iBusiness, 2012, 4, 216-221
http://dx.doi.org/10.4236/ib.2012.43027 Published Online September 2012 (http://www.SciRP.org/journal/ib)

A Hybrid Clonal Selection for the Single Row Facility Layout Problem with Unequal Dimensions

Hasan Hosseini-Nasab*, Leila Emami

Department of Industrial Engineering, Yazd University, Yazd, Iran.
Email: *hhn@yazduni.ac.ir

Received May 10th, 2012; revised May 29th, 2012; accepted July 23rd, 2012

ABSTRACT

The single row facility layout problem (SRFLP) is an important combinatorial optimization problem where a given set of facilities have to be arranged in a single row to minimize the weighted sum of the distances between all pairs of facilities. In this paper, a hybrid method for single row facility layout problem is proposed in which, the simulated annealing (SA) is embedded in the clonal selection algorithm (CSA). The performance of the proposed algorithm is tested on benchmark problems. Computational results show the efficiency of the proposed algorithm compared to other heuristics.

Keywords: Single Row Facility Layout; Clonal Selection Algorithm; Simulated Annealing

1. Introduction

The facility layout problem (FLP) is to arrange of a given number of departments while minimizing the total cost associated with the (known or projected) interactions between them. Types of the facility layout problem occur in many environments, such as hospital layout and service center layout. Generally, these problems are hard problems; most types of these problems are NP-hard. Single row facility layout problem (SRFLP) is a special case of the facility layout that it is to arrange *n* departments on a straight line to minimize the total weighted distances between all pairs of facilities. If all the facilities have the same length, the SRFLP becomes an instance of the linear arrangement problem [1,2] which is itself a special case of the quadratic assignment problem [3]. Various applications of the SRFLP have been identified in the literature. One such application is in the flexible manufacturing systems, where machines within manufacturing cells are often placed along a straight path travelled by an automated guided vehicle [4]. The minimum linear arrangement problem was proved NP-hard [5], therefore the SRFLP is NP-hard because this is a generalization of it. Exact approaches have been proposed for the problem. A branch-and-bound algorithm is presentedin [6-8]. Picard and Queyranne [9] presented dynamic programming algorithms for the SRFLP. Heragu and Kusiak [10] presented a nonlinear model. Love and Wong [11] and Amaral [12-14] proposed linear mixed-integer programs.

The paper has the following structure. Section 2 gives a literature review on SRFLP. Section 3 is the problem description. Section 4 the principles of the original CSA and SA method are briefly introduced, the proposed hybrid algorithm in more details for SRFLP described in Section 5. In Section 6 computational results are given. Finally, Section 7 contains conclusions.

2. Literature Review

Many heuristics and metaheuristic algorithms have been presented to solve the SRFLP because this problem is a NP-complete problem [15]. Neghabat [16] proposed an algorithmto reach a complete solution by adding one machine at a time to the end of the current solution. Drezner [17] presented a heuristic method based on the eigenvectors of a transformed flow matrix. Another heuristic has been presented by Heragu and Kusiak [10], in which a pair of facilities with the largest adjusted flow is initially laid and then the partial order is gradually completed through a loop adding new machines to the right and left of the order obtained in the previous iteration. Heragu and Alfa [18] proposed a simulated annealing algorithm to solve SRFLP. A constructive greedy heuristic has been developed by Kumar and Hadjinicola [19]. In this algorithm assigns facilities with the largest number of moves between them to adjacent locations on the line. Braglia [20] introduced an algorithm that is combined of simulated annealing and genetic algorithm to minimize the total backtracking in the linear ordering of machines.

*Corresponding author.

Solimanpur *et al.* [21] presented a non-linear 0 - 1 programming model for the SRFLP then was solved by an ant algorithm.

Anjos *et al.* [22], Anjos and Vannelli [23], Anjos and Kong [24], Hungerlander and Rendl [25], had presented semi-definite programming relaxation providing a lower bound on the optimum value of the SRFLP. Recently, many researchers proposed meta-heuristic methods, such as: a scatter search algorithm by Kumar [26], a hybrid algorithm based on ant colony optimization and PSO by Teo and Ponnambalam, [27], a genetic algorithm by Lin [28], a PSO algorithm by Samarghandi *et al.* [29], and genetic algorithm by Datta *et al.* [30].

In this paper, a hybrid clonal selection algorithm is presented for SRFLP. The computational experiments show the efficient performance of the proposed algorithm on different instances of various sizes available in the literature.

3. Problem Formulation

Heragu and Kusiak [4] presented a model for the problem, which they called ABSMODEL. Let be u_i^+ the distance between the centroid of department *i* and the line origin and let S_{ij} be the minimum separation between departments *i* and *j*. Their model is given by,

$$\min\left\{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}\left|u_i^+ - u_j^+\right|\right\} \tag{1}$$

$$\begin{gathered} s.t: \left|u_i^+ - u_j^+\right| \ge \frac{1}{2}\left(l_i + l_j\right) + s_{ij} \\ \left(i = 1,\cdots,n;\ j = i+1,\cdots,n.\right) \end{gathered} \tag{2}$$

As the absolute value for the distance between the centroids are used, it does not matter if department *i* is to the left or to the right of department *j*. Note that the minimum value that $u_i^+ - u_j^+$ can assume could be set greater than $\left(l_i + l_j\right)/2$ if we are given a value S_{ij}.

If $\prod_n$ the set of all permutations π of $N = \{1, 2, \cdots, n\}$. The SRFLP can be mathematically formulated in Amaral, [14]:

$$\min_{\pi\in\Pi_n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij} d_{ij}^{\pi} \tag{3}$$

where d_{ij} is the distance between departments *i* and *j* with respect to a permutation π. New modeling of the SRFLP in (3) implies that the ABSMODEL searches the set of all permutations of numbers $1, 2, \cdots, n$ to find the permutation which minimizes the objective function.

4. Clonal Selection and Simulated Annealing

4.1. AIS, CSA

Artificial immune algorithm (AIA) is a recent branch of stochastic search algorithms and classified as a population-based metaheuristic method. These properties impart a high degree of robustness and performance and attracted interest of researches in implementing it to engineering systems. This adopted engineering analogue, called AIS, emerged in the 1990s as a new computational research area [31].

Three commonly applied types of AIAs are clonal selection algorithm (CSA), immune network algorithm (INA) and negative selection algorithm (NSA). Among these models, CSA is reported to be successful in combinatorial optimization problems [32]. Therefore, a CSA is employed to solve SRFLP in this study.

CSA is treated as biological and random search based general-purpose heuristic methods. The most important component of a CSA is how to represent the solution. The feasible solutions are coded as individuals. Each individual in the population is called an antibody. A CSA starts with a set of random antibodies as an initial population. Any population in turn is a subset of the solution space. Appropriate solutions are searched within the population. The new generations are formed through copying (cloning) in proportion to concept called as the affinity value. The algorithm does not have a crossover operator as in GA. The mutation operators are utilized to search and evaluate new regions in the solution space inversely proportional to the affinity value of the solution. The mutation rates in CSA are higher compared to Gas. It can be described by the following steps [33]:

1) Initialize the antibody pool init P including the subset of memory cells (M);

2) Evaluate the fitness of all the antibodies (affinity with the antigen) in population P;

3) Select the best candidates (P_r) from population P, according to their fitness;

4) Clone P_r into a temporary antibody pool (C);

5) Generate a mutated antibody pool (C_1). The mutation rate of each antibody is inversely proportional to its fitness;

6) Evaluate all the antibodies in C_1;

7) Eliminate those antibodies similar to the ones in C, and update C_1;

8) Re-select the antibodies with better fitness from C_1 to construct memory set M. Other improved individuals of C_1 can replace certain members with poor fitness in P to maintain the antibody diversity;

9) Return back to Step 2, if a pre-set termination criterion is not met.

4.2. Simulated Annealing

Simulated annealing (SA) proposed by Kirkpatrick *et al.* [34], belongs to the class of stochastic search algorithms, known as meta-heuristics. This algorithm motivated from an analogy between the physical annealing of solid materials and optimization problems. SA has been widely applied to solve combinatorial optimization problems. It is inspired by the physical process of heating a substance and then cooling it slowly, until a strong crystalline structure is obtained. This process is simulated by lowering an initial temperature by slow stages until the system reaches an equilibrium point and no more changes occur.

5. Proposed Hybrid Algorithm

5.1. Antibody Representation and Initialization

There are various techniques for encoding solutions in an individual. In this paper, we use permutation representation. In this approach, Integer-valued *n* elements are the facilities to be considered. Each permutation indicates a layout of facilities. Each permutation allocating *n* rectangular facilities to a straight line is a feasible solution for the SRFLP therefore the solution is always feasible. Each layout (antibody) represents a potential solution and has a cost value that refers to the affinity value of that antibody.

We have used two techniques for initializing individuals. These techniques are described in the following:

- Random initializing

This approach, we generate random permutation for an individual.

- Length-Based Permutation (LBP)

This method proposed in Samarghandi *et al.* [29]. It is based on the assumption that if the flow between facilities is equal then the optimal permutations of the facilities can be given as follows:

Sort the facilities in non-descending order such that the shortest facility is denoted by 1 and the longest facility by *n*. Then, **Figure 1** shows the optimum solution when *n* is an odd number, and **Figure 2** shows the optimum solution when *n* is an even number [29].

5.2. Cloning Selection Procedure

For the selection of antibodies to constitute the mutating pool, first the best antibody is selected and to fulfill the rest, the binary tournament selection operator is used. It selects two random individuals from the population and stores a copy of the best individual (based on objective values) in the mating pool. The process is repeated (popsize-1) times.

$$\cdots(i)\cdots(3)(1)(2)\cdots(i-1)\cdots$$

Figure 1. Optimal layout when *n* is odd.

$$\cdots(i)\cdots(2)(1)(3)\cdots(i-1)\cdots$$

Figure 2. Optimal layout when *n* is even.

5.3. Affinity Maturation

After generating the clone population all of the antibodies existing in the pool undergo an operator which makes a random change in the clones. This operator is called hyper-mutation. Each antibody undergoes different rate of change based on the affinity value. The inferior antibodies undergo high rate of hyper-mutation whereas better antibodies suffer a slight change. In this study, two mutation procedure are used as follows:

As a low rate hyper-mutation, we utilize swap mutation. In swap mutation randomly chooses two facilities and exchanges their locations. As high rate hyper-mutation, we use an operator working as follows:

- One solution is randomly chosen from the 20% of the antibodies with the smallest objective values.
- Each gene of the antibody is checked against its corresponding gene in selected antibody for equality. If they contain identical values, the antibody maintains the gene; otherwise, the gene becomes empty.

Next, the values missing in the antibody are identified and inserted into the empty genes in a random manner.

After defining the mutation operators, we need to determine the condition under which we use one of the operators. We sort the antibodies in non-decreasing order of their objective function values. Swap mutation is applied for *H*% of the best antibodies and the remaining antibodies $(1-H)\%$ are generated through another operator.

For accepting the off spring we use simple simulated annealing acceptance criterion. Besides the acceptance of better offspring, inferior offspring might be accepted by the following random mechanism:

If $rnd(0,1) < \exp\left(\left(\text{off spring} - \text{creator}\right)/T\right)$

In this paper, we use exponential cooling schedule as follows:

$$T_i = \left(A/(i+1)\right) + B,\ A = \left(\left(T_0 - T_f\right)(L+1)\right)/L,\quad B = T_0 - A,\ i = 1,2,\cdots,L. \tag{5}$$

where T_0, T_f and L are initial temperature, final (stopping) temperature and desired number of temperature levels between T_0 and T_f, respectively.

5.4. Stepwise Procedure

The stepwise procedure of the proposed hybrid algorithm is as follows:

Step 1: Set the values of control parameters: *P* (antibody population size), T_0 (initial temperature), T_f (final temperature), *L* (desired number of temperature levels between T_0 and T_f), *B* (parameter of elimination ratio of antibodies at each iteration), $K \leftarrow 1$;

Step 2: Create a population of *P* antibodies based on the procedure described in Section 5.1;

Step 3: Evaluate each antibody in the population;

Step 4: Select antibodies using the tournament technique for mating pool;

Step 5: Sort the antibodies in non-decreasing order of their objective function values. Swap mutation is applied for *H*% of the best antibodies and replace that with offspring if its function improved or

If $rnd(0,1) < \exp((\text{off spring} - \text{creator})/T)$ and the remaining antibodies $(1-H)\%$ are generated through another operator;

Step 6: If generated new antibodies are less than (*B*P*) go to Step 5 otherwise proceed to Step 7, $K \leftarrow K+1$ and adopt temperature;

Step7: If *k* = *N* stop. Otherwise go to Step 4.

6. Computational Results

In this section, we present the computational results of the proposed HCSA algorithm applied to the existent problems in the literature. The proposed algorithm are coded in C++ and executed on a PC with 2.2 GHz Intel core 2 Due and 4 GB of RAM memory. The proposed hybrid algorithm includes six parameters which affect the algorithm's performance. For tuning the algorithm, extensive sensitivity analysis were conducted with different sets of parameters. Based on these observations, the following values were obtained for the parameters:

$$P = 70,\ T_0 = 35,\ N = 12n,\ L = 15n,\ B = 0.35,$$

All the problems from the literature can be divided into two major sets, those with a proven optimal solution and those without a proven optimal solution.

The first set of problems with an optimal solution includes 15 problems. Objective function value and computation time obtained by different heuristics are shown in **Table 1**. For each problem the algorithms perform 10 runs and the best result obtained by the 10 runs presented. Moreover, the computational time which the best solution appears for the first time by HCSA is reported. In **Table 1** the obtained best results of proposed algorithm are compared with the solutions of Solimanpur *et al.* [21] and with the optimal objective values given by the exact methods. As shown in **Table 1**, proposed algorithm could obtain the optimal solution for each of the problems. As mentioned the computational time in the tables are not comparable because the computers are different. Still, the computational time of meta-heuristics is considered here also as it is available in the literature.

We have also tested the algorithm on large-size problems with no proven optimal solution reported in the literature. This set of problems consist of 20 instances considered in **Table 2**. The best found solutions are reported in boldface. It is observed the proposed HCSA generates better solutions for 3 problems.

7. Conclusion

This paper considers the single row facility layout problem

Table 1. The best solutions obtained by the CSA for 15 traditional instances.

Problem no.	No. of facility	Optimal objective value	Reference	Solimanpur *et al.* (2005)		Proposed algorithm	
				OFV	Time	Solution	Time
1	4	78.00	Beghin-Picavet and Hansen (1982)	78.00	0.00	78.00	0.00
2	5	151.00	Love and Wong (1976)	151.00	0.00	151.00	0.00
3	5	1.100	Nugent *et al.* (1968)	1.100	0.00	1.100	0.00
4	6	1.990	Nugent *et al.* (1968)	1.990	0.00	1.990	0.00
5	7	4.73	Nugent *et al.* (1968)	4.73	0.00	4.73	0.00
6	8	6.295	Nugent *et al.* (1968)	6.295	0.00	6.295	0.00
7	8	2324.5	Simmons (1969)	2324.5	0.00	2324.5	0.00
8	10	2781.5	Simmons (1969)	2781.5	0.01	2781.5	0.01
9	11	6933.5	Simmons (1969)	6933.5	0.03	6933.5	0.012
10	12	23.365	Heragu and Kusiak (1991)	23.365	0.06	23.365	0.01
11	15	44.600	Heragu and Kusiak (1991)	44.600	0.18	44.600	0.022
12	20	119.710	Heragu and Kusiak (1991)	119.71	1.8	119.710	0.081
13	20	15549.0	Heragu and Kusiak (1991)	15549.0	2.30	15549.0	0.12
14	30	334.870	Heragu and Kusiak (1991)	334.870	37.30	334.870	1.11
15	30	44965.0	Heragu and Kusiak (1991)	44965.0	37.30	44965.0	0.81

Table 2. The best solutions obtained by the HCSA for 20 large-size instances.

Problem no.	No. of facility	Anjos *et al.* (2005)		Samarghandi and Eshghi (2010)		Datta *et al.* (2011)		HCSA	
		Objective function	Time (h)	OFV	Time (sec)	OFV	Time (sec)	OFV	Time (sec)
1	60	1479294.00	5	1477834.0	0.82	1477834.0	19.54	1477834.0	10.90
2	60	829792.00	5	841792.0	0.98	841792.0	22.34	841792.0	9.66
3	60	650167.00	5	648337.5	0.90	648337.5	68.81	648337.5	23.12
4	60	402214.00	5	398511.0	0.913	398468.0	20.71	**398406.0**	19.91
5	60	318805.00	5	318805.0	0.762	318805.0	26.41	318805.0	17.6
6	70	1531212.00	7	1529197.0	1.499	1528621.0	64.83	1529197.0	29.11
7	70	1440901.00	7	1441028.0	1.940	1441028.0	77.49	1441028.0	48.1
8	70	1518993.50	7	1518993.0	1.761	1518993.0	68.26	1518993.0	42.1
9	70	971090.00	7	969130.0	1.233	968796.0	100.59	969130.0	50.83
10	70	4216349.00	7	4218230.0	1.570	4218017.5	60.48	**4218002.0**	52.61
11	75	2396213.00	10	2393483.0	2.010	2393456.0	125.26	2393490.0	88.66
12	75	4325142.00	10	4321190.0	2.198	4321190.0	128.95	4321190.0	90.36
13	75	1256199.00	10	1248551.0	2.912	1248537.0	157.95	1248551.0	103.1
14	75	3941713.00	10	3942013.0	2.516	3941981.0	119.92	3942013.0	93.3
15	75	1801040.00	10	1791408.0	2.098	1791408.0	101.67	1791408.0	90.21
16	80	2104771.00	10	2069097.5	3.975	2069097.5	75.41	2069097.5	110.1
17	80	1919288.00	10	1921177.0	5.641	1921177.0	68.75	1921177.0	126.8
18	80	3291413.00	10	3251413.0	4.797	3251368.0	85.9	3251413.0	133.9
19	80	3751331.00	10	3746515.0	3.453	3746515.0	77.81	3746515.0	141.6
20	80	1593108.0	10	1589061.0	3.769	1588901.0	196.51	**1588862.0**	166.3

in which the size of facilities are different. This problem belongs to an NP-hard class and traditional approaches cannot reach to an optimal solution in a reasonable time. Therefore, in this paper a hybrid clonal selection was proposed to find optimal and near-optimal solutions for SRFLP. To evaluate the efficiency the proposed hybrid algorithm the known SRFLP test problems from the literature are used. The proposed HCSA is first tested on set of problems with a proven optimal solution in which it could obtain the optimal solution for all of them. Then the algorithm is tested for large-size instances and the results verify the efficiency of algorithm in finding good quality solutions compared other heuristics and it successfully could improve 3 of 20 solutions.

REFERENCES

[1] W. Liu and A. Vannelli, "Generating Lower Bounds for the Linear Arrangement Problem," *Discrete Applied Mathematics*, Vol. 59, No. 2, 1995, pp. 137-151. doi:10.1016/0166-218X(93)E0168-X

[2] J. E. Mitchell and B. Borchers, "Solving Linear Ordering Problems with a Combined Interior Point/Simplex Cutting Plane Algorithm," *High Performance Optimization*, Vol. 33, 2000, pp. 349-366.

[3] E. Cela, "The Quadratic Assignment Problem, Theory and Algorithms," Kluwer Academic Publishers, Dordrecht, 1998.

[4] S. S. Heragu and A. Kusiak, "Machine Layout Problem in Flexible Manufacturing Systems," *Operations Research*, Vol. 36, No. 2, 1988, pp. 258-268. doi:10.1287/opre.36.2.258

[5] M. R. Garey and D. S. Johnson, "Computers and Intractability: An Introduction to the Theory of NP-Completeness," Freeman, New York, 1979.

[6] D. M. Simmons, "One-Dimensional Space Allocation: An Ordering Algorithm," *Operations Research*, Vol. 17, No. 5, 1969, pp. 812-826.

[7] D. M. Simmons, "A Further Note on One-Dimensional

Space Allocation," *Operations Research*, Vol. 19, No. 1, 1971, p. 249. doi:10.1287/opre.19.1.249

[8] R. M. Karp and M. Held, "Finite-State Processes and Dynamic Programming," *SIAM Journal of Applied Mathematics*, Vol. 15, No. 3, 1967, pp. 693-718.

[9] J. C. Picard and M. Queyranne, "On the One-Dimensional Space Allocation Problem," *Operations Research*, Vol. 29, No. 2, 1981, pp. 371-391. doi:10.1287/opre.29.2.371

[10] S. S. Heragu and A. Kusiak, "Efficient Models for the Facility Layout Problem," *European Journal of Operational Research*, Vol. 53, No. 1, 1991, pp. 1-13.

[11] R. F. Love and J. Y. Wong, "On Solving a One-Dimensional Space Allocation Problem with Integer Programming," *INFOR*, Vol. 14, 1976, pp. 139-143.

[12] A. R. S. Amaral, "On the Exact Solution of a Facility Layout Problem," *European Journal of Operational Research*, Vol. 173, No. 2, 2006, pp. 508-518.

[13] A. R. S. Amaral, "An Exact Approach for the One-Dimensional Facility Layout Problem," *Operations Research*, Vol. 56, No. 4, 2008, pp. 1026-1033. doi:10.1287/opre.1080.0548

[14] A. Amaral, "A New Lower Bound for the Single Row Facility Layout Problem," *Discrete Applied Mathematics*, Vol. 157, No. 1, 2009, pp. 183-190. doi:10.1016/j.dam.2008.06.002

[15] G. Suresh and S. Sahu, "Multiobjective Facility Layout Using Simulated Annealing," *International Journal of Production Economics*, Vol. 32, No. 2, 1993, pp. 239-254. doi:10.1016/0925-5273(93)90071-R

[16] F. Neghabat, "An Efficient Equipment Layout Algorithm," *Operations Research*, Vol. 22, No. 3, 1974, pp. 622-628. doi:10.1287/opre.22.3.622

[17] Z. Drezner, "A Heuristic Procedure for the Layout of a Large Number of Facilities," *Management Science*, Vol. 7, No. 33, 1987, pp. 907-915. doi:10.1287/mnsc.33.7.907

[18] S. S. Heragu and A. S. Alfa, "Experimental Analysis of Simulated Annealing Based Algorithms for the Facility Layout Problem," *European Journal of Operational Research*, Vol. 57, No. 2, 1992, pp. 190-202. doi:10.1016/0377-2217(92)90042-8

[19] K. R. Kumar, G. C. Hadjinicola and T. L. Lin, "A Heuristic Procedure for the Single Row Facility Layout Problem," *European Journal of Operational Research*, Vol. 87, No. 1, 1995, pp. 65-73. doi:10.1016/0377-2217(94)00062-H

[20] M. Braglia, "Optimization of a Simulated-Annealing-Based Heuristic for Single Row Machine Layout Problem by Genetic Algorithm," *International Transactions in Operational Research*, Vol. 1, No. 3, 1996, pp. 37-49. doi:10.1111/j.1475-3995.1996.tb00034.x

[21] M. Solimanpur, P. Vrat and R. Shankar, "An Ant Algorithm for the Single Row Layout Problem in Flexible Manufacturing Systems," *Computers & Operations Research*, Vol. 32, No. 3, 2005, pp. 583-598. doi:10.1016/j.cor.2003.08.005

[22] M. F. Anjos and A. Vannelli, "Computing Globally Optimal Solutions for Single-Row Layout Problems Using Semidefinite Programming and Cutting Planes," *INFORMS Journal on Computing*, Vol. 20, No. 4, 2008, pp. 611-617. doi:10.1287/ijoc.1080.0270

[23] M. F. Anjos and C. Kong, "FLP Database," 2007. http://flplib.uwaterloo.ca

[24] P. Hungerlander and F. Rendl, "A Computational Study for the Single-Row Facility Layout Problem," 2011. http://www.optimization-nline.org/DBFILE/2011/05/3029.pdf

[25] M. F. Anjos, A. Kennings and A. Vannelli, "A Semidefinite Optimization Approach for the Single-Row Layout Problem with Unequal Dimensions," *Discrete Optimization*, Vol. 2, No. 2, 2005, pp. 113-122. doi:10.1016/j.disopt.2005.03.001

[26] S. Kumar, *et al.*, "Scatter Search Algorithm for Single Row Layout Problem in FMS," *Advances in Production Engineering and Management*, Vol. 3, No. 4, 2008, pp. 193-204.

[27] Y. T. Teo and S. G. Ponnambalam, "A Hybrid ACO/PSO Heuristic to Solve Single Row Layout Problem," 4*th IEEE Conference on Automation Science and Engineering*, Washington DC, 23-26 August 2008.

[28] M. T. Lin, "The Single-Row Machine Layout Problem in Apparel Manufacturing by Hierarchical Order-Based Genetic Algorithm," *International Journal of Clothing Science and Technology*, Vol. 21, No. 1, 2009, pp. 31-43. doi:10.1108/09556220910923737

[29] H. Samarghandi and K. Eshghi, "An Efficient Tabu Algorithm for the Single Row Facility Layout Problem," *European Journal of Operational Research*, Vol. 205, No. 1, 2010, pp. 98-105. doi:10.1016/j.ejor.2009.11.034

[30] D. Datta, A. R. Amaral and J. R. Figueira, "Single Row Facility Layout Problem Using a Permutation-Based Genetic Algorithm," *European Journal of Operational Research*, Vol. 213, No. 2, 2011, pp. 388-394.

[31] D. Dasgupta, "An Overview of Artificial Immune Systems and Their Applications," In: D. Dasgupta, Ed., *Artificial Immune Systems and Their Applications*, Springer-Verlag, Berlin, 1998, pp. 3-18.

[32] O. Engin and A. Doyen, "Artificial Immune Systems and Applications in Industrial Problems," *Gazi University Journal of Science*, Vol. 17, No. 1, 2004, pp. 71-84.

[33] X. Wang, "Clonal Selection Algorithm in Power Filter Optimization," *Proceedings of the IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications*, Espoo, 28-30 June 2005, pp. 122-127.

[34] S. Kirkpatrick, C. D. Gelatt, Jr. and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, 1983, pp. 671-680.