**Integrated JIT Lot-Splitting Model with Setup Time Reduction for Different Delivery Policy using PSO Algorithm**

**M.J. Tarokh & P. Motamedi**

***mjtarokh@kntu.ac.ir,*** *Department of Industrial Engineering, K.N.Toosi University of Technology, Tehran, Iran*

***motamedi\_p@mapnamd2.com,*** *Department of Industrial Engineering, K.N.Toosi University of Technology, Tehran, Iran*

|  |  |  |
| --- | --- | --- |
| **KEYWORDS** |  | **ABSTRACT** |
| *Joint economic lot-sizing (JELS),* *Setup time reduction, Particle swarm optimization, optimal aggregate total cost* |  | *This article develops an integrated JIT lot-splitting model for a single supplier and a single buyer. In this model we consider reduction of setup time, and the optimal lot size are obtained due to reduced setup time in the context of joint optimization for both buyer and supplier, under deterministic condition with a single product. Two cases are discussed: Single Delivery (SD) case, and Multiple Delivery (MD) case. These two cases are investigated before and after setup time reduction.**The proposed model determines the optimal order quantity (Q\*), optimal rate of setup reduction (R\*), and the optimal number of deliveries (N\*) -just for multiple deliveries case- on the joint total cost for both buyer and supplier. For optimizing our model two algorithm including Gradient Search and Particle Swarm Optimization (PSO), which is a population-based search algorithm, are applied.**Finally numerical example and sensitivity analysis are provided to compare the aggregate total cost for two cases and effectiveness of the considered algorithm. The results show that which policy for lot-sizing is leading to less total cost.* |

1. **Introduction**

**Join economic lot sizing:** (JELS), the problem of determining production and procurement quantities, is one that has to face when the supplier and the buyer has agreed to cooperate in a production system network. Goyal [1] has considered an integrated inventory model for a single product and perhaps it is the first contribution in this field.

A seminal work in the area of “integrated inventory models” is that of Banerjee [2] who proposed the concept of a joint economic lot size. Also the studies of Goyal [3] are related to this concept. He has expanded this model, where a buyer’s order quantity is delivered by the supplier in equal several shipments, as well as Kim and Ha [4]. Earlier works focused on the potential saving for both parties (the vendor and the buyer) simultaneously. A comprehensive literature review of this work is presented in Goyal & Gupta [5], Abad [6], Parlar & Wang [7], Aderohunmu & others [8], Lu [9], Goyal [10], Hill [11], Viswanathan [12], Bylka [13], and Goyal & Nebebe [14].

Since Goyal [1] introduced the integrated inventory model between a supplier and a buyer, many researchers have developed this concept for various cases, such as Banerjee [2], Goyal [3], Hill [15].

The first study on setup reduction is due to Porteus [16], with the economic order quantity (EOQ) model. Spence & Porteus [17] found the optimal rate of setup reduction in a multi-product EOQ model and Kim & others [4] for the economic manufacturing quantity (EMQ).

**Particle swarm optimization**: is a population-based swarm intelligence algorithm. It was first introduced by Kennedy and Eberhart [18] as a simulation of the social behavior of social organisms, such as bird flocking and fish schooling. PSO uses the physical movement of the individuals (particles) in the swarm and has a flexible and well-balanced mechanism to enhance and adapt to global and local exploration in continues space, while some work has been done recently in discrete domains. Recent complete surveys for PSO can be found in Banks, Vincent and Anyakoha [19,20] and Poli, Kennedy and Blackwell [21].

Several successful applications of PSO to unclear problems reported in Siqueira et al. [22]; Domingos et al. [23]; Pereira et al. [24]; Waintraub et al. [25]; in which PSO demonstrated advantages over other well-established PBM (Population Based Metaheuristic), motivated this work. PSO has been applied successfully to scheduling problems such as job shop scheduling, [26,27], flow shop scheduling [28,29], assembly scheduling [30], and resource-constraint project scheduling [31].

The wide use of PSO mainly during the last few years is due to the number of advantages that this method has compared with other optimization methods. Some of the key advantages are that this optimization method doesn't require the calculation of derivatives, that the knowledge of good solutions is retained by all particles and that the particles in the swarm share information among them. Furthermore PSO is less sensitive to the nature of the objective function, can be used for stochastic objective functions and can easily escape from local minima.

In presenting study, this paper will be organized as follows: Section 2 addresses the notations and assumptions of the proposed model. Section 3 describes the joint economic lot-sizing model, for a single supplier and a single buyer with single delivery and multiple deliveries. The description of the setup time reduction formulation is in section 4. This section explains the model of joint economic lot-sizing with the setup time reduction. In section 5 we describe that how the optimal policy for buyer and supplier can be achieved for each case by using two algorithms, Gradient Search and PSO. In section 6 numerical examples and sensitivity analysis are provided. Conclusions are summarized in section 7.

1. **Notations and Assumptions**

Joint economic lot sizing model allows the supplier and the buyer to reduce their total costs. At the other hand, small lot sizing is a way to implementing successful JIT leading to minimum supply chain costs.

In this study we extend Kim & Ha’s model [4] by considering setup time reduction as a decision variable in a joint economic lot-sizing (JELS) model with single, and several deliveries.

* 1. **Notations:**

Following notations are considered:

D: buyer’s demand rate per unit time, deterministic

P: supplier’s production rate per unit time, (P>D)

A: buyer’s ordering cost per order

S: supplier’s setup time

C: unit cost for supplier’s setup time

Q: buyer’s order quantity (production lot size)

HB: buyer’s holding cost per unit time

HS: supplier’s holding cost per unit time

F: fixed transportation cost per trip

V: unit variable cost for order handling and receiving

N: number of deliveries per batch cycle (integer number)

q: delivery size per trip, 

* 1. **Assumptions:**
1. We consider single supplier and single buyer for only one product.
2. All necessary information of the buyer and supplier are given to both sides.
3. Backorders and shortages are not allowed.
4. The buyer is assumed to pay transportation and order handling cost to facilitate frequent deliveries.
5. Product is manufactured with a finite production rate P and P>D. (if P<D, we can’t satisfy buyer’s demand and the problem would be infeasible.)
6. All cost parameters are known and constant.
7. No quantity discount is allowed and unit price is fixed. (demand rate and production rate are known, constant and deterministic.
8. HB > HS , therefore it’s not optimal to send any shipment when the buyer has some inventory.
9. The number and size of transportation vehicles has no constraints.
10. The transportation and receiving cost of each shipment is a linear function of the shipped quantities at a fixed cost.
11. There is no lead time.

*In the Single Delivery case:*

1. Every time the buyer requests an order, the supplier make the production set up on a lot for lot basis.

*In the Multiple Deliveries:*

1. When the buyer places an order, the supplier splits the order quantity into small lot sizes and send them in equal shipments.
2. **Joint Economic Lot Sizing (JELS) Model:**
	1. **Single Delivery (SD)**

In this section we first present a lot for lot inventory policy. In lot for lot model, the supplier produces optimal lot size at one setup and delivers it to the buyer at one shipment. The buyer’s total cost is composed of ordering cost, holding cost, transportation cost and order receiving cost:

 (1)

The supplier’s total cost consists of setup cost and holding cost:

 (2)

The total cost function for a joint economic lot sizing model, consists of all costs from both buyer and supplier. So, by adding Eq.(1) and Eq.(2), aggregate total cost function will be found.

 (3)

By taking the first derivative of Eq.(3), with respect to Q and set it equal to zero, optimal order quantity Q\* will be obtained.

 (4)

* 1. **Multiple Deliveries (MD)**

In multiple deliveries case, the order which is produced by quantity of Q, is delivered to buyer over N times, in small quantities q. So we have: .

Small lot sizing is a way to implementing successful JIT. All of the buyer’s total cost is:

 (5)

The supplier’s total cost consists of setup cost and holding cost:

 (6)



**Fig 1. Inventory-Time plot of buyer for MD case**



**Fig 2. Inventory-Time plot of supplier for MD case**

Adding Eq.(5) and Eq.(6) yields the joint total relevant cost for the supplier and the buyer, as follows:

 (7)

Note that if the number of deliveries, N, in Eq.(7) is one, the MD case becomes identical to Eq.(3) for SD policy. So in this case, we assume that .

According to calculations of Kim & Ha [4], by taking the first derivatives of Eq.(7) with respect to Q and N, we can obtain following formulas:

 (8)

 (9)

N\* denote the optimum integer value of N and Q\* is the optimum value of Q. if N\* in the Eq.(8) isn’t an integer number, we should choose N, which yields in Eq.(7), where N+ and N- represent the nearest integers larger and smaller than the N\*. The minimum aggregate total cost is obtained by substituting N\* and Q\* into Eq.(7).

The optimal delivery size q\*, which remains the same over multiple deliveries policy, is obtained by dividing Q\* by N\* from Eq.(8) and Eq.(9).

1. **Setup Time Reduction**

At first Porteus [16] introduces the relationship between optimal lot size and setup reduction. Following Porteus, the cost equation to reduce the setup time have been modified by Kreng and Wu [32] as follows:

 for  (10)

The rate of setup time reduction is calculated by Kreng and Wu [33]:

 for  (11)

x and y and tS are positive constants, tS is the original setup time and  is the setup time after reduction.

A fixed cost is needed to reduce setup time by a fixed percentage. This fixed percentage is , and the cost to reduce the increment of fixed percentage setup reduction is M, and fixed as well. Therefore, the Eq.(10), CS, redefined as:

 (12)

, Where R is considered as the decision variable.

* 1. **Setup time reduction in SD case:**

We know S is the supplier’s setup time and C is the unit cost for setup time. By considering “s” as the setup cost per unit time and tS as the setup time per production run before reduction, following equations are yield: CS=s  and = (1-R)

Then, the aggregate total cost for single delivery policy with considering the setup time reduction can be redefined as follows:

 (13)

K is the amortization of the setup reduction capital and a fixed reduction percentage, , can be achieved whenever the unit incremental cost of M is made.

* 1. **Setup time reduction in MD case:**

Integrating Eq.(11) and Eq.(12) to Eq.(7), the aggregate total cost for multiple deliveries with considering setup time reduction can be rearranged as follows:

 (14)

The equation above consisted of buyer’s ordering cost, buyer’s holding cost, transportation cost, order receiving cost, supplier’s setup cost after setup time reduction, supplier’s holding cost and total setup reduction capital. And the objective is to minimize the sum of these costs.

1. **Gradient Search and PSO algorithm for optimizing two cases after setup time reduction**
	1. **Gradient search algorithm for SD case:**

In order to obtain optimal Q\* and R\* in single delivery case after setup time reduction, we should take the partial derivatives of Eq.(13) with respect to Q and R, set them to equal to zero and solve both simultaneously.

 (15)

 (16)

* 1. **Gradient search algorithm for MD case:**

We can determine the optimal order quantity, Q\*, optimal rate of setup time reduction, R\*, and optimal number of deliveries from the aggregate total cost after setup time reduction from Eq.(14) with regarding that TC(Q,N,R)aggregate is a convex function. we obtain following formulas by taking partial derivatives of Eq.(14) with respect to R, N and Q, setting them equal to zero and solving for R, N and Q simultaneously.

 (17)

 (18)

 (19)

* 1. **PSO Algorithm**

In the implementation of the PSO, the population is referred to as a swarm and each individual as a particle. It is initialized with a random particles group and then searches the solution space for optima by updating generations. The general PSO algorithm is represented step by step in Fig.3.

In PSO, each particle included by social structure keeps in mind its best position and uses this as a factor affecting its speed. A particle gains speed toward its individual best position considering with how far away from that point. It also shows the same behavior for the global best position. In other words, while it is scanning the surface, it is affected by the global best position and adjusts its own speed. In the situation of that it is far from the global best position, there will be a higher chance in its speed and direction. Individuals (particles) of a swarm show inclination to change their movements by using the information below.

* Position of the *i*th particle in *k*th iteration is  (*k*=0,… *iter*max and *i*=1,…,N).
* Speed of the particle *i* in iteration *k* is  .
* Best position of the particle *i* (local best) is Pbesti.
* Best position of the particle group(global best) is gbest.

Each individual's speed changes according to the formula in Eq.(20);

 (20)

*i*th individual's speed on *k*th iteration

 *i*th individual's position on *k*th iteration

*w* inertia function

*Ci* inertia factor

Rnd random number

Pbesti individual's best position

Gbest globalbest position

**Initialization (for k=0)**

For i=1 to N

 Assign particles randomly in solution space ()

 Generate initial solutions S()

 Assign Pbesti = initial solutions S()

 Assign Gbesti = the obtained best solution among all particles

 Generate initial velocities randomly ()

 Add velocities to the corresponding particles ()

**Initialization (for k=0)**

 Determine the inertia weight ()

 For i=1 to N

 Update velocities ()

 Modify the current positions ()

 Update the Pbesti

 Update the Gbesti

**Finalize the algorithm (k=itermax)**

Assign the Gbesti=Ubest and stop

**Fig.3. Algorithmic schema for general PSO**

Inertia value of the equation changes on the each iteration. This change is based on the logic of decreasing from the value determined to minimum value according to inertia function. The objective is to converge the created speed by diminishing on the further iterations; hence more similar results can be obtained [33].

Inertia function is obtained as follows:

 (21)

*w*max first inertia force

*w*minminimum inertia force

*iter*maxmaximum iteration number

The values of *Ci* inertia factor and *w*max and *w*min inertia forces are investigated by Shi and Eberhart [34,35]. It is found that these values shouldn’t be changed from a problem to another. They fixed the values of these parameters as; *Ci*=2, *w*max=0.9 and *w*min=0.4. In this study we also used these fixed values.

Positions of the particles change by speeds as shown in Eq.(22)

 (22)

Same procedure is reiterated for each dimension.

As it can be seen above, the advantages of the PSO are easiness to implement and having few parameters to adjust. However, there are some difficulties related with applying PSO on constricted models even it has been successfully applied in many areas, such as function optimization, artificial neural network training, fuzzy system control, and other areas [36].

in our model, we also use this algorithm for optimizing obtained functions.

1. **Numerical Examples**

In this section, we use an example which originally comes from Banerjee [2]. It was modified by Kim & Ha [4] and we gathered additional values from example of Kreng & Wu [32] for analyzing our model.

We consider a buyer, a supplier and a single product. Buyer’s annual demand is 4800 units and the order cost for each order is 25$. Fixed transportation cost which buyer pays for each trip is 50$ and the unit variable cost for order handling and receiving is 1.00$/unit. Annual production capacity of supplier is 19220 units. The cost of supplier’s setup time is 400$ per unit. We assume that HB and HS are 7$ and 8$ per unit per year. The setup time before reduction policy is 4 and the setup cost per unit time is 100$, where the amortization of the setup reduction capital is 0.35, it is assumed that a rate of 30% to fixed reduction can be achieved whenever the unit incremental cost of 2000$ is made. In summary:

|  |  |  |  |
| --- | --- | --- | --- |
| A= | 20 | C×S= | 400 |
| D= | 4800 | s= | 100 |
| tS= | 4 | F= | 60 |
| M= | 2000 | V= | 1 |
| P= | 19200 | K= | 0.35 |
| HS= | 6 | HB= | 7 |
| = | 0.3 |  |  |

* 1. **Example for SD policy before and after setup time reduction**

For single delivery policy, using PSO algorithm and also Gradient Search algorithm (Eq.(13) and Eq.(14)), table 1 presents the effect of the rate of setup time reduction, R on Q\* and aggregate total cost.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | R=1-t’/t | t | t’ | t’/t | **Gradient Search** | **PSO** |
| Q\* | ΔQ | TC(Q) | ΔTC | Q\* | ΔQ | TC(Q) | ΔTC |
| **before** | 0 | 4 | 4 | 1 | 732.44 | - | 11025.75 | - | 736.18 | - | 11025.83 | - |
| **After setup time reduction** | 0.1 | 4 | 3.6 | 0.9 | 700.92 | 31.52 | 10964.63 | 61.12 | 703.37 | 32.81 | 10964.67 | 61.17 |
| 0.2 | 4 | 3.2 | 0.8 | 667.92 | 64.52 | 10915.26 | 110.49 | 661.04 | 75.14 | 10915.56 | 110.27 |
| 0.3 | 4 | 2.8 | 0.7 | 633.20 | 99.24 | 10882.19 | 143.56 | 634.54 | 101.64 | 10882.20 | 143.63 |
| 0.4 | 4 | 2.4 | 0.6 | **596.46** | 135.98 | **10872.44** | 153.31 | **592.89** | 143.29 | **10872.53** | 153.30 |
| 0.5 | 4 | 2.0 | 0.5 | 557.30 | 175.14 | 10897.44 | 128.31 | 559.17 | 177.01 | 10897.46 | 128.37 |
| 0.6 | 4 | 1.6 | 0.4 | 515.18 | 217.26 | 10977.33 | 48.43 | 515.76 | 220.42 | 10977.33 | 48.50 |
| 0.7 | 4 | 1.2 | 0.3 | 469.29 | 263.15 | 11151.87 | -126.11 | 468.11 | 268.07 | 11151.88 | -126.05 |
| 0.8 | 4 | 0.8 | 0.2 | 418.40 | 314.04 | 11515.04 | -489.29 | 416.93 | 319.25 | 11515.06 | -489.23 |
| 0.9 | 4 | 0.4 | 0.1 | 360.39 | 372.05 | 12382.32 | -1356.6 | 358.47 | 377.71 | 12382.36 | -1356.5 |
| 1 | 4 | 0 | 0.0 | 291.04 | 441.40 | - | - | 292.69 | 443.49 | - | - |

Table 1- The effects of R & Q on TC in single delivery policy

From this table, we can interfere that if the setup time is reduced from tS=4 to 0 and all other parameters remain unchanged, then by using GS algorithm, the optimal solution will be R\*= 0.4, = 2.4, Q\*= 596.46, and TC\*(Q,R)=10872.44; and by using PSO algorithm, the optimal solution will be R\*= 0.4, = 2.4, Q\*= 592.89, & TC\*(Q,R)=10872.53.

* 1. **Example for MD policy before and after setup time reduction**

Using the given parameters in 2 algorithms including PSO and GS, we calculate N\* and Q\* and then by decreasing tS from 4 to 0, we found the optimal value of R. table 2 and table 3 present this results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|   | **R=****1-t’/t** | **t** | **t’** | **t’/t** | **Gradient Search** |
| **N\*** | **N** | **Q\*** | **ΔQ** | **TC(Q,N)** | **min TC(Q,N)** | **opt(N)** | **ΔTC** |
| **before** | 0 | 4 | 4 | 1 | 2.49 | 2 | 893.05 | - | 10604.83 | 10597.83 | 3 | 0.0 |
| 3 | 973.06 | - | 10597.83 |
| **After setup time reduction** | 0.1 | 4 | 3.6 | 0.9 | 2.37 | 2 | **859.34** | 113.72 | 10592.47 | **10592.47** | **2** | 5.4 |
| 3 | 940.07 | 32.99 | 10608.01 |
| 0.2 | 4 | 3.2 | 0.8 | 2.24 | 2 | 824.25 | 148.81 | 10595.55 | 10595.55 | 2 | 2.3 |
| 3 | 905.87 | 67.19 | 10635.42 |
| 0.3 | 4 | 2.8 | 0.7 | 2.11 | 2 | 787.60 | 185.47 | 10619.37 | 10619.37 | 2 | -21.5 |
| 3 | 870.33 | 102.73 | 10685.73 |
| 0.4 | 4 | 2.4 | 0.6 | 1.96 | 2 | 749.15 | 223.91 | 10672.03 | 10672.03 | 2 | -74.2 |
| 3 | 833.28 | 139.78 | 10767.49 |
| 0.5 | 4 | 2 | 0.5 | 1.81 | 1 | 588.93 | 304.12 | 10945.41 | 10766.43 | 2 | -161.6 |
| 2 | 708.63 | 264.43 | 10766.43 |
| 0.6 | 4 | 1.6 | 0.4 | 1.63 | 1 | 545.24 | 427.82 | 11028.39 | 10924.95 | 2 | -327.1 |
| 2 | 665.64 | 307.42 | 10924.95 |
| 0.7 | 4 | 1.2 | 0.3 | 1.44 | 1 | 497.74 | 475.33 | 11206.99 | 11190.79 | 2 | -593.0 |
| 2 | 619.68 | 353.38 | 11190.79 |
| 0.8 | 4 | 0.8 | 0.2 | 1.22 | 1 | 445.19 | 527.87 | 11575.80 | 11575.80 | 1 | -978.0 |
| 2 | 570.02 | 403.04 | 11663.77 |
| 0.9 | 4 | 0.4 | 0.1 | 0.94 | 1 | 385.55 | 587.52 | 12451.54 | 12451.54 | 1 | -1853.7 |
| 1 | 4 | 0 | 0 | 0.54 | 1 | 314.80 | 658.27 | - | - | - | - |

Table 2- The effects of R & Q on TC in multiple delivery policy using GS algorithm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|   | **R=****1-t’/t** | **t** | **t’** | **t’/t** | **PSO** |
| **N\*** | **N** | **Q\*** | **ΔQ** | **TC(Q,N)** | **min TC(Q,N)** | **opt(N)** | **ΔTC** |
| **Before** | 0 | 4 | 4 | 1 | 2.49 | 2 | 891.65 | - | 10604.83 | 10599.408 | 3 | - |
| 3 | 962.98 | - | 10599.41 |
| **After setup time reduction** | 0.1 | 4 | 3.6 | 0.9 | 2.37 | 2 | **852.23** | 39.42 | 10592.67 | **10592.666** | **2** | 6.7 |
| 3 | 926.98 | -35.33 | 10610.21 |
| 0.2 | 4 | 3.2 | 0.8 | 2.24 | 2 | 818.06 | 73.59 | 10595.70 | 10595.6987 | 2 | 3.7 |
| 3 | 898.65 | -7.00 | 10636.50 |
| 0.3 | 4 | 2.8 | 0.7 | 2.11 | 2 | 779.18 | 112.47 | 10619.67 | 10619.6704 | 2 | -20.3 |
| 3 | 862.29 | 29.36 | 10686.97 |
| 0.4 | 4 | 2.4 | 0.6 | 1.96 | 1 | 732.76 | 158.89 | 11012.95 | 10691.9811 | 2 | -92.6 |
| 2 | 820.11 | 71.54 | 10769.78 |
| 0.5 | 4 | 2 | 0.5 | 1.81 | 1 | 601.63 | 290.02 | 10951.21 | 10766.8611 | 2 | -167.5 |
| 2 | 699.04 | 192.61 | 10766.86 |
| 0.6 | 4 | 1.6 | 0.4 | 1.63 | 1 | 539.53 | 352.12 | 11026.48 | 10925.1792 | 2 | -325.8 |
| 2 | 658.79 | 232.86 | 10925.18 |
| 0.7 | 4 | 1.2 | 0.3 | 1.44 | 1 | 491.48 | 400.17 | 11204.96 | 11191.1373 | 2 | -591.7 |
| 2 | 611.54 | 280.11 | 11191.14 |
| 0.8 | 4 | 0.8 | 0.2 | 1.22 | 1 | 438.18 | 453.47 | 11573.61 | 11573.6055 | 1 | -974.2 |
| 2 | 561.24 | 330.41 | 11664.21 |
| 0.9 | 4 | 0.4 | 0.1 | 0.94 | 1 | 377.43 | 514.22 | 12449.18 | 12449.1752 | 1 | -1850 |
| 1 | 4 | 0 | 0 | 0.54 | 1 | 308.58 | 583.07 | - | - | - | - |

Table 3- The effects of R & Q on TC in multiple delivery policy using PSO algorithm

Table 2 presents the results of using Gradient Search Algorithm. We can interfere that by using this algorithm, optimal value of R is 0.1. In other word, if the setup time is reduced from tS=4 to 0 and all other parameters remain unchanged, the optimal solution will be R\*= 0.1, N\*=2, = 3.6, Q\*= 859.34, and TC\*(Q,R,N)= 10592.47.

The result of Particle Swarm Optimization Algorithm is presented in table 3. We can interfere that by using this algorithm, optimal value of R is 0.1. In other word, if the setup time is reduced from tS=4 to 0 and all other parameters remain unchanged, the optimal solution will be R\*= 0.1, N\*=2, = 3.6, Q\*= 852.23, and TC\*(Q,R,N)= 10592.67.

* 1. **Sensitivity Analysis**

A sensitivity analysis is performed to study the effects of changes in the parameters of the system on the optimal order quantity, rate of setup time reduction, and number of deliveries. This analysis is performed by increasing or decreasing the parameters by 10%, 20%, and 30% taking one at a time, keeping the remaining parameters at their original values. The effects of changes in parameters on SD case and MD case are investigated. Following ratios are being calculated for different quantity of these parameters:

 (23)

 (24)

 (25)

 Which r1 shows the deviation of PSO solution from Gradient Search solution, and r2 and r3 determine the difference of optimal cost function in MD and SD case for both PSO (r2) and GS (r3) algorithms.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Change (%)** | **D** | **SD** | **MD** | **r**2 | **r**3 |
| **GS** | **PSO** | **r**1(\*103) | **GS** | **PSO** | **r**1(\*103) |
| **Q\*** | **R\*** | **TC\*(Q)** | **Q\*** | **R\*** | **TC\*(Q)** | **Q\*** | **R\*** | **TC\*(Q,N)** | **Q\*** | **R\*** | **TC\*(Q,N)** |
| -30% | 3360 | 606.1 | 0.1 | 8445.5 | 601.2 | 0.1 | 8445.7 | -0.02 | 747.2 | 0 | 8216.67 | 761.46 | 0 | 8217.54 | -0.11 | 0.03 | 0.03 |
| -20% | 3840 | 612.1 | 0.2 | 9296.9 | 618.3 | 0.2 | 9297.2 | -0.03 | 798.8 | 0 | 9031.99 | 801.29 | 0 | 9032.02 | 0.00 | 0.03 | 0.03 |
| -10% | 4320 | 610.3 | 0.3 | 10116.3 | 614.6 | 0.3 | 10116.4 | -0.01 | 815.2 | 0.1 | 9825.83 | 809.73 | 0.1 | 9825.96 | -0.01 | 0.03 | 0.03 |
| 0% | 4800 | 601.2 | 0.4 | 10912.5 | 592.9 | 0.4 | 10913.0 | -0.05 | 859.3 | 0.1 | 10592.5 | 852.23 | 0.1 | 10592.7 | -0.02 | 0.03 | 0.03 |
| 10% | 5280 | 625.0 | 0.4 | 11689.0 | 632.4 | 0.4 | 11689.4 | -0.03 | 1025.5 | 0 | 11334.6 | 1017.6 | 0.1 | 11335 | -0.03 | 0.03 | 0.03 |
| 20% | 5760 | 605.4 | 0.5 | 12448.1 | 591.6 | 0.5 | 12449.6 | -0.11 | 1039.8 | 0.1 | 12049.7 | 1035.2 | 0.1 | 12050.7 | -0.09 | 0.03 | 0.03 |
| 30% | 6240 | 624.8 | 0.5 | 13192.7 | 629.4 | 0.5 | 13192.9 | -0.01 | 1087.6 | 0.1 | 12750.3 | 1096.2 | 0.1 | 12749.6 | 0.06 | 0.03 | 0.03 |

Table 4- change in parameter D

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Change (%)** | **P** | **SD** | **MD** | **r**2 | **r**3 |
| **GS** | **PSO** | **r**1(\*103) | **GS** | **PSO** | **r**1(\*103) |
| **Q\*** | **R\*** | **TC\*(Q)** | **Q\*** | **R\*** | **TC\*(Q)** | **Q\*** | **R\*** | **TC\*(Q,N)** | **Q\*** | **R\*** | **TC\*(Q,N)** |
| -30% | 13440 | 579.7 | 0.4 | 11102.2 | 595.4 | 0.4 | 11104.1 | -0.170 | 993.7 | 0 | 10490.1 | 976.03 | 0 | 10492.9 | -0.27 | 0.06 | 0.06 |
| -20% | 15360 | 588.3 | 0.4 | 11024.0 | 601.3 | 0.4 | 11025.3 | -0.112 | 984.9 | 0 | 10535.2 | 966.42 | 0 | 10538.4 | -0.30 | 0.05 | 0.05 |
| -10% | 17280 | 595.4 | 0.4 | 10962.4 | 614.7 | 0.4 | 10965.0 | -0.241 | 978.3 | 0 | 10570.1 | 953.04 | 0 | 10575.1 | -0.48 | 0.04 | 0.04 |
| 0% | 19200 | 601.2 | 0.4 | 10912.5 | 592.9 | 0.4 | 10913.0 | -0.045 | 859.3 | 0.1 | 10592.5 | 852.23 | 0.1 | 10592.7 | -0.02 | 0.03 | 0.03 |
| 10% | 21120 | 606.1 | 0.4 | 10871.4 | 624.6 | 0.4 | 10873.7 | -0.212 | 859.3 | 0.1 | 10592.5 | 876.19 | 0.1 | 10593.5 | -0.10 | 0.03 | 0.03 |
| 20% | 23040 | 610.2 | 0.4 | 10836.8 | 608.4 | 0.4 | 10836.8 | -0.003 | 859.3 | 0.1 | 10592.5 | 871.69 | 0.1 | 10593 | -0.05 | 0.02 | 0.02 |
| 30% | 24960 | 613.8 | 0.4 | 10807.4 | 631.7 | 0.4 | 10809.5 | -0.191 | 859.3 | 0.1 | 10592.5 | 865.96 | 0.1 | 10593.5 | 0.00 | 0.02 | 0.02 |

Table 5- Change in parameter P

The following inferences can be made from the results of table 1-3, and sensitivity analysis based on table 3-4.

* By using PSO algorithm, which is a meta-heuristic algorithm and gives an approximate solution, and Gradient Search algorithm, which gives an exact solution, and comparing the results of these methods by considering r1, we can interfere that the values computed for the aggregate total cost are approximately similar.
* To compare effectiveness of different delivery policies, SD and MD, we should compare joint total costs of SD and MD case in tables 1-3. TC\*MD<TC\*SD, so the policy of frequent shipment results in less total cost than single shipment policy. TC\*MD=10592.5 by N=2, while TC\*SD=10872.5.
* From table 5 we can interfere that by increasing the value of D, while other parameters remain unchanged, the optimal joint total cost of MD policy increases, but the optimal joint total cost of SD policy decreases.
* Enlarging the parameter P, results to more joint total cost for both single delivery and multiple delivery policy.
1. **Conclusions**

The effects of setup time reduction in the integrated lot splitting strategy have been analyzed in this study. The proposed model determines optimal order quantity, optimal rate of setup reduction, and optimal number of deliveries on the integrated total relevant cost for single delivery and multiple deliveries policy, and compares the optimal value of these two cases by using two algorithms, PSO and GS.

Suggested extending studies for proposed model is considering multiple products or multiple buyers and suppliers, or probabilistic parameters.

**Appendix:**

The expression of holding cost is derived by Joglekar (1988). From Fig.1 the holding cost of supplier is derived as following:

== =



**References:**

[1] Goyal, S.K. An integrated inventory model for a single supplier - single customer problem, International Journal of Production Research. 15(1), 1976, 107-111.

[2] Banerjee, A. A joint economic-lot-size model for purchaserand vendor, Decision Sciences, 17, 1986, 292-311.

[3] Goyal, S.K. A joint economic-lot-size model for purchaser and vendor: A Comment, Decision Sciences, 19 (1), 1988, 236-241.

[4] Kim, S. and Ha, D. A JIT lot-splitting model for supply chain management: Enhancing buyer-supplier linkage, International Journal of Production Research 86, 2003, 1-10.

[5] Goyal, S.K., Gupta, Y. Integrated inventory models: The buyer–vendor coordination. European Journal of Operational Research 41, 1989, 261–269.

[6] Abad, P.L. Supplier pricing when the buyer’s annual requirements are fixed. Computers & OR. 21 (2), 1994, 155–167.

[7] Parlar, M., Wang, Q. Discounting decisions in supplier–buyer relationship with a linear buyer’s demand. IIE Transactions 26 (2), 1994, 34–41.

[8] Aderohunmu, R., Ayodele, M., Bryson, N. Joint vendor–buyer policy in JIT manufacturing. Journal of the OR Society 46, 1995, 375–385.

[9] Lu, L. A one-vendor multi-buyer integrated inventory model. European Journal of Operational Research 81 (2), 1995, 312–323.

[10] Goyal, S.K.. A one-vendor multi-buyer integrated inventory model: A comment. European Journal of Operational Research 82 (1), 1995, 209–210.

[11] Hill, R.M. The single-vendor single-buyer integrated production–inventory model with a generalized policy. European Journal of Operational Research 97, 1997, 493–499.

[12] Viswanathan, S. Optimal strategy for the integrated vendor–buyer inventory model. European Journal of Operational Research 105, 1998, 38–42.

[13] Bylka, S. A dynamic model for the single-vendor, multiple-buyer problem. International Journal of Production Economics 59, 1999, 297–304.

[14] Goyal, S.K., Nebebe, F. Determination of economic production-shipment policy for a single-vendor–single buyer system. European Journal of Operational Research. 121(1), 2000, 175–178.

[15] Hill, R.M. The single-vendor single-buyer integrated production - inventory model with a generalized policy. European Journal of Operational Research 97, 1997, 493–499.

[16] Porteus, E.L. Investigating in reduced setups in the EOQ model, Management Science. 30(8), 1985, 998-1010.

[17] Spence, A.M. and Porteus, E.L. Setup reduction and increased effective capacity, Management Science, 33(10), 1987, 1291-1301

[18] Kennedy, J., & Eberhart, R. C. Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks. 1995, 1942−1948, Piscataway, NJ, USA: IEEE Press.

[19] Banks, A., Vincent, J., & Anyakoha, C. A review of particle swarm optimization. Part I: Background and development. Natural Computing, 6(4), 2007, 467−484.

[20] Banks, A., Vincent, J., & Anyakoha, C. A review of particle swarm optimization. Part II: Hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. Natural Computing, 7, 2008, 109−124.

[21] Poli, R., Kennedy, J., & Blackwell, T. Particle swarm optimization. An overview. Swarm Intelligence. 1, 2007, 33−57.

[22] Siqueira, N.N., Pereira, C.M.N.A., Lapa, C.M.F. The Particle Swarm Optimization algorithm applied to nuclear systems surveillance test planning. 2005, In: Proceedings of the International Nuclear Atlantic Conference (INAC).

[23] Domingos, R.P., Schirru, R., Pereira, C.M.N.A. Particle Swarm Optimization in reactor core designs. Nuclear Science and Engineering. 2006, 152, 157.

[24] Pereira, C.M.N.A., Schirru, R., Lapa, C.M.F., Canedo, J.A.C., Waintraub, M., Meneses, A.A.M., Baptista, R.P., Siqueira, N.N. Particle Swarm Optimization applied to nuclear engineering problems. International Journal of Nuclear Knowledge Management. 2 (3), 2007, 313-332.

[25] Waintraub, M., Schirru, R., Pereira, C.M.N.A. Multiprocessor modeling of parallel particle swarm optimization applied to nuclear engineering problems. Progress in Nuclear Energy 51, 2009, 680-688.

[26] Xia W, Wu Z. An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Computers and Industrial Engineering, 48(2), 2005, 409-425.

[27] Sha DY, Hsu C-Y. An effective hybrid particle swarm optimization for job shop scheduling problem. Computers and Industrial Engineering, 51(4), 2006, 791-808.

[28] Lian Z, Gu X, Jiao B. A similar particle swarm optimization algorithm for permutation flow shop scheduling to minimize make span. Applied Mathematics and Computation, 175(1), 2006, 773–785

[29] Tasgetiren MF, Liang Y-C, Sevkli M, Gencyilmaz G. A particle swarm optimization algorithm for makespan and total flow-time minimization in the permutation flow shop sequencing. European Journal of Operational Research. 177(3), 2007, 1930–1947.

[30] Allahverdi A, Al-Anzi FS. A PSO and a tabu-search heuristics for the assembly scheduling problem of the two-stage distributed database application. Computers & Operations Research. 33(4), 2006, 1056–80.

[31] Zhang H, Li H, Tam CM. Particle swarm optimization for resource-constrained project scheduling. International Journal of Project Management. 24(1), 2006, 83–92.

[32] Kreng B. and Wu, S. Implementing an optimal policy for setup time reduction in an economic production quantity model, International Journal of Systems Science, 31(5), 2000, 605-612.

[33] Onut S, et.al. A particle swarm optimization algorithm for the multi-level warehouse layout design problem. Computers and Industrial Engineering 54, 2008, 783-799.

[34] Shi, Y., & Eberhart, R. C. A modified particle swarm optimizer. In Proceedings of the IEEE international conference on evolutionary computation. 1998, 69–73. Piscataway, NJ: IEEE Press.

[35] Shi, Y., & Eberhart, R. C. Parameter selection in particle swarm optimization. In Evolutionary Programming, VII: Proc. EP98, 1998, 591–600. New York: Springer-Verlag.

[36] Eberhart, R. C., Shi, Y. Comparison between genetic algorithms and particle swarm optimization. In Proceedings of 7th ICEC, 1998, 611–616.