Journal of Optimization in Industrial Engineering 13 (2013) 55-64

A Multi-level Capacitated Lot-sizing Problem with Safety Stock Deficit and Production Manners: A Revised Simulated Annealing

Esmaeil Mehdizadeh[a], Mohammad Reza Mohammadizadeh[b,*]

[a]*Assistant Professor, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.*

[b]*M.Sc.Student , Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.*

Received 2 September, 2011; Revised 19 January, 2012; Accepted 27 February, 2012

Abstract

Lot-sizing problems (LSPs) belong to the class of production planning problems in which the availability quantities of the production plan are always considered as a decision variable. This paper aims at developing a new mathematical model for the multi-level capacitated LSP with setup times, safety stock deficit, shortage, and different production manners. Since the proposed linear mixed integer programming model is NP-hard, a new version of simulated annealing algorithm (SA) is developed to solve the model named revised SA algorithm (RSA). Since the performance of the meta-heuristics severely depends on their parameters, Taguchi approach is applied to tune the parameters of both SA and RSA. In order to justify the proposed mathematical model, we utilize an exact approach to compare the results. To demonstrate the efficiency of the proposed RSA, first, some test problems are generated; then, the results are statistically and graphically compared with the traditional SA algorithm.

Keywords: Lot-sizing problem; Simulated annealing; Shortage; Safety stock deficit; Production manners

1. Introduction

Production planning problem consists in deciding how to transform raw material into final goods in order to satisfy the demands at minimum cost. The lot-sizing problem (LSP) is a crucial step and a well-known optimization problem in production planning which involves time-varying demands for a set of *N* items over *T* periods. In industrial applications, several factors may sophisticate making the best decisions. For instance, considering multi-items can lead to impossibility of satisfying demand. Moreover, safety stock is a complicating constraint as a target to reach rather than an industrial constraint to satisfy (Baker, 1990). Today, in most operational and industrial applications, one of the important questions in the field of production control being studied is to find the optimum composition of using production resources towards customer satisfaction and profiting. In reality, optimizing manners of production planning in line with practical restrictions have always been the centre of attention for industrial managers. In the field of production planning, the prospect of production is divided into three areas including short-term, mid-term, and long-term. Lot-sizing problems fall under mid-term programming prospects. The main concern of this research is Multi Level Capacitated Lot-Sizing Problem (MLCLSP). In this regard, Chen and Thizy (Chen and Thizy, 1990) introduced a question of multi-product lot-sizing problem with the capacity and setup time. To explain how the generalize ability between the exact and approximate answers is created, the Lagrangian relaxation method was utilized. They also presented a new algorithm to solve their MLCLSP.

Maes and Van Wassenhove (Maes and VanWassenhove, 1988) developed a novel algorithm to solve the problem of multi-product production with the capacity and setup time. The goal of most MLCLSPs is to find the optimum production plan, shortage, and inventory. Tempelmeier and Derstroff (Tempelmeier and Derstroff, 1996) developed a new method based on Lagrangian for the multi-product, multi-layer with lot-size and setup time. Using the Lagrangian relaxation, the multi product, multi layer problem of lots is changed into some limited one-product problems. To solve these problems of one-product a low limit is used for the value of the objective function and also high limits are results of the sense of the new method of the finite timing procedure. The quality of this method was tested in problems with various sizes. Berretta and Rodrigues (Berretta and Rodrigues, 2004) produced a method to solve the problem of the volume of multi-layer

* Corresponding author e-mail: m_mohamadizadeh@yahoo.com

production with capacity. In their model, costs and setup times were taken into consideration. Based on the complexity of the problem, the memetic algorithm was presented on the basis of meta-heuristic methods. Tang (Tang, 2004) proposed a multi-item lot-sizing model. To solve his model, simulated annealing algorithm was presented.

Absi and Kedad-Sidhoum (Absi and Kedad-Sidhoum, 2007) presented a mixed integer mathematical formulation for the MLCLSP with capacity constraint, and lower and upper bounds for productions. They developed a branch and bound algorithm for the model without considering the safety stock as well. Following this, Absi and Kedad-Sidhoum (Absi and Kedad-Sidhoum, 2008) presented a mixed integer mathematical formulation for the multi-product problem with capacity limitation, taking into consideration setup and shortage costs to make their model more realistic. Since the proposed model is NP-hard, they used a simple branch and bound heuristic to solve the problem. Atthe end, some numerical examples were presented to show the implication of the method in solving the problem. Aksen et al. (Aksen et al. 2003) presented a new mathematical formulation for MLCLSP with shortage costs. Loparic et al. (Loparic et al. 2001) presented a dynamic programming algorithm with a difficulty degree of T^2 to optimize the MLCLSP with safety stock. Sural et al. (Sural et al. 2009) presented a Lagrangian relaxation method to solve the problem of production with setup time to minimize on-hand inventory.

Nowadays, to increase customer satisfaction, using safety stock, from the supply chain viewpoint, has widely become important. Since the safety stock concept is considered to be a new trend in the literature of such problems, considering safety stock can be a contribution of the presented model. In this regard, Absi and Kedad-Sidhoum (Absi and Kedad-Sidhoum, 2009) followed their previous work with developing a new model for multi-item lot-sizing with considering setup time and shortage costs and safety stock. To solve the model, they used the dynamic programming approach and analyzed the implication of the proposed model. Han et al. (Han et al. 2009) have used the particle swarm optimization algorithm to solve the problem of multi-item lot size without capacity. To demonstrate the performance of the proposed algorithm, the comparison procedure with the genetic algorithm was analyzed. Choudhary and Shankar (Choudhary and Shankar, 2011) applied integer linear programming approach to solve a multi-period procurement lot-sizing problem for a single product that is procured from a single supplier considering rejections and late deliveries under all-unit quantity discount environment. The goal of this proposed model is to relate goals of costs and decision making in the suitable number of production and delivery scheduling to reduce total costs according to discounts, economic deals, and supply management. Moreover, the optimum model in analyzing the effect of variety on model parameters such as rejection rate, demand, inventory capacity, and holding costs is for a multi-period problem to arrange a production program. This analysis helps to make a decision to identify chances of reducing costs to a great extent. To show the implication of the proposed model, a numerical example was provided. The proposed approach provides flexibility to decision maker in multi-period procurement lot-sizing decisions through tradeoff curves and sensitivity analysis.

Recently, Wu et al. (Wu et al. 2011) proposed two new mixed linear programming models and presented a new optimization technique which reached high quality answers within a logical time. Rezaei and Davoodi (Rezaei and Davoodi, 2011) have also proposed two multi-objective mixed integer non-linear programming models for the multi-item lot size with multi-supplier problem. Each model was produced based on three objective functions including costs, quality, and service level and a group of constraints. In the first model, shortage is not allowed, but in the second model, all the demand during the stock-out period is backordered. With regards to complexity of the models, an innovative genetic algorithm is presented to obtain a set of Pareto-optimal solutions. Comparison of results indicates that, in a backordering situation, buyers are better able to optimize their objectives than a situation where there is no shortage.

Due to the variety of products in the current manner under review, each product might be produced through different manners, and the costs of each unit and the value of resources used depend on the selected production manner. Paying attention to the kind of production manner to make a mathematical model for the problem can increase the implication of the problem. In most wide industrial applications, one of the most important questions is to identify the best value of production. In this research, an integer linear programming model is developed for the multi-item lot-sizing while taking into consideration many industrial limitations. The goal is to minimize the total production cost, inventory costs, shortage costs, safety stock deficit costs, and setup costs. As another contribution, we follow to develop simulated annealing algorithm in the production planning literature. The main characteristic of the proposed algorithm are (I) initializing with a population of solutions rather than one solution, (II) generating several neighbor's solutions, and (III) considering calibrated parameters. All this facts led us to find better solutions.

The paper is organized as follows: In the next section, the MLCLSP model with considerations of setup time, holding, shortage costs, and different production manners is presented. Section 3 elaborates on a developed SA algorithm to solve the model. In Section 4, we first tune the parameters of both algorithms and then the results are analyzed in terms of graphical and statistical comparison in Section 5. The final section provides conclusions and directs for future researches.

2. Problem Formulation

In this section, first the problem, assumptions, parameters, and decision variables are thoroughly discussed, and then the proposed linear programming model is defined.
Nowadays, in most production centers, the need to answer the question of appointing a combination of the production of commodities is felt more than ever before. In order to close the gap between the conditions of the problem and the real world conditions, in this research, the multi-item lot size problem is studied with considerations of production line equilibrium and capacity limitations. Not only has there been a consideration of different production manners for products, but also the model has been designed in the conditions of having safety stock and shortage being allowed. The main goal is to present a mathematical model to optimize production, inventory, and shortage quantities as well as determine the best production manner in which summations of production, setup, inventory, and shortage costs are minimized.

2.1. Assumptions

- Demand is deterministicand occurs at a constant rate.
- Shortage is backlogged.
- Shortage and inventory costs must be taken into consideration at the end.
- Raw material resource with given capacities are considered.
- Shortage is allowed for one period.
- The quantity of inventory and shortage at the beginning of the planning horizon is zero.
- The quantity of shortage at the end of the planning horizon is zero.

2.2. Parameters

T: Number of periods in the planning horizon, t=1, …,T

N: Number of products, i=1, …,N

J: Number of production manner, j=1, …,J

d_{it}: The demand for product i in the period t

φ_{it}: Unitary shortage cost of product i in period t

y_{it}^{-} : Unitary safety stock deficit cost of product i in period t

L_{it} : The quantity of the safety stock of product i in the period t

δ_{it} : The difference of safety stock of product i in the period t and the previous period

α_{ijt}: The production cost of each unit of product i in the period t through the manner j

β_{ijt}: The setup cost of the production of product i in the period t through the method j

y_{it}^{+} : The unit holding cost of product i in the period t

C_t : The capacity of the source at hand in the period t

v_i : The quantity of the source used by each unit of the product i

f_{ij} : The quantity of wasted source for product i produced through the manner j

M : A large number

2.3. Decision Variables

X_{ijt}: Production quantity for product i in the period t through the manner j

$y_{ijt} = 0$ or 1: One is when the product i is produced in the period t through the manner j. Zero is when the something else happens.

r_{it}: The quantity of shortage of product i in the period t

S_{it}^{+} : The quantity of overstock deficit of product i in the period t

S_{it}^{-} : The quantity of safety stock deficit of product i in the period t

To show the network structure of the MLCLSP, Fig. 1 is plotted. In this figure, the nodes represent the periods of the planning prospect, and arcs relate to any of the model's decision variables, depending on the signals just discussed. Nevertheless, the production parameters of demand and the difference of safety stock in two consecutive periods is represented.

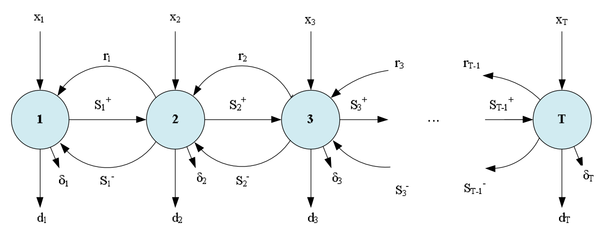

Fig. 1.Structure of the *MLCLSP*

2.4. The proposed Model

$$Min\ Z=\sum_{i=1}^{n}\sum_{t=1}^{T}[\sum_{j=1}^{J}(\alpha_{ijt}.x_{ijt}+\beta_{ijt}.y_{ijt}) + \varphi_{it}.r_{it}+y_{it}^{+}.s_{it}^{+}+y_{it}^{-}.s_{it}^{-}] \quad (1)$$

S.t.

$$s_{i,t-1}^{+}-s_{i,t-1}^{-}-r_{i,t-1}+r_{it}+\sum_{j=1}^{J}x_{ijt}=d_{it}+\delta_{it}+s_{it}^{+}-s_{it}^{-}$$

$$\forall i=1,2,...,n\ ;\ t=1,2,...,T-1 \quad (2)$$

$$s_{i,t-1}^{+}-s_{i,t-1}^{-}-r_{i,t-1}+\sum_{j=1}^{J}x_{ijt}=d_{it}+\delta_{it}+s_{it}^{+}-s_{it}^{-}$$

$$\forall i=1,2,...,n\ ;\ t=T \quad (3)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{J}(v_i.x_{ijt}+f_{ij}.y_{ijt})\le C_t \quad \forall t=1,2,...,T \quad (4)$$

$$x_{ijt}\le M.y_{ijt} \quad \forall i=1,2,...,n\ ;\ j=1,2,...,J\ ;$$

$$t=1,2,...,T \quad (5)$$

$$r_{it}\le d_{it} \quad \forall i=1,2,...,n\ ;\ t=1,2,...,T \quad (6)$$

$$s_{it}^{-}\le L_{it} \quad \forall i=1,2,...,n\ ;\ t=1,2,...,T \quad (7)$$

$$x_{ijt},r_{it},s_{it}^{+},s_{it}^{-}\ge 0\ ;y_{ijt}\in\{0,1\} \quad \forall i=1,2,...,n\ ;$$

$$j=1,2,...,J\ ;\ t=1,2,...,T \quad (8)$$

Equation (1) shows the objective function which minimizes the total cost considered by the production plans which are included unit production costs with different production manner, inventory costs, shortage costs, and setup costs. Constraints (2) are the inventory flow conservation equations through the planning horizon. Constraints (3) show the inventory flow conservation equations in the last period of planning horizon, which are written without considering the shortage. The sources capacities are shown in Constraints (4). Constraints (5) ensure the quantity of production of i in the period t should not exceed the maximum value allowed. Constraints (6) and (7) define upper bounds on respectively, the demand shortage and the safety stock deficit for product i in period t. Constraints (8) characterize the variable's domains.

3. A Revised Simulated Annealing

Since the proposed model is an NP-hard one regarding its computational complexity [12], to solve the model a simulated annealing (SA) algorithm was developed and to prove its implication, the results of this algorithm were statistically analyzed with the results of the classic SA algorithm as well as an exact approach.
The concept of SA, introduced by Kirkpatrick et al. (Kirkpatrick et al. 1983) is actually a comparison between the physical annealing process in solids and the solving of complex optimization problems. The SA algorithm is a method to improve being placed in the local optimum point. On the other hand, in local optimization algorithms, the new solution is only accepted when the objective function is improved. This is while in the SA algorithm, not only solutions which do not improve the objective function are accepted, but also unsuitable solutions are possibly accepted. This probability function can be seen in Eq. (9).

$$P(\Delta f)=\exp(\frac{-\Delta f}{T}) \quad (9)$$

Where Δf is the value of change in the objective function and T shows the temperature. Even if this possibility is larger than a random number between zero and one, it is accepted. The algorithm generally works in a way that in every repetition, the SA algorithm produces a neighbor state such as *s'* and based on a possibility, the problem moves from *s* to *s'* or stays in *s*. This process is repeated until an almost optimum solution is reached or the maximum number of repetitions is conducted. On the other hand, the T parameter should be selected in a way that it accepts most neighboring states and, at the end, with the gradual reduction in the parameter T, the algorithm reports suitable solutions.
In this article, in order to reach higher quality solutions, two new concepts in applying the SA algorithm have been taken into consideration. The first concept is in initialization phase of the algorithm in which the algorithm starts with a group of solutions as a population. The other concept states that in the neighboring production phase, for each solution, many neighbors are produced at once. Finally, after the solution evaluation phase, the best solutions are selected and are moved to the next repetition as the selected solutions group. The expansion of these two concepts and considering them as an important part of the SA algorithm increases the search ability of the algorithm in reaching high quality solutions.
Now, in order to describe the proposed algorithm, the steps to implement the algorithm have been discussed in detail in the subcategories below.

3.1. Initialization

In this section, the inputs of the algorithm include the preliminary temperature (T_0), the temperature reduction rate (β), the maximum value of repetitions (*nIt*), the number of the population (*nPop*), and the number of neighboring (*nMv*).

3.2. Solution representation

The most important part in improving the implementation of meta-heuristic algorithms is solution representation. In this paper, the illustration of the solution structure is in the single-string (A) type, which its length is equal to the total number of periods. Each part of the strand itself contains a string (B_t) which its length equals the total number of different manner of production. On the other hand, the components of the (B_t) string can define the production program, depending on the type of decision variable (e.g., accidental figures of zero and one for the setup decision variable) (Tavakkoli-Moghaddam et al. 2009). In order for the structure to become clearer, the schematic of the solution group of a specific problem with three products and three different methods of production for each of the products in five periods is shown in Fig. 2.

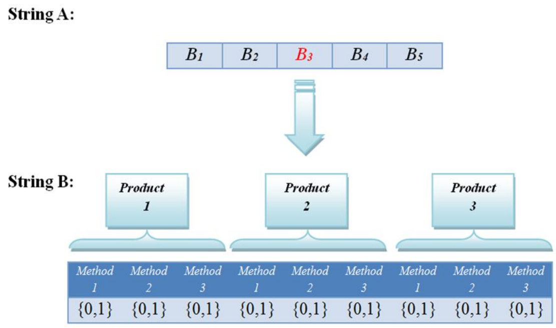

Fig. 2.An example of solution representation

3.3. Neighborhood Structure

In order to create neighboring, the structure of the replacement neighboring is adapted, that is, two random numbers are created in the integer range of one to the number of periods (e.g. Figures 2 and 5) and the place of the cell of these two figures are exchanged. Moreover, in the neighborhood structure, we consider a repair process to ensure finding feasible solution. The proposed repair process attempts to change the decision variables in order to balance the infeasible constraints. Fig. 3 illustrates the way in which the neighboring structure works.

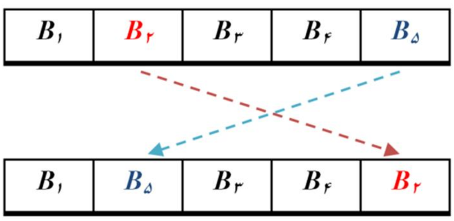

Fig. 3.An example ofneighborhood structure

3.4. Neighboring solutions evaluation

The process of evaluating produced solutions is in a way that if any neighboring solution is better than the current solution in hand, or if the possibility function is larger than the steady random number, we move to a new solution, and if the neighboring solution is not better than the current solution, or if the possibility function is smaller than the steady random number, we choose the current solution. However, in both states, there is the possibility of choosing the worse solution.

3.5. Cooling schedule

In order to reach better solution, the algorithm reduces temperature in consecutive repetitions through the use of Eq. (10) so that the convergence process is accomplished.

$$T_h = \beta \times T_{h-1} \;\; ; h > 2,\; 0 < \beta < 1 \qquad (10)$$

β represents the temperature decrease rate and h is the main counter of the algorithm's loop. Finally, after reaching a predetermined value of temperature, the algorithm will be stopped.
In the next section, experimental problems with different dimensions are implemented by the SA algorithm and we have used statistical analyses to demonstrate the performance of the proposed solving methodology.

4. Parameter Calibration

Nowadays, several procedures in the design of experiments (DOE) are implemented to calibrate the algorithms. As an alternative, in a full factorial experiment as the number of considered factors increases, the number of level combinations increases very rapidly resulting in very large computational efforts (Montgomery, 2005). To decrease the number of required experiments, a fractional factorial experiment is used in which only a portion of the total possible combinations are considered. Taguchi (Taguchi, 1986) presented a number of designs to examine a large number of factors with a very small number of observations. To determine the best level of each factor, Taguchi approach utilizes signal-to-noise (*S/N*) ratio as a measure of variations in Eq. (11).

$$S / N\ ratio = -10\log_{10}(objectivefunction)^2 \qquad (11)$$

In this paper, to obtain the optimum values of the parameters, we use the relative percentage deviation (RPD) as a common performance measure to evaluate the algorithms. RPD shows that how much an algorithm is different from the best obtained solution on average and is calculated according to the Eq. (12).

$$RPD = \frac{A\lg_{sol} - Min_{sol}}{Min_{sol}} \times 100 \qquad (12)$$

Where Min_{sol} represents the best solution obtained for each run and Alg_{sol} is the obtained solution for a run by a given algorithm. In this respect, it should be mentioned that the proposed RSA and SA algorithms contain four and two parameters, respectively. The both algorithms parameters along with their levels are provided in Tables 1-2.

Table 1
Parameters levels for SA

Factors		Levels		
		Level 1	Level 2	Level 3
T_0	(A)	750	1000	1250
α	(B)	0.9	0.95	0.99

Table 2
parameters levels for RSA

Factors		Levels		
		Level 1	Level 2	Level 3
T_0	(A)	750	1000	1250
α	(B)	0.9	0.95	0.99
n_{pop}	(C)	5	10	15
n_{Move}	(D)	2	4	6

Considering four factors in three levels there are $(3^4)=81$ different combinations for just one problem that leads to the huge computational efforts. Using Taguchi's plan, these 81 combinations are reduced to 9. In order to conduct the experiment, we consider test problems 5 and 8 for SA and test problems 3 and 8 for RSA. Hence, the algorithms are run three times, then, the average of these three run is reported.

Table 3
Experimental results obtained by RSA

NO.	A	B	C	D	RPD (TP 3)	RPD (TP 8)
1	1	1	1	1	0.057	0.187
2	1	2	2	2	0.160	0.104
3	1	3	3	3	0.053	0.039
4	2	1	2	3	0.130	0.032
5	2	2	3	1	0.307	0.087
6	2	3	1	2	0	0.026
7	3	1	3	2	0.335	0.072
8	3	2	1	3	0.164	0.094
9	3	3	2	1	0.242	0

Table 4
Experimental results obtained by SA

NO.	A	B	RPD (TP 5)	RPD (TP 8)
1	1	1	0.380	0.190
2	2	1	0.246	0.171
3	3	1	0.563	0.139
4	1	2	0.453	0.114
5	2	2	0.395	0.108
6	3	2	0.126	0.103
7	1	3	0.173	0.102
8	2	3	0	0.055
9	3	3	0.124	0

After RPDs are calculated for each combination, they are transformed into the *S/N* ratio. Then, in order to identify the significant factors, we implement an ANOVA F-test on the *S/N* ratio data with a 95% confidence limit. In order to clarify the trend of Taguchi implementation, Figs. 4-5 plot the best value of algorithm parameters. The best level of all parameters is specified in Table 5.

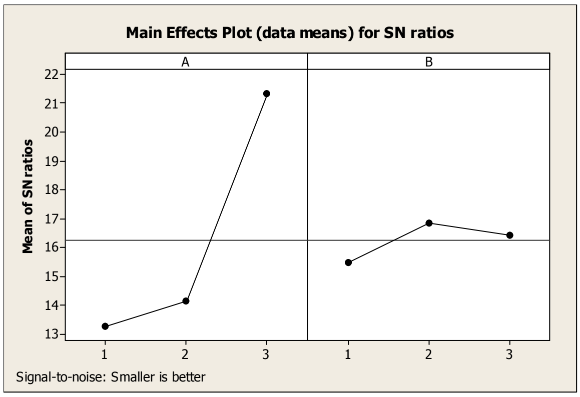

Fig. 4. Mean of S/N ratio levels for SA parameters

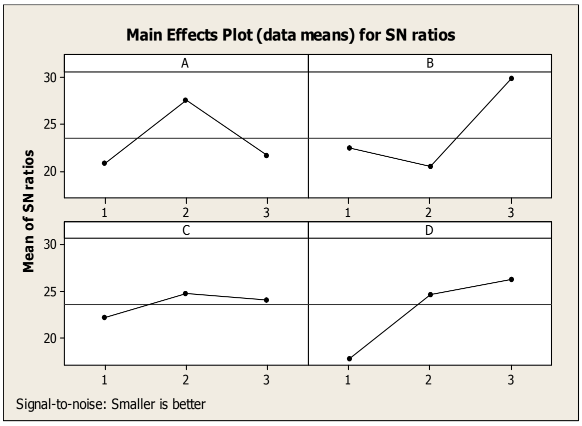

Fig. 5. Mean of S/N ratio levels for RSA parameters

Table 5
Best level for parameters

Solving methods	Parameters	Optimum amount
RSA	T_0	1000
	α	0.99
	n_{pop}	10
	n_{Move}	6
SA	T_0	1250
	α	0.95

5. Analysis of the Results

In this section, in order to prove the suitable implication of the proposed solving method, first some experimental problems with different dimensions were generated, and then the acquired computational results were analyzed through the solving methods, including the classical SA algorithm, exact approach, and RSA.
For this reason, 17 test problems in different dimensions, including problems with small, medium, and large

dimensions, were presented, and for each of the solving methods, the objective function values are achieved and compared. Table 6 shows the computational outcome obtained from the experimental problems created from each of the solving methods. To appoint the outcomes of SA and RSA, proposed algorithms were coded with the Matlab (R2010a) software and run on a notebook with four GB RAM and two GB processor.

Fig. 6 graphically depicts the way both proposed algorithms act on produced experimental problems. Generally, the picture states the *RSA* algorithm presents a better performance solution than the classical *SA* algorithm regarding all problems from the perspective of quality (especially in the case of problems with large dimensions). To prove this matter, the presented statistical analysis (the variance analysis outcome) were reported for problems with small, medium, and large dimensions, in Tables 7, 8, and 9, which according to the values of the survey (or *P-Value*) we can reach the conclusion that the algorithm has shown its usefulness in different problems as compared to the classical algorithm, and statistical results also are significantly different for problems with large dimensions. To clarify the matter, confidence distances for different sizes have been illustrated in Fig. 7, Fig. 8, and Fig. 9. The point worth noticing is that with the increase in the dimensions of the problem, the difference of values of objective functions in these two algorithms follow an increasing process as well (Fig. 10). At the end, it is important to note that although the proposed RSA algorithm searches a wider range, it also needs more computational time, but due to the features of meta-heuristic algorithms, which from the perspective of time, are superior to perspectives of exact optimization, the algorithm can be considered a useful solving process to reach higher quality answers.

Table 6
Computational results of solving methodologies

Problem No.	Problem size	product	method	period	CPU Time (Second)			Objective function value			The gap between SA & RSA
					Lingo	SA	RSA	Lingo	SA	RSA	
1	Small	2	2	3	≈0	1.44	2.47	2057072	2057072	2057072	0
2		3	2	5	≈0	2.36	4.45	4970620	6861514	5767352	1094162
3		3	3	5	1	2.40	5.51	5282963	8972893	7691883	1281010
4		5	2	6	3	8.21	12.50	10286650	15394429	14245656	1148773
5		5	3	6	15	23.46	34.58	9715882	19968815	16027097	3941718
6	Medium	5	2	12	-	31.13	47.26	-	36360170	30205808	6154362
7		5	3	12	-	32.51	61.08	-	56904449	46622287	10282162
8		10	2	12	-	40.11	86.20	-	74365993	62183899	12182094
9		10	3	5	-	43.08	54.36	-	39182321	33809806	5372515
10		10	3	12	-	37.45	88.54	-	139326141	115386577	23939564
11		12	2	12	-	48.23	95.13	-	90525497	72633105	17892392
12	Large	15	3	12	-	58.16	108.52	-	222272814	167454335	54818479
13		20	3	12	-	62.29	121.08	-	285356409	217178037	68178372
14		22	3	12	-	76.34	136.48	-	316380464	238597664	77782800
15		25	3	12	-	80.48	152.31	-	368069030	283219015	84850015
16		30	2	12	-	78.05	174.56	-	246503341	177706572	68796769
17		30	3	12	-	85.27	186.37	-	438705298	329881502	108823796

$$\alpha_{ijt} \sim Uniform[65,85];\ \beta_{ijt} \sim Uniform[200000,260000];\ \varphi_{it} \sim Uniform[100,250];\ y_{it}^{+} \sim Uniform[50,80]$$

$$y_{it}^{-} \sim Uniform[60,180];\ d_{it} \sim Uniform[1000,3000];\ L_{it} \sim Uniform[200,1000];\ f_{ij} \sim Uniform[1,5];\ V_{i} \sim Uniform[0.1,1]$$

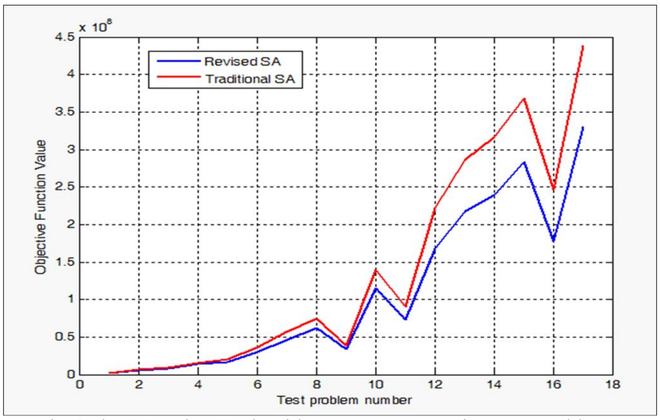

Fig.6.The SA and RSA algorithms Process on various test problems

Table 7
Analysis of variance for test problems with small size

Source	DF	SS	MS	F	P
Small Size	1	5.57361E+12	5.57361E+12	0.13	0.726
Error	8	3.37492E+14	4.21864E+13		
Total	9	3.43065E+14			

```
                                   Individual 90% CIs For Mean Based on
                                   Pooled StDev
Level  N      Mean     StDev   ---------+---------+---------+---------+
RSA    5   9157812   5855295   (--------------*---------------)
SA     5  10650945   7077317       (--------------*---------------)
                               ---------+---------+---------+---------+
                                  7000000  10500000  14000000  17500000
```

Fig.7. The output of analysis of variance for test problems with small size

Table 8
Analysis of variance for test problems with medium size

Source	DF	SS	MS	F	P
Medium Size	1	4.79095E+14	4.79095E+14	0.39	0.549
Error	10	1.24376E+16	1.24376E+15		
Total	11	1.29166E+16			

```
                               Individual 90% CIs For Mean Based on Pooled StDev
Level  N      Mean     StDev      ---+---------+---------+---------+------
RSA    6  60140247  31573733      (------------*------------)
SA     6  72777429  38608414            (------------*------------)
                                  ---+---------+---------+---------+------
                               40000000  60000000  80000000  1.00E+08
```

Fig.8.The output of analysis of variance for test problems with medium size

Table 9
Analysis of variance for test problems with large size

Source	DF	SS	MS	F	P
Large Size	1	1.78834E+16	1.78834E+16	3.45	0.093
Error	10	5.17635E+16	5.17635E+15		
Total	11	6.96469E+16			

```
                                     Individual 90% CIs For Mean Based on Pooled
                                     StDev
Level  N       Mean     StDev          ----+---------+---------+---------+-----
RSA    6  235672854  62450582          (----------*----------)
SA     6  312881226  80328275                      (----------*---------)
                                       ----+---------+---------+---------+-----
                                     2.00E+08  2.50E+08  3.00E+08  3.50E+08
```

Fig.9.The output of analysis of variance for test problems with large size

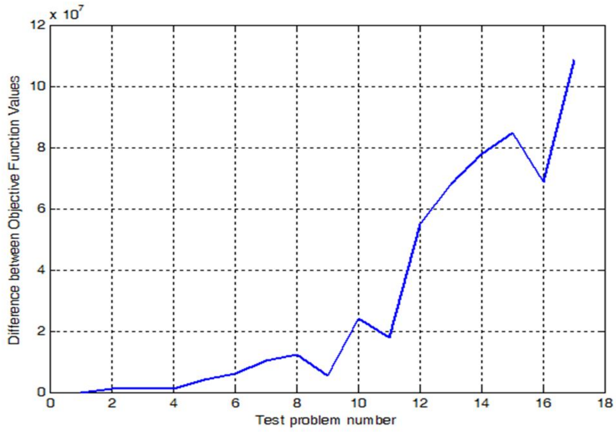

Fig. 10. The differentiation of both RSA & SA in term of OFVs

6. Conclusion and Future Researches

Due to the great importance of production planning problems in today's world, this research attempted to propose a new integer programming model in the MLCLSP. A characteristic of the presented model is that, it pays attention to different operational and industrial constraints in the production level. One of the general aspects of multi-item lot-sizing problems in the literature is matters related to inventory, shortage, and safety stock costs. Due to the proposed model being multi-product, different production manners can exist for various products. Therefore, a new mathematical model in the multi-item lot-sizing problem framework was presented considering setup time, safety stock, shortage, different production manners, and different industrial constraints. The main goal is to optimize production, inventory, and shortage quantities as well as to determine the best production manner in which summations of production, setup, inventory, and shortage costs are minimized. To solve the model, an RSA algorithm was developed in the literature of lot-sizing problems. It is implemented on experimental problems of different dimensions, and then is compared with the classical SA as well as exact approach. Computational results show the suitable performance of the proposed RSA algorithm to reach higher quality answers. For future research in the field of formulation, the problem itself can be transformed into a multi-objective model by adding other goals of service level, so that it gets closer to real world situations.

7. References

Absi, N. and Kedad-Sidhoum, S. (2007). MIP-based heuristics for multi item capacitated lot-sizing problem with setup times and shortage costs. RAIRO–Operations Research, 41, 2, 171–192.

Absi, N. and Kedad-Sidhoum, S. (2008).The multi-item capacitated lot-sizing problem with setup times and shortage costs. European Journal of Operational Research, 185, 1351-1374.

Absi, N. and Kedad-Sidhoum, S. (2009). The multi-item capacitated lot-sizing problem with safety stocks and demand shortage costs. Computer and Operations Research, 36, 2926–2936.

Aksen, D.,Altinkemer, K. and Chand, S. (2003).The single-item lot-sizing problem with immediate lost sales.European Journal of Operational Research, 147, 558–566.

Baker, K.R. (1990).Lot-sizing procedures and a standard data set: a reconciliation of the literature. Journal of Manufacturing and Operations Management, 2,199–221.

Berretta, R. and Rodrigues, L.F. (2004). A memetic algorithm for a multistage capacitated Lot-sizing problem. International Journal of Production Economics, 87, 67-81.

Chen, W. H. and Thizy, J. M. (1990), Analysis of relaxations for the multi-item capacitated lot-sizing problem. Annals of Operations Research, 26,29–72.

Choudhary, D. and Shankar, R. (2011). Modeling and analysis of single item multi-period procurement lot-sizing problem considering rejections and late deliveries. Computers & Industrial Engineering, doi: 10.1016/j.cie.(2011) 08.005.

Han,Y., Tang, J., Kaku, I. andMu, L. (2009). Solving uncapacitated multilevel lot-sizing problems using a particle swarm optimization with flexible inertial weight. Computers & Mathematics with Applications, 57, 1748-1755.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983).Optimization by Simulated Annealing. Science, 220, 671-680.

Loparic, M., Pochet, Y. andWolsey, LA. (2001). The uncapacitated lot-sizing problem with sales and safety stocks. Mathematical Programming, 89, 487–504.

Maes, J. and Van Wassenhove, L. (1988), Multi-item single-level capacitated dynamic lot-sizing heuristics: a general review, Journal of Operational Research Society, 39, 11,991–1004.

Montgomery, D.C. (2005). Design and Analysis of Experiments, Sixth ed., Hoboken, John Wiley & Sons, Inc, New Jersey.

Rezaei, J. and Davoodi, M. (2011), Multi-objective models for lot-sizing with supplier selection, International Journal of Production Economics, 130, 77-86.

Sural, H., Denize, LM. And Van Wassenhove, LN. (2009). Lagrangean relaxation based heuristics for lot-sizing with setup times. European Journal of Operational Research, 194, 1,51–60.

Taguchi, G. (1986). Introduction to Quality Engineering: Designing Quality into Products and Processes, White Plains: Asian Productivity Organization/UNIPUB, Tokyo, Japan.

Tang, O. (2004).Simulated annealing in lot sizing problems. Int. J. Production Economics, 173–181.

Tavakkoli-Moghaddam, A., Rahimi-Vahed, A.R. and Ghodratnama, A. and Siadat, A. (2009). A simulated annealing method for solving a new mathematical model of a multi-criteria cell formation problem with capital constraints, Advances in Engineering Software, 40, 268–273.

Tempelmeier, H. and Derstroff, M. (1996). A Lagrangean-Based Heuristic for Dynamic Multilevel Multi item Constrained Lot sizing with Setup Times. Management Science, 42, 738-757.

Wu, T.,Shi, L.,Geunes, J. and Akartunal, K. (2011). An optimization framework for solving capacitated multi-level lot-sizing problems with backlogging, European Journal of Operational Research, 214, 428-441.