Design Optimization of MEMS Using Constrained Multi-Objective Evolutionary Algorithm

Wenji Li
Department of Electronic Engineering
Shantou University, Guangdong, P.R. China
wenji_li@126.com

Zhun Fan
Department of Electronic Engineering
Shantou University, Guangdong, P.R. China
zfan@stu.edu.cn

Xinye Cai
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Jiangsu, P.R. China
xinye@nuaa.edu.cn

Huibiao Lin
Department of Electronic Engineering
Shantou University, Guangdong, P.R. China
13hblin@stu.edu.cn

Shuxiang Xie
Department of Electronic Engineering
Shantou University, Guangdong, P.R. China
12sxxie1@stu.edu.cn

Sheng Wang
Department of Electronic Engineering
Shantou University, Guangdong, P.R. China
12cwang2@stu.edu.cn

ABSTRACT
MEMS layout optimization is a typical multi-objective constrained optimization problem. This paper proposes an improved MOEA called cMOEA/D to solve this problem. The cMOEA/D is based on MOEA/D but also uses the frequency of individual update of sub-problems to locate the promising sub-problems. By dynamically allocating computing resources to more promising sub-problems, we can effectively improve the performance of the algorithm to find more non-dominated solutions in MEMS layout optimization. In addition, we compared two mechanisms of constraint handling, Stochastic Ranking (SR) and Constraint-domination principle (CDP). The experimental results show that CDP works better than SR and the proposed algorithm outperforms the state-of-art algorithms such as NSGA-II and MOEA/D, in terms of convergence and diversity.

Categories and Subject Descriptors
G.1.6 [**Optimization**]: Constrained optimization

General Terms
Algorithms

Keywords
Multi-objective constrained optimization; MEMS layout design

1. INTRODUCTION
Multi-objective evolutionary algorithms have been successfully applied in a large variety of optimization problems in both science and engineering, where multiple and conflicting design objectives exist [1] [2]. Without loss of generality, a multi-objective constrained optimization problem with n design variables, and m design objectives can be defined as follows:

$$\begin{cases} \min \quad F(x) = (f_1(x), f_2(x), \dots, f_m(x)); \\ \quad \text{s.t.} \quad g_i(x) \geq 0, \quad i = 1, 2, \dots, q \\ \qquad\quad h_j(x) = 0, \quad j = 1, 2, \dots p \end{cases} \tag{1}$$

Where $x = (x_1, x_2, \dots, x_n) \subset R^n$ is n-dimensional design variables, $F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \subset R^m$ is m-dimensional objective vector. $g_i(x) \geq 0$ define q inequality constraints, $h_j(x) = 0$ defines p equality constraints.

NSGA-II [3] and MOEA/D [4] are the two typical multi-objective evolutionary algorithms. They represent the two categories of fitness assignment methods, namely fitness assignment based on domination relationship, and fitness assignment using aggregation function based on decomposition. In fitness assignment based on domination relationship, the fitness is decided by non-dominated sorting and crowding distance. Representative algorithms using this type of fitness assignment method include MOGA [5], PAES-II [6], SPEA-II [7] and NSGA-II [3]. In fitness assignment based on decomposition, comparison and sorting of individuals are made via aggregation function with weights allocated specifically to all individuals. Different weight vectors associated with the aggregation function lead to different directions towards the Pareto front. To obtain as many solutions as possible in the entire Pareto front, the weight vector may be adjusted during the evolutionary search process. Typical algorithms of this category include IMMOGLS [8], UGA [9], cMOGA [10], MOGLS [11], and MOEA/D [4].

In multi-objective constrained optimization, traditional constraint handling method is adopting penalty function to penalize the constraint violation. However, it is then necessary to find proper penalty factor to balance objective function and penalty function, which is usually difficult and application-dependent in practice. Stochastic Ranking (SR) [12] and Constraint-domination principle (CDP) [13] are two very promising penalty function methods that can well balance the objective function and penalty function without using a penalty factor. Because both SR and CDP do not

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.

Copyright © 2014 ACM 978-1-4503-2881-4/14/07…$15.00.

http://dx.doi.org/10.1145/2598394.2610010

use penalty factor, so there is no need for the user to have prior knowledge about the relative importance of the objective optimization and constraint satisfaction. In SR, usually a small proportion of infeasible solutions are compared with their objective values, while in CDP, all solutions are compared only by their constraint violation levels.

The remainder of the paper is organized as follows: Section 2 explains the framework of the proposed algorithm, Section 3 introduces the MEMS layout optimization problem, Section 4 gives the experimental results of the MEMS layout optimization problem, and Section 5 concludes the paper.

2. THE ALGORITHM

Like other variants of MOEA/D [4], the algorithm proposed in this paper, namely cMOEA/D decomposes a multi-objective constrained optimization problem into N single objective constrained optimization sub-problems. In addition, the cMOEA/D identifies the promising sub-problems using the statistic information of frequency of updating their neighboring solutions, and allocates more computational resources to the promising sub-problems accordingly. In particular, the probability of the sub-problems to be selected is calculated as follows:

$$p_{i,G} = \frac{N_{i,G}}{\sum_{i=1}^{n} N_{i,G}} \tag{2}$$

$$N_{i,G} = \frac{\sum_{g=G-SG}^{G-1} n_{i,g}}{\sum_{g=G-SG}^{G-1} total_g} + \varepsilon \tag{3}$$

Where $n_{i,g}$ represents the number of solutions the i-th sub-problem replaces its neighbors in the g-th generation, $total_g$ represents the total number of solutions that all the sub-problems of the g-th generation replace their neighbors, G is the current generation, and SG is the number of generations counted in the calculation of the statistics. ε is a small constant used to avoid zero in the denominator. In this paper, ε is set to 0.002. In order to ensure the integrity of cMOEA/D, we give all the pseudo-code which may have a lot in common with MOEA/D.

At each generation, cMOEA/D uses a decomposition method named Tchebycheff to maintain:

1. a population of N points $x^1,\ldots x^N$,where x^i is the current solution to the ith subproblem;

2. $F^1,\ldots,F^N$ and $V^1,\ldots,V^N$, where F^i is the F-value of x^i , V^i is the degree of constraint violation of individual x^i .

3. $Z=(z_1,\ldots,z_m)$, where z_i is the best value found so far for objective f_i .

4. an external population (EP) , which is used to store feasible non-dominated solutions found during the search.

5. a matrix $n_{i,g}$, where $n_{i,g}$ is used as a container to store the number of subproblem updated by each subproblem in every generation.

The pseudo-code of CMOEA-D is given in Algorithm 1:

<u>Algorithm1. Pseudo-code of cMOEA/D</u>

generate a weight vector λ^i , $i=1,2,\ldots,N$ uniformly
generate an initial population $P=\{x^i, i=1,\ldots,N\}$
Calculate the T closest weight vectors for each weight vector, set $B(i)=\{\lambda^{i_1},\ldots\lambda^{i_T}\}$ where $\lambda^{i_1},\ldots\lambda^{i_T}$ are the T closest weight vectors to the weight vector λ^i
Evaluate x^i , $F(x^i)=(f_1(x^i),f_2(x^i),\ldots,f_m(x^i))$, $V(x^i)=\sum_{j=1}^{p}\left|\min(g_j(x^i),0)\right|$, $i=1,2,\ldots,N$. $V(x^i)$ denotes the degree of constraint violation of individual x^i .
Initialize $Z=(z_1,\ldots,z_m)$ by setting $z_i=\min\{f_i(x^1),f_i(x^2),\ldots,f_i(x^N)\}$;
Set gen = 1;
while stopping criterion is not satisfied **do**
  for i = 1 to N do
    select subproblem according to (2), generate an offspring y according to Algorithm2
    Evaluate y, $F(y)=(f_1(y),f_2(y),\ldots,f_m(y))$, $V(y)=\sum_{j=1}^{p}\left|\min(g_j(y),0)\right|$;
    update Z , for each $j=1,\ldots,m$, if $z_i > f_i(y)$, then set $z_i = f_i(y)$
    update neighboring solutions according to Algorithm3;
    set $n_{i,g}$ according to Algorithm3
  end for
end while

<u>Algorithm 2. Pseudo-code of generate offspring</u>

randomly generate a number named rand from (0, 1);
if rand < δ
  $P=B(i)$
else
  $P=\{1,\ldots N\}$;
end if
Set $r_1=i$ and randomly select two indexes r_2 and r_3 from P , and then generate an offspring Solution y' from x^{r_1} , x^{r_2} and x^{r_3} by a DE operator;
Perform a mutation operator on y' with probability p_m to produce a new solution y
Repair y ;

Algorithm 3. Pseudo-code of update scheme.

1: update neighboring solutions x^j , $j \in P$ of an offspring y as follows:

2: Set $c = 0$

3: **if** $c = n_r$ or P is empty then

4: set $n_{i,g} = C$ then break;

5: **else**

6: Randomly pick an index j from P;

7: randomly generate a number named r from (0, 1)

8: **if** $(V(y) = V(x^j) = 0)\text{or}(r < p_f)$ **then**

9: **if** $g^{te}(y \mid \lambda^j, Z) \le g^{te}(x^j \mid \lambda^j, Z)$ **then**

10: $x^j = y, F(x^j) = F(y), V(x^j) = V(y) \text{and} c = c + 1$

11: **end if**

12: **else**

13: **if** $V(y) < V(x^j)$ **then**

14: $x^j = y, F(x^j) = F(y), V(x^j) = V(y) \text{and} c = c + 1$

15: **end if**

16: **end if**

17: Remove j from P and go to step3;

18: **end if**

3. MEMS MODEL

3.1 Introduction to the MEMS Model

A case study in the area of MEMS design (taken from [14] [15] [16] [17]) was carried out to verify the effectiveness of the above design optimization methodology following a MOEA/D computational approach. The design problem is a comb-drive micro-resonator (see the layout in Fig. 1), with fourteen mixed-type design variables ($L_b, w_b, L_t, w_t, L_{sy}, w_{sy}, L_{sa}, w_{sa}, w_{cy}, L_c, w_c, x_0, V, N_c$), and twenty four design constraints, both linear and nonlinear. More detailed description of the design problem in terms of analytical equations is given below.

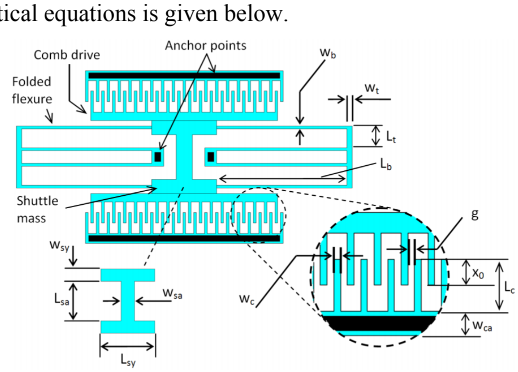

Figure 1. MEMS model (adapted from [14]).

L_b and w_b are the length and the width of the flexure beam (respectively), L_t and w_t are the length and the width of the truss beam (respectively), L_{sy} and w_{sy} are the length and the width of the shuttle yoke (respectively), L_{sa} and w_{sa} are the length and the width of the shuttle axle (respectively), w_{cy} is the width of the comb yoke, L_c and w_c are the length and the width of the comb fingers (respectively), x_0 is the comb finger overlap, V is the voltage amplitude, and N_c is the number of rotor comb fingers. The equations which are necessary to build the parametric layout are given as follows, Eqns. 4-18,

$$L_1 = (N_c - 1)(w_c + 2g) + N_c.w_c \quad (4)$$

$$L_2 = L_1 + 2(g + w_c) \quad (5)$$

$$A_s = w_{sa}.L_{sa} + 2w_{sy}.L_{sy} \quad (6)$$

$$A_b = 4(2L_b.w_b + (L_t + w_a / 2).w_t + (w_a.w_{ba}) / 2) \quad (7)$$

$$A_c = 2(L_1.w_{cy} + N_c.w_c.L_c) \quad (8)$$

$$A_t = 2(w_{ca}.L_2 + (N_c + 1).w_c.L_c) \quad (9)$$

$$B_x = \mu((A_s + 0.5A_t + 0.5A_b).(1/d + 1/\gamma) + (A_c / g)) \quad (10)$$

$$\alpha = (w_t / w_b)^3 \quad (11)$$

$$K_x = 2(E.t.w_b^3 / L_b^3).(L_t^2 + 14\alpha.L_t.L_b + 36\alpha^2 L_b^2) / (4L_t^2 + 41\alpha.L_t.L_b + 36\alpha^2 L_b^2) \quad (12)$$

$$K_y = 2(E.t.w_b^3 / L_b^3).(8L_t^2 + 8\alpha.L_t.L_b + \alpha^2 L_b^2) / (4L_t^2 + 10\alpha.L_t.L_b + 5\alpha^2 L_b^2) \quad (13)$$

$$K_{ey} = 2\varepsilon_0.N_c.V^2.x_0.t / g^3 \quad (14)$$

$$m_x = \rho(A_s + 0.25A_t + (12/35)A_b).t \quad (15)$$

$$Q = (m_x.K_x / B_x^2)^{1/2} \quad (16)$$

$$F_{ex} = 1.12\varepsilon_0.N_c.(t/g).V^2 \quad (17)$$

$$x_{disp} = Q.F_{ex} / K_x \quad (18)$$

where L_1 and L_2 are the lengths of the lower and the upper comb yokes, A_s , A_b , A_c , A_t are the layout areas of the shuttle yokes, the flexure beams, the comb finger sidewalls and the truss beams, respectively, B_x is the damping coefficient, K_x and K_y are the folded flexure spring constants, F_{ex} is the lateral component of the external electrostatic force generated by the comb drives, m_x is the effective masses, Q is the quality factor and x_{disp} is the displacement amplitude. Further details about these analytical equations and derivations of them are given in [16] [17].

3.2 Design Criteria

Several design constrains must be defined to constrain the layout synthesis of the micro resonator. There are 24 linear and nonlinear constraints defined as given in the following Eqns. 19-31.

$$g_{1,2}(x) = 0 < L_2 \le 700(\mu\ m) \quad (19)$$

$$g_{3,4}(x) = 0 < L_{sy} + 2L_b + 2w_t \le 700(\mu\ m) \quad (20)$$

$$g_{5,6}(x) = 0 < 2(w_{ca} + 2L_c - x_0 + w_{cy}) + L_{sa} + 2w_{sy} \le 700(\mu\ m) \quad (21)$$

$$g_{7,8}(x) = 4 < L_c - (x_0 + x_{disp}) \le 200(\mu\ m) \quad (22)$$

$$g_{9,10}(x) = 4 < x_0 - x_{disp} \le 200(\mu\ m) \quad (23)$$

$$g_{11}(x) = (L_t + w_a / 2) \le (L_{sa}/2 + w_{sy} + x_{disp})(\mu\ m) \quad (24)$$

$$g_{12}(x) = (L_{sa}/2 + x_{disp}) \le L_t + w_a / 2 - w_b(\mu\ m) \quad (25)$$

$$g_{13,14}(x) = 2 < L_{sy} / 2 - w_{ba} - w_{sa} / 2 \le 200(\mu\ m) \quad (26)$$

$$g_{15,16}(x) = 2 < x_{disp} \le 100(\mu\ m) \quad (27)$$

$$g_{17,18}(x) = 5 < Q \le 10^5 \quad (28)$$

$$g_{19,20}(x) = 0 < x_{disp} / L_b \le 0.1 \quad (29)$$

$$g_{21,22}(x) = 0 < K_{ey} / K_y \le 1/3 \quad (30)$$

$$g_{23,24}(x) = 4 < L_{sa} / 2 - x_{disp} - w_a / 2 \le 200(\mu\ m) \quad (31)$$

In this section, the multi-objective optimization problem (MOP), briefly described in the previous section that is related to the layout synthesis of MEMS components with respect to dynamic response (i.e. voltage) and the size of the device, is formulated. Optimum design parameters, i.e. geometrical features of the flexure beams, comb drives and the shuttle mass, are investigated to simultaneously minimize the power consumption or in other words the voltage and the area of the problem is given below,

$$\text{Minmize } f_1(\mathrm{x}): V$$
$$\text{Minmize } f_2(\mathrm{x}): A_{total} = (A_s + A_t + A_b + A_c) \tag{32}$$
$$\text{subject to: } g_i(\mathrm{x}), \text{for } i = 1,2,\ldots,24$$
$$\mathrm{x} = \{L_b, w_b, L_t, w_t, L_{sy}, w_{sy}, L_{sa}, w_{sa}, w_{cy}, L_c, w_c, x_0, V, N_c\}$$
$$\mathrm{x}_{\min} = \{2,2,2,2,2,10,2,10,10,8,2,4,0,3\};$$
$$\mathrm{x}_{\max} = \{400,20,400,20,400,400,400,400,400,400,20,400,50,50\}$$

Where $g_i(\mathrm{x})$ are the constraints given in the previous section and x is the vector of design variables. The autonomous optimization methodology to solve this nonlinear constrained optimization problem is given in the following sections.

4. EXPERIMENTAL STUDIES AND DISCUSSIONS

4.1 Experimental Settings

In order to evaluate the performance of CMOEA/D, experimental results on MEMS layout optimization problem are compared with those obtained by two state-of-the-art algorithms NSGA-II and MOEA/D. Ten independent runs with the three algorithms are made, and the following parameters are used:

The parameter settings of the compared algorithms are listed in Table 1 as follows. The population size for NSGA-II, MOEA/D and cMOEA/D is 200, crossover probability is 1.0, probability parameter *pf* = 0, mutation probability is 1/14 and maximum generation is 500.

Table 1. The parameter setting of NSGA-II, MOEA/D and cMOEA/D

	Population size	Crossover rates	Mutation rates	Maximum Generation
NSGA-II	200	1.0	1/14	500
MOEA/D	200	1.0	1/14	500
cMOEA/D	200	1.0	1/14	500

For cMOEA/D and MOEA/D, neighbor size *T*=20， neighbor selection probability $\delta = 0.9$, $n_r = 10$

For cMOEA/D, *SG*=5.

4.2 Performance Metric

Performance of a multi-objective evolutionary algorithm can be evaluated in two aspects – convergence and distribution. Convergence describes the closeness of the obtained Pareto front to the true Pareto front. Distribution on the other hand depicts how the solutions in the obtained Pareto are distributed. For the MEMS layout optimization problem, because we do not know the true Pareto front in advance, we select two metrics – Coverage (C) [18] and Hypervolume (HV) [18]. Detailed definitions of them are as follows:

Coverage Metric

C can manifest domination relationship between two approximate Pareto fronts. Assume that A and B are two Pareto fronts obtained with two different multi-objective evolutionary algorithms, C can be defined as:

$$C(A,B) = \frac{\left|\{u \in B \mid \exists v \in A : v \succeq u\}\right|}{|B|} \tag{33}$$

Here, $|B|$ represents the number of items (solutions) in *B*. $v \succeq u$ means v dominates u or v equals u .Normally $C(A,B) \neq 1 - C(B,A)$. $C(A,B) = 1$ means that all solutions in B are dominated by some solutions from *A*. On the other hand, $C(A,B) = 0$ indicates that no individual in *B* is dominated by any solution from *A*. When $C(A,B) > C(B,A)$, *A* is considered superior to *B*.

Hypervolume（*HV*）Metric

HV simultaneously considers the distribution of the obtained Pareto front *P* and its vicinity to the true Pareto front. *HV* is defined as the volume enclosed by *P* and the reference vector $r = (r_1, r_2, \ldots, r_m)$. *HV* can be defined as:

$$HV(P) = \bigcup_{i \in P} vol(i) \tag{34}$$

Here, $vol(i)$ represents the volume enclosed by solution $i \in P$ and the reference vector r . In this experiment, $r = (55,50)$.

4.3 Experimental Result

SR and CDP are two mechanisms of constraint handling. Moreover, when pf = 0 the two mechanisms are equivalent [19]. Figure 1 shows different pf values in cMOEA/D when solving MEMS layout optimization problem.

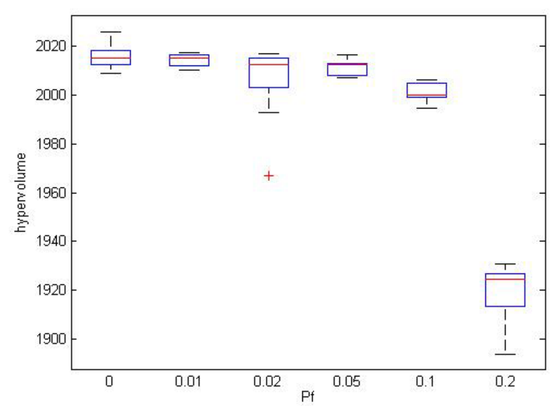

Figure 1. Box-plot of HV-metric using different pf values in 10 independent running of cMOEA/D

Table 2. HV-metric values of the solutions found by cMOEA/D using different pf value.

HV-metric	cMOEA/D	
Parameter Pf	Best	Mean
0	**2025.9**	**2015.8**
0.01	2017.3	2014.3
0.02	2017.1	2006.0
0.05	2016.7	2011.4
0.1	2006.4	2001.0
0.2	1931.0	1918.9

From figure 1 and table 2, we can know that CDP (the Pf value equals zero) works better than SR in cMOEA/D.

In the following experiments, we use CDP to deal with constraints. Figure 2 is the nondominated solutions with the maxmam H-metric in 10 runs of NSGA-II, MOEAD and CMOEAD where their constraint handling is CDP.

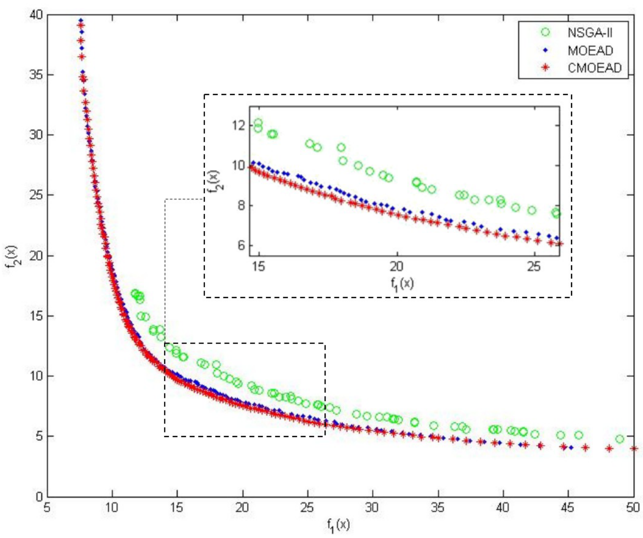

Figure 2. Plots of the nondominated solutions with the maxmam HV-metric in 10 run of NSGA-II, MOEA/D and cMOEA/D

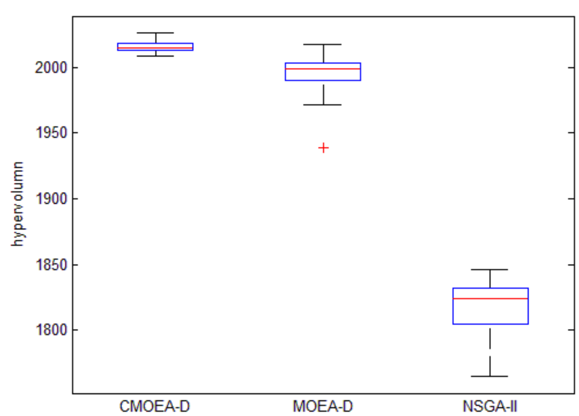

Figure 3. the boxplot about HV-metric in 10 runs of NSGA-II, MOEA/D and cMOEA/D

Table 3. HV-metric values of solutions found by cMOEA/D, MOEA/D and NSGA-II

HV-metric	cMOEA/D		MOEA/D		NSGA-II	
Instance	Best	Mean	Best	Mean	Best	Mean
MEMS	**2025.9**	**2015.8**	2017.2	1992.2	1846.3	1816.4

Figure 3 and table 3 describes the H-metric information in 10 runs of NSGA-II、MOEAD and CMOEAD.

Table 4. Average set coverage among cMOEA/D (A), MOEA/D （B） and NSGA-II（N）

C-metric	**C(A,B)**	**C(B,A)**	**C(A,N)**
	0.82	0.015	1
	C(N,A)	**C(B,N)**	**C(N,B)**
	0	1	0

Table4 presents the best of the C-metric values of the final approximations obtained by the three algorithms. The experimental results show that the proposed algorithm (cMOEA/D) outperforms the state-of-art algorithms such as NSGA-II and MOEA/D, in terms of convergence and diversity.

5. CONCLUTION

This paper proposes an improved MOEA called cMOEA/D to solve MEMS layout optimization problem. The cMOEA/D uses the frequency of individual update of sub-problems to locate the promising sub-problems. By dynamically allocating computing resources to more promising sub-problems, we can effectively improve the performance of the algorithm to find more non-dominated solutions in MEMS layout optimization. In addition, we compared two mechanisms of constraint handling, Stochastic Ranking (SR) and Constraint-domination principle (CDP). The experimental results show that CDP works better than SR and the proposed algorithm outperforms the state-of-art algorithms such as NSGA-II and MOEA/D, in terms of convergence and diversity.

The future work includes combinations of cMOEA/D with other mechanisms of constraint handling to further improve the performance of the algorithm, and test it in new applications.

6. ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China (Grant No. 61175073, 51375287 and 61332002).

7. REFERENCES

[1] Rosenberg B, Richards M, Langton J T, et al. Applications of multi-objective evolutionary algorithms to air operations mission planning[C]//Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation. ACM, 2008: 1879-1886.

[2] Tapia M G C, Coello C A C. Applications of multi-objective evolutionary algorithms in economics and finance: A survey[C]//IEEE congress on evolutionary computation. 2007, 7: 532-539.

[3] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. Evolutionary Computation, IEEE Transactions on, 2002, 6(2): 182-197.

[4] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. Evolutionary Computation, IEEE Transactions on, 2007, 11(6): 712-731.

[5] Fonseca C M, Fleming P J. Genetic Algorithms for Multiobjective Optimization: FormulationDiscussion and Generalization[C]//ICGA. 1993, 93: 416-423.

[6] Corne D W, Jerram N R, Knowles J D, et al. PESA-II: Region-based selection in evolutionary multiobjective optimization[C]//Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001. 2001.

[7] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm[J]. 2001.

[8] Ishibuchi H, Murata T. A multi-objective genetic local search algorithm and its application to flowshop scheduling[J]. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 1998, 28(3): 392-403.

[9] Leung Y W, Wang Y. Multiobjective programming using uniform design and genetic algorithm[J]. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2000, 30(3): 293-304.

[10] Murata T, Ishibuchi H, Gen M. Specification of genetic search directions in cellular multi-objective genetic algorithms[C]//Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, 2001: 82-95.

[11] Jaszkiewicz A. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-A comparative experiment[J]. Evolutionary Computation, IEEE Transactions on, 2002, 6(4): 402-412.

[12] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization[J]. Evolutionary Computation, IEEE Transactions on, 2000, 4(3): 284-294.

[13] Deb K. Multi-objective optimization using evolutionary algorithms[M]. Chichester: John Wiley & Sons, 2001.

[14] Tutum C C, Fan Z. Multi-criteria layout synthesis of mems devices using memetic computing[C]//Evolutionary Computation (CEC), 2011 IEEE Congress on. IEEE, 2011: 902-908.

[15] Fan Z, Liu J, Sorensen T, et al. Improved differential evolution based on stochastic ranking for robust layout synthesis of MEMS components[J]. Industrial Electronics, IEEE Transactions on, 2009, 56(4): 937-948.

[16] Fedder G K, Mukherjee T. Physical design for surface-micromachined MEMS[C]//Proceedings of the Fifth ACM/SIGDA Physical Design Workshop. 1996: 53-60.

[17] Fedder G K, Iyer S, Mukherjee T. Automated optimal synthesis of microresonators[C]//Solid State Sensors and Actuators, 1997. TRANSDUCERS'97 Chicago., 1997 International Conference on. IEEE, 1997, 2: 1109-1112.

[18] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach[J]. Evolutionary Computation, IEEE Transactions on, 1999, 3(4): 257-271.

[19] Jan M A, Khanum R A. A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D[J]. Applied Soft Computing, 2013, 13(1): 128-148.