Received: September 12, 2019. Revised: January 30, 2020. Accepted: February 26, 2020. Published: March 5, 2020.

A Particle Swarm Optimization and Golden Section Search Based Hybridization Scheme to Solve Economic Lot Scheduling Problem Using Basic Period Approach

Syed HasanAdil
Department of Computer Science
Iqra University
Karachi, Pakistan
hasan.adil@iqra.edu.pk

Kamran Raza
Department of Computer Science
Iqra University
Karachi, Pakistan
kraza@iqra.edu.pk

Syed SaadAzhar Ali
Department of Computer Science
Iqra University
Karachi, Pakistan
saadazhar@iqra.edu.pk

***Abstract*—In this paper we suggest a hybridization scheme to solve Economic Lot Scheduling Problem (ELSP) using basic period approach. We proposed a hybrid approach based on Particle Swarm Optimization(PSO)to find the optimum value of k_i's and Golden Section Search (GSS) to find the optimum value of basic period *T*. The proposed hybridized scheme is compared with the best known Genetic Algorithm (GA) on Bomberger'sdataset. This hybridapproach is found competitive and efficient in solving Economic Lot Scheduling Problem and outperform the Genetic Algorithm on problems with lower machine utilization as well as higher machine utilization.**

Keywords-Economic Lot Scheduling Problem; Basic Period Approach;Particle Swarm Optimization; Golden Section Search.

I. INTRODUCTION

The Economic Lot Scheduling Problem (ELSP) has been under research for more than four decades. The problem is computationally very complex and has been classified as NP-hard problem [1]. Despite its complexity the ELSP has been encountered in most production planning scenarios. Due to NP hardness of the problem many researchers have developed heuristic solutions to the problem. There are four approaches to solve the ELSP problem: common cycle [7]; basic period [4]; extended basic approach [3]; and time varying lot size approach [6].

As the ELSP is generally viewed as NP-hard, the focus of most research efforts has been towards generating near optimal repetitive schedule(s). To date, several heuristic solutions [4, 9, 10, 11, 12, 18] have been proposed using any one of the common cycle, basic period, extended basic approach, or time-varying lot size approaches. The common cycle approach always produces a feasible schedule and is the simplest to implement, however, in some cases the solution when compared to the lower bound is of poor quality [16]. Unlike the common cycle approach, the basic period approach allows different cycle times for different products, however, the cycle times must be an integer multiple of a basic period. Although the basic period approach generally produces a better solution to ELSP than common cycle approach, but getting a feasible schedule is NP-hard [1]. The BP approach assumes that the production runs of all products shall be made in each basic period. Then,the basic period must be long enough to accommodate the production of all the products. This is a rather restrictive condition which usually results in suboptimal solutions. The extended basic period approach removes this restriction and admits the possibility that in any basic period only a subset of the products shall be produced [14, 15]. This obviates the waste of capacity of the production facility. Lastly, the time-varying lot size approach is more flexible than the other two approaches, allowing for different lot sizes for the different products in a cycle [16]. Dobson [6] showed that the time-varying lot size approach always produced a feasible schedule as well as giving a better quality solution.

The proposed research is motivated by the recent success [4, 9, 10, 11, 12, 18] of the meta-heuristics to solve ELSP.Therefore, this research investigates the use of meta-heuristics to solve the ELSP problem using basic period approach. We applied Particle Swarm Optimization with Golden Section Search (GSS)to find the solution and compared with existing Genetic Algorithm (GA) [4] based best known solution. The two meta-heuristics will be compared in order to calibrate their performance in regards to solution quality produced and computation time needed.

II. BASIC PERIOD APPROACH TO ELSP

We present ELSP model [1] which is based on the basic period approach. We have to produce *m* distinct products on single production facility with the following assumptions.

- The competing products for production facility do not have any precedence over each other.
- Back-orders are not allowed.
- An item is considered for production only if its inventory is depleted to the zero level. This rule is known as Zero-Switching-Rule (ZSR).
- The production facility is assumed to be failure free and to always produce perfect quality products.

The solution of the ELSP is based on specifying an inventory cycle for each part, subject to following conditions:

- The quantity of a part produced during its cycle must be sufficient to meet demand over the cycle.
- The length of the cycle must be sufficient to permit the production of other parts scheduled during the cycle.

A schedule is feasible if the above conditions are met. This feasible solution becomes optimal if the total costminimizes.

The following notations are used:

i	:	An item index, i={1,2, …,n}
D_i	:	Annual demand for item i (units/ year)
P_i	:	Annual production rate for item i (units/year)
H_i	:	Holding cost for item i ($/unit-year)
S_i	:	Setup cost for item i ($/setup)
τ_i	:	Setup time for item i (years)
Q_i	:	Production quantity for item i, a decision variable (units)
T_i	:	Cycle time for item i, a decision variable (in days)
TC_i	:	Total annual holding and setup cost for item i ($/year)
TC	:	Total annual holding and setup cost for all item ($/year)

The total cost for an item $i$is:

$$TC_i = \frac{Q_i}{2}\left(1 - {}^{D_i}/_{P_i}\right)H_i + \left({}^{D_i}/_{Q_i}\right)S_i \tag{1}$$

The total annual cost of all n items is:

$$TC = \sum_{i=1}^{n}\left[\frac{Q_i}{2}\left(1 - {}^{D_i}/_{P_i}\right)H_i + \left({}^{D_i}/_{Q_i}\right)S_i\right] \tag{2}$$

The ELSP is formulated as follows:

Minimize TC

$$\text{Subject to} \sum_{i=1}^{n}\left(\left({}^{D_i}/_{P_i}\right)\tau_i + \frac{D_i}{P_i}\right) \leq 1 \tag{3}$$

No two items are produced at the same time (4)

The first constraint ensures that the time spent setting up the machine and producing the items does not exceed the time available. Solving the unconstrained problem results a loose lower bound known as the independent solution (IS). The optimal order quantity for item $i$is:

$$Q_i^* = \sqrt{\frac{2\,D_i S_i P_i}{H_i\left(1 - {}^{D_i}/_{P_i}\right)}} \tag{5}$$

Substituting from equation (5) into equation(2) gives IS lower bound on the ELSP as follows:

$$TCIS = \sum_{i=1}^{n}\sqrt{\frac{2\,D_i S_i H_i}{(P_i - D_i)P_i}} \tag{6}$$

Alternatively, a tighter lower bound (TCL) can be obtained by minimizing the total cost (TC) subject to constraint in equation (3):

$$Q_i^* = \sqrt{\frac{2\,D_i P_i (S_i + \lambda\,\tau_i)}{H_i(P_i - D_i)}} \tag{7}$$

Andsatisfying:

$$\lambda\left(\sum_{i=1}^{n}\frac{\tau_i D_i}{Q_i} + \sum_{i=1}^{n}\frac{D_i}{P_i} - 1\right) = 0 \tag{8}$$

In case if the production facility in under-utilized, the capacity constraint will not be binding and TCL will be same as TCIS. However, with the higher utilization, TCL is higher than the IS lower bound. The increase in TC and TCL relative to TCIS at high utilization is due to production quantities becoming larger to reduce the time spend on setup, which substantially increases the holding cost.

Now, we discuss an analytical approach which allows achieving the optimal solution to a restricted version of the original problem mentioned in [2, 3]. The approach is called basic period approach. In basic period approach, the cycle time for every item i is an integer multiple k_i of a fundamental cycle T. Thus, the cycle time for an item $i$is:

$$T_i = k_i T \tag{9}$$

Also the production quantity for an item i will becomes:

$$Q_i = T_i D \tag{10}$$

The total cost incurred under basic period approach (TCBP) is obtained from substituting T_i and $Q_i$into equation (2). Thus, the total cost is:

$$TCBP = \sum_{i=1}^{n} T k_i D_i\left(1 - \frac{D_i}{P_i}\right)\frac{H_i}{2} + \frac{S_i}{T\,k_i} \tag{11}$$

TCBP established in Equation (11) is now a function of T and k_i's. Once TCBP is established, the ELSP under BP approach is:

Minimize TCBP

$$\text{Subject to} \sum_{i=1}^{n}\left(\tau_i + \frac{D_i\,T\,k_i}{P_i}\right) \leq T \tag{12}$$

The constraint in the above optimization problem ensures that the fundamental cycle is long enough to accommodate the production of all items even though not every item has to be produced during every fundamental cycle. The constraint guarantees the feasibility but may result in a suboptimal solution to the original problem. In [1], it is shown that the above problem can be formulated and solved as a Dynamic Programming (DP) problem. The main idea of [1] was to fix T, and solve the DP problem to obtain the optimal k_i's and then use the information to get a better estimate of the optimal T. Thus, this approach requires solving a number of DP problems to find the optimal T.

In a nutshell this approach requires a one-dimensional search on T. In each of the iteration of the search, a DP problem must be solved. Thus, a more precise estimate of the optimal T requires larger number of the DP problems to be solved that makes the use of meta-heuristics even more attractive alternate to solve the problem. The above formulation very well suits meta-heuristics. GA [4] suggested that both the T and k_i'sare simultaneously determined leaving no need to solveDP problems repeatedly with different values of T. Furthermore, the curse of dimensionality due to DP is not encountered in using GA. In this research;ParticleSwarm Optimization (PSO) is used to find an optimal T using a one-dimensional nonlinear programming method for a combination of k_i's.

III. PARTCILE SWARM OPTIMIZATION

Particle swarm optimization is a population based swarm intelligence algorithm. It was originally proposed by Kennedy [5, 8, 17] as a simulation of the social behavior of social organisms, such as bird flocking and fish schooling. PSO uses the physical movements of the individuals (particles) in the swarm and has a flexible and well balanced mechanism to enhance and adapt to global and local exploration abilities. The PSO algorithm is widely used in many optimization problems due to the intrinsic simplicity of the algorithm itself. It does not require mathematical computation like derivatives or complex encoding like Genetic Algorithm. PSO maintain best solution of each particle along with the global best solution of the whole population and therefore it is less sensitive to local minima problem.

The PSO algorithm works by selecting a set of P particles and initialized by placing it into random positions in the solution space. The position of each particle represents a solution to the problem and its performance is evaluated by objective function specific to a particular problem. The velocity of the each particle v_j is defined as the change of its position. The direction of movement of each particle is the active interaction of individual and whole swarm flying experiences. Each particle adjusts its path towards the solution based on its own previous best position and previous best position of the whole population, namely p_j and p_g. The velocities and positions of particles are updated using the following formulas:

$$v_j(t+1) = v_j(t) + c_j rand_j \left(p_j - s_j(t)\right) + c_g rand_g \left(p_g - s_j(t)\right) \quad (13)$$

$$s_j(t+1) = s_j(t) + v_j(t+1) \quad (14)$$

Where t is the previous iteration and t+1 is the current iteration to compute; c_j and c_g are the acceleration coefficients; $rand_j$, $rand_g$ are random numbers between 0 and 1 inclusive associated with the best solution of a particular particle and the best solution of the whole swarm. c_j and c_g are used to provide the maximum distance a particle will move in a single iteration. The objective function is than computed using particles placed in new positions at iteration t+1. The same equations (13) and (14) are repeated until the maximum iteration becomes reached or until a convergence criterion has been met. At the end of all iterations the best solution found by the whole swarm is returned.

IV. PROPOSED HYBRIDIZATION SCHEME

The proposed hybridized Particle Swarm Optimization (PSO) with Golden Section Search (GSS) algorithm is discussed below:

- The nonlinear objective function given in equation (11) is minimized subject to constraint given in equation (12).
- Computes lower and upper bounds of T and k_i's using following equations [4],

$$T^{LB} = \sum_{i=1}^{n} 0.25\, Q_i^*/P_i \quad (15)$$

$$T^{UB} = \max\left\{ \sqrt{(2(\sum_{i=1}^{n} S_i)/\sum_{i=1}^{n} H_i D_i (1 - D_i P_i)),} \atop (\sum_{i=1}^{n} \tau_i)/(1 - D_i P_i) \right\} \quad (16)$$

$$k_i^{LB} = 1 \quad (17)$$

$$k_i^{UB} = \left\lceil (5\,(Q_i^*/D_i)/T)(\sum_{i=1}^{n} \frac{D_i}{P_i}) \right\rceil, i = 1, 2, \ldots, n \quad (18)$$

- Initialize k_i'srandomly between $[k_i^{LB}, k_i^{UB}]$, i = 1, 2, …, n
- Given the initial k_i's, the TCBP subject to constraint (12) can be minimized by performing a one dimensional search on Tbased on Golden section search usingMatlab*fminbnd*[13] function.
- Repeat the following steps until the maximum number of iteration is reached or until a convergence criterion is met.
- Apply PSO algorithm as discussed in section 3 using equation (13) and (14) to generate the new positions of

P particles in k-dimensional search space. Here, the position of each particle in each dimension represents one $k_i$and the whole particle represents one complete possible solution to ELSP problem.

- Updatesk_i*'s* associated with each particle that do not fulfill lower and upper bound requirements with randomly generated values between $[k_i^{LB}, k_i^{UB}]$.
- Given newly generated k_i*'s*associated with each particle in k-dimensional search space; apply a one dimensional search on T based on Golden section search using Matlab*fminbnd*[13] function to minimize TCBP subject to constraint (12).
- Update current best k_i*'s*and T that minimize TCBP.
- Update best position (solution) p_j of each particle in the swarm.
- Update best position p_g of the whole swarm using best solution of all the particles in the swarm.

Table 1: Data of Bomberger's problem

Product index, i	1	2	3	4	5	6	7	8	9	10
Base Demand	24,000	24,000	48,000	96,000	4800	4800	1440	20,400	20,400	24,000
Setup cost (Si): $	15	20	30	10	110	50	310	130	200	5
Production rate (Pi): units/day	30,000	8000	9500	7500	2000	6000	2400	1300	2000	15,000
Setup time (τi) : h	1	1	2	1	4	2	8	4	6	1
Holding cost (Hi): $/unit-year	0.00065	0.01775	0.01275	0.01000	0.27850	0.02675	0.15000	0.59000	0.09000	0.00400

Table 2: Comparison of TSIS, TCL, GA, and Hybrid PSO solutions for Bomberger's problem

	Total Annual Costs					
Utilization (%)	**TSIS**	**TCL**	**GA**	**Hybrid PSO**	**Best Cost**	**Best Algorithm(s)**
50	5960.445	5960.445	6038.410	6036.513	6036.513	PSO
55	6218.253	6218.253	6328.670	6328.086	6328.086	PSO
60	6459.905	6459.905	6621.750	6618.572	6618.572	PSO
65	6687.131	6687.131	6914.700	6914.837	6914.700	GA
66.18	6738.810	6738.810	7024.110	7024.100	7024.100	GA, PSO
70	6901.335	6901.335	7395.460	7395.466	7395.460	GA, PSO
75	7103.674	7103.674	7789.630	7794.202	7789.630	GA
80	7295.114	7295.114	8096.010	8085.485	8085.485	PSO
83	7405.090	7405.090	8250.290	8250.290	8250.290	GA, PSO
86	7511.593	7511.593	8553.310	8483.945	8483.945	PSO
88.24	7588.934	7588.934	8782.420	8782.289	8782.289	PSO
89	7614.763	7614.763	8874.550	8874.803	8874.550	GA
92	7714.729	7714.729	9745.800	10086.443	9745.800	GA
95	7811.608	8418.885	12018.080	11949.646	11949.646	PSO
97	7874.534	11290.966	17143.000	17134.260	17134.260	PSO
98	7905.510	15681.535	24533.820	24457.541	24457.541	PSO
99	7936.166	29942.667	55544.470	47550.735	47550.735	PSO

Table 3: Comparison of Relative Deviation from TCL, Improvement over GA, and CPU time taken by algorithms for Bomberger's problem

	% Relative Deviation from TCL		% Improvement over GA	CPU time (sec.)
Utilization (%)	**GA**	**Hybrid PSO**	**Hybrid PSO**	**Hybrid PSO**
50	1.308	1.276	0.031	15.189
55	1.776	1.766	0.009	15.019
60	2.505	2.456	0.048	15.489
65	3.403	3.405	0	15.690
66.18	4.234	4.234	0	15.972
70	7.160	7.160	0	16.059
75	9.656	9.721	0	16.060
80	10.979	10.834	0.130	15.561
83	11.414	11.414	0	16.205
86	13.868	12.945	0.811	14.813
88.24	15.727	15.725	0.001	13.786
89	16.544	16.547	0	13.465
92	26.327	30.743	0	11.131
95	42.751	41.939	0.569	11.075
97	51.829	51.752	0.051	11.283
98	56.450	55.964	0.311	11.140
99	85.503	58.806	14.392	11.063
Average	21.261	19.805	0.962	14.059
Min.	1.308	1.276	0	11.063
Max.	85.503	58.806	14.392	16.205
σ	23.939	20.042	3.469	2.078

Table 4: Detail comparison of GA and Hybrid PSO results for Bomberger's problem

Utilization	**Meta-heuristic**	
	GA	**Hybrid PSO**
50	T= 28.183 and K=[5,1,2,1,2,4,10,1,3,1]	T = 28.594 and K=[3,2,2,1,2,4,8,1,3,1]
55	T = 28.762 and K=[5,2,2,1,2,4,8,1,2,1]	T = 29.439 and K=[5,2,2,1,2,4,9,1,2,1]
60	T = 28.863 and K=[4,1,1,1,2,4,9,1,2,2]	T = 29.306 and K=[5,1,1,1,2,4,8,1,2,2]
65	T = 30.828 and K=[2,1,1,1,2,3,7,1,2,1]	T = 30.838 and K=[2,1,1,1,2,3,7,1,2,1]
66.18	T = 30.443 and K=[2,1,1,1,2,2,6,1,2,1]	T = 30.449 and K=[2,1,1,1,2,2,6,1,2,1]
70	T = 33.42 and K=[2,1,1,1,1,2,3,1,2,1]	T = 33.42 and K=[2,1,1,1,1,2,5,1,2,1]
75	T = 31.794 and K=[3,1,1,1,2,3,7,1,1,1]	T = 32.11 and K=[3,1,1,1,2,4,6,1,1,1]
80	T = 34.438 and K=[2,1,1,1,1,3,6,1,1,1]	T = 35.28 and K=[3,1,1,1,1,3,6,1,1,1]
83	T = 34.951 and K=[1,1,1,1,1,2,5,1,1,1]	T = 34.961 and K=[2,1,1,1,1,2,5,1,1,1]
86	T = 37.131 and K=[1,1,1,1,1,1,5,1,1,1]	T = 38.371 and K=[1,1,1,1,1,2,4,1,1,1]
88.24	T = 38.442 and K=[1,1,1,1,1,1,3,1,1,1]	T = 38.436 and K=[1,1,1,1,1,1,3,1,1,1]
89	T = 41.748 and K=[1,1,1,1,1,1,3,1,1,1]	T = 41.758 and K=[1,1,1,1,1,1,3,1,1,1]
92	T = 53.904 and K=[1,1,1,1,1,1,2,1,1,1]	T = 53.914 and K=[1,1,1,1,1,1,1,1,1,1]
95	T = 75.809 and K=[1,1,1,1,1,1,1,1,1,1]	T = 75 and K=[1,1,1,1,1,1,1,1,1,1]
97	T = 125.08 and K=[1,1,1,1,1,1,1,1,1,1]	T = 125 and K=[1,1,1,1,1,1,1,1,1,1]
98	T = 188.14 and K=[1,1,1,1,1,1,1,1,1,1]	T = 187.5 and K=[1,1,1,1,1,1,1,1,1,1]
99	T = 439.45 and K=[1,1,1,1,1,1,1,1,1,1]	T = 375 and K=[1,1,1,1,1,1,1,1,1,1]

V. RESULTS

The results obtained from detailed analysis are shown in Table 2, Table 3, and Table 4. Table 2 compares the cost obtained by solving [1] problem as shown in Table 1 using Hybrid PSO and GA [4] algorithms. Table 3 compares the (i) relative deviation from tighter lower bound (TCL), (ii) improvement achieved through Hybrid PSO algorithm over results obtained through GA algorithm [4], (iii) efficiency in terms of execution time taken by Hybrid PSO algorithm. Table 4 compares the detailed solution found by Hybrid PSO with GA solution [4].

Table 2 shows that 71% of Hybrid PSO solutions are either better or similar to best result obtained from GA algorithm, while only 41% of GA solution are better or similar to best result obtained from Hybrid PSO algorithm. So, in majority of cases Hybrid PSO performed better than GA algorithm.

Table 3 shows that average relative deviation from TCL is 19.805% using Hybrid PSO and worst average relative deviation from TCL is 21.261% using GA algorithm, average improvement over GA is 0.962% using PSO, and average CPU utilization time is 14.049 sec using PSO. It is also important to note that GA differs with PSO algorithm for high utilization as well as low utilization cases. GA found worst relative deviation from TCL for higher utilization but results for lower utilization cases are comparatively closed to PSO algorithm.

Table 4 shows the detail comparison of values for T and k_i(i.e., i=1,2,…10) using Hybrid PSO with GA algorithm. For low utilization cases 50 to 92 k_i have different values but for high utilization cases 95 to 99 all k_i have same value '1'. Hybrid PSO found same value for T and $k_i$which gives low deviation from TCL. GA found the same value for k_i but failed to found value for T similar to Hybrid PSO algorithm and therefore it results in high deviation from TCL.

VI. CONCLUSION

This research presented hybridization scheme based on Particle Swarm Optimization and Golden Section Search to solve the ELSP problem under basic period approach. This Hybrid PSO technique used PSO optimization to find the optimum value of k_i*'s* and followed by GSS to find the basic period T. The feasibility of the solution is guaranteed with a constraint that ensures the items assigned in each period can be produced within the length of the period. The experimental results indicate following outcomes:

- The hybridization scheme was able to find comparatively better BP solutions than GA [4]for the low utilization problems.
- The hybridization scheme was also able to find comparatively better BP solutions than GA [4] for the high utilization problems.

References

[1] E. Bomberger,"A dynamic programming approach to a lot size scheduling problem," Management Science, vol. 12, no. 11,pp. 778–784, July 1966.

[2] S. E. Elmaghraby, "The Economic Lot Scheduling Problem (ELSP): Review and Extensions," in Management Science, vol. 24, no. 6, pp. 587–598, February 1978.

[3] S. E.Elmaghraby, "An Extended Basic Period Approach to the Economic Lot Scheduling Problem (ELSP)," Production and Industrial Systems: Future Development and the Role of Industrial and Production Engineering, Taylor and Francis, pp. 649–662.

[4] M. Khouja, Z. Michalewicz, and M. Wilmot, "The use of genetic algorithms to solve the economic lot size scheduling problem," European Journal of Operational Research, vol. 110, no. 3, pp. 509-524, November 1998.

[5] K. Thanushkodi and K. Deeba, "On performance analysis of hybrid algorithm (Improved PSO with Simulated Annealing) with GA, PSO for multiprocessor job scheduling," WSEAS Transaction on Computers, vol. 10, no. 9, pp. 287-300, September 2011.

[6] G. Dobson, "The Economic Lot-Scheduling Problem: Achieving Feasibility Using Time-Varying Lot Sizes," Operation Research, vol 35, no. 5,pp. 764-771, September 1987.

[7] F.Hanssmann, "Operations Research in Production and Inventory," John Wiley and Sons, pp. 158-160, 1962.

[8] J. Kennedy andR. C.Eberhard, "Swarm intelligence,"Morgan Kaufmann Publishers,2001.

[9] H. Aytug, M. Khouja, and F. E.Vergara,"Use of genetic algorithm to solve production and operations management problems: A review," International Journal of Production Research, vol. 41, no. 17, pp. 3955–4009, November 2003.

[10] M. Ben-daya and M. Al-Fawzan, "A tabu search approach for the flow shop scheduling problem," European Journal of Operational Research, vol. 109, no. 1, pp. 88–95, August 1998.

[11] R. W. Eglese, "Simulated annealing: A tool for operational research," European Journal of Operational Research, vol. 46, no. 3, pp. 271–281, June 1990.

[12] L. Gaafar,"Applying genetic algorithms to dynamic lot sizing with batch ordering," Computers & Industrial Engineering, vol. 51, no. 3, pp. 433–444, November 2006.

[13] MathWorks, "Find minimum of single-variable function on fixed interval - MATLAB", http://www.mathworks.com/help/matlab/ref/fminbnd.html, September 2012.

[14] H. Sun, H. Huang, and W. Jaruphongsa, "A genetic algorithm for the economic lot scheduling problem under extended basic period and power-of-two policy," CIRP Journal of Manufacturing Science and Technology, vol. 2, no. 1, pp. 29-34, 2009.

[15] M. F. Tasgetiren, O. Bulut, and M. M. Fadiloglu, "A Discrete Harmony Search Algorithm for the Economic Lot Scheduling Problem with Power of Two Policy," IEEE World Congress on Computational Intelligence, June 2012.

[16] S. A. Raza and A. Akgunduz, "A comparative study of heuristic algorithms on Economic Lot Scheduling Problem," Computer & Industrial Engineering, vol. 55, no. 1, pp. 94-109, August 2008.

[17] R. Kubota and H. Tamukoh, "A modified particle swarm optimization considering component combined with personal best positions," Recent Researches in Circuits, Communications and Signal Processing, pp. 55-58, 2013.

[18] M. F. Tasgetiren, O. Bulut and M. M. Fadiloglu, "A discrete harmony search algorithm for the economic lot scheduling problem with power of two policy," IEEE World Congress on Computational Intelligence, 2012.