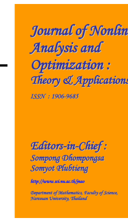
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 2, No. 1, (2011), 197 - 209
ISSN : 1906-9605
http://www.sci.nu.ac.th/jnao



A COMPARATIVE STUDY ON CLASSICAL AND META-HEURISTIC OPTIMIZATION METHODS IN CELL FORMATION PROBLEM

M. HASANNEJAD[1,*], S.E. HOSSEINI[2]

[1]Industrial engineering department, Mazandaran university of science & technology, Babol, Iran

[2]Master of science in industrial management, Islamic azad university, Semnan branch, Iran

ABSTRACT. The Cell Formation (CF) problem determines the decomposition of manufacturing cells, in which parts are grouped into part families, and machines are allocated into machine cells to take advantages of minimum inter-cellular movements and the maximum number of parts flow. In this paper, we compare two classical and meta-heuristic optimization methods for solving the manufacturing CF problem. Hence, a dynamic integer model of CF with three sub-objective functions is considered. Also, a set of 20 test problems with various sizes is solved, once by using of Lingo software as a classical optimization method and another with proposed Modified Self-adaptive Differential Evolution (MSDE) algorithm as a meta-heuristic. The result of this comparative study indicates that MSDE algorithm performs more effective for all test problems. Furthermore, due to the fact that CF is a NP-hard problem, classical optimal method needs a long computational time and so not reliable.

KEYWORDS : Cell formation problem; Modified self-adaptive differential evolution; Classical optimization method.

---

[*] *Corresponding author.*
Email address : hasannejadm@yahoo.com (M. Hasannejad) and ebrahim_356@yahoo.com (S.E.Hosseini).
Article history : Received 6 December 2010. Accepted 31 January 2011.

1. Introduction

1.1. **Cell formation problem.** With increasing market pressure for shorter product life cycle and need of responding to environmental issues, the dynamic aspects of manufacturing play a key role in designing and optimizing of a manufacturing system. Cellular manufacturing system (CMS) is a production approach aimed at increasing production efficiency and flexibility by utilizing the process similarities of the parts in order to minimize the intracellular cost of parts. It involves grouping similar parts into part families and the corresponding machines into machine cells. The success of CMS is rooted in their ability to reduce set-up times, reduced work-in-process inventories, reduced material handling costs, simplified scheduling, shorter lead-times, faster response to internal and external changes, better production control and etc. [1]. Also, some disadvantages have point out in prior research for the CMS, such as: reduced flexibility in compare of a job shop, reduced machine utilization due to dedication of machines to cells and impact of machine breakdown on due date [2]. There are three important steps in CMS design: 1) cell formation (CF), 2) machine layout and 3) cell layout, that Cell formation is the first major step in designing a CMS, which involves identification of machine cells and part families. Most of the CF studies have focused on independence of cells, and a few number of them point out the inter-cell movements [3]. In the last three decades of research in CF, researchers have mainly used zero-one machine component incidence matrix as the input data for the problem [4]. In addition, many procedures have been proposed for CF problem such as: data array reordering, hierarchical clustering, non-hierarchical clustering, heuristic and meta-heuristic algorithms, mathematical programming, graph theory, artificial neural networks and fuzzy sets [5].

1.2. **Differential evolution (DE) algorithm.** Evolutionary algorithms are population based stochastic optimization techniques that search for solution using a simplified model of natural genetics in non-continuous, non-convex, nonlinear, time-dependent or solution spaces. The DE is one of the most frequently used evolutionary algorithms, which was first successfully applied by Storn and Price in optimization of some well known nonlinear, non-differentiable and non-convex functions [6]. In recent years, DE has gradually become more popular and has been applied successfully to solve analytically non-solvable problems with non-continuous, nonlinear, noisy, flat, and multi-dimensional objective functions [7]. Due to its simplicity, effectiveness and robustness, DE has successfully applied in different practical applications [8], image processing [9], data clustering [10], optimal design [11] and scheduling [12] problems. The DE algorithm begins from a primary population which generated randomly and consists of four stages process named: initialization, mutation, crossover and selection. In mutation process of a DE algorithm, the weighted difference between two randomly selected population members is added to a third member to generate a mutated solution. Then, a crossover operator follows to combine the mutated solution with the target solution so as to generate a trial solution. Thereafter, a selection operator is applied to compare the fitness function value of both competing solutions, namely, target and trial solutions to determine who can survive for the next generation. This process is repeated over several generations resulting in an evolution of the population to optimal values. The convergence properties of DE are strongly related to its stochastic nature and DE uses random sequence for its parameters. Most of DE studies are related with the mutation operator and just two variants being currently used, so called binomial and exponential crossover.

This paper presents the MSDE algorithm, which is improved of DE algorithm. MSDE in compare of DE algorithm rectifies the mutation rules, thus it select randomly three chromosomes from solution area and uses the environs of the best selected chromosome for any mutant chromosome. In this paper, the performance of MSDE is compared as a meta-heuristic optimizing method with the classical one. Therefore, a dynamic integer model of CF with nonlinear objective function is first presented and then, it solved by Lingo software and MSDE algorithm to achieve the optimum solution. The results have compared from two aspects: the time to get optimum solution (CPU time) and the optimum objective function.

The remainder of this paper is organized as follows: In Section 2 we review on previous work on related areas. The dynamic integer CF model is presented in Section 3. Section 4 describes the MSDE algorithm for CF problem. In section 5 we show the computational results on test problems with various sizes, and Section 6 concludes the paper.

2. Literature review

In this section, we review the previous research on CF problem. Therefore, we can classify these studies by date of research. Before 1995, integer programming formulations of CFP solved by non-traditional methods such as simulated annealing (SA) [13], genetic algorithm (GA) [14], tabu search (TS) [15] and Neural network [16]. Since 1996 until 2000, researches have dedicated the use of multi-criteria CF problem on special areas of science. For instance, evaluate and compare the performance of CMS [17], CM scheduling problems [18] and fuzzy evolutionary algorithms in CF problem [19]. Most studies after year 2000, have been on designing of a CMS [20] and evaluate the performance of a CMS using machine reliability [21]. Also, the GA and TS algorithm and utilize of sequence data have considered for inter cellular movements [22]. In order to evaluate the goodness of the cell formation, a good number of performance measures have been proposed in the literature [23]. In 2006, Geonwook & Herman formed part families and designed machine cells by using of GA under demand changes [24]. Also, Wu et al., designed a CMS Concurrently using GA [25]. In 2007, Mahdavi et al., designed a new mathematical model for CMS based on cell utilization [26], Tavakkoli-Moghaddam et al., designed a facility layout problem in CMS with stochastic demands [27], and, Das et al., worked on machine reliability and preventive maintenance planning in a CMS[28]. Recently, Wu et al., used SA algorithm in a CMS [29] and Safaei et al., proposed a hybrid SA for solving an extended model of dynamic CMS [30].

The related previous work on DE algorithm was started by Wang & Chou in 1997. They used the DE algorithm in control problems and the time of distinguish algebraic equations problems [31]. Afterwards, DE utilized extensively for scientific approach, fuzzy logic and neural networks in 1998 [32]. In 2000, DE algorithms have been used in several areas such as electrical energy distribution [33], pharmacy [34], science of aviation and magnetic [35], chemical engineering [36] and environmental science [37]. Also, it applied in group technology and linear-programming models in 2001 [38], medical science in 2003, optimizing non-linear process in 2006 and in 2007 for optimizing functions with unknown parameters. It has been shown to perform better than GA [39] or particle swarm optimization (PSO) [40] over several numerical benchmarks [41]. Recently, some researchers succeeded to use this algorithm for optimizing of traveling salesman problem (TSP) [42], machines layout [43] and Flow Shop Scheduling [44] problems.

3. Dynamic CF model

In this section, we present a dynamic integer CF model including three sub-objective functions as cost parameters.

3.1. **Assumptions.** The following assumptions are considered in the proposed dynamic cell formation problem:

(i) The operating times for all part type operations on different machine types are known.
(ii) Each machine type can perform several operations (machine flexibility).
(iii) Operating cost of each machine type per hour is known.
(iv) Machines are available at the start of each period (no installation time).
(v) Investment or purchase cost of each type of machine in each period is known.
(vi) Bounds and quantity of machines in each cell are constant.
(vii) Inter-cell relocation costs are constant for all moves regardless of the distance traveled.
(viii) Machine relocation from one cell to another is performed between periods with no time.
(ix) Parts are moved between any two cells in batches and the inter-cell cost per batch between cells is known and constant.
(x) Batch size is constant for all productions in all periods.
(xi) The demand for each part type in different period is dynamic.
(xii) Setup times are not considered.

3.2. **Indexing sets.** j: index for operations j = 1,. . . ,O_p
p: index for parts p = 1,. . . ,P
m: index for machines m = 1,. . . ,M
c: index for cells c = 1,. . .,C
h: index for periods h = 1,. . .,H

3.3. **Input parameters.** C_m : purchase cost of machine type m
IC : inter-cell material handling cost per batch
D_{ph} : demand for product p in period h
B_{int} : batch size for inter-cell material handling
MC_m : relocation cost of machine type m
L_B : lower bound of cell size
U_B : upper bound of cell size
T_m : available time for each machine type m
t_{jp} : time required to perform operation j on product p

3.4. **Decision variables.**

$$B_{jpch} : \begin{cases} 1 & \text{if part type } p \text{ remains in cell } c \text{ after operation } j \text{ in period } h \\ 0 & \text{other wise} \end{cases}$$

$$X_{jpch} : \begin{cases} 1 & \text{if operation } j \text{ of part type } p \text{ is done in cell } c \text{ in period } h \\ 0 & \text{other wise} \end{cases}$$

N_{mch} : number of machines of type m devoted in cell c during period h
$K^{+}{}_{mch}$: number of machines of type m added in cell c during period h

$K^{-}{}_{mch}$: number of machines of type m removed from cell c during period h

3.5. **Mathematical formulation.**

$$Min \sum_{m=1}^{M}\sum_{c=1}^{C}\sum_{h=1}^{H}(C_m N_{mch}) \quad + \quad \sum_{j=1}^{Op}\sum_{p=1}^{P}\sum_{c=1}^{C}\sum_{h=1}^{H}(IC \times B_{jpch} \times \frac{D_{ph}}{B_{int}}) + \sum_{m=1}^{M}\sum_{c=1}^{C}\sum_{h=1}^{H} MC_m(k^{+}_{mch} + k^{-}_{mch}) \tag{3.1}$$

s.t:

$$\sum_{c=1}^{C} x_{jpch} = 1, \qquad \forall p, j, h, \tag{3.2}$$

$$\sum_{p=1}^{P}\sum_{j=1}^{O_p} D_{ph} t_{jp} x_{jpch} \le T_m N_{mch}, \qquad \forall m, c, h, \tag{3.3}$$

$$l_B \le \sum_{m=1}^{M} N_{mch} \le u_B, \qquad \forall c, h, \tag{3.4}$$

$$N_{mch} = N_{mc(h-1)} + k^{+}_{mch} - k^{-}_{mch}, \qquad \forall m, c, h, \tag{3.5}$$

$$x_{jpch} + x_{(j+1)pch} - B_{jpch} \le 1, \qquad \forall j, p, c, h, \tag{3.6}$$

$$B_{jpch}, x_{jpch} \in \{0, 1\},$$

$N_{mch}, k^{+}_{mch}, k^{-}_{mch} \ge 0$ and integer.

3.6. **Sub-objective functions.** *According to the proposed dynamic CF problem, multiple costs are considered in design of objective function. Meanwhile, it is not possible to consider all costs in the model due to the complexity and computational time required. Here, objective function consists of three cost parameters. The objective is to minimize the sum of the following costs:*

(i) *Machine annual cost: This cost includes investment and amortization cost per period and calculates based on the number of machines of each type used for specific period. Increasing of this cost could lead to the high total cost of companies.*

$$\min f_1 = \sum_{m=1}^{M}\sum_{c=1}^{C}\sum_{h=1}^{H}(C_m N_{mch}), \tag{3.7}$$

(ii) *Inter-cell handling costs: One of the major problems in design of a CF, is Inter-cell moves. The cost of transferring parts between cells incurred when parts cannot be produced completely by a machine type or in a single cell. Inter-cell moves decrease the efficiency of cellular manufacturing by complicating production control and increasing material handling requirements and flow time.*

$$\min f_2 = \sum_{j=1}^{Op}\sum_{p=1}^{P}\sum_{c=1}^{C}\sum_{h=1}^{H}(IC \times B_{jpch} \times \frac{D_{ph}}{B_{int}}), \tag{3.8}$$

(iii) *Machine relocation cost during different period: The cost of relocating machines from one cell to another between periods. In dynamic and stochastic production systems, the best CF design for one period may not be an efficient design for subsequent periods. By rearranging the manufacturing cells, the CF can continue operating efficiently as the product mix and demand change. Moving machines from cell to cell requires effort and can lead to the disruption of production.*

$$\min f_3 = \sum_{m=1}^{M}\sum_{c=1}^{C}\sum_{h=1}^{H} MC_m(k^+_{mch} + k^-_{mch}). \tag{3.9}$$

3.7. **Model constraints.** Primary constraints of model include size of cells and binary decision variables. Constraint (3.2) ensures that each part operation is assigned to one machine and one cell. Constraint (3.3) ensures that machines capacities have not exceeded and can satisfy the demand. Constraints (3.4) specify the lower and upper bounds of cells. Constraint (3.5) ensures that the number of machines in current period is equal to the aggregated number of machines in the previous plus the number of machines being moved in and subtracted by the number of machines being moved out. In other words, they ensure conservation of machines over the horizon. In constraint (3.6), if at least one of the operations of part p is proceed in cell c in period h, then the value of B_{jpch} will be equal to one; otherwise it is equal to zero.

4. Differential Evolution Algorithm

4.1. **MSDE optimization process.** MSDE algorithm rectifies the mutation rules, which has done by random selection of three chromosomes within solution space and using the environment of the best selected chromosome for any mutant chromosome. This algorithm also improves the scale factor of F_i by computing the mutation probability of P_r from normal distribution. This algorithm consists of four steps of initialization, mutation, crossovers and selection, which are repeated until receiving of near-optimal solution. One of the important characteristics of proposed MSDE algorithm in this paper is that refreshes itself in each generation number for k=1,. . . ,P_S by vacating of useless information, which can accelerate the process of algorithm. These steps are explained as follows:

A. Initialization:

The first step in MSDE algorithm is to create an initial population of candidate solutions by assigning random values to each decision variable of each chromosome for the population. In this step also the parameters of N_C (number of chromosomes) which i=1,2,. . . ,N_C, P_S and CR have adjusted and the scale factor of F_i initializes for the first population with normal distribution, N(0.5,0.15), where 0.5 is the mean and 0.15 is the standard deviation and will going to generate random number between (0,1].

B. Mutation:

The principal idea of MSDE is a new scheme for generating trial parameter vectors by adding the weighted difference vector between two population members to a third member. The mutation operator is in charge of introducing new parameters into the population. Therefore, it creates mutant vectors according to Eq. (4.1) by computing the scale factor of F_i according to Eq. (4.2), where (X_{r1}, X_{r2}, X_{r3}) are three chromosomes which selected randomly from S={$X_1{}^k$, $X_2{}^k$,..., $X_{Nc}{}^k$}and X_{rb} is the best of them. Also, (F_{r4}, F_{r5}, F_{r6}) are three factor scale vectors which selected randomly from current population that are not similar for any two chromosomes.

$$V_i^{(K)} = X_{rb} + F_i^{(K)}(X_{r2}^{(K)} - X_{r3}^{(K)}), \tag{4.1}$$

$$F_i^{(K)} = F_{r4}^{(K)} + N(0.05)(F_{r5}^{(K)} - F_{r6}^{(K)}), \qquad r_4 \neq r_5 \neq r_6, \tag{4.2}$$

C. Crossover:
The crossover operator creates the trial vectors, which are used in selection process. A trail vector ($U_{i,j}$) is a combination of a mutant vector and a parent vector that compared against the crossover constant called CR for each gene of population chromosomes. If the value of the random number is less or equal than CR, the parameter will select from the mutant vector ($V_{i,j}$), otherwise from the parent vector ($X_{i,j}$) as given in Eq. (4.3). The crossover operator maintains diversity in the population, preventing from local minimum convergence.

$$u_{i,j} = \begin{cases} V_{i,j} & if\ \ rand(0,1) \leq Pc \\ X_{i,j} & otherwise \end{cases} \qquad j = 1...n, \tag{4.3}$$

D. Selection:
The selection operator chooses the vectors that are going to compose the population in the next generation. This operator compares the fitness of the trial vector {f($U_{i,j}$)} corresponding to target vector {f($X_{i,j}$)} and selects the one that has better target function for k=1,. . . ,P_S according to Eq. (4.4).

$$f(U_{i,j}) < f(X_{i,j}). \tag{4.4}$$

Chromosome structure for CF problem
A chromosome or feasible solution proportional to the described CF model consists of the following genes in each period.

(i) (a) The gene related to the number of available machines in each cell in each period, is called [N_{mch}] matrix. The gene is limited to 0 up to U_B and has three dimensions of *m*, *c* and *h*.
(b) The gene related to the number of moved machines inside or outside of cell in each cell in each period, is called [K_{mch}] matrix. The gene is limited to L_B up to U_B. This gene could take positive or negative integer values and has three dimensions of *m*, *c* and *h*.
(c) The gene for distinguishing inter-cell transportation, is called [B_{jpch}] matrix. This gene is binary and has four dimensions of *j*, *p*, *c* and *h*, that by remaining of part *P* after operation *j* in cell, get the number 1 and lead to the inter-cell transportation cost, otherwise get the number zero and has no cost.

5. Computational results

In this section, in order to make the similar comparative condition of study for two computational optimizing methods, we utilized a 20 set of test problem, which generated randomly with various dimensions and then the proposed CF model would solved once by Lingo software as a classical method and another, by MSDE algorithm as a meta-heuristic optimizing method. It should be mentioned that due to increase of the reliability of assess process, we procure the crossover constant called P_c by several experiment. Therefore, we have selected the two experimental test problem first, and then have solved by different size of P_c in order to get the best quantity of P_c.

5.1. **Comparative result of optimizing methods.** Considering to the input parameters of proposed CF model, we have selected 20 test problems with different sizes up to the maximum number of defined. The proposed CF model have solved by optimizing methods in similar conditions and limitations in order to compare optimization methods exactly in two factors named, the time to get result and value of objective function. The coding language of MSDE algorithm has been C^{++} and utilized by the processor of 3.00 GB with 2.00 GB of RAM to compute the CF model. Table 1 dedicates the comparative result of test problems with Lingo 8 and MSDE algorithm separately.
The CF input parameters used in all problems are as follows:

(i) Purchase cost of machine type m (C_m) = [500,1000]
(ii) Inter-cell material handling cost per batch (IC) = 50
(iii) Demand for product p in period h (D_{ph}) = [100,200]
(iv) Batch size for inter-cell material handling (B_{int}) = 50
(v) Relocation cost of machine type m (MC_m) = [100,150]
(vi) The min/max capacity of cell: $LB = 1$, $UB = M/2$, where M = number of machines
(vii) Available time for each machine type m (T_m) = [120,600] s
(viii) Time required to perform operation j on product p (t_{jp}) = [5,30] s
(ix) $CR = 0.2$, $N_C = 1000$, $P_S = 100$

According to the table, the decline percent by using of MSDE has grown with large scale of input parameters. Decline percent is on range of 70% for small tests that has been received up to 97% for larger than 500000 dimensions tests.

Table 1. Comparative result of optimizing methods

No	***Problem Dimensions*** *j-p-m-c-h*	***Lingo result*** *runtime (s) function*		***MSDE result*** *runtime (s) function*		*Decline percent in MSDE*
1	4-4-3-2-2	3	1730	4	1800	-0.33%
2	5-4-4-3-2	28	3900	8	4300	71%
3	6-5-4-3-3	50	10600	14	10600	72%
4	9-8-8-4-2	490	25100	26	25100	94.7%
5	10-10-8-4-3	2400	54700	49	54700	98%
6	12-10-10-3-3	2250	56300	44	48700	98%
7	14-12-10-4-3	2900	92500	75	93700	97.4%
8	15-15-12-4-4	5040	170500	126	170500	97.5%
9	18-18-17-3-3	10700	128200	99	129500	99%
10	20-20-18-4-3	>10h	-	164	229100	-
11	24-20-20-4-4	>10 h	-	251	373100	-
12	25-25-22-5-3	>10 h	-	320	458000	-
13	30-25-25-4-5	>10 h	-	458	736000	-
14	30-30-30-5-5	>10 h	-	678	1108400	-
15	35-32-30-5-5	>10 h	-	803	1383000	-
16	35-35-32-6-5	>10 h	-	1058	1818800	-
17	40-38-38-5-6	>10 h	-	1308	2257100	-
18	42-42-40-6-6	>10 h	-	1784	3148200	-
19	50-45-42-6-5	>10 h	-	1830	3347200	-
20	50-50-48-6-5	>10 h	-	2095	3724300	-

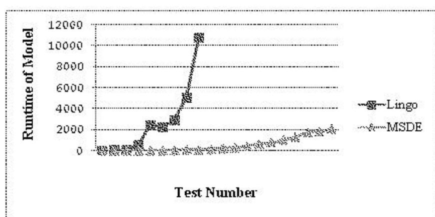


Diagram 1. Compare the runtime of CFP for two methods

As the diagram shows, the trend of runtime for CF problem has increased rapidly by Lingo software with large dimension parameters. These trends dedicate that although using of classical optimizing method cannot be able to solve nonlinear and non-differentiable functions efficiently, but evolutionary algorithm such as DE can be utilized effectively, specially for large scale problems.

5.2. **The convergence of MSDE algorithm.** Due to distinguish the behavior of MSDE algorithm at finding the optimum solution, we have proceeded to depict the diagrams which can construe the algorithm reliability for different positions of input parameters. Therefore, we selected the two test problems with different dimensions, as the problem with 6 operations, 5 parts, 4 machines, 3 cells and 3 periods and 1080 dimensions as a small test problem and the problem with 35 operations, 35 parts, 32 machines, 6 cells and 5 periods and 1176000 dimensions as a large dimension one. The convergence of MSDE algorithm has considered for two test problems by different quantity of generation number and calculated for

N_g= 100, 200, 400, 600, 800, 1000 according to the objective function. The results have shown in diagrams 2 and 3 for two problems separately.

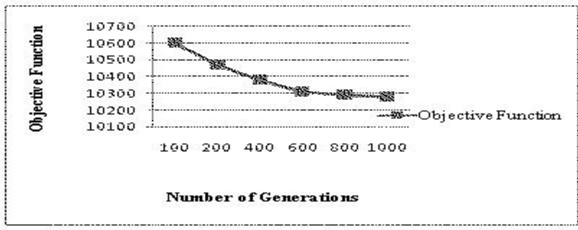


Diagram 2. Convergence diagram for test problem with (6,5,4,3,3) as (j,p,m,c,h)

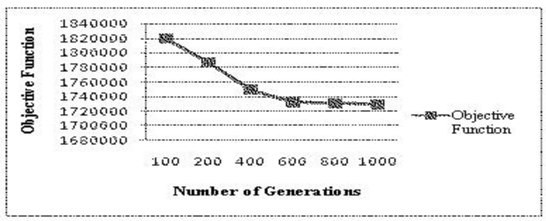


Diagram 3. Convergence diagram for test problem with (35,35,32,6,5) as (j,p,m,c,h)

6. Conclusion

The differential evolution algorithm is one the newest and important evolutionary algorithm, which can used property for optimizing of variety functions. According to the result, MSDE decreases highly the computational time. Whereas the cell formation problem is a hard-problem, so using classical optimizing method increases the computational time and cannot be reliable. Furthermore, due to the diversity of products and application of cellular manufacturing systems, evolutionary algorithms such as DE can be lucrative. Therefore, it is recommended that utilize evolutionary algorithm for solving non-linear and multi-criteria functions such as cell formation problem.

References

[1] Askin, R. G., & Estrada, S. (1999). Investigation of cellular manufacturing practices. In S. A. Irani (Ed.), Handbook of cellular manufacturing systems (pp. 25–34). New York: John Wiley and Sons (chapter1).

[2] Boughton, N. J., & Arokiam, I. C. (2000). The application of the cellular manufacturing: A regional small to medium enterprise perspective. Proceedings of the Institution of Mechanical Engineers, 214(Part B), 751–754.

[3] Vakharia, A. J., & Wemmerlov, U. (1990). Designing a cellular manufacturing system: a materials flow approach based on operations sequences. IIE Trans, 22, 84–97.

[4] Lee S. D., & Chen, Y. L. (1997). A weighted approach for cellular manufacturing design: minimizing inter-cell movement and balancing workload among duplicate machines. Int J Prod Res, 35, 1125–1146.

[5] Jong, D., & Kenneth, A. (2006). Evolutionary computation. The MIT Press, p256.

[6] Price, K.V., Storn, R., & Lampinen, J. (2005). Differential Evolution: A Practical Approach to Global Optimization. Springer, Berlin.

[7] Bastürk, A., & Günay, E. (2009). Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm. Expert System with Applications, 36(3.8), 2645–2650.

[8] Ursem, K., & Vadstrup, P. (2003). Parameter identification of induction motors using differential evolution, in: Proceedings of CEC 2003. IEEE Computer Press, 2, 790–796.

[9] Omran, M. G. H., Engelbrecht, A. P., & Salman, A. (2005). Differential evolution methods for unsupervised image classification, in: Proceedings of IEEE Congress on Evolutionary Computation, IEEE Computer Press, 2, 966–973.

[10] Paterlini, S., & Krink, T. (2005). Differential evolution and particle swarm optimization in partitional clustering. Comput. Stat. Data Anal., 50(3.5), 1220–1247.

[11] Babu, B.V., & Munawar, S.A. (2007). Differential evolution strategies for optimal design of shell-and-tube heat exchangers. Chem. Eng. Sci., 62(**??**), 2739–3720.

[12] Nearchou, A.C., & Omirou, S.L. (2006). Differential evolution for sequencing and scheduling optimization. J. Heuristics, 12, 395–411.

[13] Venugopal, V., & Narendran, T. T. (1992). Cell formation in manufacturing systems through simulated annealing: an experimental evaluation. International Journal of Production Research, 63(3.3), 409–422.

[14] Joines, J. A. (1993). Manufacturing cell design using genetic algorithms. MSc Thesis. Raleigh, NC: North Carolina State University.

[15] Sun, D., Lin, L., & Batta, R. (1995). Cell formation using tabu search. Computers and Industrial Engineering, 28, 485–494.

[16] Chen, S. J., & Cheng, C. S. (1995). A neural network based cell formation algorithm in cellular manufacturing. International Journal of Production Research, 33(3.2), 293–318.

[17] Jeon, G., Leep, H. R., & Parsei, H. R. (1998). Cellular manufacturing system based on new similarity coefficient which considers alternative routes during machine failure. Computers and Industrial Engineering, 34(**??**), 21–36.

[18] Lockwood, W., Mahmoodi, F., & Mosier, C. (2000). Scheduling unbalanced cellular manufacturing systems with lot splitting. International Journal of Production Research, 38, 951–966.

[19] Suresh, N. C., Slomp, J., & Kaparthi, S. (1999). Sequence-dependent clustering of parts and machines, a Fuzzy ART neural network approach. International Journal of Production Research, 37(4.3), 2793–2816.

[20] Poornachandra, R., Panchalavarapu, V., & Chankong. (2005). Design of cellular manufacturing systems with assembly considerations. Computers and Industrial Engineering, 48(3.3), 449-469.

[21] Seifoddini, S., & Djassemi, M. (2001). The effect of reliability consideration on the application of quality index. Computers and Industrial Engineering, 40(1–2), 65–77.

[22] Wu, T. H., Low, C., & Wu, W. T. (2004). A tabu search approach to the cell formation problem. International Journal of Advanced Manufacturing Technology, 23, 916–924.

[23] Zolfaghari, S., & Liang, M. (2003). A new genetic algorithm for the machine/part grouping problem involving processing times and lot sizes. Computers and Industrial Engineering, 45, 713–731.

[24] Geonwook, J., & Herman, R. L. (2006). Forming part families by using genetic algorithm and designing machine cells under demand changes. Computers & Operations Research, 33, 263–283.

[25] Wu, X. D., Chu, C. H., Wang, Y. F., & Yan, W. L. (2006). Concurrent design of cellular manufacturing systems: A genetic algorithm approach. International Journal of Production Research, 44(3.6), 1217–1241.

[26] Mahdavi, I., Javadi, B. F., Alipour, K., & Slomp, J. (2007). Designing a new mathematical model for cellular manufacturing system based on cell utilization. Applied Mathematics and Computation, 190, 662–670.

[27] Tavakkoli-Moghaddam, R., Javadian, N., Javadi, B., & Safaei, N. (2007). Design of a facility layout problem in cellular manufacturing systems with stochastic demands. Applied Mathematics and Computation, 184, 721–728.

[28] Das K., Lashkari, R. S., & Sengupta, S. (2007). Machine reliability and preventive maintenance planning for cellular manufacturing systems. European Journal of Operational Research, 183, 162–180.

[29] Wu, T. H., Chang, C. C., & Chung, S. H. (2008). A simulated annealing algorithm for manufacturing cell formation problems. Expert Systems with Applications, 34, 1609–1617.

[30] Safaei, N., Saidi-Mehrabad, M., & Jabal-Ameli, M. S. (2008). A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system. European Journal of Operational Research, 185, 563–592.

[31] Wang, F. S., & Chiou, J. P. (1997). Optimal control and optimal time location problems of differential-algebraic systems by differential evolution. Ind Industrial and Engineering Chemistry Research, 36, 5348– 5357.

[32] Schmitz, G. J., & Aldrich, C. (1998). Neuro fuzzy modeling of chemical process systems with ellipsoidal radial basis function neural Networks and genetic algorithms. Computers & Chemical Engineering, 1001–1004.

[33] Chang, T., & Chang, H. C. (2000). An efficient approach for reducing harmonic voltage distortion in distribution systems with active power line conditioners. IEEE Transactions on Power Delivery, 15(3.3), 990–995.

[34] Lameh, J., Wang, P., Elgart, D., Meredith, D., Shafer, SL., & Loew, GH. (2000). Unraveling the identity of benzodiazepine binding sites in rat hippocampus and olfactory bulb. European Journal of Pharmacology, 400(2 –3), 167–176.

[35] Stumberger, G., Pahner, U., & Hameyer, K. (2000). Optimization of radial active magnetic bearings using the _nite element technique and the differential evolution algorithm. IEEE Transactions on Magnetics, 36(3.4), 1009–1013.

[36] Wang, F., & Sheu, J. (2000). Multi objective parameter estimation problems of fermentation processes using a high ethanol tolerance yeast. Chemical Engineering Science, 55(**??**), 3685–3695.

[37] Booty, W., Lam, C., Wong, I., & Siconol, P. (2000). Design and implementation of an environmental decision support system. Environmental Modelling and Software, 16(3.5), 453–458.

[38] Hendtlass, T. (2001). A combined swarm differential evolution algorithm for optimization problems. Lecture Notes in Computer Science, vol. 2070, Berlin, Springer.

[39] Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Harbor.

[40] Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. Proc. IEEE Int. conf. Neural Networks, p, 1942-1948.

[41] Vesterstrøm, J., & Thomson, R. (2004). A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, in Proc. Sixth Congress on Evolutionary Computation, (CEC-2004), IEEE Press.

[42] Onwubolu, G. C. (2003). Optimizing CNC drilling machine operations: traveling salesman problem. differential evolution approach.

[43] Nearchou, A. C. (2006). Meta-heuristics from nature for the loop layout design problem. International Journal of Production Economics, 101(3.2), 312–328.

[44] Onwubolu, G. O., & Davendra, D. (2006). Scheduling flow shops using differential evolution. European Journal of Operational Research, 169, 1176–1184.