JOURNAL OF MODERN MANUFACTURING SYSTEMS AND TECHNOLOGY

Homepage: http://journal.ump.edu.my/jmmst
ISSN (Online): 2636-9575

An implementation of differential evolution algorithm for a single product and single period multi-echelon supply chain network model

Ayda Emdadian[1], S. G. Ponnambalam[2]*, G. Kanagaraj[3+]

*Correspondence
sgponnambalam@ump.edu.my
+This author contributes equally to this work

[1] School of Engineering
Monash University Malaysia
46150 Bandar Sunway, Selangor, Malaysia
ayda1369e@yahoo.com
[2] Faculty of Manufacturing Engineering,University Malaysia Pahang,26600 Pekan, Pahang, Malaysia
[3]Department of Mechatronics Engineering, Thiagarajar College of Engineering, Madurai, India
gkmech@tce.edu

Articles Info:
Received **11 June 2018**
Received in revised form **7 August 2018**
Accepted **11 Sept 2018**
Available Online **13 Sept 2018**

Keywords:
Multi-echelon supply chain architecture
Evolutionary approach
Differential Evolution algorithms
Swarm intelligence

ABSTRACT

In this paper, five variants of Differential Evolution (DE) algorithms are proposed to solve the multi-echelon supply chain network optimization problem. Supply chain network composed of different stages which involves products, services and information flow between suppliers and customers, is a value-added chain that provides customers products with the quickest delivery and the most competitive price. Hence, there is a need to optimize the supply chain by finding the optimum configuration of the network in order to get a good compromise between several objectives. The supply chain problem utilized in this study is taken from literature which incorporates demand, capacity, raw-material availability, and sequencing constraints in order to maximize total profitability. The performance of DE variants has been investigated by solving three stage multi-echelon supply chain network optimization problems for twenty demand scenarios with each supply chain settings. The objective is to find the optimal alignment of procurement, production, and distribution while aiming towards maximizing profit. The results show that the proposed DE algorithm is able to achieve better performance on a set of supply chain problem with different scenarios those obtained by well-known classical GA and PSO.

INTRODUCTION

In today's global competition in manufacturing and distribution, the successes of an industry are depended on cost-effective supply chain management under various markets, logistics and production uncertainties. Uncertainties in the supply chain usually decrease profit, i.e. increase total supply chain cost [1]. The key issues in supply chain management can broadly be divided into main categories: (i) supply chain design (ii) supply chain planning and (iii) supply chain control. In a supply chain, the flow of goods between a supplier and customer passes through several stages, and each stage may consist of many facilities thus making it difficult to manage its whole integration. One of these critical decisions may involve integrating the supply chain in which the companies participate to enable the ability to make business decisions jointly. These strategic decisions lead to costly, time-consuming investment as the facilities located today, are expected to remain in operation for a long time. Hence, it is very important to design an efficient supply chain to facilitate the movements of goods to increase the competitiveness introduced by the market globalization [2].

A supply chain network (SCN) typically comprised of suppliers, producers, distributors, and customers. SCN establish the distribution channels, and the number of materials and items to consume, produce, and ship from suppliers to customers. The SCN design problem is one of the most comprehensive strategic decision problems that need to be optimized for long-term efficient operation of a whole supply chain. It determines the number, location, capacity and type of plants, warehouses, and distribution centers to be used. These problems typically increase as the number of supply chain layers increase, the time period increases, and the number of products and purchase orders increase. These cause the network search space and time required to obtain a solution to increase markedly. Therefore, the SCN design problem is an NP-complete problem [3]. According to the findings of Ebikake et al [4], finding the best solution for supply chain management is NP-hard problem, so it must be strategically dealt with, on a case to case basis developing an efficient methodology that can find out the optimal or near-optimal solution in minimum computational time. It is intractable to solve this kind of problem in the real world as it often incurs expensive computational efforts.

Meta-heuristics are kind of near-optimal algorithms that were proposed in the two recent decades to integrate basic heuristic methods in higher-level structures in order to effectively and efficiently search a solution space. Nowadays, these algorithms have a large number of applications in an optimization of different hard-to-solve problems. In this paper, we use DE variants for solving constrained multi-echelon supply chain network problems of Kadadevaramath [1]. The performance of the DE variants used in this paper has been compared with genetic algorithm and particle swarm optimization algorithm. The results indicated that the DE method can obtain a better quality solution compared to classical GA and PSO.

Rest of the paper is organized as follows. Section 2 deals with the work that is done previously in the related field. Section 3 explains problem description and mathematical formulation of three stage multi-echelon supply chain network model. The implementation of DE algorithm is given in section 4 followed by numerical illustration given in section 5. The results are given in section 6 and finally, conclusions are given in section 7.

LITERATURE REVIEW

A large amount of literature on supply management places great emphasis on the integration of different components of the chain. Most of the research in this area is based on the classic work of Clark and Scarf [5-6] more discussion of two-echelon models may be found in [7]. Bora and Grossmann [8] formulated the problem as a multistage stochastic program with decision dependent elements where investment strategies are considered to reduce uncertainty, and time-varying distributions are used to describe uncertainty and proposed a new mixed-integer/disjunctive programming model.

Cohen and Moon [9] extend Cohen and Lee [10] developed a constrained optimization model; called PILOT, to investigate the effects of various parameters on supply chain cost, and consider the additional problem of determining which manufacturing facilities and distribution centers should be open. The objective function of the PILOT model is a cost function, consisting of fixed and variable production and transportation costs, subject to supply, capacity, assignment, demand, and raw material requirement constraints. Based on the results of their example supply chain system, the authors conclude that there are a number of factors that may dominate supply chain costs under a variety of situations and that transportation costs play a significant role in the overall costs of supply chain operations.

Goh et al [11] focused on the operational issues of a Two-echelon Single-Vendor–Multiple-Buyers Supply chain problem under vendor managed inventory mode of operation and proposed PSO and hybrid GA to solve this problem. Che [12] developed a decision methodology for the production and distribution planning of a multi-echelon unbalanced supply chain problem. He proposed a mathematical model to determine the best pattern of the supply chain system by integrating cost and time criteria and simultaneously considering multiple products, production loss, transportation loss, quantity discount, production capacity, and starting-operation quantity.

Ishii et al. [13] develop a deterministic model for determining the base stock levels and lead times associated with the lowest cost solution for an integrated supply chain on a finite horizon. The stock levels and lead times are determined in such a way as to prevent stock out, and to minimize the amount of obsolete ("dead") inventory at each stock point. Their model utilizes a pull-type ordering system, which is driven by, in this case, linear (and known) demand processes.

Pyke and Cohen [14] develop a mathematical programming model for an integrated supply chain, using stochastic sub-models to calculate the values of the included random variables included in the mathematical program. The authors consider a -level supply chain, consisting of one product, one manufacturing facility, one warehousing facility, and one retailer. The model minimizes total cost, subject to a service level constraint, and holds the setup times, processing times, and replenishment lead times constant. The model yields the approximate economic (minimum cost) reorder interval, replenishment batch sizes, and the order-up-to product levels (for the retailer) for a particular production network.

Work of Babu and Gujarathi [15] focused on solving three stage supply chain problems using multi-objective differential evolution (MODE) algorithm. In their study, three cases of objective functions were considered and Pareto-optimal solutions were obtained for each case. The results were compared with those reported using a non-dominated sorting genetic algorithm (NSGA-II) in the literature. Minimizing total cost has been the primitive objective in most of the SCN design models [16-17]. But for a supply chain, producing products at minimum cost is not the only objective, a satisfying customer is also equally important. Later some researchers started incorporating more than one competing objectives such as improving customer service and reducing cost in their models.

Many of the aforementioned articles use Lagrangian relaxations or heuristic methods to solve the model. Recently, Atamtürk et al. [18] have shown how to formulate different variants of the joint inventory-location problems in a supply chain comprising of a distribution center and retailers as conic mixed-integer problems. Mehrdad et al [19] extend the work by providing a novel conic integer reformulation for a joint inventory-location problem in a four echelon supply chain. Nowadays, as the use of a computer is rapidly increasing, many evolutionary computation methods for solving optimization problems have been introduced. Probably, among them, Differential Evolution (DE) is the most well-known class of evolutionary algorithms. It has taken a lot of attention of researchers in the several years. In this paper, the performance of five DE variants are examined for solving constrained multi echelon supply chain network problems of Kadadevaramath [1].

MATHEMATICAL MODEL FORMULATION FOR THREE STAGE SINGLE PRODUCT, SINGLE PERIOD SUPPLY CHAIN NETWORK

This section develops a mathematical model to quantify the relationship among all the decision variables involved in three stage multi-echelon supply chain network. Problem description, the notations, assumptions, decision variables used in this formulation are given below.

PROBLEM DESCRIPTION

A three echelon two-stage supply chain network considered in this paper is as shown in Fig.1. The first level consists of three suppliers which are suppliers of raw materials to plants for the manufacturing of products. The second level consists of two plants where products are manufactured and shipped to distribution centers. The third level consists of six distribution centers where products are sold to retailers. A product is manufactured from three different components can be supplied from any supplier to any of two plants. Plants may produce any product limited by its production and delivery capacity or decided by its strategy for each product. The final products are shipped to distribution centers based on demands. For the given cost data set the problem is to find the optimal alignment of procurement, production, and distribution while aiming at maximizing profit throughout the supply chain.

Assumptions

- A single product (made up of three components) flows through the supply chain network
- Distribution centers face random customer demand and demand distribution is assumed to be uniform
- The Quantity of goods at any installation takes integer values
- Linear holding cost rates exist only for manufacturing plants in the supply chain
- Shortages are not permitted (no shortage cost)
- Transportation costs are directly proportional to the quantity shipped
- Manufacturing costs are directly proportional to the number of products produced
- All installations have a finite capacity

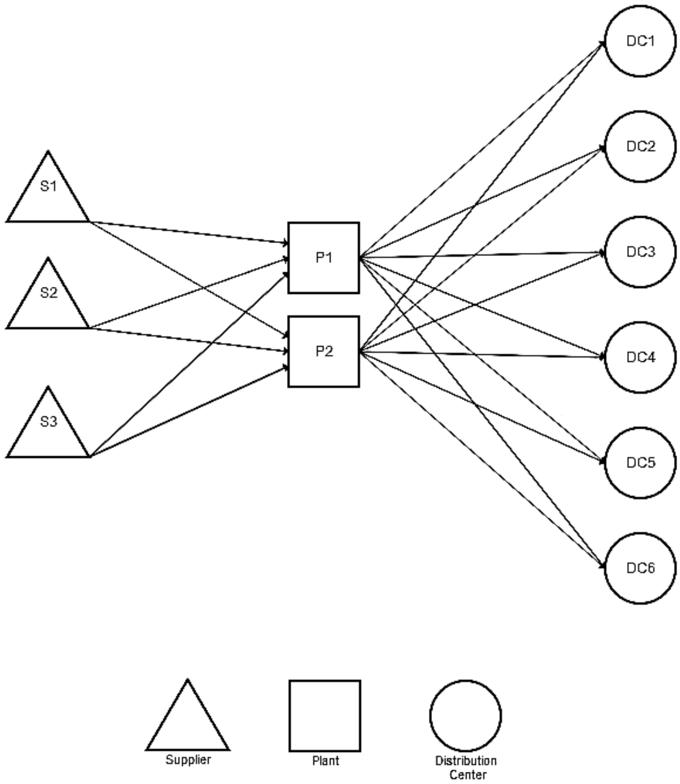

Figure 1. Three echelon supply chain network

Sets

Let C be the set of components $c = 1,2,3, \dots \dots . . C$
Let S be the set of suppliers $s = 1,2,3, \dots \dots . . S$
Let P be the set of plants $p = 1,2,3, \dots \dots . . P$
Let D be the set of distribution centers $d = 1,2,3, \dots \dots . . D$

Parameters

$L_{c,s}$ be the capacity of supplier s for component c
$CS_{c,s}$ be the cost of making a component c by supplier s
$STC_{c,s,p}$ be the transportation cost of a component c from supplier s to plant p /unit
U_p be the capacity of plant p
MC_p be the manufacturing cost of plant p /unit
IC_p be the inventory cost at plant p /unit/period
$PTC_{p.d}$ be the plant transportation cost from plant p to distribution centre d
D_d be the demand at distribution centre d
SP_d be the selling price at distribution centre d /unit

Decision variables

$X_{c,s,p}$ be the amount of component shipped c from supplier s to plant p
$Y_{p,d}$ be the amount of product shipped from plant p to distribution centre d

Mathematical model

The objective function aims at maximizing the profit of overall supply chain distribution network which implies maximize the revenue and minimize the total supply chain cost

Maximize Profit = Revenue – Total supply chain cost (TSCC)
Maximize Revenue $=\sum_d (D_d \times SP_d)$
Minimize TSCC = TSMC+TMC+TTC

Three components of total supply chain cost (TSCC) are:

(a)Total supplier material cost (TSMC) = $\sum_c \sum_s \sum_p (CSc,s \ \times Xc,s,p)$

(b)Total manufacturing cost (TMC) = $\sum_p \{(MCp) \times (\sum_d Yp,d)\} + \sum_p \{(ICp) \times (\sum_d Ic,p)\}$

(c)Total transport cost (TTC) = $\sum_c \sum_s \sum_p (Xc,s,p \ \times STCc,s,p) + \sum_p \sum_d (Yp,d \ \times PTCp,d)$

The objective function is given by

$$\text{Maximize Profit} = \sum_d (D_d \ \times SP_d) - \{\sum_c \sum_s \sum_p (CSc,s \ \times Xc,s,p) + \sum_p \{(MCp) \times (\sum_d Yp,d)\} + \sum_p \{(ICp) \times (\sum_d Ic,p)\} + \sum_c \sum_s \sum_p (Xc,s,p \ \times STCc,s,p) + \sum_p \sum_d (Yp,d \ \times PTCp,d)\} \quad (1)$$

Subject to

Supplier capacity constraint

$$\sum_P Xc,s,p \ \leq Lc,s \quad \forall\, c,s \quad (2)$$

Plant capacity constraint

$$\sum_P Yp,d \ \leq Up \quad \forall\, p \quad (3)$$

Demand constraint

$$\sum_P Yp,d \ = Dd \quad \forall\, d \quad (4)$$

Inventory balancing constraint

$$\sum_v Xc,s,p - \ 3\sum_d Yp,d = 0 \quad \forall\, c,s \quad (5)$$

The objective function (1) maximizes the profit by minimizing the overall supply chain operating cost. Constraint (2) ensures that the total amount of any of the three components shipped from a supplier cannot exceed the supplier's capacity for that component. Constraint (3) specifies that the total production quantities do not exceed plant capacities individually. Constraint (4) ensures the products shifted from plant to distribution centers should be more than or equal to the demand raised by the distribution centers. Constraint (5) ensures that the components moved from the suppliers should be more than the products to be manufactured to meet the required demand.

IMPLEMENTATION OF DIFFERENTIAL ALGORITHM

Differential evolution (DE) was first proposed by Storn and Price in 1995 [20] as a powerful heuristic method for solving non-linear, non-differentiable and multimodal optimization problems. This technique has been structured based upon a combination of simple arithmetic operators, the classical crossover, mutation, and selection operators. In this method, the purpose of mutation and crossover is to generate new vectors, and the vectors will survive to the next generation are determined upon selection operator [21].

The theoretical framework of DE is very simple and DE is computationally inexpensive in terms of memory requirements and CPU times. Thus, nowadays, DE has gained much attention and wide application in a variety of fields [22-23]. Due to its simplicity, easy implementation, fast convergence, and robustness, the programming and operation of DE are also quite easy because it requires the settings of only three control parameters: population size, scaling factor, and crossover constant rate in crossover operator. These advantages facilitate the wide usage of DE. DE is a population-based search technique which utilizes NP variables as a population of D dimensional parameter vectors for each generation. Four main steps are involved in DE known as, initialization, mutation, recombination, and selection. DE produces an initial population by randomly sampling several points (each point is called a target vector) from the search space.

$$\underset{x\ i,0}{\rightarrow} = (x_{i,1,0}, x_{i,2,0}, \ldots\ldots, x_{i,D,0}), \qquad i = 1,2, \ldots\ldots, NP$$

Where NP denotes the population size and D denotes the number of variables. At each generation G, a mutant vector $\underset{v\ i,0}{\rightarrow} = (v_{i,1,G}, v_{i,2,G}, \ldots\ldots, v_{i,D,G})(i \in 1,2, \ldots\ldots, NP)$ is produced by the mutation operator for each target vector $\underset{x\ i,G}{\rightarrow}$. Afterward, the crossover operator is implemented on the mutant vector and the target vector to generate a trial vector $\underset{u\ i,G}{\rightarrow} = (u_{i,1,G}, u_{i,2,G}, \ldots\ldots, u_{i,D,G})(i \in 1,2, \ldots\ldots, NP)$. The crossover operator and the mutation operator together are called a trial vector generation strategy. In the selection phase, the function value of the trial vector is compared to the function value of the target vector and the target vector for the next generation is updated using equation (11). If the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector with which it was compared. In addition, the best parameter vector is evaluated for every generation in order to keep track of the progress that is made during the optimization process.

The DE algorithm adopted in this paper is presented below,
Initialize variable: NP = 30, D = 30, F = 0.5, C_R = 0.9, g = 1000
Initialize randomly generated population of Np target vector (X_{ig}), each with length D
While g ≤ 1000
    // Mutation operation to generate donor vector (V_{ig}) based on random selected target vector
    for i = 1 to NP
        //Select different mutation relations depending upon the variant. For Example, DE/Rand/E is included in this algorithm.
    Randomly choose 3 distinct vectors $X_{r1,g}$, $X_{r2,g}$, $X_{r3,g}$
    $V_{i,g} = X_{ip,g} + F * (X_{iq,g} – X_{ir,g})$
    end
    // Crossover operation to generate trial vectors ($y^{j}_{i,k}$), consisting of both target and donor elements
    for i = 1 to Np
    Generate a randomly distributed number R_i (0,1)
        for j = 1 to D
            if $R_i \le C_R$
            $y^{j}_{i,k} = V_{i,j,g}$
            else
            $y^{j}_{i,k} = X_{i,j,g}$
            end
        end
    end
    // Evaluation of target and trial vectors
    // Selection operation to detect target vector for next generation ($X_{i,g+1}$)
    for i = 1 to Np
        if $f(y^{j}_{i,k}) > f(X_{i,g})$
        $X_{i,g+1} = y^{j}_{i,k}$
        else
        $X_{i,g+1} = X_{i,g}$
    end
    g = g+1;
end

Mutation operation

The mutation operation is based on the difference of different individuals (solutions), to produce a mutant vector $V_{i,G}$ with respect to each individual $X_{i,G}$, in the current population. This main operation is founded on the differences between randomly sampled pairs of solutions in the population. For each target vector $X_{i,G}$, i =1, 2,…, NP, a mutant vector $V_{i,G}$ can be made by the following mutation operators. In all types, the scale factor F is a positive control parameter for scaling the difference vector. Mutation is carried out by the mutation scheme. For each vector xi at any time or generation g, three randomly chosen vectors X_p, X_q and X_r, and then resulting donor vector generated by any one of the following mutation strategies.

Strategy 1: DE/Rand/1/Bin

$$V_i^{g+1} = X_p^g + F(X_q^g - X_r^g) \tag{6}$$

Strategy 2: DE/Best/1/Bin

$$v_i^{g+1} = X_{best}^g + F(X_q^g - X_r^g) \tag{7}$$

Strategy 3: DE/Rand-to-Best/1/bin

$$V_i^{g+1} = X_p^g + F\left(X_{best}^g - X_p^g\right) + F\left(X_q^g - X_r^g\right) \tag{8}$$

Strategy 4: DE/Rand/2/bin

$$V_i^{g+1} = X_p^g + F\left(X_q^g - X_r^g\right) + F\left(X_s^g - X_t^g\right) \tag{9}$$

Strategy 5: DE/Current-to-Best/1/bin.

$$V_i^{g+1} = X_p^g + F\left(X_{best}^g - X_p^g\right) + F\left(X_q^g - X_r^g\right) \quad (10)$$

Here, F is the scale factor used to scale differential vector. X best is the solution with the best fitness value in the current population.

Crossover operation

In order to increase the diversity of the perturbed parameter vectors, a crossover is introduced after the mutation operation. Crossover operation is employed to generate a temporary or trial vector by replacing certain parameters of the target vector by the corresponding parameters of a randomly generated donor vector. To get each individual's trial vector, $U_{i,,G+1,}$ crossover operation is performed between each individual and its corresponding mutant vector. The crossover operator is applied to obtain the trial vector $y_{i,k}^{j}$ from $V_{i,g}$ and $X_{i,g}$. The crossover is defined by

$$y_{i,k}^{j} = \begin{cases} V_{i,g} \text{ if } R^j \leq C_R \text{ or } j = i \\ X_{i,g} \text{ if } R^j > C_R \text{ and } j \neq i \end{cases} \quad (11)$$

where i is a randomly chosen integer in the set$i$, i.e., $i \in I$ = {1,2,..., D}; the superscript j represents the j-th component of respective vectors;$R^j \in (0,1)$, drawn randomly for each j. The ultimate aim of the crossover rule is to obtain the trial vector $y_{i,k}^{j}$ with components coming from the components of the target vector $X_{i,g}$ and the mutated vector$V_{i,g}$. This is ensured by introducingC_R and the set I. Notice that for C_R = 1 the trial vector $y_{i,k}^{j}$ is the replica of the donor vector $V_{i,g}$. The targeting process (mutation and crossover) continues until all members of $X_{i,g}$ are considered.C_R assumed to be 0.9.

Selection operation

The selection operator of DE adopts a one-to-one competition between the target vector $X_{i,g}$ and the trial vector_$U_{i,g}$. If the objective function value of the trial vector is less than or equal to that of the target vector, then the trial vector will survive into the next generation, otherwise, the target vector will enter the next generation: To generate the new individual for the next generation, selection operation is performed based on equation (12) between each individual and its corresponding trial vector.

$$X_{i,g+1} = \begin{cases} y_{i,k}^{j}, if\ f(\ y_{i,k}^{j}) > f(X_{i,g}) \\ X_{i,g}\ , if\ f(\ y_{i,k}^{j}) \leq f(X_{i,g}) \end{cases} \quad (12)$$

Termination criteria

DE algorithm will give the final objective value after the number of iteration has been chosen for a problem. In this study, the number of iteration assumed as 1000 generation and results published in the next section are based on 1000 generation.

NUMERICAL ILLUSTRATION

The DE starts the search by generating a population of candidate solutions. In our implementation, this population is randomly generated according to uniform distributions. That is, the parameters (gene values) *Ri* is randomly generated according to uniform distributions U [R_{imin},R_{imax}] where R_{imin} and R_{imax} are the minimum and the maximum possible values of *Ri*. In DE algorithm, a solution can be represented as a vector of decision variables. The number of vectors representing the SCN is called population size (NP). In this study, each vector consists of 30 variables in a population size of 30.

Fig. 2 shows representation of a solution vector consists of two sets of decision variables X_{cvp} and Y_{Pd}. X_{csp} represents the number of component 'c' shipped from supplier 's' to plant 'p', and Y_{Pd} represent the number of product shipped from plant 'p' to distribution centre 'd'. For example, X_{232} represent the number of component 2 shipped from supplier 3 to plant 2.

Component 1 from s_1, s_2, s_3						Component 2 from s_1, s_2, s_3						Component 2 from s_1, s_2, s_3					
X_{111}	X_{112}	X_{121}	X_{122}	X_{131}	X_{132}	X_{211}	X_{212}	X_{221}	X_{222}	X_{231}	X_{232}	X_{311}	X_{312}	X_{321}	X_{322}	X_{331}	X_{332}
137	47	75	72	73	65	115	19	36	82	114	83	126	160	42	20	113	4

a) Xcsp

Product P_1 to all DCs						Product P_2 to all DCs					
Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}	Y_{16}	Y_{21}	Y_{22}	Y_{23}	Y_{24}	Y_{25}	Y_{26}
39	68	32	29	55	62	43	30	23	28	34	26

b) Y_{Pd}

Fig. 2 Vector representation in DE algorithm for multi stage SCN architecture

Step 1: Initialization

Assume a population of five vectors, which are generated randomly and provided in Table 1.

Table 1, Target vectors representing the initial population

173	19	155	43	168	1	76	40	40	203	20	166	207	68	104	34	107	64	34	43	65	2	72	10	62	52	2	49	2	45
5	191	229	2	65	232	34	56	161	127	8	133	287	71	0	126	136	110	28	40	8	25	52	13	68	55	59	26	22	42
78	54	45	119	112	39	65	13	246	4	59	112	228	24	36	102	57	144	27	48	51	39	43	41	69	47	16	12	31	14
10	161	135	115	22	176	62	46	313	78	89	18	77	173	46	20	76	145	78	38	67	4	24	28	18	57	0	47	50	27
110	74	62	132	13	208	20	103	167	62	39	114	342	13	62	57	152	63	51	86	36	22	40	39	45	9	31	29	34	16

Step 2: Mutation

Once the initial population is generated, mutation operation is done to generate donor vectors. As explained in section 4.1, five different mutations are used to have five variants of DE. Among the five variants, DE/Rand/1/Bin is performing better. Hence, this section explains the implementation of DE/Rand/1/Bin mutation strategy using equation 6.

First, randomly select three vectors from the population. For example, vectors corresponding to row 5, 1 and 3 are selected randomly from Table 1 and F is assumed as 0.5. The selected vectors are given in Table 2.

Table 2, Vectors undergo mutation

Xp	5	110	74	62	132	13	208	20	103	167	62	39	114	342	13	62	57	152	63	51	86	36	22	40	39	45	9	31	29	34	16
Xq	1	173	19	155	43	168	1	76	40	40	203	20	166	207	68	104	34	107	64	34	43	65	2	72	10	62	52	2	49	2	45
Xr	3	78	54	45	119	112	39	65	13	246	4	59	112	228	24	36	102	57	144	27	48	51	39	43	41	69	47	16	12	31	14

Table 3, Donor vector after mutation

157.5	56.5	117	94	41	189	25.5	116.5	64	161.5	19.5	141	331.5	35	96	23	177	23	54.5	83.5	43	3.5	54.5	23.5	41.5	11.5	24	47.5	19.5	31.5

Table 4, Corrected Donor vector

158	57	117	94	41	189	26	117	64	162	20	141	332	35	96	23	177	23	55	84	43	4	55	24	42	12	24	48	20	32

The generated donor vector ($V_{i,g}$) using DE/Rand/1/Bin mutation strategy is given in Table 3. Since the vectors should only contain integer and positive values, each negative value should be replaced by absolute value and non-integer values should be replaced by integer using round function. The corrected donor vector is given in Table 4.

Step 3: Crossover

A trial vector is generated using a crossover operator. The binominal crossover is used in this paper. In this stage, the trial vector ($y^{j}_{i,k}$) is calculated by combining donor vector ($V_{i,g}$) and target vector ($X_{i,g}$) based on equation (11). C_R represents the crossover probability. R is a random number between 0 and 1, if R is smaller than C_R the variable of mutation vector will be selected as a variable of the trial vector. Otherwise, the variable of a target vector will be selected as the variable of a trial vector. C_R assumed to be 0.9. For example, for the first position of trial vector; if R = 0.1669 the donor vector value, which is 158 copied into trail vector in the first position. For the second position of a trial vector; if R = 0.9351, then target vector value, which is 19 copied into trail vector in the second position similar way the whole trial vectors are generated and are shown in Table 5.

Table 5, Trial vector after crossover

158	57	117	94	41	189	26	117	64	162	20	141	332	35	96	23	177	23	55	84	43	4	55	24	42	12	24	48	20	32
160	17	55	83	185	49	72	10	110	67	25	186	293	29	65	109	73	104	5	51	50	38	67	32	91	45	17	13	7	23
26	171	144	122	28	261	40	91	274	107	79	19	134	168	59	3	124	105	90	57	60	5	23	27	6	38	8	56	52	28
63	92	7	170	15	227	15	90	270	38	59	87	353	9	28	91	127	103	48	89	29	41	26	55	49	7	38	11	49	1
42	134	182	71	100	73	90	15	250	149	80	44	10	201	67	9	54	146	70	17	82	6	40	14	27	79	15	57	34	42

Step 4: Selection

At this stage, trial vector ($y^{j}_{i,k}$), is compared with the target vector ($X_{i,g}$) to select vectors for the next generation. The selection will be done using equation (12), which is based on objective function value comparison. If the objective function value of a trial vector is better than the value of the target vector, the trial vector will be chosen as a new trial vector ($X_{i,g+1}$) for the next generation. Otherwise, the target vector will be chosen for the next generation. The target vectors for the next generation are shown in Table 6. The mutation, crossover, and selection operations are repeated until the termination condition is satisfied.

Table 6, Target vectors for next generation

173	19	155	43	168	1	76	40	40	203	20	166	207	68	104	34	107	64	34	43	65	2	72	10	62	52	2	49	2	45
160	17	55	83	185	49	72	10	110	67	25	186	293	29	65	109	73	104	5	51	50	38	67	32	91	45	17	13	7	23
78	54	45	119	112	39	65	13	246	4	59	112	228	24	36	102	57	144	27	48	51	39	43	41	69	47	16	12	31	14
10	161	135	115	22	176	62	46	313	78	89	18	77	173	46	20	76	145	78	38	67	4	24	28	18	57	0	47	50	27
110	74	62	132	13	208	20	103	167	62	39	114	342	13	62	57	152	63	51	86	36	22	40	39	45	9	31	29	34	16

RESULTS AND DISCUSSION

The performance of the DE variants was evaluated by solving the supply chain problems considered by Kadadevaramath et al 2012. The DE parameters used in this paper are: number of generations = 1000, number of vectors per population = 30, mutation scale factor = 0.5 and crossover rate = 0.9. To assess the quality of five DE variants, a series of computational experiments were conducted. The computational experiments were done using a PC with Intel Core 2 Duo, 2.6 GHz and 3GB RAM under Mat lab environment.

For all twenty scenarios, demand rates followed the same as previous research for purpose of result comparison. All the twenty scenarios are considered to evaluate the performance of five DE variants. In order to compare the results of each variant with others, the same termination condition has been used which is based on the number of generation.

Table 7 Optimal profit given by DE variants for all 20 scenarios

Scenario	DE variants				
	DE/Rand/1/bin	DE/Best/1/bin	DE/Rand-to-best/1/bin	DE/Rand /2/bin	DE/Current-to-Best/1/bin
1	693821	672086	676171	671019	675129
2	716444	707093	722314	694190	717853
3	682453	663375	675756	674255	681772
4	668867	649099	675036	657576	665509
5	650618	628437	643188	645505	640557
6	662956	641070	655714	657326	665230
7	675353	649113	663800	673513	667640
8	642256	629189	638198	636307	639562
9	644235	635217	638419	639312	639725
10	620780	584334	610262	614915	614153
11	618662	591943	612987	620885	617209
12	637072	604984	632456	629973	626318
13	693816	688345	683107	686424	689643
14	645156	630811	637692	633012	633621
15	679268	660562	678428	675553	674401
16	588173	579418	580990	583522	570255
17	598663	564032	588137	595366	583811
18	585288	564770	559161	583529	573938
19	581227	565709	577483	573947	561016
20	671595	657652	666832	662277	670745

Table 8. Best Profit given by DE/Rand/1/Bin and GA Profit comparison

Scenario	DE/Rand/1/Bin profit	GA Profit	Difference	Percentage of improvement
1	693821	378179	315642	45.49
2	716444	423343	293101	40.91
3	682453	456036	226417	33.17
4	668867	368648	300219	44.88
5	650618	369655	280963	43.18
6	662956	349633	313323	47.26
7	675353	428033	247320	36.62
8	642256	345236	297020	46.24
9	644235	326946	317289	49.25
10	620780	310455	310325	49.98
11	618662	308154	310508	50.19
12	637072	321455	315617	49.54
13	693816	439865	253951	36.60
14	645156	302345	342811	53.13
15	679268	416633	262635	38.66
16	588173	294967	293206	49.85
17	598663	284643	314020	52.45
18	585288	301764	283524	48.44
19	581227	361047	220180	37.88
20	671595	374955	296640	44.16

The performances of the DE variants are analyzed by the quality of the solution yielded and the computational effort required to obtain the best solution. The profit obtained after running each variant for 1000 generations are reported in Table 7. The results clearly show that among the variants used in this study, the DE/Rand/1/bin gives the best performance within 1000 generations, which means the highest profit with lowest TSCC.

Table 8-10 compares the best profit gave by DE/Rand/1/bin with GA, NLIWD-PSO best profits reported by Cardenas et al 2014 respectively. From Table 8-10 we observe that there is an improvement about 40% compared to GA and there is an improvement about 20% compared to NLIW-PSO and about 10% compared to optimal solution reported by [24]. Decision variables for best profit obtained by DE/Rand/1/bin variant for all 20 scenarios considered in this study are provided in Tables 11. The optimal procurement of component1, component 2, component 3 from supplier 1, supplier 2, supplier 3, and optimal product manufacturing and distribution from plant 1 and plant 2 to all six distribution centers to satisfy the demand obtained by DE/Rand/1/bin variant for all 20 scenarios are listed in Table 12- 14 respectively.

Table 9. Best Profit gave by DE/Rand/1/Bin and NLIWD-PSO Profit comparison

Scenario	DE/Rand/1/Bin profit	NLIWD-PSO Profit	Difference	Percentage of improvement
1	693821	549973	143848	20.73
2	716444	577768	138676	19.36
3	682453	547611	134842	19.76
4	668867	538945	129922	19.42
5	650618	519753	130865	20.11
6	662956	531895	131061	19.77
7	675353	540576	134777	19.96
8	642256	512192	130064	20.25
9	644235	518223	126012	19.56
10	620780	489435	131345	21.16
11	618662	491591	127071	20.54
12	637072	502506	134566	21.12
13	693816	557971	135845	19.58
14	645156	512355	132801	20.58
15	679268	543166	136102	20.04
16	588173	462378	125795	21.39
17	598663	473687	124976	20.88
18	585288	461293	123995	21.19
19	581227	461254	119973	20.64
20	671595	535868	135727	20.21

Table 10. Best Profit gave by DE/Rand/1/Bin and Optimal Profit comparison

Scenario	DE/Rand/1/Bin profit	Optimal Profit	Difference	Percentage of improvement
1	693821	617267	76554	11.03
2	716444	646013	70431	9.83
3	682453	613867	68586	10.05
4	668867	601763	67104	10.03
5	650618	583815	66803	10.27
6	662956	598785	64171	9.68
7	675353	610023	65330	9.67
8	642256	578901	63355	9.86
9	644235	582062	62173	9.65
10	620780	555707	65073	10.48
11	618662	560013	58649	9.48
12	637072	571454	65618	10.30
13	693816	625820	67996	9.80
14	645156	577047	68109	10.56
15	679268	613513	65755	9.68
16	588173	528866	59307	10.08
17	598663	536631	62032	10.36
18	585288	526438	58850	10.05
19	581227	523400	57827	9.95
20	671595	603272	68323	10.17

Table 11. Best values of Decision variables to the 20 scenarios given by DE/Rand/1/Bin

Scenario	X_{111}	X_{112}	X_{121}	X_{122}	X_{131}	X_{132}	X_{211}	X_{212}	X_{221}	X_{222}	X_{231}	X_{232}	X_{311}	X_{312}	X_{321}	X_{322}	X_{331}	X_{332}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}	Y_{16}	Y_{21}	Y_{22}	Y_{23}	Y_{24}	Y_{25}	Y_{26}
1	3	8	4	5	2	13	42	64	347	2	176	38	222	139	120	17	185	65	60	96	22	28	71	88	23	2	33	29	18	2
2	6	5	10	2	3	28	11	64	354	30	124	69	53	323	64	80	240	10	66	36	42	53	67	19	30	25	18	28	25	77
3	11	5	2	14	2	13	84	45	222	164	108	67	343	32	58	39	192	21	63	28	41	77	41	82	25	42	13	5	35	13
4	7	1	4	1	13	28	38	56	328	47	77	127	281	56	81	18	195	21	58	58	62	73	52	38	10	2	33	20	3	44
5	17	0	4	7	9	33	101	36	191	92	146	30	313	34	81	14	193	27	29	53	52	87	52	78	35	26	11	11	4	4
6	4	18	14	5	5	8	8	26	133	182	180	68	393	6	11	106	192	28	56	89	15	73	44	32	5	6	41	16	31	45
7	6	18	5	13	1	2	74	60	229	131	124	109	240	109	87	45	149	33	38	82	19	68	36	56	44	7	71	3	20	26
8	14	12	6	3	3	9	67	76	302	4	111	19	127	151	120	30	103	133	45	71	61	45	1	61	17	3	2	14	97	12
9	10	19	1	15	8	16	87	22	144	38	92	119	391	3	122	10	107	142	68	80	43	39	65	25	5	6	22	39	13	33
10	2	7	1	2	5	0	67	64	307	13	21	81	293	98	139	3	107	88	72	44	63	36	44	51	9	32	6	30	25	7
11	2	6	12	3	9	5	12	101	348	48	133	73	208	28	83	23	182	25	71	35	65	36	50	71	15	23	28	17	11	1
12	10	7	5	2	3	6	71	1	198	113	111	30	161	230	88	50	170	80	55	27	46	45	60	37	41	68	21	6	14	18
13	2	1	6	7	8	57	88	46	217	149	115	41	350	8	42	94	178	12	91	64	61	12	49	57	0	12	11	57	42	16
14	5	1	1	3	3	1	23	104	225	119	155	61	169	140	89	47	203	21	75	62	14	85	12	43	15	2	40	4	67	11
15	2	24	1	8	3	9	62	29	150	196	28	159	334	44	64	80	217	13	78	64	28	14	9	87	2	8	33	71	62	9
16	6	6	8	9	10	3	84	8	145	174	19	15	251	104	128	9	229	13	90	32	29	44	48	46	1	34	35	25	6	10
17	5	2	2	6	2	6	65	45	223	13	62	63	240	143	67	67	183	55	60	75	43	76	5	19	12	10	10	1	52	44
18	6	16	6	1	6	7	16	26	321	7	101	34	343	11	83	15	19	197	42	78	49	53	51	25	18	22	19	5	11	27
19	3	4	0	5	16	5	27	71	96	63	186	50	264	36	28	97	172	55	62	63	15	31	38	55	20	4	43	25	34	2
20	6	2	2	3	2	8	21	102	188	141	157	72	125	222	67	27	212	8	48	69	15	12	69	47	33	12	80	51	12	7

Table 12.Optimal procurement of component 1 from suppliers for DE/Rand/1/bin

Scenario	Plant1 supplier1	Plant1 supplier2	Plant1 supplier3	Plant2 supplier1	Plant2 supplier2	Plant2 supplier3
1	3	8	4	5	2	13
2	6	5	10	2	3	28
3	11	5	2	14	2	13
4	7	1	4	1	13	28
5	17	0	4	7	9	33
6	4	18	14	5	5	8
7	6	18	5	13	1	2
8	14	12	6	3	3	9
9	10	19	1	15	8	16
10	2	7	1	2	5	0
11	2	6	12	3	9	5
12	10	7	5	2	3	6
13	2	1	6	7	8	57
14	5	1	1	3	3	1
15	2	24	1	8	3	9
16	6	6	8	9	10	3
17	5	2	2	6	2	6
18	6	16	6	1	6	7
19	3	4	0	5	16	5
20	6	2	2	3	2	8

Table 13. Optimal procurement of component 2 from suppliers for DE/Rand/1/bin

Scenario	Plant1 supplier1	Plant1 supplier2	Plant1 supplier3	Plant2 supplier1	Plant2 supplier2	Plant2 supplier3
1	42	64	347	2	176	38
2	11	64	354	30	124	69
3	84	45	222	164	108	67
4	38	56	328	47	77	127
5	101	36	191	92	146	30
6	8	26	133	182	180	68
7	74	60	229	131	124	109
8	67	76	302	4	111	19
9	87	22	144	38	92	119
10	67	64	307	13	21	81
11	12	101	348	48	133	73
12	71	1	198	113	111	30
13	88	46	217	149	115	41
14	23	104	225	119	155	61
15	62	29	150	196	28	159
16	84	8	145	174	19	15
17	65	45	223	13	62	63
18	16	26	321	7	101	34
19	27	71	96	63	186	50
20	21	102	188	141	157	72

Table 14. Optimal procurement of component 3 from suppliers for DE/Rand/1/bin

Scenario	Plant 1 Supplier 1	Plant 1 Supplier 2	Plant 1 Supplier 3	Plant 2 Supplier 1	Plant 2 Supplier 2	Plant 2 Supplier 3
1	222	139	120	17	185	65
2	53	323	64	80	240	10
3	343	32	58	39	192	21
4	281	56	81	18	195	21
5	313	34	81	14	193	27
6	393	6	11	106	192	28
7	240	109	87	45	149	33
8	127	151	120	30	103	133
9	391	3	122	10	107	142
10	293	98	139	3	107	88
11	208	28	83	23	182	25
12	161	230	88	50	170	80
13	350	8	42	94	178	12
14	169	140	89	47	203	21
15	334	44	64	80	217	13
16	251	104	128	9	229	13
17	240	143	67	67	183	55
18	343	11	83	15	19	197
19	264	36	28	97	172	55
20	125	222	67	27	212	8

CONCLUSION

During the past fifteen years, differential evolution (DE) which is an efficient and robust evolutionary algorithm has become a hot spot in the community of evolutionary computation. In this paper, the performance of five DE variants has been investigated by solving three stage multi-echelon supply chain network optimization problem for twenty demand scenarios with each supply chain settings. Five variants of DE are proposed to solve the three echelons SCN architecture and the results were compared with GA, PSO and optimal solutions reported by Cardenas *et.al*,[24]. Computational results demonstrate the efficiency of variant DE/Rand/1/bin to solve the SCN problem and superior performance over GA and PSO in all the problem instances. Future research can be interesting to build and solve a mathematical model for multiple states, multiple products with multi periods. Also, recent local search methods can be extended because of their potential to enhance the performance of the proposed solving approach to search the near-optimal solution in the reasonable time.

ACKNOWLEDGEMENT

The authors would like to acknowledge the financial support from the University Malaysia Pahang through the research grant with the grant number RDU180374.

REFERENCES

[1] Kadadevaramath, R. S., Chen, J. C., Shankar, B. L., & Rameshkumar, K. (2012). *Application of particle swarm intelligence algorithms in supply chain network architecture optimization*. Expert Systems with Applications, 39(11), 10160-10176

[2] Amiri, A. (2006). *Designing a distribution network in a supply chain system: Formulation and efficient solution procedure*. European journal of operational research, 171(2), 567-576.

[3] Feyzian-Tary, I., Razmi, J., & Sangari, M. S. (2018). *A variational inequality formulation for designing a multi-echelon, multi-product supply chain network in a competitive environment*. Annals of Operations Research, 264(1-2), 89-121.

[4] Ebikake, E., Sassanelli, C., & Terzi, S. (2018). *PSS design through Design for Supply Chain: State of the art review*.

[5] Clark, A. J., & Scarf, H. (1960). *Optimal policies for a multi-echelon inventory problem*. Management science, 6(4), 475-490.

[6] Clark, A. J., & Scarf, H. (1962). *Approximate solutions to a simple multi-echelon inventory problem*. Studies in applied probability and management science, 88-110.

[7] Diks, E. B., De Kok, A. G., & Lagodimos, A. G. (1996). *Multi-echelon systems: A service measure perspective*. European Journal of Operational Research, 95, 241-263.

[8] Tarhan, B., & Grossmann, I. E. (2008). *A multistage stochastic programming approach with strategies for uncertainty reduction in the synthesis of process networks with uncertain yields*. Computers & Chemical Engineering, 32 (4-5), 766-788.

[9] Morris, A. C., & Sanqwon, M. (1990). *Impact of production scale economies, manufacturing complexity, and transportation costs on supply chain facility networks* (No. REP-12849. CIMMYT.).

[10] Cohen, M. A., & Lee, H. L. (1989). *Resource deployment analysis of global manufacturing and distribution networks*. Journal of manufacturing and operations management, 2(2), 81-104.

[11] Sue-Ann, G., Ponnambalam, S. G., & Jawahar, N. (2012). *Evolutionary algorithms for optimal operating parameters of vendor managed inventory systems in a two-echelon supply chain*. Advances in Engineering Software, 52, 47-54.

[12] Che, Z. H. (2012). *A particle swarm optimization algorithm for solving unbalanced supply chain planning problems*. Applied Soft Computing, 12(4), 1279-1287.

[13] Ishii, K., Takahashi, K., & Muramatsu, R. (1988). *Integrated production, inventory and distribution systems*. The International Journal Of Production Research, 26(3), 473-482.

[14] Pyke, D. F., & Cohen, M. A. (1990). *Push and pull in manufacturing and distribution systems*. Journal of Operations Management, 9(1), 24-43.

[15] Babu, B. V., & Gujarathil, A. M. (2007, September). *Multi-objective differential evolution (MODE) for optimization of supply chain planning and management*. In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on (pp. 2732-2739). IEEE.

[16] Cohen, M. A., & Lee, H. L. (1989). *Resource deployment analysis of global manufacturing and distribution networks*. Journal of manufacturing and operations management, 2(2), 81-104.

[17] Tsiakis, P., Shah, N., & Pantelides, C. C. (2001). *Design of multi-echelon supply chain networks under demand uncertainty*. Industrial & Engineering Chemistry Research, 40(16), 3585-3604.

[18] Atamtürk, A., Berenguer, G., & Shen, Z. J. (2012). *A conic integer programming approach to stochastic joint location-inventory problems*. Operations Research, 60(2), 366-381.

[19] Shahabi, M., Akbarinasaji, S., Unnikrishnan, A., & James, R. (2013). *Integrated inventory control and facility location decisions in a multi-echelon supply chain network with hubs*. Networks and Spatial Economics, 13(4), 497-514.

[20] Storn, R., & Price, K. (1997). *Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces*. Journal of global optimization, 11(4), 341-359.

[21] Damak, N., Jarboui, B., Siarry, P., & Loukil, T. (2009). *Differential evolution for solving multi-mode resource-constrained project scheduling problems*. Computers & Operations Research, 36(9), 2653-2659.

[22] Doolun, I. S., Ponnambalam, S. G., Subramanian, N., & Kanagaraj, G. (2018). *Data driven hybrid evolutionary analytical approach for multi objective location allocation decisions: Automotive green supply chain empirical evidence*. Computers & Operations Research.

[23] Thoucharee, S., Pitakaso, R., & Phannikul, T. (2017). *Application of Modified Differential Evolution for the supply chain management of rice in the lower North-Eastern Region of Thailand*. Songklanakarin Journal of Science & Technology, 39(5).

[24] Cárdenas-Barrón, L. E., & Treviño-Garza, G. (2014). *An optimal solution to a three echelon supply chain network with multi-product and multi-period*. Applied Mathematical Modelling, 38(5-6), 1911-1918.