Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 12 (2014) 186 – 195

Procedia Technology

The 7th International Conference Interdisciplinarity in Engineering (INTER-ENG 2013)

Sensitivity analysis of dynamic cell formation problem through meta-heuristic

Mohammad Hassannezhad*, Marco Cantamessa, Francesca Montagna, Fahad Mehmood

Department of Management and Production Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

Abstract

In spite of many researches in literature investigating dynamic of cell formation (CF) problem, further research needs to be elaborated to assay hidden aspects of cellular manufacturing system (CMS), due to inherent complexity and uncertainty on optimizing this problem. In this paper, sensitivity analysis of modified self-adaptive differential evolution (MSDE) algorithm is proposed for basic parameters of CF problem, considering to the graphical representation supported by statistical analysis. Hence, a dynamic integer model of CF problem is first presented as the NP-hard problem. Then, the two basic test CF problems are introduced thereby the performance of MSDE algorithm assessed by diverse problems sizes through 140 runs from aspects of the average runtime of algorithm and the best local optimum objective function. Finally, statistical analysis is implemented on behavior of objective function values in order to validate our computational results graphically as well as statistically, giving some insights related to importance of CF parameters on designing CMS.

Keywords: Meta-heuristic; Modified self-adaptive differential evolution algorithm; Sensitivity analysis; Cell formation problem.

* Corresponding author. Tel.: +39-011-0907295; fax: +39-011-5647299.
E-mail address: hassannezhad@polito.it

2212-0173 © 2013 The Authors. Published by Elsevier Ltd.
Selection and peer-review under responsibility of the Petru Maior University of Tirgu Mures.
doi:10.1016/j.protcy.2013.12.474

1. Introduction

Nowadays, according to fast technology improvement, shorter lifetime of products and speedy introduction of new products, managers are seeking to production styles that have higher efficiency and flexibility than traditional systems. Therefore, CMS has been introduced as a mixture of the work area of work-shop manufacturing and line-production system. One of the difficult steps of cell manufacturing design is CF problem in which parts with similar manufacturing processes will be made in one cell. As a NP-hard problem [1,2], most of recent researches attended to solve CF problem through meta-heuristics [3, 4, 5]. In spite of many researches on applying evolutionary algorithms to solve CF problem, the literature on differential evolution (DE) is rather poor than other algorithms [6], which motivated us to propose the modified version of DE and apply it on CF problem, aiming to assess the validity of this algorithm on CMS.

DE algorithm, initially proposed by Price and Storn [7], is a population-based algorithm that is applied for optimizing non-differentiable, non-convex, nonlinear and multi-objective functions [8-12]. It is a simple, powerful, parallel and direct search method with good convergence and fast implementation properties [7]. DE uses a simple mutation operator based on differences between pairs of solutions with aim of finding a search direction in current population. It also constitutes a rather efficient way to self-adapt the mutation operator, where the newly generated offspring competes against its corresponding parent and replaces it if the offspring has a higher fitness value. Moreover, DE rectifies the problem of premature convergence which previously observed in genetic algorithm (GA), where the population converges to some local optima of a multi-objective function [13]. It has been preferred to many other evolutionary techniques like GA and particle swarm optimization (PSO) due to its attractive characteristics such as its simple structure, convergence speed, versatility and robustness with only a few parameters required to be set by a user [14]. This paper proposes MSDE algorithm as the modified version of DE, trying to tackle the limits of the original version. Several researches in literature attempted to modify the original DE algorithm. For instance, Tasgetiren et al. [15] solved the generalized traveling salesman problem (TSP) using discrete DE. Das et al. [16] used a modified version of DE in pattern recognition of adaptive clustering. Ali and Torn [17] introduced a modified version of DE algorithm to improve efficiency and robustness. The modified DE algorithm proposed by Babu and Jehan [18] utilizes only one set of population against the two sets of original DE algorithm. Brest et al. [19] introduced self-adapting control parameter settings of DE algorithm.

Since the performance of meta-heuristics on multi-objective functions are affected by complexity of problems, due to the nature of these algorithms, the main contribution of this paper would be analysing validity of a modified version of DE for parameters of CF model by means of the two test problems to prove the convergence power of aforementioned algorithm. Since real-life cases mostly deal with large-scale problems with numerous variables, we aim to assess the performance of MSDE algorithm by varying the basic parameters of CF problem. Our focus is on behaviour of algorithm, especially in large-scale problems which will be complemented through statistical analysis. To our knowledge, this is the first research with focus on sensitivity of DE algorithm for dynamic multi-objective CF model so that the result can be used on construction of multi-objective CF models. The paper is organized as follows: The multi-objective integer CF model is described in section 2. Section 3 includes the proposed MSDE algorithm for solving the CF problem. Section 4 indicates the computational results and statistical analysis and finally, the paper is concluded in section 5.

2. Dynamic CF model

2.1. Assumptions

The following assumptions are considered in the proposed dynamic cell formation problem:

- The operating times for all part type operations on different machine types are known.
- The demand for each part type in different period is dynamic and deterministic.
- Each machine type can perform several operations (machine flexibility).
- Operating cost of each machine type per hour is known and constant in each period.
- Machines are available at the start of each period (no installation time).

- Investment or purchase cost of each type of machine in each period is known.
- Bounds and quantity of machines in each cell are constant.
- Inter-cell relocation costs are constant for all moves regardless of the distance travelled.
- Machine relocation from one cell to another is performed between periods with no time.
- Parts are moved between two cells in batches and inter-cell cost per batch between cells is known and constant.
- Batch size is constant for all productions in all periods.
- Setup times are not considered.

2.2. Indexing sets

j : index for number of operations; $j = 1,\ldots,O_p$
p : index for number of parts; $p = 1,\ldots,P$
m : index for machines' types; $m = 1,\ldots,M$
c : index for number of cells; $c = 1,. . .,C$
h : index for number of periods; $h = 1,. . .,H$

2.3. 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

2.4. Decision variables

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

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

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

2.5. Mathematical formulation

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

$$Subject\ to:\ \sum_{p=1}^{C} x_{jpch} = 1, \qquad \forall p, j, h \tag{2}$$

$$\sum_{p=1}^{P}\sum_{j=1}^{O_p} D_{ph} t_{jp} x_{jpch} \le T_m N_{mch} 1, \qquad \forall m,c,h \tag{3}$$

$$l_B \le \sum_{m=1}^{M} N_{mch} \le u_B, \qquad \forall c,h \tag{4}$$

$$N_{mch} = N_{mc(h-1)} + k_{mch}^{+} - k_{mch}^{-}, \qquad \forall m,c,h \tag{5}$$

$$x_{jpch} + x_{(j+1)pch} - B_{jpch} \le 1, \qquad \forall j,p,c,h \tag{6}$$

$$B_{jpch}, x_{jpch} \in \{0,1\}, N_{mch}, k_{mch}^{+}, k_{mch}^{-} \ge 0 \textit{ and integer.}$$

2.6. Sub-objective functions

Machine annual cost includes investment and amortization cost per period and calculates based on number of machines of each type used for specific period. Increasing this cost can lead to high total cost of companies.

$$\min f_1 = \sum_{m=1}^{M}\sum_{c=1}^{C}\sum_{h=1}^{H}\left(C_m N_{mch}\right) \tag{7}$$

Inter-cell handling costs as cost of transferring parts between cells incurred when parts cannot be produced completely by one machine type or in a single cell. Inter-cell moves decrease the efficiency of manufacturing cells by complicating production control and increasing material handling requirements and flow time.

$$\min f_2 = \sum_{j=1}^{O_p}\sum_{p=1}^{P}\sum_{c=1}^{C}\sum_{h=1}^{H}\left(IC \times B_{jpch} \times \frac{D_{ph}}{B_{\text{int}}}\right) \tag{8}$$

Machine relocation cost during different period. In dynamic production systems, the best CF design for one period may not be an efficient design for subsequent periods. By rearranging the manufacturing cells, CF can continue operating efficiently as the product composition and demand change.

$$\min f_3 = \sum_{m=1}^{M}\sum_{c=1}^{C}\sum_{h=1}^{H} MC_m \left(k_{mch}^{+} + k_{mch}^{-}\right) \tag{9}$$

2.7. Model constraints

Primary constraints of model include size of cells and binary decision variables. Constraint (2) ensures that each part operation is assigned to one machine and one cell. Constraint (3) ensures that machines capacities have not exceeded and can satisfy the demand. Constraints (4) specify the lower and upper bounds of cells. Constraint (5) ensures that number of machines in current period is equal to aggregated number of machines in previous plus the number of machines being moved in and subtracted by number of machines being moved out. In other words, they ensure conservation of machines over the corresponding period. In constraint (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.

3. MSDE algorithm for the dynamic CF problem

Unlike other meta-heuristics, DE has a few control parameters, such as scale factor (F_i), crossover probability (CR) and population size (P_S), which are fixed during the optimization process in original DE model. The robustness and effectiveness of DE algorithm directly depend on settings of control parameters, especially on mutation operation as its central procedure. MSDE in compare of DE algorithm rectifies the mutation rules, thus it selects randomly three chromosomes from solution area and uses surrounding 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 M_P from normal distribution. One of the important features of the proposed algorithm is that it refreshes itself in each generation for $k=1,...,P_S$, by vacating of useless information which can accelerate the process of algorithm. In other words, we contrive a recycle bin in program for non-dominated solutions that automatically empties itself by going to the next generation ($k=k+1$), preventing to transfer extra information to the next generation in computing.

3.1. Chromosome structure (Solution coding)

Number of available machines in each cell in each period, $[N_{mch}]$ matrix. The gene is limited to *0* up to U_B and has three dimensions of *m*, *c* and *h*.

Number of moved machines inside or outside of cell in each cell in each period, $[K_{mch}]$ matrix. The gene is limited to L_B up to U_B and can take positive or negative integer values and has three dimensions of m, c, h.

Distinguishing inter-cell transportation, $[B_{jpch}]$ matrix. This gene is binary and has four dimensions of *j*, *p*, *c* and *h*, so as 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.

3.2. Fitness function

Generally, fitness function provides an evaluation of each individual, allowing all individuals to be mapped into a totally ordered set. By considering the explained CF model, new solution is accepted when its objective function value is less than compared with its parents.

3.3. MSDE optimization cycle

Initialization creates an initial population of candidate solutions by assigning random values to each decision variable for each chromosome. In this step also the parameters of N_C (number of chromosomes) for $i=1,2,...,N_C$, P_S and *CR* have adjusted and the scale factor of F_i initializes for the first population by normal distribution of *N(0.5,0.15)* which generates random numbers between (0,1].

Mutation: The mutation operator is in charge of introducing new parameters into population. Hence, it creates mutant vectors according to Eq. (10) by computing scale factor of F_i according to Eq. (11), where (X_{r1}, X_{r2}, X_{r3}) are three chromosomes which selected randomly from $S=\{X_{1k}, X_{2k},..., X_{Nck}\}$and X_{rb} is the best of them. Also, (F_{r4}, F_{r5}, F_{r6}) are factor scale vectors which selected randomly from current population and differ for any two chromosomes.

$$V_i^{(K)} = X_{rb} + F_i^{(K)}\left(X_{r2}^{(K)} - X_{r3}^{(K)}\right) \tag{10}$$

$$F_i^{(K)} = F_{r4}^{(K)} + N(0.05)\left(F_{r5}^{(K)} - F_{r6}^{(K)}\right), \qquad r_4 \neq r_5 \neq r_6 \tag{11}$$

Crossover creates trial vectors that maintain diversity in population, preventing from local minimum convergence. 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 random number is less or equal than *CR*, the parameter will select from mutant vector ($V_{i,j}$), otherwise from the parent vector ($X_{i,j}$) as given in Eq. (12).

$$U_{i,j} = \begin{cases} V_{i,j} & if \quad randj(0,1) \leq CR \\ X_{i,j} & otherwise \end{cases} \qquad j = 1,2,\ldots,n \tag{12}$$

Selection operator compares the fitness of trial vector $\{f(U_{i,j})\}$ corresponding to target vector $\{f(X_{i,j})\}$ and selects the vector with better objective function for $k=1,\ldots,P_S$ according to Eq. (13).

$$f(U_{i,j}) < f(X_{i,j}) \tag{13}$$

3.4. Stopping criteria

Considering to the literature of evolutionary algorithms, there are two types of stopping region, as number of generations and time interval. In this study, we specify the number of generations for stopping algorithm ($k=P_S$).

4. Computational results and analysis

4.1. Problem description

Following the goal of our study we aim to analyse the parameters of CF model based on their effect on basic model computed by MSDE. The reason on proposing meta-heuristic for CF problem is investigated extensively in literature, as mentioned before in section 1. In spite of good performance of classical optimization methods like branch and bound on finding optimal solutions for small-scale problems, solving NP-hard models using classical optimization methods needs long computational time and makes these methods useless. In this way, by using proposed meta-heuristic, two different dimensions test problems are used to take the efficiency of MSDE into account. For each, we vary the parameters up to 300 percentage for all CF input parameters and assay sensitivity of MSDE algorithm from aspects of average runtime of algorithm and the best objective function value (OFV) for each test problem. As the computational time of evolutionary algorithms is varied by different operation systems, due to the nature of these algorithms, we run each test problem for four times and consider the average runtime as well as the best OFV of the tests. Statistical analysis is meanwhile implemented exclusively for local optimum solutions (OFVs). All computations are done on a PC Pentium IV (3.00 GHz and 2.00 GB RAM) and the meta-heuristic is codified by visual C^{++}. In order to enhance the validity of evaluation process, we procure the crossover constant (*CR*) for MSDE algorithm through several experiments. Therefore, we selected the two experimental test problem first, and then solved by different size of *CR* in order to get the best quantity for crossover constant. The two basic test problems have been suggested with input parameters as below:

Model *a*; Small-scale test model with 12 operations, 8 parts, 6 machines, 4 cells in 2 periods

Model *b*; Large-scale test model with 24 operations, 22 parts, 18 machines, 6 cells in 4 periods

4.2. Computational results for small-scale problem (model a)

In this section, dealing with sensitivity analysis of CF parameters and the relationship between them is considered. Hence, a problem with 12 operations, 8 parts, 6 machines, 4 cells in 2 periods considered and tested in a range of up to 300 percentage increase, subject to change one parameter meaningfully along with fix other parameters. Therefore, we have 140 runs with 35 outputs for each test problem, as the result is shown in Table 1. Graphical demonstrations for OFVs are done in the scale of (1/1000), due to increase of accuracy. For instance, the 150% variation is applied to the number of periods in test problem model *a*, thereby became to a problem with 12 operations, 8 parts, 6 machines, 4 cells in 5 periods. MSDE algorithm found the local optimal OFV of 8950 (8.95 in Table 1) in average time of 69 seconds after four runs for this test problem which lead to 61% increase on runtime of algorithm. By integrating the result for each parameter, the comparison between components would be noticeable. Also, to provide a better understanding of behaviour for algorithm, the graphical representation is provided for both runtime and OFV of CF problem, as Fig. 1 a) and b) show. As expected theoretically, number of periods and cells has the most effect on CF structure. The reason can be increase of variables and the total cost of CF model by number of cycles, based on mathematical formulation. Logically, increasing the number of cells leads to the high level of inter-cellular costs as well as movement between cells to complete the product cycle (see section *3.6*). After that, number of parts and operations are highlighted and finally, number of machines which has a little different trend from the others. The reason can be laid on the model assumption since we fix the number of machines in each period. This led to rather monotonic behaviour of machine especially in sense of OFVs, while the other parameters demonstrate similar trend by increasing the problem variables for OFV. As a result, an analogous behaviour of different parameters for OFV of the CF model can interpret the robustness of MSDE algorithm on finding near-optimal solution even in large-scale problems. Besides, we understd that the role of each parameter is heavily depended on type of objective function of CF model as well as its constraints.

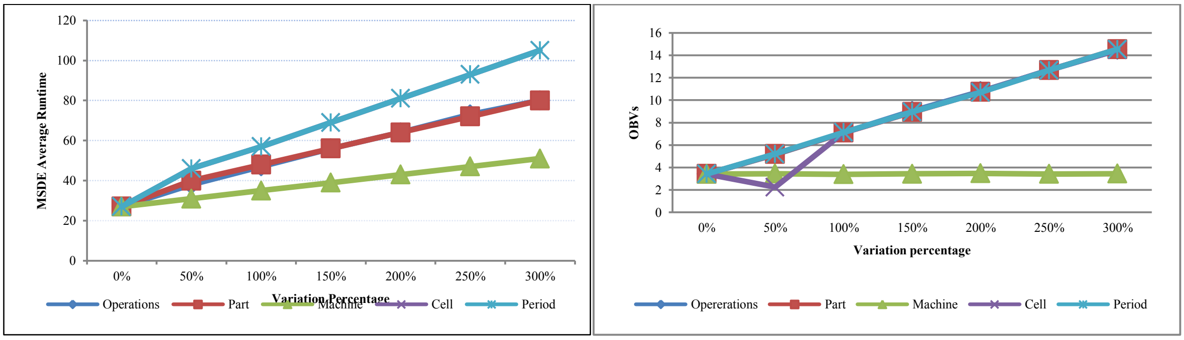
Table 1. Computational results for sensitivity analysis of CF by model *a*

Basic Test Problem: $j = 12$, $p = 8$, $m = 6$, $c = 4$, $h = 2$

Variation Domain	Operations (j)		Part (p)		Machine (m)		Cell (c)		Period (h)	
	Time	OFV	Time	OFV	Time	OFV	Time	OFV	Time	OFV
0%	27	3.42	27	3.42	27	3.42	27	3.42	27	3.42
50%	38*	5.17	40	5.19	31	3.45	46	5.24	46	5.21
100%	47	7.11	48	7.10	35	3.40	57	7.11	57	7.13
150%	56	9.03	56	8.93	39	3.43	69	8.96	69	8.95
200%	64	10.79	64	10.73	43	3.48	81	10.73	81	10.69
250%	73	12.67	72	12.67	47	3.42	93	12.69	93	12.66**
300%	80	14.47	80	14.55	51	3.45	105	14.47	105	14.55

* The average runtime through MSDE algorithm for the problem as: $j = 18$, $p = 8$, $m = 6$, $c = 4$, $h = 2$.

** The best objective function value through MSDE algorithm for the problem as: $j = 12$, $p = 8$, $m = 6$, $c = 4$, $h = 5$.



a) b)

Fig. 1. a) Graphical demonstration for sensitivity analysis of model a by runtime; b) Graphical demonstration for sensitivity analysis of model a by OFV.

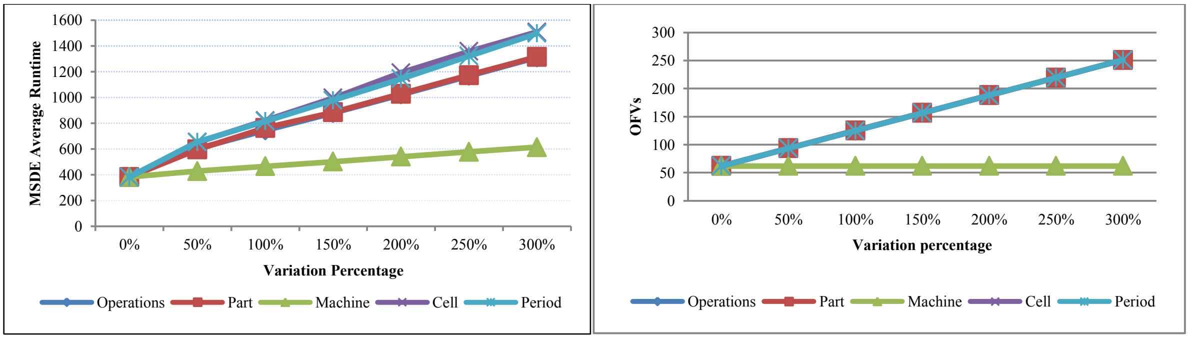
4.3. Computational results for large-scale problem (model b)

As the number of variables and complexity of model heavily affect optimization methods, we made a similar analysis for a large-scale problem, as the result is indicated in Table 2 as well as Fig. a) and b). In spite of similar behaviour by MSDE on both aspects of runtime and OFV, higher level of runtime consumed to find a near-optimal solution is noticeable for all parameters. This homogeneous behaviour on two-size problems convincingly can interpret as convergence speed and validity of MSDE on optimizing of complex models. These observations from the result of MSDE can, as similar as small-scale problem, give us an integrated view on understanding importance of each factor. However, the results obtained on the two test problems are greatly depended on the objective function formulation (see section *3.5*). So, based on the discussion above and on the analogous behaviour of MSDE algorithm, regardless of the size of problem, number of periods and cells should be accurately nominated in order to prevent extra costs of manufacturing system. This analysis also can validate the basis of CMS structures from another vision on assigning part families to machine families in different cells which should be done with the objective of minimizing total product cost with lowest number of cells.

Table 2. Computational results of CF parameters variation for model *B*

Basic Test Problem: $j = 24$, $p = 22$, $m = 18$, $c = 6$, $h = 4$

Variation Domain	Operations (j)		Part (p)						Machine (m)		Cell (c)		Period (h)
	Time	OFV	Time	OFV				Time	OFV	Time	OFV	Time	OFV
0%	382	62.07	382	62.07				382	62.07	382	62.07	382	62.07
50%	602	93.53	595	93.51				428	62.06	651	93.53	655	93.56
100%	744	124.88	763	125.34				465	62.06	821	124.87	815	124.87
150%	878	156.54	884	156.54				502	62.11	995	156.53	977	156.56
200%	1022	188.04	1028	188.05				540	62.07	1192	188.08	1145	188.03
250%	1166	219.47	1172	219.44				577	62.15	1356	219.38	1320	219.39
300%			1310	250.98	1315	251.01	614	62.08	1507	251.01	1496		251.05



a) b)

Fig. 2. a) Graphical demonstration for sensitivity analysis of model b by runtime; b) Graphical demonstration for sensitivity analysis of model b by OFV.

4.4. Statistical analysis

In order to verify the previous findings, in this section, statistical analysis is discussed, with respect to correlation between parameters as distinct sets. Due to variability of runtime for meta-heuristics by different PCs, we consider only OFVs for statistical analysis of independent observations. Moreover, as number of machines assumed constant in each period in CF model, the result indicates considerable distinction between machine and other parameters which forced us to eliminate this parameter from statistical analysis process. If we suppose output of MSDE algorithm as independent observations, we need to convert them in a same scale for comparative study. Derivative function seemed to us the best way which can apply into observations. Fig. 3.a indicates the result as well as graphical comparative demonstration, considering to the following phrases;

i : Index of variation percentage (i = 1,2,...,7)
j : Index of CF parameter (j = 1,...,4)
X_i : Amount of variation percentage
Y_{ij} : Independent observation based on variation percentage of i for CF parameter of j

R_{ij} : Derivative change rate of OFV, while $R_{ij} = \frac{Y_{ij} - Y_{(i-1)j}}{X_i - X_{(i-1)}}$ (14)

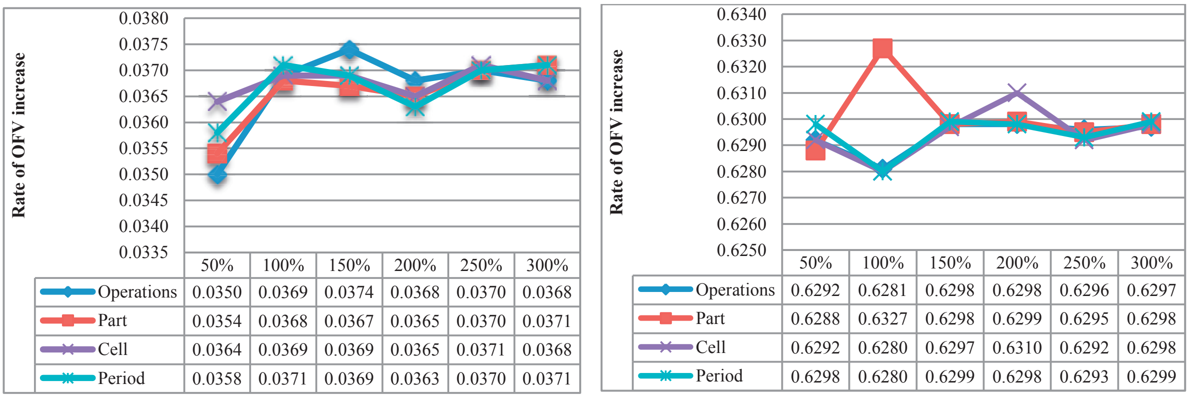
It is clear that we can compute R_{ij} for $i \geq 2$ ($X_i = 50$). For instance, for 50% variation of the number of operations in test model a, the following information is supposed;

$$X_1 = 50\ ;\ X_0 = 0\ ;\ Y_{11} = 5.17\ ;\ Y_{01} = 3.42$$

So the derivative change rate of OFV for 50% variation in number of operations would be the result of following equation.

$$R_{11} = \frac{5.17 - 3.42}{50 - 0} = 0.035$$

The other components of the Fig. 3 a) and b) can compute in a similar way based on information of the Tables 1 and 2 for OFVs. By converting MSDE outputs into the same scale, it is promising comparison of parameters behaviour by increasing the size of original test problem. In spite of the similar behaviour of all parameters based on OFV demonstration (see Fig. 1.b and Fig. 2.b), statistical analysis reveals that number of periods and cells are eligible enough to consider more on designing CMS due to high level of effect on total cost of CF problem. Moreover, number of parts which are allocated to different cells can play a key role on reducing handling costs either for inter- or between cells, while the number of operations has more uniform trend by the size of problems. As the result, applying MSDE to CF problem heavily verified previous researches on CF theories; however, it seems cogent dependency of our analysis on type of objective functions, especially on model assumptions.



	50%	100%	150%	200%	250%	300%
Operations	0.0350	0.0369	0.0374	0.0368	0.0370	0.0368
Part	0.0354	0.0368	0.0367	0.0365	0.0370	0.0371
Cell	0.0364	0.0369	0.0369	0.0365	0.0371	0.0368
Period	0.0358	0.0371	0.0369	0.0363	0.0370	0.0371

a)

	50%	100%	150%	200%	250%	300%
Operations	0.6292	0.6281	0.6298	0.6298	0.6296	0.6297
Part	0.6288	0.6327	0.6298	0.6299	0.6295	0.6298
Cell	0.6292	0.6280	0.6297	0.6310	0.6292	0.6298
Period	0.6298	0.6280	0.6299	0.6298	0.6293	0.6299

b)

Fig. 3. a) Derivative change rate of OFV for test model a; b) Derivative change rate of OFV for test model b.

5. Conclusions

In this paper, the dynamic multi-objective CF problem has been considered from a new aspect. Albeit, designing and optimizing of CF problems investigated broadly in literature, sensitivity analysis of CF parameters has been never studied before. This study can help researches on better understanding of inter-relations between dynamic CF parameters, giving some insights on designing CMSs. So first, the performance of meta-heuristic assessed by exemplifying two test models as small- and large-scale problems and then, the results validated through statistical analysis. Analysis of result revealed that the type of objective function can affect indirectly our observations as well as model constraints; however, our findings verified basic rules of designing CMS that is because of the power of MSDE algorithm. Regarding to our analysis, attempts should be made to allocate a certain number of periods and cells on designing CMS which is sequentially depended on elaborating the number of part*machine allocation to each cell. However, applying the proposed meta-heuristic on CF problem based on machine relocation or including setup time can be beneficial and suggested for further research.

References

[1] Dimopoulos C, Mort N. A hierarchical clustering methodology based on genetic programming for the solution of simple cell-formation problems. International Journal of Production Research 2001; 39:1-19.

[2] Selim MH, Askin RG, Vakharia AJ. Cell formation in group technology: review, evaluation and directions for future research. Computers and Engineering 1998; 34:3-20.

[3] Geonwook J, Herman RL. Forming part families by using genetic algorithm and designing machine cells under demand changes. Computers & Operations Research 2006; 33:263-283.

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

[5] Safaei N, Saidi-Mehrabad M, Jabal-Ameli, MS. A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system. European Journal of Operational Research 2008; 185:563-592.

[6] Noktehdan A, Karimi B, Husseinzadeh-Kashan A. A differential evolution algorithm for the manufacturing cell formation problem using group based operators. Expert Systems with Applications 2010; 37(7):4822-4829.

[7] Price KV, Storn RM, Lampinen JA. Differential Evolution: A Practical Approach to Global Optimization. Berlin Heidelberg: Springer-Verlag; 2005.

[8] Chang TT, Chang HC. An efficient approach for reducing harmonic voltage distortion in distribution systems with active power line conditioners. IEEE Transactions on Power Delivery 2000; 15(3):990-995.

[9] Booty WG, Lam DCL, Wong IWS, Siconolfi P. Design and implementation of an environmental decision support system. Environmental Modeling and Software 2000; 16(5):453-458.

[10] Cheng SL, Hwang C. Optimal approximation of linear systems by a differential evolution algorithm. IEEE Transactions on Systems, Man and Cybernetics 2001; 31(6):698-707.

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

[12] Rahnamayan S, Tizhoosh HR, Salama MMA. Opposition-based differential evolution algorithms. In: Proc. IEEE congress of evolution computation 2010-2017; 2006.

[13] LampinenJA, Zelinka I. On stagnation of the differential evolution algorithm, in: Pavel Osmera, (ed.) Proc. of MENDEL 2000, 6th International Mendel Conference on Soft Computing 76-83, Brno, Czech Republic; 2000.

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

[15] Tasgetiren MF, Suganthan PN, Pan QK. An ensemble of discrete differential evolution algorithms for solving the generalized traveling salesman problem. Applied Mathematics and Computation 2010; 215:3356-3368.

[16] Das S, Abraham A, Konar A. Adaptive clustering using improved differential evolution algorithm. IEEE Transactions on Systems, Man and Cybernetics - Part A, IEEE Press, USA 2008; 38(1):218-237.

[17] Ali MM, Torn A. Population set-based global optimization algorithms: Some modifications and numerical studies. Computers & Operation Research 2004; 31(10):1703-25.

[18] Babu BV, Jehan MML. Modified differential evolution (MDE) for optimization of non-linear chemical process. Computer Chemistry Engineering 2006; 30:989-1002.

[19] Brest J, Greiner S, Boskovic B, Mernik M, Zumer V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Trans Evolution Computation 2006; 10(6):646-57.