arizona) has shown promise as an effective and efficient optimization technique for calibrating watershed models. Experience with the method has indicated that the effectiveness and efficiency of the algorithm are influenced by the choice of the algorithmic parameters.

23) Differential evolution – an easy and efficient evolutionary algorithm for model optimization-D.G.Mayer, A.A.Archer, Agricultural Systems, Volume 83, Issue 3, March 2005

Differential evolution (DE) is a simple variant of an evolutionary algorithm. DE has only three or four operational parameters, and can be coded in about 20 lines of pseudo-code. DE has the advantage of incorporating a relatively simple and efficient form of self-adapting mutation. This is one of the main advantages found in evolution strategies.

24) New few parameters differential evolution algorithm with application to structural identification-Rita Greco, Ivo Vanzi, Journal of Traffic and Transportation Engineering (English Edition), Volume 6, Issue 1, February 2019

New schemes for both mutation and crossover operators of standard differential evolution algorithm (DEA). An adaptive mutation operator is proposed. A crossover operator replacing binomial formulation is proposed.

OBJECTIVES

As far as the project is concerned deciding the objective is of utmost important. The objective of every supply chain should be to maximize the overall value generated.

The objectives of this project are

- To observe and understand features of the supply chain model conceived in this study.
- To introduce two algorithms similar to GA, modify and coded using MATLAB for managing the supply chain cost during manufacturing.
- To fix the different parameters for the best value, of each algorithm by continuously obtaining results and comparing values.
- To compare the Total Supply Chain Costs obtained from the three algorithms for different supply chain settings and demand sets.
- Plot the results obtained and make a detailed comparison.
- To introduce new algorithms which are better than GA for optimising the inventory costs and improving the profit of company.

In order to get the proper idea of supply chain management, various metaheuristic approaches and methodology that need to be adopted we conducted literature review online and textbooks which is explained in the next chapter.

METHODOLOGY

Initially we proposed Genetic Algorithm to optimize the base-stock levels with the objective of minimizing the sum of holding and shortage costs in the entire supply chain. Simulation is used to evaluate the base-stock Levels generated by the GA. The proposed GA is evaluated with the consideration of a variety of supply chain settings in order to test for its robustness of performance across different supply chain scenarios. To test the effectiveness of the proposed GA, we decided to develop two more similar algorithms and to compare the optimized base stock levels which are obtained as results and to select the best algorithm out of the three.

In this study, we model a serial supply chain consisting of a retailer, distributor, manufacturer, and supplier and address the problem of optimization of inventory (i.e. base-stock levels) in a supply chain operating with a periodic-review inventory system.

Model Assumptions

1. A single product flows through the supply chain.
2. All the members in the supply chain operate under a periodic-review base-stock policy, where the review period is a unit of time.
3. Lead time for information or order processing is zero or negligible.
4. Processing lead time and transportation lead time are combined accordingly at each stage and considered together as one component, called replenishment lead time for that stage or member of the supply chain.
5. Retailer faces random customer demand, which is assumed to be stationary.
6. There is no lot-size or discount policy for members in the supply chain.
7. Base-stock level at every member or installation in the supply chain takes discrete integer values.
8. Every member in the supply chain has his/her own holding and shortage cost rates, i.e. local or installation cost rates.
9. If the demand exceeds on-hand inventory, then the excess demand is backlogged.
10. Transportation cost is assumed to be directly proportional to the quantity shipped.
11. All installations have infinite capacity.

12. The source of supply of raw materials to the most upstream member, namely the supplier, is assumed to have infinite raw material availability.

Supply chain inventory level and customer service play a decisive role in determining the effectiveness of supply chain inventory policy.

GA is a search algorithm based on the mechanics of natural selection and natural genetics. Simplicity of operation and power of effect are two of the main attractions of the GA approach. The reason for the choice of GA is that GA can be applied to a wide range of problems and GA makes no assumptions about the functions to be optimized.

Mechanics of the proposed GA

1. **Chromosome representation**

GAs work on a population or collection of solutions to the given problem. Each individual in the population is called a chromosome. In the current study the problem is to optimize the installation base-stock levels across all members in the supply chain to reduce TSCC. The chromosome is represented as a phenotype, i.e. the actual values of the base-stock levels are used to code all the genes in a chromosome.

S_4	S_3	S_2	S_1

2. **Generation of initial population**

The initial population is created by generating solutions (i.e. base-stock levels) randomly within the base-stock levels limits, i.e. every base-stock level is bounded by upper and lower limits Base-stock level (S_i) at member 'i' will vary between these upper and lower limits. These limits are computed by taking into account the customer demand, minimum replenishment lead time, and maximum replenishment lead time with respect to the corresponding member. the maximum replenishment lead time with respect to member 1 is the sum of replenishment lead times from member 1 to the ultimate upstream member, namely member N. Member 1 may face this maximum replenishment lead time when all the upstream members have zero on-hand inventory. The minimum replenishment lead time for member 1 is the replenishment lead time for member 1 itself. Similarly, all members in the supply chain will have their respective minimum and maximum replenishment lead times.

3. **Evaluation and selection of chromosomes for mating pool**

Every chromosome in the parent population is evaluated through simulation and the respective objective function ($TSCC_k$) value is obtained through simulation of the supply chain. Fitness value f_k is computed for the k th chromosome by using the objective function value, i.e. f_k = $1/(1 + TSCC_k)$, where k = 1 to n. With such f_k values, chromosomes are considered for placement in the mating pool in order to allow the chromosomes for crossover.

4. **Crossover operation**

A single point crossover operator is used in this study. The first two chromosomes in the mating pool are selected for crossover operation. Crossover is performed with the probability of CR, by sampling a uniform random number, u. The chromosomes are crossed over if u is less than or equal to CR; otherwise, both chromosomes are directly placed in initial population as an intermediate population consisting of offspring.

5. **Mutation operation**

A gene-wise mutation is employed in this study because every gene in a chromosome represents the base-stock level of the corresponding member. Since the chromosome is represented in a phenotypic manner, the value of each gene is varied within the corresponding member's upper and lower limits with respect to the base-stock levels. All genes in initial population are mutated with the probability of MR by sampling a uniform random number, u. If u ≤ MR, then the value of the gene is altered as given below:

$$S_i^{new} = S_i^{old}(1 - x) + S_i^{old} * 2 * x * u$$

where u is a uniform random number and x denotes the fraction of S_i^{old}. It is to be noted that if the computed S_i^{new} takes a non-integer value, then it is rounded off to the nearest integer.

After going through multiple journals and literatures followed by a series of brain storming we selected the two algorithms which are similar to Genetic Algorithm and made the necessary adaptations.

Crow Search Algorithm

Crows are considered as one of the most intelligent birds. They contain the largest brain relative to their body size. Crows have been known to watch other birds, observe where the other birds hide their food, and steal it once the owner leaves. If a crow has committed thievery, it will take extra precautions such as moving hiding places to avoid being a future victim.

The principles of CSA are listed as follows:

- Crows live in the form of flock.
- Crows memorize the position of their hiding places.
- Crows follow each other to do thievery.
- Crows protect their caches from being pilfered by a probability.

It is assumed that there is a d-dimensional environment including a number of crows. In our study since we are considering a Four member supply chain, we took d as 4. The number of crows (flock size) is N and the position of crow i at time (iteration) $iter$ in the search space is specified by a vector $X^{i,iter}$ (i = 1,2,….,N ; $iter$ = 1,2,…., $iter_{max}$) where $X^{i,iter}$ = [$X_1^{i,iter}$, $X_2^{i,iter}$,….., $X_d^{i,iter}$] and $iter_{max}$ is the maximum number of iterations. Each crow has a memory in which the position of its hiding place is memorized. At iteration $iter$, the position of hiding place of crow i is shown by $m^{i,iter}$. This is the best position that crow i has obtained so far. Indeed, in memory of each crow the position of its best experience has been memorized. Crows move in the environment and search for better food sources (hiding places). Assume that at iteration $iter$, crow j wants to visit its hiding place, $m^{j,iter}$. At this iteration, crow i decides to follow crow j to approach to the hiding place of crow j. In this case, two states may happen:

State 1: Crow j does not know that crow i is following it. As a result, crow i will approach to the hiding place of crow j. In this case, the new position of crow i is obtained as follows:

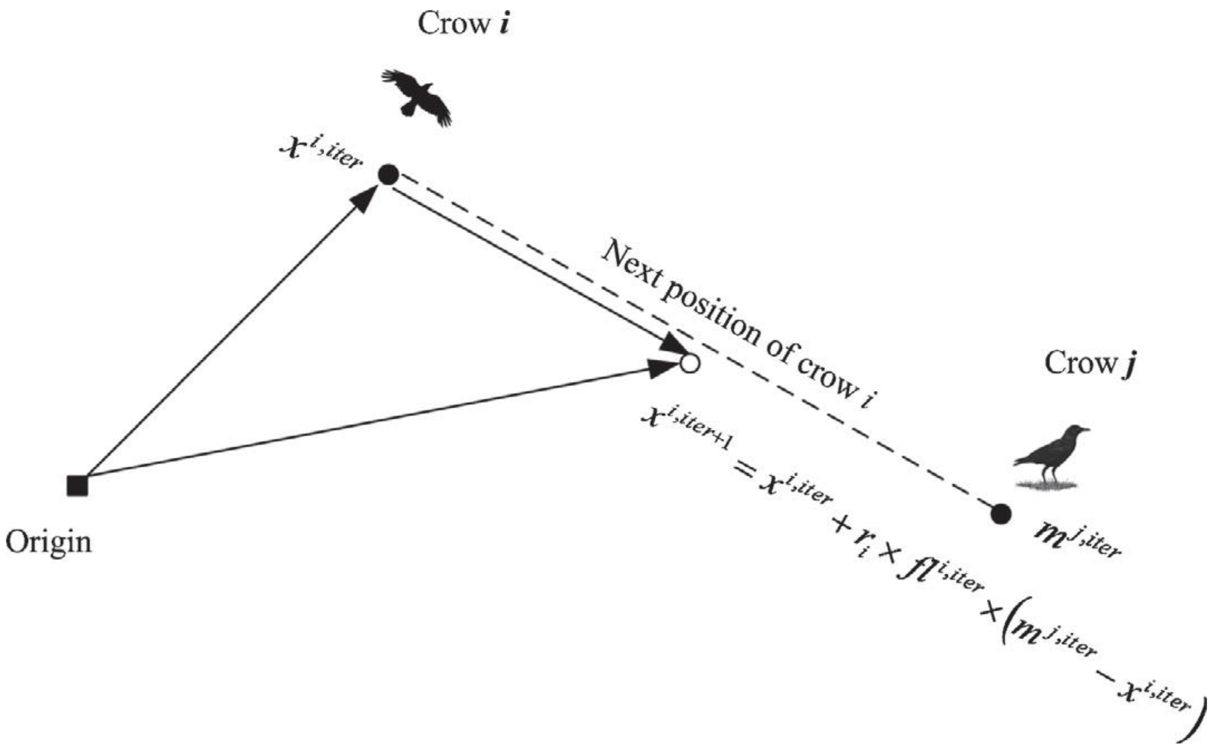
$$X^{i,iter+1} = X^{i,iter} + r_i * fl^{i,iter} * (m^{i,iter} - X^{i,iter})$$

Where r_i is a random number with uniform distribution between 0 and 1 and $fl^{i,iter}$ denotes the flight length of crow i at iteration iter. Small values of fl leads to local search and large values results in global search.

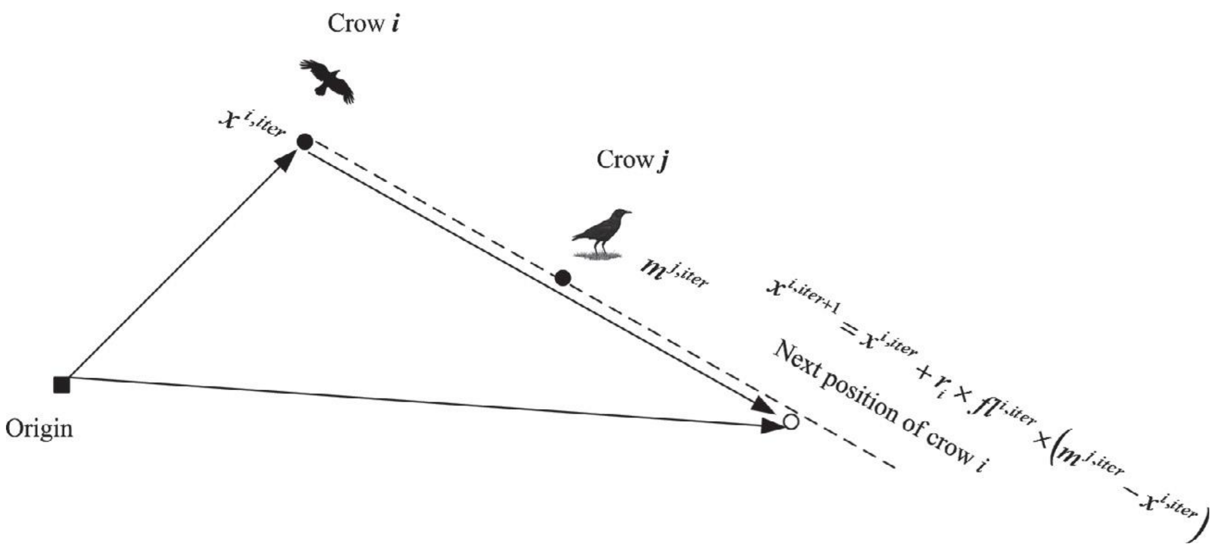
State 2: Crow j knows that crow i is following it. As a result, in order to protect its cache from being pilfered, crow j will fool crow i by going to another position of the search space.

$$X^{i,iter+1} = \begin{cases} X^{i,iter} + r_i * fl^{i,iter} * (m^{i,iter} - X^{i,iter}) & r_j \geq AP^{j,iter} \\ a\ random\ position & otherwise \end{cases}$$

Where r_j is a random number with uniform distribution between 0 and 1 and $AP^{j,iter}$ denotes the awareness probability of crow j at iteration iter.



(a) *fl* < 1



(b) *fl > 1*

CSA implementation for optimization

Step 1: **Initialize problem and adjustable parameters**
The optimization problem, decision variables and constraints are defined.

Then, the adjustable parameters of CSA (flock size (N), maximum number of iterations ($iter_{max}$), flight length (fl) and awareness probability (AP) are valued.

Step 2: **Initialize position and memory of crows**
N crows are randomly positioned in a d-dimensional search space as the members of the flock. Here in our study, we are taking d as 4 since we are considering a four member supply chain, each crow denotes a feasible solution of the problem and d is the number of decision variables.

Step 3: **Evaluate fitness (objective) function**
For each crow, the quality of its position is computed by inserting the decision variable values into the objective function.

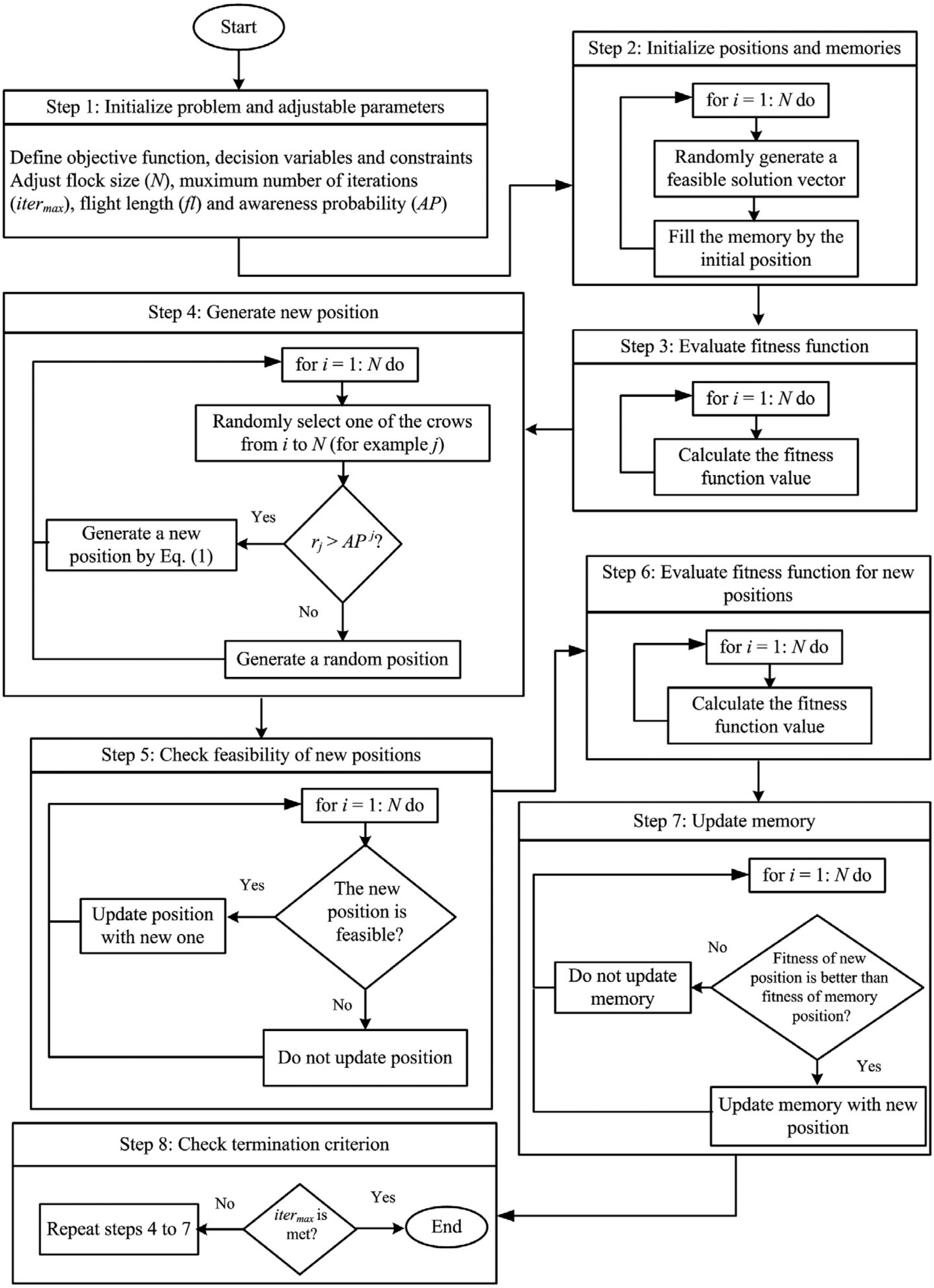
Step 4: **Generate new position**

Step 5: **Check the feasibility of new positions**
The feasibility of the new position of each crow is checked. If the new position of a crow is feasible, the crow updates its position. Otherwise, the crow stays in the current position and does not move to the generated new position.

Step 6: **Evaluate fitness function of new positions**
The fitness function value for the new position of each crow is computed.

Step 7: **Update memory.**
If the fitness function value of the new position of a crow is better than the fitness function value of the memorized position, the crow updates its memory by the new position.

Step 8: **Check termination criterion**
Steps 4–7 are repeated until $iter_{max}$ is reached. When the termination criterion is met, the best position of the memory in terms of the objective function value is reported as the solution of the optimization problem.



Start
Step 1: Initialize problem and adjustable parameters
Define objective function, decision variables and constraints
Adjust flock size (N), muximum number of iterations ($iter_{max}$), flight length (fl) and awareness probability (AP)
Step 2: Initialize positions and memories
for i = 1: N do
Randomly generate a feasible solution vector
Fill the memory by the initial position
Step 3: Evaluate fitness function
for i = 1: N do
Calculate the fitness function value
Step 4: Generate new position
for i = 1: N do
Randomly select one of the crows from i to N (for example j)
$r_j > AP^j$?
Yes
Generate a new position by Eq. (1)
No
Generate a random position
Step 5: Check feasibility of new positions
for i = 1: N do
The new position is feasible?
Yes
Update position with new one
No
Do not update position
Step 6: Evaluate fitness function for new positions
for i = 1: N do
Calculate the fitness function value
Step 7: Update memory
for i = 1: N do
Fitness of new position is better than fitness of memory position?
No
Do not update memory
Yes
Update memory with new position
Step 8: Check termination criterion
$iter_{max}$ is met?
No
Repeat steps 4 to 7
Yes
End

Gravitational search algorithm

The basic physical theory from which GSA is inspired is Newton's law of universal gravitation. The GSA performs search by employing a collection of agents (candidate solutions) that have masses proportional to the value of a fitness function. During iteration, the masses attract each other by the gravity forces between them. The heavier the mass, the bigger the attractive force. Therefore, the heaviest mass, which is possibly close to the global optimum, attracts the other masses in proportion to their distances.

This algorithm is formulated as follows:
Every mass has a position in search space as follows:

$$x_i = (x_i^1, \ldots x_i^d, \ldots x_i^n), \text{ i = 1, 2,.....,N}$$

Where N is the number of masses, n is the dimension of the problem, and x_i^d is the position of the i^{th} agent in the d^{th} dimension. Here, in our study case we are taking d as 4 since we are considering a four member supply chain.
The algorithm starts by randomly placing all agents in a search space. During all epochs, the gravitational force from agent j on agent i at a specific time t is defined as follows:

$$F_{ij}^d(\text{t}) = \text{G(t)}\,\frac{M_{pi}(t) * M_{aj}(t)}{R_{ij}(t) + \in}\,(x_j^d(\text{t}) - x_i^d(\text{t}))$$

where M_{aj} is the active gravitational mass related to agent j, M_{pi} is the passive gravitational mass related to agent i, G(t) is a gravitational constant at time t, ∈ is a small constant, and R_{ij}(t) is the Euclidian distance between two agents i and j.

The gravitational factor (G) and the Euclidian distance between two agents i and j are calculated as follows:

$$\text{G(t)} = G_0 * \exp\left(-\propto * \frac{iter}{maxiter}\right)$$

$$R_{ij}(t) = X_i(t), X_j(t)_2$$

where ∝ is the coefficient of decrease, G_0 is the initial gravitational constant, *iter* is the current iteration, and *maxiter* is the maximum number of iterations.

The total force that acts on agent i is calculated by the following equation:

$$F_i^d(\mathrm{t}) = \sum_{j\in kbest, j\neq i}^{N} rand_j \; F_{ij}^d(t)$$

Where $rand_j$ is a random number in the interval [0,1] and $kbest$ is the set of first K agents with the best fitness value and biggest mass.
To improve the performance of GSA by controlling exploration and exploitation only the Kbest agents will attract the others. Kbest is a function of time, with the initial value K_0 at the beginning and decreasing with time. In such a way, at the beginning, all agents apply the force, and as time passes, Kbest is decreased linearly and at the end there will be just one agent applying force to the others.

$Kbest$ for the t^{th} iteration is defined as,

$$Kbest(t) = final_{per} + \left(\frac{1-t}{\max\_it}\right) * (100 - final_{per})$$

Here, $\max\_it$ is maximum number of iterations and $final_{per}$ is the percent of objects which apply force to others. The equation of $Kbest$ shows that its value decreases linearly over iterations.

Newton's law of motion has also been utilised in this algorithm, which states that the acceleration of a mass is proportional to the applied force and inverse to its mass, so the accelerations of all agents are calculated as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}$$

where d is the dimension of the problem, t is a specific time, and M_{ii} is the inertial mass of agent i.

The velocity and position of agents are calculated as follows:

$$v_i^d(t+1) = rand_i * v_i^d(t) + a_i^d(t)$$
$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$$

where d is the problem's dimension and is a random number in the interval [0,1].

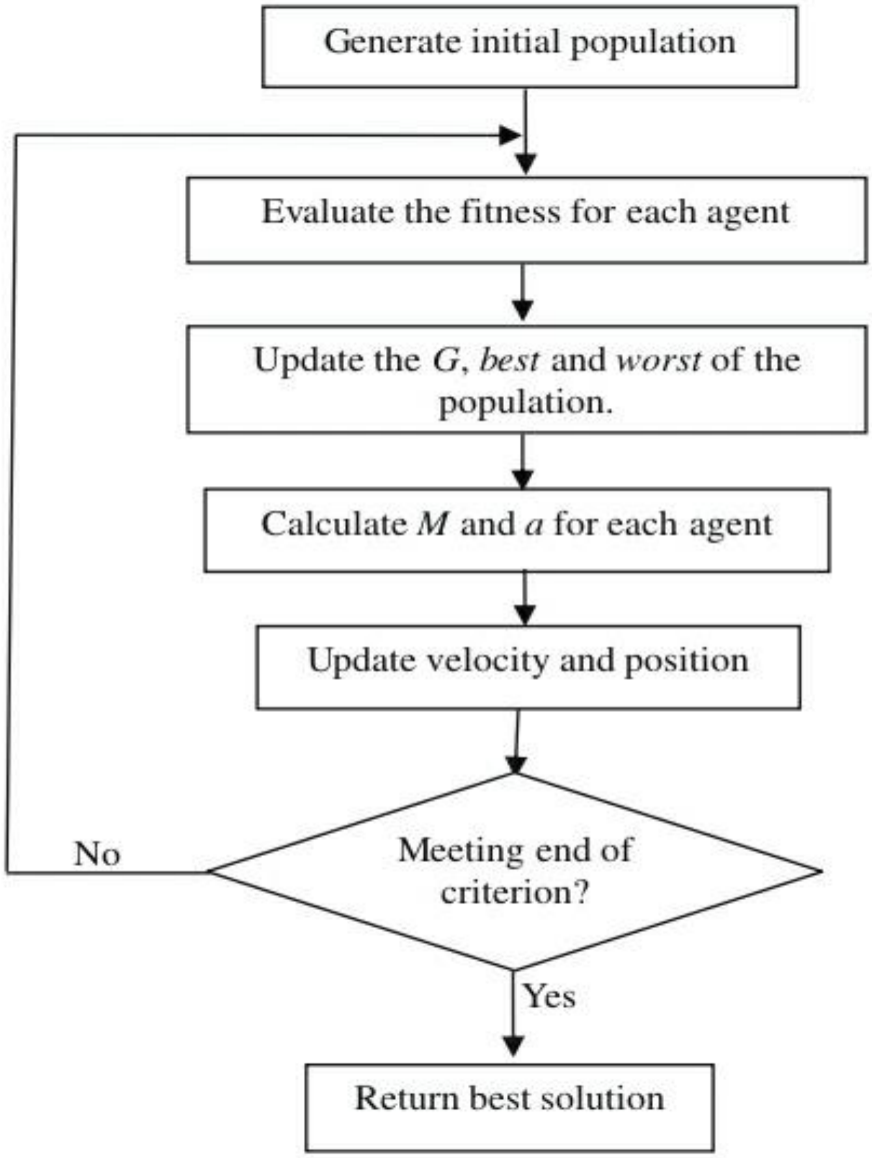
Since agents' masses are defined by their fitness evaluation, the agent with heaviest mass is the fittest agent. According to the above equations, the heaviest agent has the highest attractive force and the slowest movement. Since there is a direct relation between mass and the fitness function, a normalisation method has been adopted to scale masses as follows:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}$$

where fit_i (t) is the fitness value of the agent i at time t, best (t) is the fittest agent at time t, and worst (t) is the weakest agent at time t.

In the GSA, at first, all agents are initialised with random values. Here, we assume that the gravitational and the inertia masses are the same. A bigger inertia mass provides a slower motion of agents in the search space and hence a more precise search. Conversely, a bigger gravitational mass causes a higher attraction of agents. This permits a faster convergence.



Adaptive gbest guided Gravitational Search Algorithm

This algorithm is used here to propose a method to overcome the problem of slow exploitation.

Slow Exploitation

In GSA, the gravitational constant (G) defines the speed at which solutions change their location in solution space. high value of G results in high intensities of gravitational forces and resulting rapid movement in earlier iterations. However, G is progressively decreasing and this, combined with the slow movement of increasingly heavy agents, helps GSA during exploitation. Unfortunately, heavy masses with slow movement and less intensity of attractive force significantly degrade the speed of convergence as well. Therefore, it seems that GSA suffers from slow search speed in the exploitation phase.

GSA has no memory for saving the best solution obtained so far so the best solution might be lost as the best mass is attracted away by other less fit masses. All these problems motivated us to adapt the method discussed in this journal, for improving the exploitation.

Improving the Exploitation

The basic idea of the proposed method is to save and use the location of the best mass to speed up the exploitation phase. There are two benefits in this method:

- Accelerating the movement of particles towards the location of the best mass, which may help them to surpass it and be the best mass in the next iteration.
- Saving the best solution attained so far.

Following equation is proposed for mathematically modelling the proposed method:

$$V_i(t+1) = rand * V_i(t) + c_1' * ac_i(t) + c_2' * \left(gbest - X_i(t)\right)$$

where $V_i(t)$ is the velocity of agent i at iteration t, c_1' and c_2' are accelerating coefficients, rand is a random number between 0 and 1, $ac_i(t)$ is the acceleration of agent i at iteration t, and $gbest$ is the position of the best solution acquired so far.

The first component $rand * V_i(t) + c_1' * ac_i(t) + c_2'$ is the same as that of GSA, in which the exploration of the masses is emphasised. The second component $c_2' * \left(gbest - X_i(t)\right)$ is responsible for attracting masses towards the best masses obtained so far. The distance of each mass from the best mass is calculated by $gbest - X_i(t)$. We adaptively decrease c_1' and increase c_2' so that the masses tend to accelerate towards the best solution as the algorithm reaches the exploitation phase. Values of c_1' and c_2' are given by,

$$c_1' = \left(-2t^3 / T^3\right) + 2$$

$$c_2' = \left({2t^3}/{T^3}\right)$$

Where t is the current iteration and T is the maximum no of iterations.

The proposed method uses a memory (gbest) for saving the best solution obtained so far, in contrast to the unmodified GSA, so it is accessible at any time and will not be lost. The effect of gbest is emphasised in the exploitation phase by adapting c_1' and c_2'. Also the computational cost of this method is extremely low.

EXPERIMENTAL ANALYSIS

The supply chain considered in the current study is a single-product four-stage serial supply chain, whose members include retailer, distributor, manufacturer, and supplier. The assumption about the supply chain model conceived here is already discussed in the Methodology chapter.

A downstream member, say the retailer, places an order to the upstream member, namely the distributor, only when the retailer's on-hand inventory is depleted because of customer demand. Downstream members at any time have some quantity of inventory stock, known as inventory on-hand. The outstanding orders, which have been ordered by the downstream member to the upstream member, but have not been physically delivered by the upstream member to the downstream member are termed as on-order inventory and the quantity that has been ordered by the customer, but not fulfilled, is referred to as the back-order quantity. If the inventory position falls below the base-stock level, a replenishment order is placed for a quantity that will bring the inventory position back to the prespecified base-stock level. Also, every member is associated with the local holding and shortage cost rates; the sum of such costs incurred by all members in the chain is referred to as the TSCC. All members have their respective replenishment order lead times that can vary anywhere from the minimum replenishment lead time to the maximum replenishment lead time, depending upon the on-hand inventory of the upstream members. The most downstream member, i.e. the retailer faces the end customer demand.

Different supply chain settings considered in the study

Eight different supply chain settings are selected for the purpose of performance analysis and optimization using algorithms. These supply chain settings are selected in such a way that there exist different holding and shortage cost ratios and demand range for different members. These holding cost rates, shortage cost rates and demand range are considered in order to check for the robustness of the proposed algorithms in terms of the consistent and good performance of these across various supply chain settings and scenarios.

SETTING - A1				
	Retailer	Distributor	Manufacturer	Supplier
Holding Cost	2	2	2	2
Shortage Cost	4	0	0	0

Lead Time	1	1	1	1
Max Demand	100			
Min Demand	0			

SETTING - A2				
	Retailer	Distributor	Manufacturer	Supplier
Holding Cost	2	2	2	2
Shortage Cost	4	0	0	0
Lead Time	1	1	1	1
Max Demand	1000			
Min Demand	0			

SETTING - A3				
	Retailer	Distributor	Manufacturer	Supplier
Holding Cost	8	6	4	2
Shortage Cost	16	0	0	0
Lead Time	1	1	1	1
Max Demand	100			
Min Demand	0			

SETTING - A4				
	Retailer	Distributor	Manufacturer	Supplier
Holding Cost	8	6	4	2
Shortage Cost	16	0	0	0
Lead Time	1	1	1	1
Max Demand	1000			
Min Demand	0			

SETTING - A5				
	Retailer	Distributor	Manufacturer	Supplier
Holding Cost	100	75	50	25
Shortage Cost	200	0	0	0
Lead Time	1	1	1	1
Max Demand	100			
Min Demand	0			

SETTING - A6				
	Retailer	Distributor	Manufacturer	Supplier
Holding Cost	100	75	50	25
Shortage Cost	200	0	0	0
Lead Time	1	1	1	1
Max Demand	1000			
Min Demand	0			

SETTING - A7				
	Retailer	Distributor	Manufacturer	Supplier
Holding Cost	100	100	100	100
Shortage Cost	200	0	0	0
Lead Time	1	1	1	1
Max Demand	100			
Min Demand	0			

SETTING - A8				
	Retailer	Distributor	Manufacturer	Supplier
Holding Cost	100	100	100	100
Shortage Cost	200	0	0	0
Lead Time	1	1	1	1
Max Demand	1000			
Min Demand	0			

PARAMETER SETTING FOR GA

Crossover rate, Mutation rate and 'x' are the three parameters that are needed to be specified for employing GA. Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next and it occurs during evolution according to a user-definable mutation rate. Crossover is a genetic operator used to combine the genetic information of two parents to generate new offspring. Crossover rate indicate a ratio of how many couples will be picked for mating. 'x' denotes the fraction of base stock level of mutating gene before mutation. The crossover rate (CR), mutation rate (MR), and 'x' for the proposed GA are fixed based on the study conducted by J Sudhir Ryan Daniel and Chandrasekharan Rajendran (2005).

PARAMETER SETTING FOR CSA

Awareness Probability and Flight Length (fl) are the two parameters that are to be fixed for employing Crow Search Algorithm. Flight Length denotes the distance travelled by the crow in given time. Small values of fl leads to local search and large values results in global search. Awareness Probability indicates the probability by which the crow is aware about other crow following it. In CSA, intensification and diversification are mainly controlled by the parameter of awareness probability (AP). Use of small values of AP, increases intensification and use of large values of AP increases diversification. All parameters are checked for first supply chain setting, A1.

BEST VALUES FOR DIFFERENT FLIGHT LENGTH

fl = 0.5		fl = 2		fl = 5		fl =10	
Best chromosome	TSCC	Best chromosome	TSCC	Best chromosome	TSCC	Best chromosome	TSCC
17 44 65 95	127642	3 32 86 100	125724	4 30 87 100	125722	5 27 70 119	125764
1 39 8 188	126872	0 34 68 129	126146	0 34 67 130	126146	0 32 71 127	126160
14 34 59 127	134100	0 44 85 100	133326	1 43 85 100	133336	1 40 70 118	133378
17 42 52 119	129084	1 45 81 100	128084	1 45 82 99	128088	1 45 80 100	128114
18 45 69 91	127014	1 39 81 100	124910	1 39 77 104	124918	2 40 69 110	124942
17 44 57 112	124420	0 28 89 108	122906	0 28 94 103	122906	1 27 83 115	122920
14 34 59 127	129202	1 34 68 124	128118	1 36 60 130	128114	0 39 70 118	128112
14 34 59 127	137304	0 40 86 100	135710	0 38 87 101	135716	1 37 78 110	135738
16 44 63 102	123536	0 29 96 100	121746	1 27 95 102	121750	1 26 84 114	121776
14 34 59 127	123972	0 45 86 100	123208	1 45 82 103	123222	0 33 80 118	123268

17 42 57 115	129012	0 36 87 100	126648	2 31 87 102	126662	1 36 80 104	126690
19 48 56 102	128380	0 40 84 100	126320	1 39 80 104	126326	0 20 95 109	126378
18 45 68 91	133742	0 33 84 104	131714	0 33 69 119	131714	0 29 69 123	131722
17 44 66 95	141120	0 38 80 99	138782	1 27 89 101	138816	0 28 90 99	138802
15 42 53 122	127212	2 38 89 100	126502	0 40 89 100	126506	1 25 98 104	126546
MEAN TSCC =	**129507.5**		**127989.6**		**127996.1**		**128020.7**

On comparing the Mean TSCC values of 15 sets of demand for different values of fl, we found that for out study case , the optimum value of fl is 2 and TSCC increases on lowering and increasing flight length value.

BEST VALUES FOR DIFFERENT AWARENESS PROBABILITIES OF CSA

AP = 0.1		**AP = 0.15**		**AP = 0.2**		**AP = 0.5**	
BEST CHROMOSOME	**TSCC**	**BEST CHROMOSOME**	**TSCC**	**BEST CHROMOSOME**	**TSCC**	**BEST CHROMOSOME**	**TSCC**
100 119 93 159	**563374**	**100 119 93 159**	**563374**	**101 118 92 160**	**563419**	**105 103 100 162**	**564811**
93 104 104 158	**598032**	**94 104 104 158**	**598049**	**93 104 106 157**	**598110**	**100 96 103 160**	**598670**
91 116 97 160	**613808**	**93 115 98 160**	**613820**	**94 119 93 161**	**614006**	**104 102 104 156**	**615440**
96 115 100 160	**591577**	**99 113 100 160**	**591593**	**96 114 101 160**	**591565**	**107 109 98 160**	**592134**
83 107 97 158	**618572**	**98 92 103 157**	**619075**	**83 107 97 158**	**618572**	**90 100 100 157**	**618854**
97 115 106 161	**602086**	**97 115 107 161**	**602090**	**97 115 106 161**	**602086**	**99 114 107 160**	**602126**
98 93 100 164	**562053**	**96 93 101 164**	**562007**	**103 93 99 163**	**562561**	**97 93 99 166**	**562056**
96 113 96 163	**615355**	**98 110 97 163**	**615379**	**95 113 97 163**	**615367**	**104 100 103 161**	**615933**

96 101 99 162	571280	96 99 101 162	571254	97 98 100 163	571323	96 98 103 162	571380
90 121 94 158	588899	91 119 95 158	588899	92 118 96 158	588959	100 97 110 157	591244
79 126 92 161	584996	81 124 93 161	585018	80 125 92 162	585041	77 121 96 162	585341
103 107 103 159	603623	104 108 102 159	603626	105 107 102 160	603669	105 102 107 158	603771
95 109 106 163	620237	100 104 108 162	620229	98 107 106 163	620274	100 108 105 164	620597
98 115 95 154	598058	96 116 95 154	598015	98 114 96 154	598082	99 110 95 157	598167
100 116 108 160	581668	101 116 108 160	581694	100 115 108 161	581696	103 115 105 163	582068
MEAN TSCC =	594241.2		594274.8		594315.3		596839.4

On comparing the Mean TSCC values of 15 sets of demand for different values of AP, we found that for out study case the optimum value of AP is 0.1 and TSCC increases on increasing awareness probability value.

PARAMETER SETTING FOR GSA

Initial Gravitational constant (G_0) , Alfa($\propto$) and Kbest are the three parameters that are to be fixed for employing Crow Search Algorithm. The gravitational constant, G, is initialized at the beginning and will be reduced with time to control the search accuracy. In other words, G is a function of the initial value (G0) and time (t):

$$G(t) = G(G_0,t)$$

$\propto$ is the coefficient of decrease . Kbest is the set of first K agents with the best fitness value and biggest mass. Kbest is a function of time, with the initial value K_0 at the beginning and decreasing with time. In such a way, at the beginning, all agents apply the force, and as time passes, Kbest is decreased linearly and at the end there will be just one agent applying force to the others. All parameters are checked for first supply chain setting, A1.

BEST VALUES FOR DIFFERENT INITIAL GRAVITATIONAL CONSTANT OF GGSA

G0 = 100		G0 = 500		GO = 1000	
BEST CHROMOSOME	TSCC	BEST CHROMOSOME	TSCC	BEST CHROMOSOME	TSCC
4 33 84 100	125722	4 33 84 100	125722	4 33 84 100	125722
0 34 85 112	126146	0 34 78 119	126146	0 34 80 117	126146
0 39 90 100	133336	0 44 74 111	133348	0 44 85 100	133326
1 45 81 100	128084	1 45 81 100	128084	0 46 79 102	128090
1 39 81 100	124910	1 39 81 100	124910	1 39 81 100	124910
0 29 86 111	122906	0 28 91 106	122906	0 28 89 108	122906
1 38 79 109	128110	1 38 55 133	128110	1 38 68 120	128110
0 40 86 100	135710	0 40 82 104	135718	0 40 82 104	135718
1 28 96 100	121744	1 28 96 100	121744	1 24 100 100	121752
1 31 97 102	123242	0 43 88 100	123212	0 43 88 100	123212
0 35 86 101	126648	0 34 88 100	126650	0 36 84 103	126650
1 39 84 100	126318	1 39 84 100	126318	1 37 86 100	126322
0 33 64 124	131714	0 32 74 115	131716	0 33 81 107	131714
0 38 80 99	138782	0 38 80 99	138782	0 37 81 99	138784
2 38 89 100	126502	2 38 89 100	126502	2 38 89 100	126502
MEAN TSCC	**127991.6**		**127991.2**		**127990.9**

G0 = 1500		G0 = 2000	
BEST CHROMOSOME	TSCC	BEST CHROMOSOME	TSCC
4 32 85 100	125722	4 32 85 100	125722
0 34 73 124	126146	0 34 84 113	126146
0 44 85 100	133326	0 42 87 100	133330
1 45 81 100	128084	1 45 76 105	128094
1 39 81 100	124910	0 40 80 101	124914
0 28 95 102	122906	0 29 91 106	122906
1 38 75 113	128110	1 38 75 113	128110
0 40 86 100	135710	0 34 92 100	135722
0 29 81 115	121776	1 28 72 124	121792
0 37 94 100	123224	0 44 79 108	123226
0 36 85 102	126648	0 35 87 100	126648
1 39 84 100	126318	1 37 86 100	126322
0 35 40 146	131722	0 27 81 113	131726
0 38 80 99	138782	0 37 78 102	138790
2 38 89 100	126502	2 38 89 100	126502
MEAN TSCC	**127992.4**		**127996.7**

On comparing the Mean TSCC values of 15 sets of demand for different values of G_0, we found that for out study case the optimum value of G_0 is 1000.

BEST VALUES FOR DIFFERENT ALFA VALUES OF GGSA

Alfa = 4		Alfa = 10		Alfa = 15		Alfa = 2	
chromosome	**tscc**	**chromosome**	**tscc**	**chromosome**	**tscc**	**chromosome**	**tscc**
4 33 84 100	125722	4 33 81 103	125728	3 25 93 100	125734	4 32 85 100	125722
0 34 80 117	126146	0 34 83 114	126146	0 33 87 111	126148	0 33 62 136	126154
0 44 85 100	133326	0 44 85 100	133326	0 25 104 100	133364	0 45 70 114	133358
0 46 79 102	128090	1 45 75 106	128096	1 37 89 100	128100	0 26 101 100	128126
1 39 81 100	124910	1 39 66 115	124940	0 43 78 100	124924	1 38 81 101	124914
0 28 89 108	122906	0 29 74 123	122930	0 29 83 114	122912	0 25 92 108	122912
1 38 68 120	128110	1 38 57 131	128110	1 32 64 130	128122	1 36 57 133	128114
0 40 82 104	135718	0 39 63 124	135760	0 32 81 113	135752	0 40 68 118	135746
1 24 100 100	121752	1 28 90 106	121756	1 28 87 109	121762	0 29 94 102	121750
0 43 88 100	123212	0 45 70 116	123240	1 34 96 100	123232	0 26 104 101	123248
0 36 84 103	126650	0 36 85 102	126648	1 24 84 114	126698	0 30 92 100	126658
1 37 86 100	126322	0 40 84 100	126320	1 35 59 129	126384	1 38 84 101	126322
0 33 81 107	131714	0 33 53 135	131714	0 28 88 105	131724	0 33 71 117	131714
0 37 81 99	138784	0 38 69 110	138804	1 32 67 117	138832	1 37 73 106	138800
2 38 89 100	126502	2 38 82 107	126516	2 35 83 109	126526	2 38 84 105	126512
	127990.9		**128002.3**		**128014.3**		**128003.3**

On comparing the Mean TSCC values of 15 sets of demand for different values of Alfa , we found that for out study case the optimum value of Alfa is 1000.

CONVERGENCE OF ITERATIONS

CONVERGENCE LIMIT IN GA (500 ITERATIONS)

BEST CHROMOSOME	TSCC	POINT OF CONVERGENCE (ITERATION NO)
7 27 87 100	125750	393
0 34 67 130	126146	438
0 41 88 100	133332	150
4 39 84 100	128132	471
4 35 81 101	124938	451
5 18 74 128	122978	358
0 40 55 132	128116	370
0 27 99 100	135736	245
0 31 93 102	121758	495
3 39 34 155	123354	475
0 36 73 113	126676	488
0 28 81 114	126384	499
0 33 75 113	131714	474
2 35 78 102	138804	429
3 36 76 114	126542	457
AVG		**413**

We found that the average point of convergence of 500 iterations for GA is 413.

CONVERGENCE LIMIT IN CSA (500 ITERATIONS)

BEST CHROMOSOME	TSCC	POINT OF CONVERGENCE (ITERATION NO)
3 32 86 100	125724	232
0 34 68 129	126146	397
0 44 85 100	133326	481
1 45 81 100	128084	460
1 39 81 100	124910	321
0 29 87 110	122906	469
1 34 68 124	128118	287
1 39 86 100	135714	495
0 29 96 100	121746	221
0 45 83 103	123214	428
0 36 87 100	126648	332
0 40 84 100	126320	476
0 33 84 104	131714	309
0 38 80 99	138782	196
2 38 89 100	126502	462
AVG		**371**

We found that the average point of convergence of 500 iterations for CSA is 371.

CONVERGENCE LIMIT IN GGSA (500 ITERATIONS)

BEST CHROMOSOME	TSCC	POINT OF CONVERGENCE (ITERATION NO)
4 19 83 115	125752	241
1 32 75 123	126152	237
1 33 74 121	133398	249
3 29 76 119	128176	238
1 27 46 147	125028	245
0 28 53 145	122976	328
5 38 63 121	128160	269
2 33 78 113	135760	252
2 18 89 116	121810	240
2 24 80 125	123310	245
1 26 94 101	126668	254
2 39 63 120	126372	245
4 34 80 130	142304	296
1 36 71 109	138808	240
1 39 62 127	126558	237
AVG		**255**

We found that the average point of convergence of 500 iterations for GGSA is 255.

RESULTS AND ANALYSIS

COMPARISON OF GA,CSA AND GGSA

Thirty different demand sets were generated corresponding to the eight supply chain setting mentioned in the experimental analysis chapter.. All three algorithms are tested for each demand sets using supply chain settings. Results of the comparative study are shown below:

SETTING – A1					
GA		GSA		CSA	
BEST CHROMOSOME	TSCC	BEST CHROMOSOME	TSCC	BEST CHROMOSOME	TSCC
4 31 86 100	125722	4 33 84 100	125722	3 32 86 100	125724
0 34 67 130	126146	0 34 80 117	126146	0 34 68 129	126146
0 41 88 100	133332	0 44 85 100	133326	0 44 85 100	133326
2 36 89 100	128110	0 46 79 102	128090	1 45 81 100	128084
3 35 81 102	124926	1 39 81 100	124910	1 39 81 100	124910
4 19 74 128	122968	0 28 89 108	122906	0 28 89 108	122906
0 40 55 132	128116	1 38 68 120	128110	1 34 68 124	128118
0 27 99 100	135736	0 40 82 104	135718	0 40 86 100	135710
0 31 93 101	121756	1 24 100 100	121752	0 29 96 100	121746
3 42 37 149	123336	0 43 88 100	123212	0 45 86 100	123208
0 36 74 113	126670	0 36 84 103	126650	0 36 87 100	126648
2 37 85 100	126330	1 37 86 100	126322	0 40 84 100	126320
0 33 75 113	131714	0 33 81 107	131714	0 33 84 104	131714
2 35 78 102	138804	0 37 81 99	138784	0 38 80 99	138782
3 36 76 114	126542	2 38 89 100	126502	2 38 89 100	126502
0 52 64 115	129014	1 48 82 100	129002	1 49 81 100	129000
0 45 79 103	120938	4 43 80 100	120920	4 43 80 100	120920
2 40 35 151	129552	1 40 25 162	129548	1 39 33 155	129548
0 30 75 122	126502	0 32 82 113	126498	0 32 73 122	126498
0 34 36 157	130950	1 33 85 108	130850	1 33 93 100	130834
3 25 77 113	135308	2 34 82 100	135256	2 34 82 100	135256

3 41 52 133	124466	3 43 83 100	124396	3 43 83 100	124396
0 20 93 117	129432	0 40 90 100	129358	0 43 87 100	129352
2 32 42 150	124460	1 32 29 164	124450	1 32 62 131	124450
0 28 96 101	129472	0 32 79 114	129490	0 32 93 100	129462
3 40 40 142	116772	1 39 86 99	116684	0 43 83 99	116680
2 28 93 100	122454	1 31 70 121	122488	1 31 91 100	122446
0 31 99 105	127828	0 50 72 113	127818	0 50 56 129	127818
3 36 83 101	132264	3 34 86 100	132262	3 38 82 100	132262
5 38 30 157	132316	2 40 88 100	132174	2 40 88 100	132174
MEAN TSCC =	**127731.2**		**127701.9**		**127698**

SETTING – A2					
GA		**GGSA**		**CSA**	
BEST CHROMOSOME	**TSCC**	**BEST CHROMOSOME**	**TSCC**	**BEST CHROMOSOME**	**TSCC**
11 316 853 1041	1316034	1 284 943 996	1315880	1 349 879 996	1315750
11 323 795 1020	1267654	1 333 667 1144	1267652	4 330 815 996	1267594
0 245 660 1305	1313708	4 349 864 993	1312860	4 348 863 994	1312864
34 386 813 1053	1339610	23 410 853 998	1339430	22 411 852 998	1339432
10 342 901 1004	1303894	1 363 900 993	1303812	2 363 899 993	1303814
30 405 755 1048	1206310	34 423 785 995	1206146	34 423 785 995	1206146
22 309 866 1044	1281168	10 340 769 1122	1281082	10 340 732 1159	1281082
10 344 793 1090	1309000	1 437 800 997	1308610	2 436 801 997	1308608
17 274 776 1151	1249920	9 340 828 1042	1249774	9 339 380 1490	1249774
26 348 505 1449	1236406	12 353 966 997	1235410	12 353 966 997	1235410
28 246 945 1127	1231276	14 462 877 995	1230422	13 463 878 995	1230422
0 246 993 1037	1254108	1 463 817 998	1253550	1 500 780 998	1253476

19 334 843 1002	1249666	26 327 846999	1249646	26 327 846 999	1249646
9 448 769 1038	1321620	0 328 943 996	1321704	0 473 799 996	1321414
21 327 861 995	1250138	14 339 854998	1250082	16 337 854 998	1250090
49 270 896 1051	1247594	14 302 951997	1247334	14 302 951 997	1247334
8 307 834 1071	1322006	0 321 806 1095	1321966	0 321 826 1075	1321966
20 263 1008 997	1308398	7 423 862 996	1308050	7 423 865 996	1308050
13 205 844 1205	1158488	11 411 847 999	1157636	11 411 847 999	1157636
16 329 908 996	1233136	0 348 900 996	1233000	0 348 900 996	1233000
9 411 798 1008	1239726	741 784 1016	1239706	8 418 790 1010	1239710
26 141 960 1139	1297426	8 398 862 998	1296594	8 398 862 998	1296594
0 296 831 1100	1309280	1 292 667 1265	1309274	1 308 6311285	1309242
35 361 878 1004	1333738	35 440 808 996	1333716	35 366 882 996	1333716
13 408 525 1329	1263406	0 426 751 1102	1262890	0 426 853 1000	1262686
24 308 884 1002	1201842	11 388 823 1000	1201652	11 388 823 1000	1201652
10 239 528 1466	1251910	1 328 883 1030	1250794	1 328 914 999	1250732
21 253 1024 999	1361416	12 401 891 997	1361080	13 400 891 997	1361084
14 277 815 1178	1349546	4 364 875 1042	1349348	4 364 481 1436	1349348
24 370 594 1338	1243842	18 383 922 999	1243090	18 383 922 999	1243090
MEAN TSCC =	**1275076**		**1274740**		**1274712**

SETTING – A3					
GA		**GSA**		**CSA**	
BEST CHROMOSOME	**TSCC**	**BEST CHROMOSOME**	**TSCC**	**BEST CHROMOSOME**	**TSCC**
42 51 55 89	457846	41 52 54 90	457804	41 52 54 90	457804
45 50 55 93	463092	45 50 55 93	463092	45 50 55 93	463092

48 51 58 92	479572	48 49 60 92	479552	47 50 60 92	479564
46 54 57 90	465516	45 52 59 90	465510	46 54 57 90	465516
43 38 62 91	453682	43 38 62 91	453682	43 38 62 91	453682
43 46 61 90	453174	43 46 61 90	453174	43 46 61 90	453174
46 28 76 91	468154	44 50 56 92	463326	44 50 56 92	463326
43 57 53 91	491862	42 57 54 91	491810	42 57 54 91	491810
36 44 67 91	453558	36 44 66 92	453546	37 43 66 92	453546
51 39 63 92	444382	51 44 57 93	444334	51 40 61 93	444326
43 48 56 93	464510	43 48 56 93	464510	45 46 57 93	464602
43 47 56 94	461246	43 48 56 93	461132	43 48 56 93	461132
44 46 59 89	480614	43 47 59 89	480606	45 46 59 89	480630
47 34 61 93	507232	46 39 58 92	506950	46 39 58 92	506950
45 52 56 91	460162	45 53 55 92	460154	45 53 55 92	460154
44 42 64 93	464758	44 42 65 92	464732	44 42 65 92	464732
36 47 64 92	450658	36 47 63 93	450640	37 46 63 93	450662
41 50 59 93	474450	41 50 59 93	474450	41 50 59 93	474450
38 41 70 91	467144	39 40 70 91	467144	37 43 70 90	467144
48 46 60 92	473506	48 46 60 92	473506	49 45 60 92	473520
46 43 55 91	493518	46 43 55 91	493518	46 44 54 91	493518
50 49 56 92	445754	49 50 56 92	445752	49 50 56 92	445752
43 55 58 93	467668	41 57 58 93	467658	43 54 59 93	467658
35 53 61 91	454432	36 53 61 90	454410	36 53 61 90	454410
39 54 53 92	468062	39 54 54 91	468042	40 54 54 91	468060
41 50 56 91	427460	41 50 55 92	427412	43 48 55 92	427480
31 56 55 93	453266	31 56 55 93	453266	31 56 55 93	453266
36 53 65 93	465036	40 51 65 93	465058	36 53 65 93	465036
51 51 51 91	474640	51 52 50 91	474638	51 52 50 91	474638
51 49 59 90	474698	51 48 60 90	474678	51 48 60 90	474678
MEAN TSCC =	**465321.7**		**465136.2**		**465143.7**

SETTING – A4					
GA		**GSA**		**CSA**	
BEST CHROMOSOME	**TSCC**	**BEST CHROMOSOME**	**TSCC**	**BEST CHROMOSOME**	**TSCC**
473 411 603 924	4818564	465 412 590 935	4818374	465 412 590 935	4818374
459 425 561 881	4683790	461 426 558 883	4683758	461 426 558 883	4683758
409 438 586 920	4804284	411 434 590 919	4804208	411 434 590 919	4804208
464 488 576 918	4848966	468 486 580 913	4848814	431 515 577 916	4849246
348 573 574 914	4766614	345 576 573 912	4766542	349 574 563 922	4766584
421 414 633 899	4429356	422 414 630 902	4429316	433 408 625 905	4429450
427 492 584 883	4634178	423 483 584 887	4633830	423 483 584 887	4633830
447 482 532 934	4772372	442 482 540 928	4772248	442 482 539 929	4772254
348 452 632 922	4657124	349 451 635 919	4657072	349 451 635 919	4657072
488 497 571 930	4470548	488 494 570 933	4470478	488 494 570 933	4470478
516 430 613 939	4450380	511 433 613 939	4450284	511 433 613 939	4450284
393 553 548 925	4616040	456 497 550 926	4616432	393 553 550 923	4616000
414 471 567 910	4543372	416 466 572 909	4543322	412 478 569 909	4543334
460 552 567 885	4734090	461 551 568 885	4734042	461 551 568 885	4734042
433 380 637 904	4600786	433 381 634 905	4600762	433 381 634 905	4600762
495 416 633 901	4528218	498 417 629 902	4528126	498 417 630 901	4528126
411 505 584 900	4791224	408 496 596 899	4791100	408 496 596 899	4791100
442 469 601 938	4791828	434 471 597 942	4791422	434 484 584 942	4791516
388 427 639 909	4310858	372 440 623 918	4310604	372 440 623 918	4310604
405 416 661 918	4522384	407 426 649 919	4522202	407 430 645 919	4522202
431 246 791 896	4596416	386 397 687 898	4596690	388 397 684 899	4596698
462 485 598 902	4667704	461 481 597 908	4667560	461 481 597 908	4667560
401 485 573 921	4870544	407 472 577 923	4870234	409 471 577 923	4870250
454 538 523 923	4764432	455 537 526 921	4764388	455 535 527 921	4764392
348 523 626 906	4624666	350 519 625 909	4624582	350 519 625 909	4624582
384 420 615 913	4455780	387 404 630 913	4455108	396 395 630 913	4455098
417 391 629 943	4639460	431 379 632 942	4639288	431 379 632 942	4639288
542 507 574 906	4905488	542 509 573 904	4905404	542 509 573 904	4905404
426 538 569 930	4892194	428 547 562 928	4892032	428 547 562 928	4892032
507 371 655 955	4574900	507 373 657 954	4574722	507 373 657 954	4574722
MEAN TSCC =	**4658885**		**4658765**		**4658775**

SETTING – A5					
GA		**GSA**		**CSA**	
BEST CHROMOSOME	**TSCC**	**BEST CHROMOSOME**	**TSCC**	**BEST CHROMOSOME**	**TSCC**
42 51 55 89	5723075	41 52 54 90	5722550	41 52 54 90	5722550
45 50 55 93	5788650	45 50 55 93	5788650	45 50 55 93	5788650
48 51 58 92	5994650	48 49 60 92	5994400	47 50 60 92	5994550
46 54 57 90	5818950	45 52 59 90	5818875	46 54 57 90	5818950
43 38 62 91	5671025	43 38 62 91	5671025	43 38 62 91	5671025
43 46 61 90	5664675	43 46 61 90	5664675	43 46 61 90	5664675
46 28 76 91	5851925	44 50 56 92	5791575	44 50 56 92	5791575
43 57 53 91	6148275	42 57 54 91	6147625	42 57 54 91	6147625
36 44 67 91	5669475	36 44 66 92	5669325	37 43 66 92	5669325
51 39 63 92	5554775	51 44 57 93	5554175	51 40 61 93	5554075
43 48 56 93	5806375	43 48 56 93	5806375	45 46 57 93	5807525
43 47 56 94	5765575	43 48 56 93	5764150	43 48 56 93	5764150
44 46 59 89	6007675	43 47 59 89	6007575	45 46 59 89	6007875
47 34 61 93	6340400	46 39 58 92	6336875	46 39 58 92	6336875
45 52 56 91	5752025	45 53 55 92	5751925	45 53 55 92	5751925
44 42 64 93	5809475	44 42 65 92	5809150	44 42 65 92	5809150
36 47 64 92	5633225	36 47 63 93	5633000	37 46 63 93	5633275
41 50 59 93	5930625	41 50 59 93	5930625	41 50 59 93	5930625
38 41 70 91	5839300	39 40 70 91	5839300	37 43 70 90	5839300
48 46 60 92	5918825	48 46 60 92	5918825	49 45 60 92	5919000
46 43 55 91	6168975	46 43 55 91	6168975	46 44 54 91	6168975
50 49 56 92	5571925	49 50 56 92	5571900	49 50 56 92	5571900
43 55 58 93	5845850	41 57 58 93	5845725	43 54 59 93	5845725
35 53 61 91	5680400	36 53 61 90	5680125	36 53 61 90	5680125
39 54 53 92	5850775	39 54 54 91	5850525	40 54 54 91	5850750
41 50 56 91	5343250	41 50 55 92	5342650	43 48 55 92	5343500
31 56 55 93	5665825	31 56 55 93	5665825	31 56 55 93	5665825
36 53 65 93	5812950	40 51 65 93	5813225	36 53 65 93	5812950

51 51 51 91	5933000	51 52 50 91	5932975	51 52 50 91	5932975
51 49 59 90	5933725	51 48 60 90	5933475	51 48 60 90	5933475
MEAN TSCC =	**5816522**		**5814203**		**5814297**
SETTING – A6					
GA		**GSA**		**CSA**	
BEST CHROMOSOME	**TSCC**	**BEST CHROMOSOME**	**TSCC**	**BEST CHROMOSOME**	**TSCC**
473 411 603 924	60232050	465 412 590 935	60229675	465 412 590 935	60229675
459 425 561 881	58547375	461 426 558 883	58546975	461 426 558 883	58546975
409 438 586 920	60053550	411 434 590 919	60052600	411 434 590 919	60052600
464 488 576 918	60612075	468 486 580 913	60610175	431 515 577 916	60615575
348 573 574 914	59582675	345 576 573 912	59581775	349 574 563 922	59582300
421 414 633 899	55366950	422 414 630 902	55366450	433 408 625 905	55368125
427 492 584 883	57927225	423 483 584 887	57922875	423 483 584 887	57922875
447 482 532 934	59654650	442 482 540 928	59653100	442 482 539 929	59653175
348 452 632 922	58214050	349 451 635 919	58213400	349 451 635 919	58213400
488 497 571 930	55881850	488 494 570 933	55880975	488 494 570 933	55880975
516 430 613 939	55629750	511 433 613 939	55628550	511 433 613 939	55628550
393 553 548 925	57700500	456 497 550 926	57705400	393 553 550 923	57700000
414 471 567 910	56792150	416 466 572 909	56791525	412 478 569 909	56791675
460 552 567 885	59176125	461 551 568 885	59175525	461 551 568 885	59175525
433 380 637 904	57509825	433 381 634 905	57509525	433 381 634 905	57509525
495 416 633 901	56602725	498 417 629 902	56601575	498 417 630 901	56601575
411 505 584 900	59890300	408 496 596 899	59888750	408 496 596 899	59888750
442 469 601 938	59897850	434 471 597 942	59892775	434 484 584 942	59893950
388 427 639 909	53885725	372 440 623 918	53882550	372 440 623 918	53882550
405 416 661 918	56529800	407 426 649 919	56527525	407 430 645 919	56527525
431 246 791 896	57455200	386 397 687 898	57458625	388 397 684 899	57458725
462 485 598 902	58346300	461 481 597 908	58344500	461 481 597 908	58344500
401 485 573 921	60881800	407 472 577 923	60877925	409 471 577 923	60878125
454 538 523 923	59555400	455 537 526 921	59554850	455 535 527 921	59554900
348 523 626 906	57808325	350 519 625 909	57807275	350 519 625 909	57807275
384 420 615 913	55697250	387 404 630 913	55688850	396 395 630 913	55688725
417 391 629 943	57993250	431 379 632 942	57991100	431 379 632 942	57991100

542 507 574 906	61318600	542 509 573 904	61317550	542 509 573 904	61317550
426 538 569 930	61152425	428 547 562 928	61150400	428 547 562 928	61150400
507 371 655 955	57186250	507 373 657 954	57184025	507 373 657 954	57184025
MEAN TSCC =	**58236066.7**		**58234560**		**58234687.5**
SETTING – A7					
GA		**GSA**		**CSA**	
BEST CHROMOSOME	**TSCC**	**BEST CHROMOSOME**	**TSCC**	**BEST CHROMOSOME**	**TSCC**
4 31 86 100	6286100	4 33 84 100	6286100	3 32 86 100	6286200
0 34 67 130	6307300	0 34 80 117	6307300	0 34 68 129	6307300
0 41 88 100	6666600	0 44 85 100	6666300	0 44 85 100	6666300
2 36 89 100	6405500	0 46 79 102	6404500	1 45 81 100	6404200
3 35 81 102	6246300	1 39 81 100	6245500	1 39 81 100	6245500
4 19 74 128	6148400	0 28 89 108	6145300	0 28 89 108	6145300
0 40 55 132	6405800	1 38 68 120	6405500	1 34 68 124	6405900
0 27 99 100	6786800	0 40 82 104	6785900	0 40 86 100	6785500
0 31 93 101	6087800	1 24 100 100	6087600	0 29 96 100	6087300
3 42 37 149	6166800	0 43 88 100	6160600	0 45 86 100	6160400
0 36 74 113	6333500	0 36 84 103	6332500	0 36 87 100	6332400
2 37 85 100	6316500	1 37 86 100	6316100	0 40 84 100	6316000
0 33 75 113	6585700	0 33 81 107	6585700	0 33 84 104	6585700
2 35 78 102	6940200	0 37 81 99	6939200	0 38 80 99	6939100
3 36 76 114	6327100	2 38 89 100	6325100	2 38 89 100	6325100
0 52 64 115	6450700	1 48 82 100	6450100	1 49 81 100	6450000
0 45 79 103	6046900	4 43 80 100	6046000	4 43 80 100	6046000
2 40 35 151	6477600	1 40 25 162	6477400	1 39 33 155	6477400
0 30 75 122	6325100	0 32 82 113	6324900	0 32 73 122	6324900
0 34 36 157	6547500	1 33 85 108	6542500	1 33 93 100	6541700
3 25 77 113	6765400	2 34 82 100	6762800	2 34 82 100	6762800
3 41 52 133	6223300	3 43 83 100	6219800	3 43 83 100	6219800
0 20 93 117	6471600	0 40 90 100	6467900	0 43 87 100	6467600
2 32 42 150	6223000	1 32 29 164	6222500	1 32 62 131	6222500
0 28 96 101	6473600	0 32 79 114	6474500	0 32 93 100	6473100
3 40 40 142	5838600	1 39 86 99	5834200	0 43 83 99	5834000
2 28 93 100	6122700	1 31 70 121	6124400	1 31 91 100	6122300

0 31 99 105	6391400	0 50 72 113	6390900	0 50 56 129	6390900
3 36 83 101	6613200	3 34 86 100	6613100	3 38 82 100	6613100
5 38 30 157	6615800	2 40 88 100	6608700	2 40 88 100	6608700
MEAN TSCC =	**6386560**		**6385097**		**6384900**
SETTING – A8					
GA		**GSA**		**CSA**	
BEST CHROMOSOME	**TSCC**	**BEST CHROMOSOME**	**TSCC**	**BEST CHROMOSOME**	**TSCC**
11 316 853 1041	65801700	1 284 943 996	65794000	1 349 879 996	65787500
11 323 795 1020	63382700	1 333 667 1144	63382600	4 330 815 996	63379700
0 245 660 1305	65685400	4 349 864 993	65643000	4 348 863 994	65643200
34 386 813 1053	66980500	23 410 853 998	66971500	22 411 852 998	66971600
10 342 901 1004	65194700	1 363 900 993	65190600	2 363 899 993	65190700
30 405 755 1048	60315500	34 423 785 995	60307300	34 423 785 995	60307300
22 309 866 1044	64058400	10 340 769 1122	64054100	10 340 732 1159	64054100
10 344 793 1090	65450000	1 437 800 997	65430500	2 436 801 997	65430400
17 274 776 1151	62496000	9 340 828 1042	62488700	9 339 380 1490	62488700
26 348 505 1449	61820300	12 353 966 997	61770500	12 353 966 997	61770500
28 246 945 1127	61563800	14 462 877 995	61521100	13 463 878 995	61521100
0 246 993 1037	62705400	1 463 817 998	62677500	1 500 780 998	62673800
19 334 843 1002	62483300	26 327 846 999	62482300	26 327 846 999	62482300
9 448 769 1038	66081000	0 328 943 996	66085200	0 473 799 996	66070700
21 327 861 995	62506900	14 339 854 998	62504100	16 337 854 998	62504500
49 270 896 1051	62379700	14 302 951 997	62366700	14 302 951 997	62366700
8 307 834 1071	66100300	0 321 806 1095	66098300	0 321 826 1075	66098300
20 263 1008 997	65419900	7 423 862 996	65402500	7 423 865 996	65402500
13 205 844 1205	57924400	11 411 847 999	57881800	11 411 847 999	57881800
16 329 908 996	61656800	0 348 900 996	61650000	0 348 900 996	61650000
9 411 798 1008	61986300	7 419 784 1016	61985300	8 418 790 1010	61985500

26 141 960 1139	64871300	8 398 862 998	64829700	8 398 862 998	64829700
0 296 831 1100	65464000	1 292 667 1265	65463700	1 308 631 1285	65462100
35 361 878 1004	66686900	35 440 808 996	66685800	35 366 882 996	66685800
13 408 525 1329	63170300	0 426 751 1102	63144500	0 426 853 1000	63134300
24 308 884 1002	60092100	11 388 823 1000	60082600	11 388 823 1000	60082600
10 239 528 1466	62595500	1 328 883 1030	62539700	1 328 914 999	62536600
21 253 1024 999	68070800	12 401 891 997	68054000	13 400 891 997	68054200
14 277 815 1178	67477300	4 364 875 1042	67467400	4 364 481 1436	67467400
24 370 594 1338	62192100	18 383 922 999	62154500	18 383 922 999	62154500
MEAN TSCC =	**63753777**		**63736983**		**63735603**

Here in our study case, the performance of GA, CSA and GGSA were analyzed for equal number of generations. All results are taken for the given number of generations. The analysis is conducted by comparing the mean TSCC values of 30 demand sets in each supply chain setting, for all three algorithms. On analyzing the results we came to the knowledge that, in all supply chain settings CSA and GGSA performed better than GA. For setting A2 and setting A3, GSA performed better than CSA and hence the best. For all other settings, CSA was better and gave minimum value of Mean TSCC. All results are satisfactory and we got the results as expected.

CONCLUSION

The goal of the project was to optimize the inventory in serial supply chain using different algorithms. We were able to successfully introduce two algorithms, Crow Search Algorithm and Adaptive gbest- Gravitational Search Algorithm along with Genetic Algorithm into the supply chain. Our prime objective of the project was to fix the different parameters for the best value, by comparing the obtained results for each parameter of CSA and GGSA and the values are fixed based on minimum TSCC obtained. We were able to achieve our prime objective successfully. Our next objective was to compare the Total Supply Chain Costs obtained from the three algorithms for different supply chain settings and demand sets and to plot the results. We choose 8 supply chain settings for the comparison. Thus we were able to complete all the objectives of the project and successfully compare and analyze the results.

REFERENCES

1) G.R. Harik ; F.G. Lobo ; D.E. Goldberg ; The compact genetic algorithm ,Publisher: IEEE.

2) Ujjwal Maulik, Sanghamitra Bandyopadhyay, Presented by Hu Shu-chiung (2004.05.27) ; Genetic algorithm-based clustering technique.

3) Ahmad Sayed Saif-Eddine Amin Kamel El-Kharbotly (2018.09.002) ; An improved genetic algorithm for optimizing total supply chain cost in inventory location routing problem.

4) UmutTosun , AhmetCosar (2013) ;A New Parallel Genetic Algorithm for Reducing the Bullwhip Effect in an Automotive Supply Chain.

5) N.Jawahara, A.N Balajib , European Journal of Operational Research Volume 194, Issue 2, 16 April 2009, Pages 496-537,O.R. Applications; A genetic algorithm for the two-stage supply chain distribution problem associated with a fixed charge

6) J. Sudhir Ryan Daniel , Chandrasekharan Rajendran (January 2005) ; A simulation-based genetic algorithm for inventory optimization in a serial supply chain.

7) Alireza Askarzadeh (2016) ; A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm ; Department of Energy Management and Optimization, Institute of Science and High Technology and Environmental Sciences, Graduate University of Advanced Technology, Kerman, Iran

8)Mert SinanTurgut Deniz TürselEliiyi (May 2020); Island-based Crow Search Algorithm for solving optimal control problems ; Applied Soft Computing Volume 90, 106170

9) Farid Mohammadi Hamdi Abdi (October 2018) ; A modified crow search algorithm (MCSA) for solving economic load dispatch problem. Applied Soft Computing Volume 71, Pages 51-65

10) Shalini Shekhawat, Akash Saxena (6 September 2019) ; Development and applications of an intelligent crow search algorithm based on opposition based learning.

11) Hoda Zamania Amir H.Gandomi (December 2019) ; CCSA: Conscious Neighborhood-based Crow Search Algorithm for Solving Global Optimization Problems ; Applied Soft Computing Volume 85, 105583.

12) Norlina Mohd Sabri, Mazidah Puteh, Mohamad Rusop Mahmood; An overview of Gravitational Search Algorithm utilization in optimization problems.

13) Esmat Rashedi, Hossein Nezamabadi-pour, Saeid Saryazdi(8 March 2009); GSA: A Gravitational Search Algorithm. Information Sciences.

14) Jianhua Jiang, Keqin Li (15 April 2020); SCGSA: A sine chaotic gravitational search algorithm for continuous optimization problems Expert Systems with Applications, Volume 144 **.**

15) Mohamed Atef Mos(May 2020); A novel hybrid particle swarm optimization and gravitational search algorithm for multi-objective optimization of text mining, Applied Soft Computing, Volume 90.

16) **Osman Özkaraca, Ali Keçebaş (1 April 2019);** Performance analysis and optimization for maximum exergy efficiency of a geothermal power plant using gravitational search algorithm- Energy Conversion and Management, Volume 185.

17) Seyedali Mirjalili, Andrew Lewis (May 2016); The Whale Optimization Algorithm- Advances in Engineering Software, Volume 95.

18) Yan Jiang, Xianing Wu (1 October 2007); Particle swarm optimization: An improved particle swarm optimization algorithm-, Applied Mathematics and Computation, Volume 193, Issue 1.

19) Christian Blum (December 2005); Ant colony optimization: Introduction and recent trends- Physics of Life Reviews, Volume 2, Issue 4.

20) P. Dinakar, Prasad Reddy, T.Gowri Manohar (May 2016); Application of flower pollination algorithm for optimal placement and sizing of distributed generation in Distribution systems-, Journal of Electrical Systems and Information Technology, Volume 3, Issue 1.

21) Weifeng Gao, Lingling Huang (May 2012); A global best artificial bee colony algorithm for global optimization-, Journal of Computational and Applied Mathematics, Volume 236, Issue 11.

22) Qingyun Duan, Vijai K.Gupta (15 June 1994); Schuffled complex evolution algorithm:Optimal use of the SCEUA global optimization method for calibrating watershed models, Journal of Hydrology, Volume 158, Issues 3–4.

23) D.G.Mayer, A.A.Archer (March 2005); Differential evolution – an easy and efficient evolutionary algorithm for model optimization-Agricultural Systems, Volume 83, Issue 3.

24) Rita Greco, Ivo Vanzi (February 2019); New few parameters differential evolution algorithm with application to structural identification- Journal of Traffic and Transportation Engineering (English Edition), Volume 6, Issue 1.