International Journal of Engineering Research and General Science Volume 3, Issue 5, September-October, 2015
ISSN 2091-2730

Large Capacity Constrained Multi Product, Multi Level Lot Sizing Optimization Using binary particle swarm optimization

V.V.D.Sahithi[1], P.Sai Krishna[2] , K.Lalith kumar[3] C.S.P.Rao[4]

1 Research assistant Professor, mechanical Engineering Department, VNR Vignana Jyothi Institute of Engineering And Technology, Hyderabad 500090, India

2 ,3 Research Student, mechanical Engineering Department, VNR Vignana Jyothi Institute of Engineering And Technology, Hyderabad 500090, India

4 Professor Department of Mechanical Engineering, National Institute of Technology Warangal, Warangal 506004, India

E-mail: Sahithi.vaka@gmail.com[1] , Saikrishna.parikirala@gmail.com [2] lalith.143143@gmail.com[3]
csp_rao@rediffmail.com[4]

Abstract

Lot sizing is one of the most important and one of the most difficult to solve problems in production planning and belong to NP hard class of problems. The Capacity Constrained Multi Product, Multi Level Lot Sizing (CC-MPMLLS) belongs to those problems that production industries face in preparing material requirement planning (MRP) systems for executing their production plans is much more complex and is a combinatorial optimization problem aims to find the lot sizes that achieve cost effectiveness. It minimizes the total setup cost and holding cost by finding optimum lot sizes. As the costs of both dependent and independent items of all levels are varying with time. The subjects of single level lot sizing with variants have been addressed by several methods in the literature. Many heuristic methods have been developed to solve lot sizing problems, but most of them are applicable for small instances. Very few approaches are implemented for MPMLLS problems. In this paper we developed a Binary particle swarm optimization (BPSO) programming technique which can easily handle the large with complex product structures with in a reasonable CPU time .The effectiveness of algorithm is tested with variety of simulation experiments by taking both cost effectiveness and computational time into consideration. And Feasibility of BPSO algorithm is investigated by comparing results with binary genetic algorithm (BGA).

Keywords: Binary particle swarm optimization (BPSO), capacitated lot sizing, production planning

1. Introduction

In a Manufacturing production systems, as end products are usually made up of many intermediate items which consists in a combination of purchased parts and raw materials. The end item is therefore described by a bill of material (BOM), which is the recipe of product. For complex product structures number of levels in the BOM is more (multi level structures). So issue of satisfying external and inter demands becomes more complex which is taken care by Material Requirement Planning (MRP) which plays a very important role in coordinating replenishment decisions for complex goods in production system. Its basic philosophy is to ensure that the right numbers of components are available at right time. Lot sizing is one of the important decisions to be taken while preparing MRP.Lot sizing decisions give rise to the problem of identifying when and how much of product to produce such that setup, production and holding costs are minimized. Making the right decisions in lot sizing will affect directly the system performance and its productivity, which are important for manufacturing firm's ability to compete in the market [1].Therefore, developing and improving solutions for lot sizing problems, is very important.

Lot sizing problem attracted the attention because of its impact on the inventory levels and hence the total cost of production. It is basically concerned with finding order quantities of different items in the bill of material structure to minimize the setup cost and holding cost. Lot size might be the amount of production or purchase quantity depending on the demand at different time buckets to ensure and satisfy customer requirements. Minimizing total production cost is always a tradeoff decision between ordering and holding cost. So order quantity in particular period may be (i) requirement of that period or (ii) requirement of that period including with group of requirements of periods ahead or (iii) zero [2].

Lot sizing problems are mainly divided into 2 types like (a) single level lot sizing (SLLS), and (b) multi level lot sizing (MLLS). The number of final products in a production system is another important characteristic that affects the modeling and complexity of production planning problems. There are two principle types of production system in terms of number of products. In single item production planning there is only one end item (final product) for which the planning activity has to be organized, while in multi item production planning, there are several end items. The complexity of multi item problem is much higher than that of single item problems. Resources or capacities in a production planning systems include manpower, equipment, machines, budget, etc.when there is no restriction on resources, the problem is said to be uncapacitated, and when capacity constraints are explicitly stated, the problem is named capacitated. Capacity restriction is important, and directly affects problem complexity. Problem solving will be more difficult when capacity constraints exist. Wagner and Whitin [3] proposed an algorithm in 1958 for single level lot sizing based on dynamic programming to find optimum lot size .a heuristic technique was proposed by silver and meal [4] in 1973 for minimizing the total cost. Mc Knew and Coleman [5] proposed a part period algorithm for minimizing setup and holding cost over different periods. Hernández, W. and G. Süer, [6] proposed a genetic algorithm (GA) for solving single level uncapacitated lot sizing problem with no shortages are allowed. And then they also implemented GA procedure for capacitated multi level problems successfully. N.Dellart, J.Jeunet, N.Jonard [7] successfully applied genetic algorithm for solving large multi level multi level lot sizing problems. Taˇsgetiren and Liang [8] presented a technique particle swarm optimization (2003) to minimize the inventory setup and holding cost for setup and holding cost minimization of simple product structures. Klorklear Wajanawichakon& Rapeepan Pitakaso [9] implemented binary PSO (2011) for multi level unconstrained problems of general product structures.

In this paper, the authors have considered a very large complex product structure of a multi product multi level lot sizing problem and Capacity constraints are also taken into consideration. Thus a class of CC-MPMLLS problems were considered and attempted to solve by modeling and simulations using Binary Particle Swarm Optimization Algorithm.

The Paper is organized in six sections: section2: mathematical formulation of CC-MPMLLS problem section3: Binary Particle Swarm Optimization (BPSO) model representation .section 4: numerical example .finally section5: problem illustration section6: conclusion.

2. Mathematical Formulation of CC-MPMLLS problem

The lot sizing problem in this can be described as follows. There are N items to be produced in T periods in a planning horizon such that a demand forecast would be attained. In multi stage production systems, the planning of each item depends on the production of other items, which are situated at lower hierarchical levels in bill of material structure. When we decided to produce one item a fixed cost and time is incurred. The resources for production and setup are limited. Lead times are assumed to be zero.

Let **N** be the number of types items to be produced (i=1, 2, 3…..n), **T** represents the number of periods in planning horizon (t=1, 2, 3…T). $\mathbf{C_{it}}$ is unit production cost of item **i** in period **t**, $\mathbf{S_{it}}$ means the setup cost of item **i** in period **t**,$\mathbf{H_{it}}$ the holding cost of item **i** in period **t**, $\mathbf{r_{ij}}$=number of units of item **i** required to produce 1 unit of j, $\mathbf{d_{it}}$=external demand for item i in period t, $\mathbf{C_{it}}$=unit production cost of item **i** in period **t**,$\mathbf{CA_{it}}$=available capacity of item **i** in period **t**. $\mathbf{X1_{it}}$ is the lot size of item **i** produced in period **t**. I_{it} is the inventory of item **I** in period **t**. $\mathbf{V_{ikt}}$ represents amount of item **k** required to produce item **i** in period **t**.$\mathbf{f_{it}}$ is fixed amount of item **i** required to produce in period **t**.

Min (Total cost) =$min[f(x)] = \sum_{i=1}^{n} \sum_{t=1}^{T} (C_{it}X1_{it} + S_{it}B_{it} + H_{it}I_{it})$ …………… 1

$$I_{i,t-1} + X_{it} - I_{it} = d_{it} + \sum_{j \in S(i)} r_{ij} X_{jt} \qquad \text{....………… 2}$$

i=1, 2, 3…..N; t=1, 2, 3…T,

$$B_{it} = 0 \quad \text{if } X1_{it}=0$$
$$=1 \quad \text{if } X1_{it}>0 \qquad \text{………….…….…. 3}$$

$$I_{it,}\ X1_{it} \geq 0 \qquad \text{……………….... 4}$$

$$\sum_{i=1}^{N} (V_{ikt}X1_{it} + f_{ikt}B_{it}) \leq CA_{it} \qquad \text{................….... 5}$$

i=1, 2, 3…..N; t=1, 2, 3…T.

Here the objective function i.e.Equation-1 represents to minimize sum of production, setup, and inventory holding cost of all n items in **T** periods. Equation-2 represents an inventory balance constraints which describe the relation between inventory and production at the beginning and the end of periods. Equation-5 represents the capacity limitations of production and setup. Equation-3 is for binary variable which represents setup is made in period t or not for item i.e. Equation-6 represents that variables must be positive.

Several factors like ordering cost, holding cost, shortage cost, capacity constraints, minimum and maximum order quantity etc... Combination of these factors result in different models to be analyzed like capacitated or uncapacitated, single level or multi level, single item or multi item models.simple single product structures can be solved easily using mathematical equations .as CCMPMLLS problems are having very large solution space they are considered as NP-hard problems that does not have solution with polynomial time. So soft computing techniques are necessary to compute optimum values of lot sizes.

Taˇsgetiren and Liang [8] presented a technique particle swarm optimization (2004) to minimize the inventory setup and holding cost for setup and holding cost minimization of simple product structures. Klorklear Wajanawichakon& Rapeepan Pitakaso [9] implemented binary PSO (2011) for multi level unconstrained problems of general product structures.

In this paper authors have made an attempt to solve very large complex product structure of capacity constrained multi product multi level lot sizing problem. A binary PSO approach is used to model and simulate CC-MPMLLS problem and solved the same with time and solution efficiency. The authors have solved the same problem using Genetic Algorithm. The results of GA and BPSO are compared for the same set of problems under consideration.

3. Binary particle swarm optimization model representation

(a)Initial solution representation

Solution representation of particle p, X^{pk}_{id}, for BPSO $_{is}$ given in Table1.This representation is due to Hernández and Suer (1999). Where each swarm contains 'P' number of particles referring to d dimensions and 'i' items .here 'k' represents iteration number.

A population of binary values (0 or 1) are randomly assigned for't' dimensions of 'i' items in the MLLS problem for all 'p' particles which gives the information about where setups are made.

If $R_{id} > 0.5$ then $X_{id}=1$

Else $X_{id}=0$;

R_{id}=random value.

i= item number=1, 2, 3…n

k=iteration number=1, 2, 3…..k

d=period=1, 2, 3……t

For initial generation k=0, i.e. $X_{id} = X^0_{id}$

Table1.Representation of Particle

	1	**2**	**3**	**4**	**5**	**………..**	**12**
X^{pk}_{1d}	1	0	1	1	0	……….	1
X^{pk}_{2d}	-	-	-	-	-	……….	-
X^{pk}_{3d}	-	-	-	-	-	-	-
.	-	-	-	-	-	-	-

Lot size:

According to particle solution lot sizes are calculated as shown in Table 2.Time periods where demand is not "0" there set up has been made. So order quantity in particular period may be (i) requirement of that period or (ii) requirement of that period including with group of requirements of periods ahead or (iii) zero.

	1	**2**	**3**	**4**	**5**	**………..**	**12**
L^{pk}_{1d}	140	0	155	175	0	…………	115
L^{pk}_{2d}	-	-	-	-	-	-	-
L^{pk}_{3d}	-	-	-	-	-	-	-
.	-	-	-	-	-	-	-

Table2. Lot size according to particle dimension

L^K_{id} =lot size of item i ordered in period d at iteration k of particle p.

(b) Velocity of initial generation particles

After assigning particle dimensions, velocity values need to be calculated as shown in Table3, to find next generation population. This velocity calculation is of 2 types i.e. 1) velocity calculation for initial generation (2) Velocity calculations for remaining generations.

Velocity values are restricted to some minimum and maximum namely

$V^{pk}_{id} = [V_{mini}, Vmaxi] = [-5, 5]$.

V^{pk}_{id} =velocity of particle of period d at iteration k

For initial genetaion velocity values are calculated using following formula

$$V^{0p}_{id} = V_{mini} + (V_{maxi} - V_{mini}) * R$$

R=a random value within 0 to 1, which is generated using rand ().

Table 3. Velocity matrix of particle

	1	2	3	4	5	………..	12
V^{pk}_{1d}	-1.8	3.7	2.9	-0.69	-3.1	………..	1.2
V^{pk}_{2d}	-	-	-	-	-	-	-
V^{pk}_{3d}	-	-	-	-	-	-	-
V^{pk}_{id}	-	-	-	-	-	-	-

(c)Particle best and global best

Particle having best fitness value [$f(x^k_p)$] is assigned to global best .As it is the initial generation all particle best values are equal to particle values as shown in Table 4.

Table 4. Particle and global best matrices

	1	2	3	4	5	………..	12
PB^{pk}_{1d}	1	0	1	1	0	……….	1
PB^{pk}_{2d}	-	-	-	-	-	……….	-
PB^{pk}_{3d}	-	-	-	-	-	-	-
PB^{pk}_{id} PB^{K}_{id}	-	-	-	-	-	-	-

	1	2	3	4	5	………..	12
GB^{k}_{1d}	1	0	1	1	0	……….	1

(d)Updating parameters for next generations:

(i)Updating velocity (V^{pk}_{id}):

(I) new velocity = V^{pk}_{id}=P ($V^{p,k-1}_{id} + \Delta V^{p,k-1}_{id}$)

Where $\Delta V^{p,k-1}_{id}$= c1 R1 ($PB^{p,k-1}_{id} - X^{p,k-1}_{id}$)+ c2 R2($GB^{K-1}_{id} - X^{p,k-1}_{id}$)

C1, c2 are social and cognitive parameters, R1& R2 are uniform random numbers between (0, 1)

Here Piece wise linear function [P (V^{pk}_{id})]

P (V^{pk}_{id}) = Vmaxi if V^{pk}_{id} > Vmaxi

=V^{pk}_{id} if $|V^{pk}_{id}| \leq V_{maxi}$

=Vmini if V^{pk}_{id} < Vmini

(ii)Updating position (X^{pk}_{id}) by sigmoid function:

X^{K}_{id}= 1 if R< S (V^{pk}_{id})

=0 otherwise

Sigmoid function S (V^{pk}_{id}):

This function forces velocity values to be in the limits of '0' to '1'.It helps to update next generation

X^{pk}_{id} values. **S (V^{pk}_{id}) =**$\frac{1}{1+e^{-Vpk\,id}}$

(iii)Updating particle best and global best (PB^{pk}_{id} ,GB^{K}_{id})

After each and every iteration update particle best and global best values according to the fitness values of particles in the newly generated swarm.

(e) Terminination: If the number of iterations reaches a predetermined value, called maximum number of iterations then stop searching, other wise go to (d).

4. Numerical Example:

In this example two items are there in which item-2 is having independent demand and other item (i.e.item-1) demand depends on first one.Table5a and 5b represents the demands and also depicts different costs involved in the problem.

Figure 1 represents BOM structure.

Table 5a Product demand

Period	1	2	3	4	5	6
Demand	20	100	50	30	10	100

Table 5b setup and holding cost

Item No.	Setup cost	Holding cost
1	500	50
2	100	10

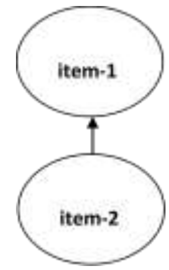


Figure 1.BOM structure

The problem is mapped and executed in terms of BPSO and the steps are given below:

Step1: initial generation

Particle 1

Item1	1	0	0	0	1	1
	200	0	0	0	10	100
	2.9	-1.8	3.5	-1.2	0.7	3.8
Item2	1	0	0	0	1	0
	200	0	0	0	110	0
	-1.5	1.6	-2.7	3.3	1.8	-3.9

Fitness fuction=$f(x^0_1)$=17200

Particle 2

Item1	1	1	1	0	0	1
	20	100	90	0	0	100
	-2.5	1.7	3	-4	1.6	2.3
Item2	1	0	0	0	0	1
	210	0	0	0	0	100
	2.2	-3.9	1.4	3	-4.6	2

Fitness function=$f(x^0_2)$ =7500

Particle 3

Item1	1	0	1	0	1	1
	120	0	80	0	10	100
	3	-4.6	2	-1.2	0.7	3.8
Item2	1	0	1	0	1	0
	120	0	80	0	110	0
	-3	-4	1.6	2.9	-1.8	3.5

Fitness fuction=$f(x^0_2)$ =9800

Step 2: As it is initial generation, assign each particle in the swarm to particle best (PB)

$PB^{1,0}_{1,1}=X^{1,0}_{1,1}$, $PB^{1,0}_{1,2}=X^{1,0}_{1,2}$………………………… $PB^{1,0}_{1,12}=X^{1,0}_{1,12}$

$PB^{1,0}_{2,1}=X^{1,0}_{2,1}$, $PB^{1,0}_{2,2}=X^{1,0}_{2,2}$………………………… $PB^{1,0}_{2,12}=X^{1,0}_{2,12}$

	D	1	2	3	4	5	6	Fitness
$PB^{1,0}_{id}$	i=1	1	0	0	0	1	1	17200
	i=2	1	0	0	0	1	0	
$PB^{2,0}_{id}$	i=1	1	1	1	0	0	1	7500
	i=2	1	0	0	0	0	1	
$PB^{3,0}_{id}$	i=1	1	0	1	0	1	1	9800

	i=2	1	0	1	0	1	0	

Assign best fitness particle to global best

	D	1	2	3	4	5	6	Fitness
GB_{id}^{0}	i=1	1	1	1	0	0	1	7500
	i=2	1	0	0	0	0	1	

Step 3: Updating velocity using piece wise function

Assume C1=C2=1, r1=r2=0.5;

Update particle dimension

$\Delta V^{1,0}_{12}= c1\ R1\ (PB^{1,0}_{12} - X^{1,0}_{12}) + c2\ R2(GB^{0}_{12} - X^{1,0}_{12})$

$\Delta V^{1,0}_{12}=1*0.5(0-0)+1*0.5(1-0)= 0.5$

$V^{1,1}_{12}=P\ (V^{1,0}_{12}+\Delta V^{1,0}_{12})=P(-1.8+0.5)=-1.3$

Updating particle position:

R (0, 1)=0.11 < Sigmoid(-1.3)=0.21

So new dimension value= $X^{1}_{12}=1$

- After completion of velocity calculations of all dimensions, particles are updated as follows

	D	1	2	3	4	5	6	fitness
$X^{1,1}_{id}$	i=1	1	1	1	1	0	1	6400
	$V^{1,1}_{1d}$	2.9	-1.3	4	-1.2	0.2	3.8	
	$Sig(V^{1,1}_{1d})$	0.94	0.21	0.98	0.23	0.54	0.97	
	Random	0.72	0.11	0.4	0.2	0.67	0.8	
	i=2	1	0	0	0	0	1	
	$V^{1,1}_{2d}$	-1.5	1.6	-2.7	3.3	1.3	3.3	
	$Sig(V^{1,1}_{2d})$	0.18	0.83	0.06	0.96	0.78	0.96	
	Random	0.10	0.91	0.10	0.99	0.80	0.91	
$X^{2,1}_{id}$	i=1	1	1	1	1	0	1	3500
	i=2	1	1	1	1	0	1	
$X^{3,1}_{id}$	i=1	1	0	0	0	1	1	9800
	i=2	1	0	0	0	1	0	

Updated particle best matrix

	D	1	2	3	4	5	6	Fitness
$PB^{1,1}_{id}$	i=1	1	1	1	1	0	1	6400
	i=2	1	0	0	0	0	1	
$PB^{2,1}_{id}$	i=1	1	1	1	1	0	1	3500
	i=2	1	1	1	1	0	1	
$PB^{3,1}_{id}$	i=1	1	0	1	0	1	1	9800
	i=2	1	0	1	0	1	0	

Global best matrix

	D	1	2	3	4	5	6	Fitness
GB_{id}^{1}	i=1	1	1	1	1	0	1	3500
	i=2	1	1	1	1	0	1	

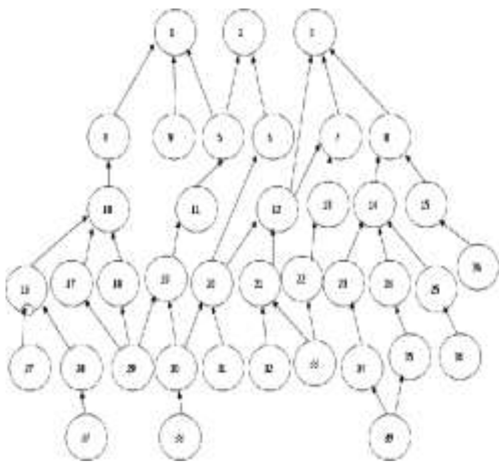
Step 4: Termination

Repeat this procedure (step3) until iteration number k< max iteration.

5. Problem Illustration:

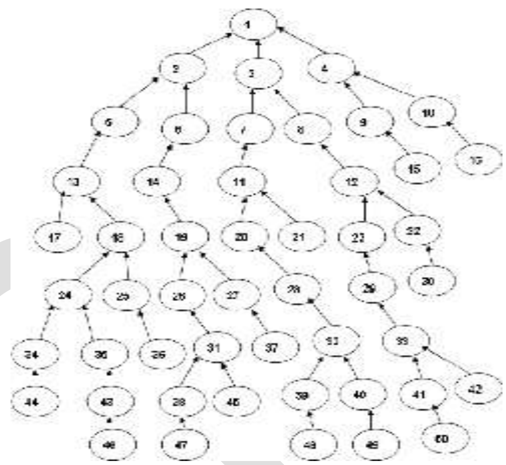
5.1. Problem

Problems shown in Figure 2a and 2b as M×T are taken for modeling and simulation of CC MLLS problem. Here M represents the total number of items involved in the BOM structure and T represents the number of periods. Table 6 represents different costs involved and Table 7and 8 carries information regarding demand and available capacity. Figure 2a is a BOM of single product where it contains 50×12 contains 50 different items in 9levels and, Figure 2b is a BOM of a multi product contains 39×12 structure with 39 different items in6 levels. Table 6 gives the information regarding the setup cost and holding costs of different items of both 50 ×12and39×12 problems. Table 7 gives the information regarding the demand and availability of end product of single product problem. Table 8 gives the information of demand and availability of end products of multi product problem.



50×12

Figure 2a Product structures of a single product



39×12

Figure2b Product structures of multi product

Table 6.Setup and Holding costs of different items in 50×12, 39×12 structures

S.No	50*12 problem		39*12 problem		S.NO	50*12 problem		39*12		S.NO	50*12		39*12	
	H.C	S.C	H.C	S.C		H.C	S.C	H.C	S.C		H.C	S.C	H.C	S.C
1	97.83	780	40.08	490	18	23.71	510	7.13	860	35	6.38	160	4.83	690
2	45.19	200	35.27	450	19	15.32	910	8.82	850	36	3.47	290	3.44	430
3	43.82	590	59.66	90	20	20.58	830	10.6	670	37	1.97	420	0.91	60
4	5.82	710	25.42	140	21	8.71	730	6.02	370	38	1.76	160	2.64	760
5	26.04	890	10.42	880	22	3.14	850	2.78	360	39	6.41	450	2.65	180
6	18.87	610	22.64	440	23	0.94	450	2.95	310	40	7.17	340		
7	27.03	920	22.31	70	24	13.02	370	9.32	440	41	2.97	750		
8	15.64	210	19.53	430	25	7.34	390	0.31	590	42	0.25	140		
9	2.67	490	1.34	930	26	7.53	540	1.45	580	43	3.22	430		
10	1.86	920	25.12	650	27	4.36	160	3.63	650	44	1.85	890		
11	23.5	520	9.46	740	28	18.52	480	4.35	450	45	3.84	610		
12	12.59	540	17.48	680	29	5.81	410	3.29	820	46	0.41	860		
13	25.13	510	4.32	800	30	1.93	140	5.04	620	47	0.37	860		
14	16.42	500	14.28	220	31	6.71	390	2.53	580	48	3.84	350		
15	0.84	300	2.56	850	32	15.35	370	3.3	340	49	3.95	610		
16	1.02	450	10.07	400	33	4.36	520	0.61	340	50	1.63	350		
17	0.62	440	4.59	650	34	3.28	700	2.52	80					

Table 7.Demand and Availability of end product in 50×12 problem

Period	1	2	3	4	5	6	7	8	9	10	11	12
Demand	15	5	15	110	65	165	125	25	90	15	140	115
Available	1000	2000	1000	0	5000	1000	0	500	800	500	1000	200

Table 8.Demand and Availability of end products in 39×12 problem

Period	1	2	3	4	5	6	7	8	9	10	11	12

Item1	10	100	10	130	115	150	70	10	65	70	165	125
available	1500	2000	0	1000	800	5000	0	800	500	1000	2000	200
Item2	175	15	85	90	85	90	75	150	75	10	150	15
available	0	1000	2000	1000	900	0	800	1200	500	500	1000	100
Item3	135	165	15	105	25	120	50	60	5	140	60	10
available	1000	2000	900	800	0	1000	1200	300	500	800	100	100

5.2. Experimental Parameters:

The Binary particle swarm optimization for capacitated large size lot sizing problem is coded in c language and run on Intel® Core™ Duo processors 667 MHz Front Side Bus and 2M Smart L2 Cache with 2GB RAM. Performance of BPSO is compared with a traditional Binary Genetic Algorithm.

Binary Genetic Algorithm Parameters:

In solving MLLS problems using Binary GA, multi point cross over is considered .Cross over of different types like product and periodic crossovers have been applied and single bit mutation is used. The cross over and mutation ratios considered are 0.80 and 0.10 respectively and 0.1 percentage of reproduction is taken.

Binary Particle swarm optimization:

For better convergence population size of BPSO should be at least twice the number of periods that are considered i.e. 24.In this paper population size is taken as 40 i.e. swarm size .means 40 different particles are considered ,as solution proceeds further particle fly around the solution space with different velocities and tries to reach optima.

Following Table 9 shows the effect of social cognitive parameters on the total cost i.e. fitness

Table 9.Effect of social cognitive parameters on cost

C1	C2	Avg cost
5	1	213882.63
1	1	207562.63
1	5	207507.96
3	2	202810.70
1	3	201254.17
2	5	201267.15
5	2	101047.38
5	1	201047.38
4	3	194724.83
3	5	191770.14
2	2	191520.84
3	3	190495.42

From this table 9 we can understand that the average cost obtained with different c_1, c2 values are more or less similar. So difference of various social cognitive values and the result of average total cost had no effect. Thus c1=c2=2 is chosen.

5.3. Simulation Results at different iterations tested in large –CCMPMLLS:

Table 10 shows the average total costs obtained by BPSO algorithm at different iterations of capacity constrained 39×12 MPMLLS problem.

Table 10.simulation results of different iterations

Particle No	300iterations	400 iterations	500 iterations	600 iterations
1	287163.81	296827.84	271354.18	252984.90
2	336654.96	367410.31	289081.71	315492.96
3	293118.93	269643.84	245523.12	284151.09
4	306763.81	302162.75	262019.62	243960.09
5	345834.90	248838.14	314560.21	329510.21
6	282012.65	266686.15	256661.42	281553.00
7	315013.18	272906.25	261492.73	352466.06
8	296122.84	348605.81	283168.21	288089.43
9	248314.07	301639.96	305530.84	254905.98
10	247896.62	257145.29	303904.59	243397.90
11	277441.78	269856.78	384417.81	292368.90
12	362870.81	306533.00	276282.09	298579.78
13	321958.31	371249.90	294672.75	275554.75
14	278471.87	271883.00	274612.28	296049.25
15	260861.73	311140.62	297362.34	272623.87
16	285929.62	286662.00	269972.37	299126.06
17	263018.00	289455.09	351597.21	266942.46
18	278906.21	264557.53	284343.62	267243.21
19	302362.43	298466.21	285951.65	253033.96
20	314437.37	274489.31	295188.09	240238.76
21	304807.06	304971.06	392710.34	266432.03
22	259932.62	261123.39	328282.68	284568.34
23	271942.18	259024.79	261638.71	357821.31
24	232535.56	267100.62	326993.50	294543.37
25	313567.40	259005.15	280591.90	306715.31
26	296707.34	268255.06	243848.54	358286.62
27	277164.31	281731.18	264352.62	276793.18
28	261792.07	274739.06	289422.71	299171.06
29	380921.25	256552.51	277678.18	258015.42
30	292137.40	266886.78	279697.71	256473.42
31	271272.21	280590.34	287545.06	319100.50
32	249626.34	265390.56	328519.46	262788.46
33	306842.96	251897.54	350235.21	315292.06
34	264361.06	294248.43	247796.82	241895.12
35	269949.12	326534.81	337587.68	248329.37
36	321372.84	248451.96	243114.26	241233.20
37	295246.15	261236.87	256835.37	219248.84
38	272370.71	296141.87	338252.03	323139.43
39	275238.12	346146.93	258783.17	285475.75
40	326699.65	259180.64	281134.03	283344.40
Avg	291241.00	285134.20	288971.5	282673.50

5.4. Comparison of results:

Following Tables11 and 12 shows the comparison of binary GA and PSO algorithms at different iterations of given CC-MLLS problems. Table 11 and Figure3 give the comparision of solution efficiency of single product problem between BGA and BPSO. Following table 12and Figure4 gives the comparision of solution efficiency of multi product problem between GA and BPSO.

Table 11.comparision 50×12 problem results between BGA and BPSO

Iteration No.(K)	**BGA**	**BPSO**

1	386785.09	250295.00
10	380891.31	243797.00
50	350503.75	203956.09
100	322136.16	193128.11
200	279484.72	192017.59
500	249875.41	189013.95
1000	234587.08	186579.11
2000	234587.08	186543.84
5000	234489.03	185042.16
10000	229484.6	184629.19
15000	229484.6	183973.11
20000	204240.90	181685.31
30000	204140.90	181685.31

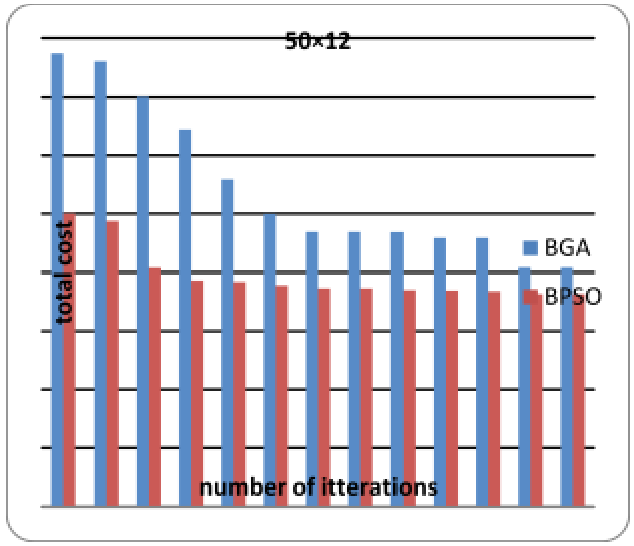


Figure3. BGA and BPSO comparison

Table 12.comparision 39×12 problem results between BGA and BPSO

Iteration No.(K)	BGA	BPSO
1	377421.19	246901.17
10	327867.12	217583.656
50	242463.20	204084.98
100	221525.29	202884.17
200	199022.79	194724.843
500	197410.34	193219.70
1000	197410.34	185691.15

2000	197410.34	185691.15
5000	197410.34	175684.78
10000	197410.34	172682.56
15000	197410.34	172682.56

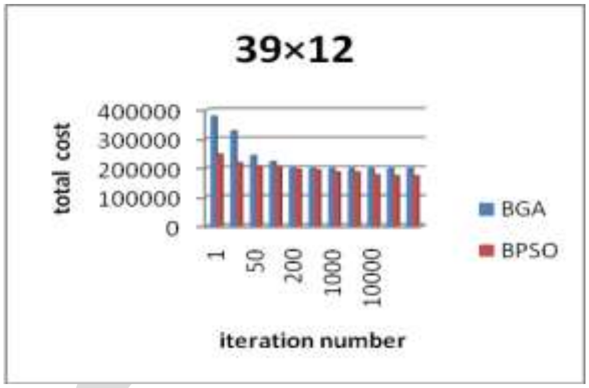


Figure4. BGA and BPSO comparison

Figure 5 represents the convergence of BPSO algorithm for two different problems of capacity constrained multi level lot sizing. The optimum solution i.e. totalcost for both single product (50×12) and Multi product (39×12) problems are given in Table 13and the following conclusions are drawn.

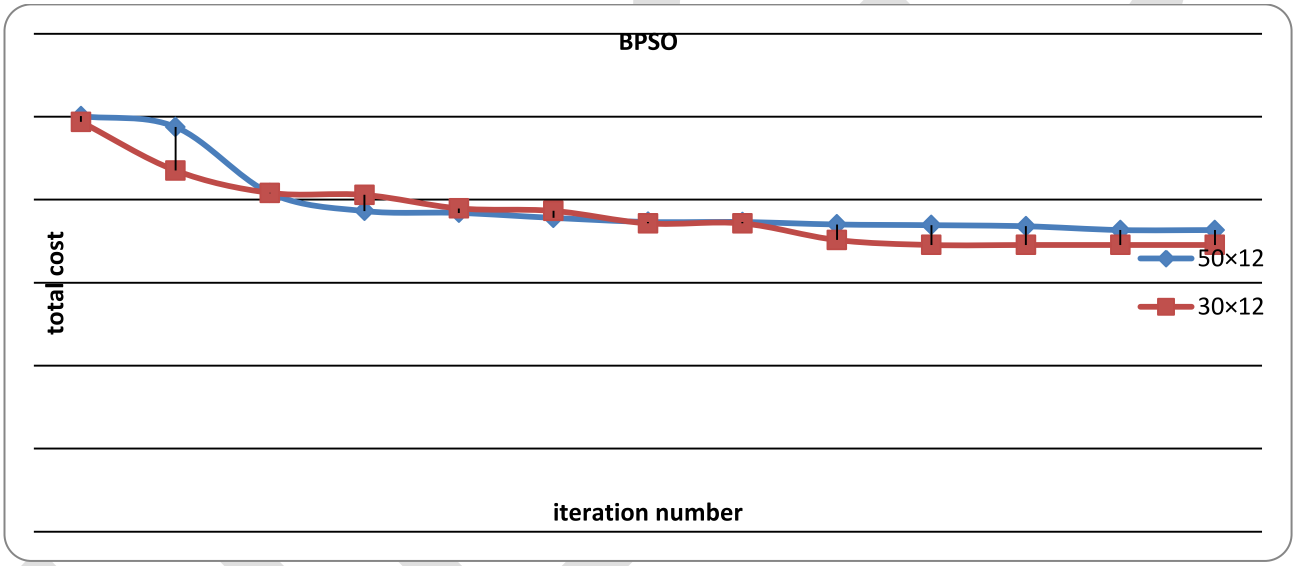


Figure 5. Convergence of BPSO

Following Table 13 gives the information about the Best solutions obtained for both single product & multi product problems of multi level using BGA and BPSO.

Table 13.Optimum solution Obtained by BPSO for MLLS problems

	BGA total cost	BPSO total cost	% of improvement
50 × 12	204,140.90	181,685.31	11
39 × 12	197,410.34	172,682.56	12.5

6. Conclusions

1. BPSO technique has been successfully employed to model and simulate CC-MPMLLS problem to minimize total cost. Two problems are single product with multi level multi item nature and other problem with three end products, multi item multi levels were considered and tested thoroughly with BPSO algorithm.BGA method is also implemented to solve the above problems and to compare with BPSO method
2. BPSO algorithm converges with in 10000 iterations for the problem under consideration and thus the developed algorithm is time efficient.

3.The solution obtained for the two problems under consideration by BGA and BPSO methods, it is observed that through both BGA and BPSO are successful methods in obtaining solutions, the solution obtained by BPSO is more efficient i.e. 11% improved in the case of 50×12 problem and 12.5% in39×12 problem

4. Computational experience show that the methodology can be implemented as a separate optimization module for solving all types of lot sizing problems in any MRP-II based package.

REFERENCES:

[1]B.Karimi, S.M.T. Fatemi Ghomi, J.M.Wilson," The capacitated lot sizing problem: a review of models and algorithms", the international journal of Management science, Omega 31(2003) 365-378.

[2] N.Dellart, J.Jeunet, Randomized multi-level lot-sizing heuristics for general product structures, European Journal of Operational Research 148 (2003) 211–228.

[3] Wagner, H. and Whitin, T. (2004). Dynamic version of the economic lot size model. Management Science, 50(12):1770–1774.

[4] Silver, E. and Meal, H. (1973). A heuristic for selecting lot size requirements for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. Production and Inventory Management, 14(2):64–74.

[5] Coleman, B.J., McKnew, M.A., 1991. An improved heuristic for multilevel lot sizing in material requirements planning. Decision Sciences 22, 136–156.

[6] Hernández, W. and G. Süer, "Genetic Algorithms in Lot Sizing Decisions", Proceedings of the Congress on Evolutionary Computing (CEC99), Washington DC, July 6-9, 1999.

[7] N.Dellart, J.Jeunet, A genetic algorithm to solve the general multi-level lot sizing problem with time varying costs. International journal of production Economics68 (2000)241-257.

[8]M.Fatih Tasgetiren and Yun-Chia Liang(2003),A binary particle swarm optimization algorithm for lot sizing problem, journal of Economic and Social Research5(2),1-20.

[9] Klorklear Wajanawichakon, Rapeepan Pitakaso, Solving large unconstrainted multi level lot-sizing problem by a binary particle swarm optimization. International Journal of Management Science and Engineering Management, 6(2): 134-141, 2011