Multi-objective Optimization with Combination of Particle Swarm and Extremal Optimization for Constrained Engineering Design

CHEN-LONG YU, YONG-ZAI LU and JIAN CHU
Research Institute of Cyber-Systems and Control
Zhejiang University
38, Zhe-Da Road, Hangzhou, 310027
CHINA
clyu@iipc.zju.edu.cn

Abstract: Engineering optimization problems usually have several conflicting objectives, such that no single solution can be considered optimum with respect to all objectives. In recent years, many efforts have focused on hybrid metaheuristic approaches for their robustness and efficiency to solve the above-mentioned multi-objective optimization problems (MOPs). This paper proposes a novel hybrid algorithm with the integration of particle swarm optimization (PSO) and bio-inspired computational intelligence extremal optimization (EO) for constrained engineering design, which combines the superior functionalities of PSO for search efficency and extremal dynamics oriented EO for global search capability. The performance of proposed PSO-EO algorithm is further tested on several benchmark MOPs in comparison with reported results. The simulations show that the PSO-EO is effective in solving MOPs, could result in faster convergence and better spread.

Key-Words: Multi-objective optimization, Evolutionary algorithm, Particle swarm optimization, Extremal optimization, Pareto dominance, Engineering design

1 Introduction

Optimization problems with two or more objectives are very common in engineering and many other disciplines, such as product and process design, finance, aircraft design, the oil and gas industry, automobile design, or wherever optimal decisions need to be taken in the presence of trade-offs among several conflicting objectives. The process of optimizing a collection of objective functions systematically and simultaneously is called multi-objective optimization. The solution of such problems is difficult due to the large number of conflict objectives and the rough landscape with multiple local minima. The Operations Research community has developed various mathematical programming techniques to solve MOPs since the 1950s. However, there are several limitations for traditional mathematical programming techniques when tackling MOPs, for example, many of them failed when the shape of the Pareto front is concave or disconnected. Also, for most of them, only one solution can be detected per optimization run [1]. The inherent difficulty and the heavy computational cost of mathematical programming techniques promote the development of more efficient and effective methods.

Evolutionary algorithms (EAs) are suitable for MOPs due to the capability of searching for multiple Pareto optimal solutions synchronously and performing better global exploration of the search space [2-3]. Furthermore, EAs can be easily extended to maintain a diverse set of solutions with the help of population mechanism [4], and are less susceptible to the shape or continuity of the Pareto front. During the past two decades, a considerable amount of multi-objective evolutionary algorithms have been presented to solve various types of MOPs [5-14]. However, evolutionary algorithms have their weakness in slow convergence and providing a precise enough solution because of the failure to exploit local information. During the last decades, a particular class of global-local search hybrids named “memetic algorithms” (MAs) are proposed. MAs are a class of stochastic heuristics for global optimization which combine the global search nature of EA with local refinement to improve individual solution. The motivation behind hybridization concept is usually to obtain better performing systems that exploit and unite advantages of the individual pure strategies, i.e., such hybrids are believed to benefit from synergy.

Under the conceptual umbrella of MA, this paper developed a novel hybrid multi-objective optimization algorithm with the integration of the popular particle swarm optimization (PSO) and recently proposed extremal optimization (EO), called “PSO-EO”. The hybrid algorithm can combine the capability of PSO in search efficiency with the advanced feature of EO in global search, and complement their individual weak points, thus outperform either one used alone. The effectiveness of the proposed PSO-EO algorithm is tested on five engineering design MOPs and three constrained benchmarks, and the comparison with some published results shows that the proposed approach is highly competitive in convergence and spread. That is precisely the aim of the study.

The rest of the paper is organized as follows: The general problem formulation for MOPs is described in Section 2. Then, the fundamental and algorithms of PSO and EO are introduced briefly, and the hybrid PSO-EO oriented multi-objective optimization solution is presented in Section 3, and the simulation studies of proposed PSO-EO in engineering design are illustrated in Section 4. Finally, the concluding remarks are addressed in Section 5.

2 Problem Formulation

A multi-objective optimization problem with minimization as an example can be generally defined as follows [15]:

Find the decision vector $\vec{x}^* = [\vec{x}_1^*, \vec{x}_2^*, \cdots, \vec{x}_n^*]^T$, which satisfies:

$$\begin{aligned} Minimize\ \ & \vec{f}(\vec{x}) = [\vec{f}_1(\vec{x}), \vec{f}_2(\vec{x}), \cdots, \vec{f}_k(\vec{x})]^T \\ s.t.\ \ & g_i(\vec{x}) \ge a_i \quad i = 1, 2, \cdots, q \ (inequality\ constrains) \\ & h_j(\vec{x}) = b_j \quad j = 1, 2, \cdots, r \ (equality\ constrains) \end{aligned}$$

For the above mentioned MOPs, there is rarely a single solution that simultaneously optimizes all the objective functions. People usually look for "trade-offs", rather than a single solution when dealing with MOPs. The notion of "optimality" is therefore, different. The most commonly adopted notion of optimality is called "Pareto optimality", some related concepts can then be defined:

Definition 1 *Pareto Optimality*: A solution $\vec{x}^* \in \Omega$ is called "Pareto optimal point" if and only if for all $\vec{x} \in \Omega$ and $I = \{1, 2, \cdots, k\}$, either $\forall_{i \in I}(\vec{f}_i(\vec{x}) = \vec{f}_i(\vec{x}^*))$ or, there is at least one $i \in I$ such that $\vec{f}_i(\vec{x}) > \vec{f}_i(\vec{x}^*)$.

Definition 2 *Pareto Dominance*: A vector $\vec{u} = [u_1, \cdots, u_k]$ is said to dominate another vector $\vec{v} = [v_1, \cdots, v_k]$ (denoted by $\vec{u} \preceq \vec{v}$) if and only if $\vec{u}$ is partially less than $\vec{v}$, i.e. $\{\forall i \in \{1, \cdots, k\}, u_i \le v_i\} \wedge \{\exists i \in \{1, \cdots, k\}, u_i < v_i\}$.

Definition 3 *Pareto-optimal set*: The Pareto optimal set P_s^* is defined as the set of all Pareto optimal solutions, i.e. $P_s^* = \{\vec{x} \in \Omega | \neg\exists \vec{y} \in \Omega : F(\vec{y}) \prec F(\vec{x})\}$.

Definition 4 *Pareto Front*: For a given MOP $\vec{f}(\vec{x})$ and Pareto optimal set P_s^*, the Pareto front P_f^* is defined as $P_f^* := \{\vec{u} = \vec{f}(\vec{x}) = (f_1(x), \cdots f_k(x)) | x \in P_s^*\}$.

3 Hybrid PSO-EO Multi-objective Optimization

In this section, the development of the proposed PSO-EO method is described in detail. First, the basic conception of PSO and EO is briefly introduced. Then the detailed issues of proposed hybrid multi-optimization algorithm including workflow, Selection mechanism of non-dominated sorting, Dynamic External Archive, Diversity Preservation, Constraints handling, etc, are discussed.

3.1 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) proposed by Kennedy and Eberhart is inspired by the social behavior of animals [16], in which the solution is called "particles". Each particle has a position **Pos**=(*Pos*^{1}, *Pos*^{2},…) and a velocity **Vel**=(*Vel*^{1}, *Vel*^{2},…) in the variable space. At each iteration, the velocity and the position are updated by:

$$Vel_{j,gen+1}^i = wVel_{j,gen}^i + c_1R_1(pBest_{j,gen}^i - Pos_{j,gen}^i) + c_2R_2 (gBest_{j,gen} - Pos_{j,gen}^i) \quad (1)$$

$$Pos_{j,gen+1}^i = Pos_{j,gen}^i + Vel_{j,gen+1}^i \quad j = 1, \ldots, ChromLength \quad (2)$$

where w is the inertia weight of the particle, c_1 & c_2 are two positive constants, R_1 & R_2 are random values in the range *[0,1]*. i is the index of a particle, and *gen* denotes the generation index, *pBest* and *gBest* are the personal best and the global best of the population, respectively.

3.2 Extremal Optimization

The Extremal Optimization (EO) proposed by Boettcher and Percus [17] is derived from the fundamentals of statistical physics and self-organized criticality (SOC) [18] based on Bak-Sneppen (BS) model which simulates far-from equilibrium dynamics in statistical physics and co-evolution. SOC states that large interactive systems evolve to a state where a change in one single of their elements may lead to avalanches or domino effects that can reach any other element in the system. For an optimization problem with n decision variables, EO proceeds as follows [17]:

1. Initialize a configuration S *at will, set* $S_{best} = S$.
2. For the current solution S .
  (a) Evaluate the fitness for each decision variable x_i .
  (b) Rank all the components by their fitness and find the component with the "worst fitness".
  (c) Choose one solution S' *in the neighborhood of* S *, i.e., such that the worst component* x_j *must change its state.*
  (d) Accept $S = S'$ *unconditionally.*
  (e) If $F(S) < F(S_{best})$ *, set* $S_{best} = S$.
3. Repeat step-2 as long as desired.
4. Return S_{best} *and* $F(S_{best})$.

Generally, EO is particularly applicable in dealing with large complex problems with rough landscape, phase transitions passing "easy-hard-easy" boundaries or multiple local optima. It is less likely to be trapped in local minima than traditional gradient-based search algorithms. Benefited from its generality and ability to explore complicated configuration spaces efficiently, EO and its derivatives have been successfully applied in solving multi-objective combinatorial hard benchmarks and real-world optimization problems.

3.3 Hybrid PSO-EO Multi-objective Optimization

As mentioned above, mathematical programming techniques often fail in solving complex MOPs. On the contrary, many evolutionary-based optimization methods are good at global search, but relatively poor in fine-tuned local search. According to so-called "No-Free-Lunch" Theorem, the performance of a search algorithm strongly depends on the quantity and quality of the problem knowledge it incorporates. This fact clearly underpins the exploitation of problem knowledge intrinsic to the hybrid metaheuristics. Under the framework of MAs, the global character of the search is given by the evolutionary nature of computational intelligence approaches while the local search is usually performed by means of constructive methods, intelligent local search heuristics or other search techniques.

Moreover, since the natural link between hard optimization and statistical physics, the dynamic properties and computational complexity of the optimization have been attractive fundamental research topics in physics society within the past two decades. It has been recognized that one of the real complexities in optimization comes from the phase transition, e.g., "easy-hard-easy" search path. Phase transitions are found in many combinatorial optimization problems, and have been observed in the region of continuous parameter space containing the hardest instances. Unlike the Equilibrium approaches such as simulated annealing (SA), EO as a general-purpose method inspired by non-equilibrium physical processes shows no sign of diminished performance near the critical point, which is deemed to be the origin of the hardest instances in terms of computational complexity. This opens a new door for development of high performance hybrid multi-objective optimization algorithm with the integration of PSO and EO. The proposed PSO-EO in this paper relies on the capability of PSO in search efficiency with the advanced feature of EO in global search. Fig.1 shows the flowchart of the proposed algorithm. At first, the positions and velocities of all particles in the generation are randomly initialized, and the local best of each particle is set as the current position of itself. The archive is set to a null set. Then the evaluation of particle position includes the calculations of positions and the pair-wise comparisons of all particles to get the relationship of dominating. The algorithm terminates until the stopping conditions are satisfied.

We will illustrate the fundamental and innovation of our method from four aspects: selection mechanism of non-dominated sorting, dynamic external archive, diversity preservation, constraints handling.

3.3.1 Selection Mechanism of Non-dominated Sorting

As known, obtaining a set of non-dominated solutions as closely as possible to the real Pareto front (P_f) and maintaining a well-distributed-solution set along P_f are the two key principles in solving MOPs. To be efficient, we employ the non-dominated sorting approach proposed by Fonseca and Fleming [19]. The approach selects the solutions in the better fronts, hence providing the necessary selection pressure to push the population towards P_f. All new positions of particles, which generated at each iteration, will be evaluated whether they dominate the current solutions by comparing their fitness values.

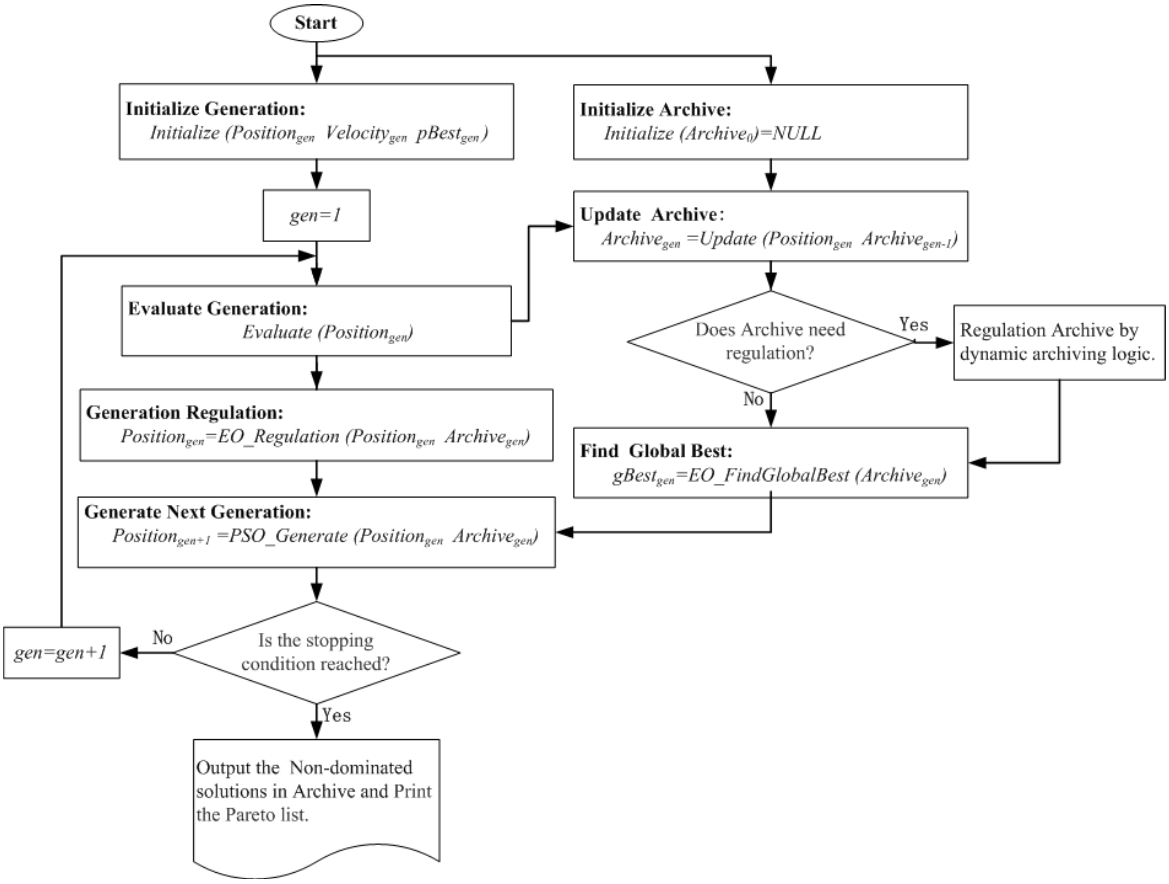


Fig.1, The pseudo-code of PSO-EO algorithm

3.3.2 Dynamic External Archive

To preserve good non-dominated solutions in the search process, a novel dynamic external archive (A^*) is introduced. Characterized by dynamic, A^* discriminates against the density estimation in NSGA-II [6] or the external archive in MOEO [20]. A^*, which provides the elitist mechanism for PSO-EO, consists of two main components as follows:

In current population, if there are N particles dominated by others. Select the ones which are far away from the current archive Pareto front, and then evolve the selected particles by *EO*. That is, for each of them (DP^*):

If *Random_number* ≤ *EO_Probability*

1) Select a Non-dominated solution NS^* from the archive by EO, according to the crowding metric in the archive, the lesser crowded region the NS^* locates, the more likely to be selected.
2) Generate a mutation on one gene of NS^* randomly to create a new particle (NP^*)..
3) Let the position of DP^* equal to NP^*, while maintain the velocity of DP^* unchanged.

End

Fig.2, Flow chart of EO Regulation

Dynamic archiving logic: The logic is used to determine whether the newly founded solutions in the search process should be added to A^*, and it works as follows:

1) If some solutions of A^* are dominated by S^*, all these dominated solutions are eliminated from A^* and S^* is added to A^*.
2) If there is at least one solution of A^* dominates S^*, A^* does not need to be updated.
3) If S^* and any solution of A^* do not dominate each other, S^* is added to A^*.

Regulation and Global Best selection: Accelerating the searching procedure is the main purpose of the regulation and selection. Here, a crowding distance metric was employed to judge whether the current solution locates in a lesser crowded region of the archive, as shown in Fig.2.

3.3.3 Diversity Preservation

For multi-objective optimization, there is strong desire to maintain a good spread of solutions besides convergence to the real Pareto front. In this paper, an adaptive lattice method is proposed for diversity preservation and well-distributed solutions of the archive.

The workflow of the adaptive lattice is shown in Fig.3. At the beginning of each iteration, the PSO-

EO will check whether the archive needs to update, as mentioned before. If yes, the adaptive lattice takes effect; if no, go to step (1). For each objective dimension the PSO-EO will find the so-called "extreme solutions" with the maximum value of each objective dimensions (for instance, the circle points on the position *1* and *6* in Fig.3). Then generate a virtual Pareto front with several virtual points distributed uniformly (the diamond points on the position *1-6*). Finally, if there is a solution with the shortest distance to the virtual Pareto front, add the solution into the archive. And then goes to the next iteration.

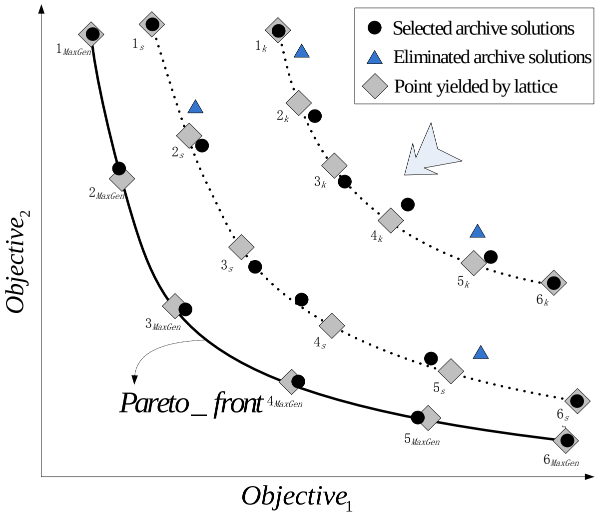


Fig.3, Flow chart of the adaptive lattice method for diversity preservation. Generation : $k < s < MaxGen$

3.3.4 Constraints handling

A simple scheme is applied to handle constraints [15]. For the comparison of two solutions, the PSO-EO will check both their objective function values and their constraints. There are three cases:

1) If both solutions are feasible, choose the one with better objective function value.
2) If one solution is feasible and the other is infeasible, choose the feasible one.
3) If both solutions are infeasible, choose the one with smaller overall constraint violations.

4 Applications of Hybrid PSO-EO-MO for Engineering Design

In this section, the proposed PSO-EO is tested on five nonlinear engineering design problems and three benchmarks, which can be classified into the following categories:

1) **Group 1:** including four bar truss design ("Four Bar") [21], two bar truss design ("Two Bar") and welded beam design ("Welded Beam") proposed by Deb et al. [22] shown in Table 1. The three engineering problems in Group 1 all have continuous variables, two objectives and connected Pareto fronts.
2) **Group 2:** including machine tool spindle design (Tool Spindle) [23], I-beam design ("I-Beam") [24], as shown in Table 2.
3) **Group 3:** including three benchmark test functions, i.e., TNK reported in [6], Hole [25], and WATER [26], as shown in Table 3.

4.1 Performance Measures

The performance index 'Front Spread' (*FS*) proposed by Bosman and Thierens is used to evaluate the performance of our approach. It indicates the size that covered by the non-dominated solutions set (S) in the objective space. The *FS* is defined as the maximum Euclidean distance inside the smallest m-dimensional bounding-box that contains S [12]. It can be calculated as follows:

$$FS(S) = \sqrt{\sum_{i=0}^{m-1} \max_{(z^0, z^1) \in S \times S} (f_i(z^0) - f_i(z^1))^2}$$

4.2 Experimental Settings

The PSO-EO parameters on the test benchmarks are initially set up as follows:

1) Population size: 50 candidate solutions and an external archive of size 100 are employed by PSO-EO to deal with all problems except WATER, which uses a population of size 80 at each iteration and an external archive of size 200.
2) Group 1: PSO-EO is applied to solve these problems and the simulation results are compared with MOEO, NSGA-II, SPEA2, PAES under the same conditions in Chen and Lu [20]. All approaches are run for a maximum of 40000, 30000, 40000 fitness function evaluations (FFE) on Four Bar, Two Bar and Welded Beam respectively, and have 50 independent runs for each. It might also be noted that, PSO-EO only takes a maximum of 15000 FFE on Four Bar.
3) Group 2 & Group 3: For PSO-EO, 20000 FFE is adopted to all problems, and the same conditions as in Baykasoglu [27] are given on I-Beam and Tool Spindle, and the same conditions as in Liu [26] are given on WATER.

4.3 Experimental Results and Discussion

Group 1:

Table 4 shows the experimental results comparisons of MOEO, NSGA-II, SPEA2, PAES and the proposed PSO-EO.

As presented in Table 4, PSO-EO is able to find a wider distributed set of non-dominated solutions than other algorithms on problem Four Bar and Welded Beam, while a little inferior to SPEA2 on problem Two Bar. In all the three problems with PSO-EO, the standard deviation of *FS* metric in 50 runs is also small, especially on Four Bar with less FFE, but a little worse than SPEA2 and NSGA-II on problem Two Bar and Welded Beam.

Group 2:

The published results of MOEO, NSGA-II, SPEA2, PAES and the proposed PSO-EO on problems in group 2 are listed in Table 5. The extreme solutions in two dimensions and *FS* are employed as the performance index to evaluate the above-mentioned algorithms. From the comparisons, we can see that the PSO-EO performs better for the first testing problem ("Tool Spindle" with discrete decision variables), both in terms of extreme solutions and *FS*. As for the second problem ("I-Beam"), the PSO-EO derives the best extreme solutions in terms of f_2 , while the other indexes (extreme solutions in terms of f_1 and *FS*) are a little inferior to GA, but still much better than other algorithms (MOTS, GA (Binary), Monte Carlo). Based on the comparisons, we can find the proposed PSO-EO is highly competitive to the state-of-the-art Methods on engineering design problems, especially for those with discrete variables and multiple disconnected Pareto fronts.

The non-dominated solutions found by the proposed PSO-EO ("red pluses") and the domains of existing feasible designs ("blue shaded area") are shown in Fig.4. It is obvious that PSO-EO performs well in convergence to the blue area and in spread of non-dominated solutions on both problems, especially for Tool Spindle, which has multiple disconnected Pareto fronts.

Group 3:

In group 3, we used three typical benchmark problems, namely TNK, Hole and WATER, to further test the proposed hybrid PSO-EO. Both the TNK and Hole have real domains of feasible designs; while the TNK is non-convex, which makes the problem hard to be solved; for the problem Hole, we select the most tough case, in which the parameter *h* is set to 6 [25]. The simulation results of proposed PSO-EO and published methods on TNK and Hole are shown in Fig.5.

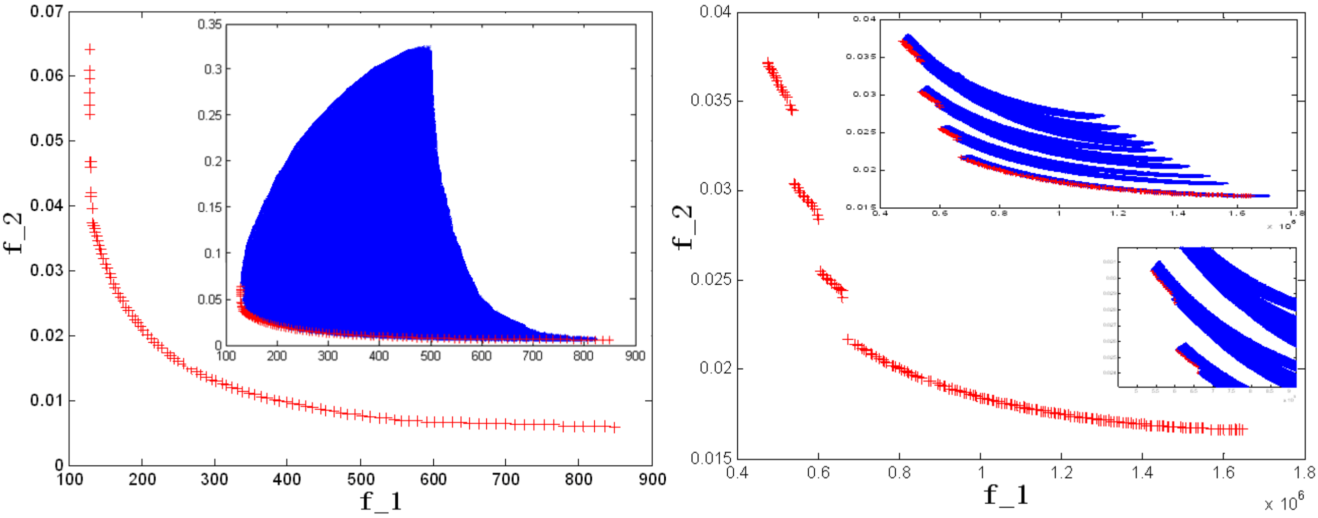
The PSO-EO results in an excellent set of non-dominated solutions for both problems in Group 3, as shown in Fig.5. It can be observed that PSO-EO performs well both in convergence and spread of solutions. This encourages the application of PSO-EO to more complex MOPs which are pretty hard, disconnected, non-convex in real world.

Table 4 Comparisons of *FS* metric in Group 1 (boldface is the best)

Algorithm	Four Bar		Two Bar		Welded Beam	
	Mean	*St.Dev.*	*Mean*	*St.Dev.*	*Mean*	*St.Dev.*
PSO-EO	**1648.52**	**0**	91044.19	429.41	**38.88**	2.84
MOEO [20]	1559.24	24.27	84758.66	4023.16	37.45	3.22
NSGA-II [20]	1648.45	0.067	91515.14	104.13	33.62	1.89
SPEA2 [20]	**1648.52**	7.4E-5	**91548.70**	**84.46**	33.81	**1.52**
PAES [20]	1647.58	4.23	88045.80	3699.86	31.34	4.89

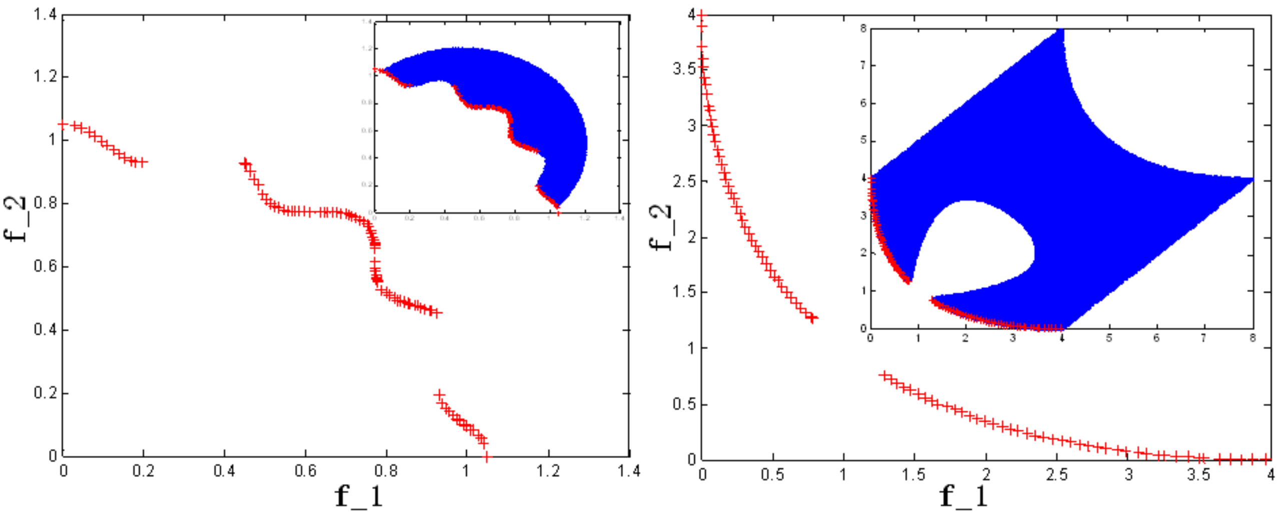
Table 5 Comparisons of *FS* metric in Group 2 (boldface is the best)

Algorithm	I-Beam	Tool Spindle
	min $f_1(X)$ min $f_2(X)$ *FS*	min $f_1(X)$ min $f_2(X)$ *FS*
PSO-EO	(127.71, 0.06424)--(**850.00, 0.00590**)--722.29	(**474653.67**, **0.037186**)--(**1646089.55**, **0.016613**)--**1171435.88**
MOTS [27]	(143.52, 0.03700)--(678.21, 0.00664)--534.69	(497644.10, 0.037839)--(1485169.00, 0.016894) --987524.90
GA (FP) [27]	(**127.46, 0.06034**)--(**850.00, 0.00590**)--**722.54**	(1124409.37, 0.017951)--(1637052.38, 0.016615)--512643.01
GA (Binary) [27]	(128.27, 0.05241)--(848.41, 0.00591)--720.14	(494015.44, 0.038087)--(1643777.68, 0.016613)--1149762.24
Monte Carlo[27]	(188.65, 0.06175)--(555.22, 0.00849)--366.57	(606765.47, 0.032463)--(1457748.36, 0.019242)--850982.89
Literature[27]	(128.47, 0.06000)--(850.00, 0.00590)--721.53	(531183.70, 0.030215)--(694200.03, 0.023101)--163016.33



(a) **(b)**

Fig.4, Non-dominated solutions on (a) I-Beam and (b) Tool Spindle with PSO-EO



(a) **(b)**

Fig.5, Non-dominated solutions on (a) TNK and (b) Hole with PSO-EO

The WATER problem is chosen as a high dimensional benchmark with five objective functions, the comparison of the proposed PSO-EO and published results are listed in Table 6, with the range of the solutions in the archive.

It is easy to find that PSO-EO can obtain broader boundary in terms of most objective functions. In other words, PSO-EO can explore wider searching region.

In Fig.6, the non-dominated solutions evolved by PSO-EO are compared with that of DMOEA [26] under the same conditions (Upper diagonal plots are for PSO-EO and lower diagonal plots are for DMOEA. The ranges of all the objectives are shown in the diagonal boxes). The value of each objective function can be obtained by checking the corresponding diagonal boxes and their ranges. It can be observed that the solutions evolved by PSO-EO generate a larger number of non-dominated points along the frontier, which means, a better convergence and spread of solutions.

Table 6 Min and max value of non-dominated solutions of Water (boldface is the best)

Algorithm	**f_1**	**f_2**	**f_3**	**f_4**	**f_5**
PSO-EO	**0.798~0.956**	**0.027~0.900**	0.095~**0.951**	**0.029**~1.531	**9.028E-04~3.125**
DMOEA [26]	0.798~0.918	0.028~0.900	0.095~0.951	0.031~1.036	9.028E-04~3.124
NSGA-II [26]	0.798~0.920	0.027~0.900	0.095~0.951	0.031~1.110	0.001~3.124
Ray-Tai-Seow [26]	0.810~**0.956**	0.046~0.834	0.067~0.934	0.036~**1.561**	0.211~3.116

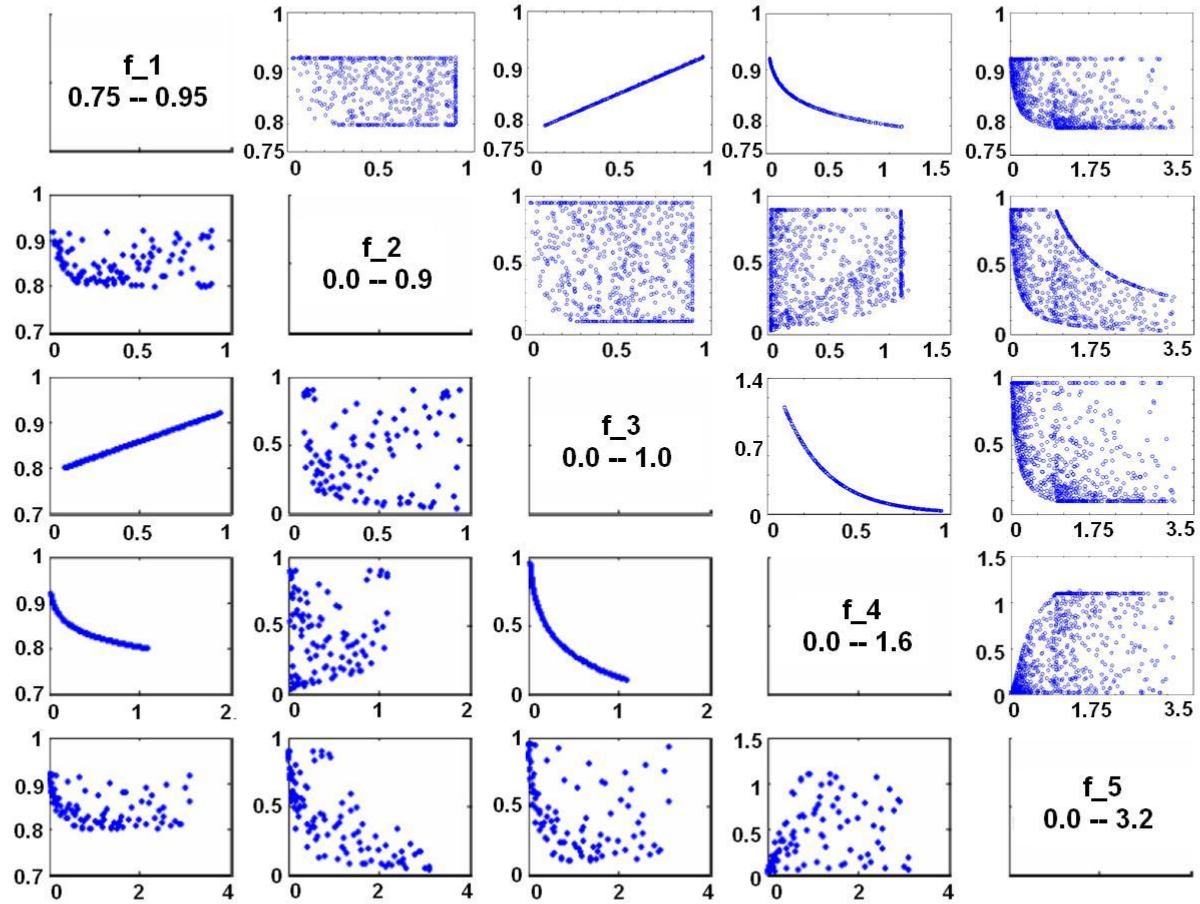


Fig.6, Non-dominated solutions obtained using PSO-EO and DMOEA for WATER

As a general remark on the simulation results and comparisons above, PSO-EO outperforms other state-of-the-art methods in terms of convergence and spread of solutions. It should be noted that the factors contributing to the performance of the proposed PSO-EO method are the capability of PSO in search efficiency with the advanced feature of EO in global search.

5 Conclusions and Future Works

In this paper, a novel Hybrid PSO-EO algorithm is proposed to solve MOPs in engineering design, of which the traditional mathematical programming techniques will fail when the shape of the Pareto front is concave or disconnected. The hybrid method combines the superior functionalities of PSO for search efficency and extremal dynamics oriented EO for global search capability, which results in better convergence and well distributed sets of non-dominated solutions. Those advantages have been clearly demonstrated by the comparison with some state-of-the-art methods over several benchmark problems.

Acknowledgment

The authors would like to thank the financial support from the National Creative Research Groups Science Foundation of China (No. 60721062), and Chen Peng, Chen Min-rong and anonymous reviewers for their helpful remarks.

References:

[1] K. E. Parsopoulos, D. K. Tasoulis and M. N. Vrahatis, Multiobjective optimization using parallel vector evaluated particle swarm optimization, *Proc. of the IASTED International Conference on Artificial Intelligence and Applications*, vol. 2, 2004, pp. 823-828.

[2] C. A. C. Coello, D. A. V. Veldhuizen and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Press, 2002.

[3] K. Deb, Multi-objective genetic algorithms: Problem difficulties and construction of test problems, *Evolutionary Computation*, vol. 7, 1999, pp. 205-230.

[4] D. A. V. Veldhuizen, J. B. Zydallis and G. B. Lamont, Considerations in engineering parallel multiobjective evolutionary algorithms, *IEEE*

Trans. on Evolutionary Computation, vol. 7, 2003, pp. 144-173.

[5] E. Zitzler, M. Laumanns and L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm for multiobjmive Optimization, Technical Report 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.

[6] K. Deb, A. Pratab, S. Agrawal and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. on Evolutionary Computation*, vol. 6, 2000, pp. 182-197.

[7] J. D. Knowles and D. W. Corne,Approximating the Non-dominated Front Using the Pareto Archived Evolution Strategy, *Evolutionary Computation*, vol. 8, 2000, pp. 149-172.

[8] C. A. C. Coello, Evolutionary multiobjective optimization: A historical view of the field, *IEEE Computational Intelligence Magazine*, vol. 1, 2006, pp. 28-36.

[9] J. Balicki, An Adaptive Quantum-based Evolutionary Algorithm for Multiobjective Optimization, *WSEAS Trans. on Systems and Control*, Issue 12, vol. 4, 2009, pp. 603-612.

[10] A. A. Keller, Fuzzy multiobjective optimization modeling with Mathematica, *WSEAS Trans. on Systems*, Issue 3, vol. 8, 2009, pp.368–378.

[11] M. Reyes-Sierra and C. A. C. Coello, Multi-objective particle swarm optimizers: A survey of the state-of-the-art, *International Journal of Computational Intelligence Research*, vol. 2, 2010, pp. 287-308.

[12] M. R. Chen, Y. Z. Lu and G. K. Yang, Multiobjective extremal optimization with applications to engineering design, *Journal of Zhejiang University Science A*, vol. 8, 2007, pp. 1905-1911.

[13] S. Elaoud, T. Loukil and J. Teghem, The Pareto fitness genetic algorithm: Test function study, *European Journal of Operational Research*, vol. 177, 2007, pp. 1703-1719.

[14] C. L. Yu, Y. Z. Lu and J. Chu, An “load forecasting - dispatching” integration system for multiple boilers in thermal power plants, *IEEE International Conference on Power Engineering and Automation Conference*, vol.3, 2011, pp.5-10.

[15] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Trans. on Evolutionary Computation*, vol. 8, 2002, pp. 256-279.

[16] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *IEEE Trans. on Neural Networks*, vol. 4, 1995, pp. 1942-1948.

[17] P. Bak and K. Sneppen, Punctuated Equilibrium and Criticality in a Simple Model of Evolution, *Physical Review Letters*, vol. 71, 1993, pp. 4083-4086.

[18] P. Bak, C. Tang and K. Wiesenfeld, Self-Organized Criticality, *Physical Review Letters*, vol. 59, 1987, pp. 381-384.

[19] C. M. Fonseca and P. J. Fleming, Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization, *Proc. of International Conference on Genetic Algorithms*, 1993, pp. 416-423.

[20] M. R. Chen and Y. Z. Lu, A novel elitist multiobjective optimization algorithm: Multiobjective extremal optimization, *European Journal of Operational Research*, vol. 188, 2008, pp. 637-651.

[21] W. Stadler and J. Dauer, Multicriteria Optimization in Engineering: a Tutorial and Survey, Structural Optimization: Status and Future, *American Institute of Aeronautics and Astronautics*, 1999, pp. 209-249.

[22] K. Deb, A. Patrap and S. Moitra, Mechanical Component Design for Multi-objective Using Elitist Non-dominated Sorting GA, KanGAL Report No. 200002, Indian Institute of Technology Kanpur, India, 2000.

[23] C. A. C. Coello, An empirical study of evolutionary techniques for multiobjective optimization in engineering design, Ph.D. Thesis, Department of Computer Science, Tulane University, New Orleans, LA, 1996.

[24] C. A. C. Coello and A. D. Christiansen, MOSES: a multiple objective optimization tool for engineering design, *Journal of Engineering Optimization*, vol. 31, 1999, pp. 337–368.

[25] E. Rigoni, Hole functions problem, ESTECO Technical Report, 2004, http://www.esteco.com.

[26] M. Z. Liu, X. F. Zou and L. S. Kang, An Effective Dynamical Multi-objective Evolutionary Algorithm for Solving Optimization Problems with High Dimensional Objective Space, *Proc. of International Symposium on Intelligence Computation and Applications*, 2007, pp. 80-89.

[27] A. Baykasoglu, Applying multiple objective Tabu search to continuous optimization problems with a simple neighbourhood strategy, *International Journal for Numerical Methods in Engineering*, vol. 65, 2005, pp. 406-424.

Table 1 Engineering design problems in Group 1.

Problem	*n*	Objective functions	Constraints
Four Bar	**4**	$f_1(X)=L(2x_1+\sqrt{2}x_2+\sqrt{x_3}+x_4)$ $f_2(X)=\frac{FL}{E}\left(\frac{2}{x_1}+\frac{2\sqrt{2}}{x_2}-\frac{2\sqrt{2}}{x_3}+\frac{2}{x_4}\right)$ $F=10kN, E=2\times10^5\,kN/cm^3$ $L=200cm, \sigma=10\,kN/cm^2$ $x_1,x_4\in[\frac{F}{\sigma},\frac{3F}{\sigma}]\quad x_2,x_3\in[\frac{\sqrt{2}F}{\sigma},\frac{3F}{\sigma}]$	
Two Bar	**3**	$f_1(X)=x_1\sqrt{16+y^2}+x_2\sqrt{1+y^2}$ $f_2(X)=\max(\sigma_{AC},\sigma_{BC})$ $\sigma_{AC}=20\sqrt{16+y^2}/yx_1, \sigma_{BC}=80\sqrt{1+y^2}/yx_2$ $x_1,x_2\ge0,\quad 1\le y\le3$	$g_1(X)=\max(\sigma_{AC},\sigma_{BC})\le10^5$
Welded Beam	**4**	$f_1(X)=1.10471h^2\ell+0.04811tb(14.0+\ell)$ $f_2(X)=\delta(X)=2.1952/t^3b$ $\tau'=6000/\sqrt{2}hl$ $\tau''=\frac{6000(14+0.5l)\sqrt{0.25(l^2+(h+t)^2)}}{2\{0.707hl(l^2/12+0.25(h+t)^2)\}}$ $0.125\le h,b\le5.0,\quad 0.1\le\ell,t\le10.0$	$\tau(X)=\sqrt{(\tau')^2+(\tau'')^2+(l\tau'\tau'')/\sqrt{0.25(l^2+(h+t)^2)}}$ $g_1(X)=13600-\tau(X)\ge0$ $g_2(X)=30000-504000/(t^2b)\ge0$ $g_3(X)=b-h\ge0$ $g_4(X)=64746.022(1-0.0282346t)tb^3-6000\ge0$

Table 2 Engineering design problems in Group 2.

Problem	n	Objective functions	Constraints
Tool Spindle	**4**	$f_1(X)=\frac{\pi}{4}\left[a(d_a^2-d_o^2)+l(d_b^2-d_o^2)\right]$ $f_2(X)=\frac{Fa^3}{3EI_a}\left(1+\frac{l}{a}\frac{I_a}{I_b}\right)+\frac{F}{c_a}\left[\left(1+\frac{a}{l}\right)^2+\frac{c_a a^2}{c_b l^2}\right]$ $I_a=0.049(d_a^4-d_o^4),\quad I_b=0.049(d_b^4-d_o^4)$ $c_a=35400\lvert\delta_{ra}\rvert^{\frac{1}{9}}d_a^{\frac{10}{9}},\quad c_b=35400\lvert\delta_{rb}\rvert^{\frac{1}{9}}d_b^{\frac{10}{9}}$ $l_k\le l\le \mathrm{l_g},\quad d_{om}\le d_o,\quad d_a,d_b\in\text{discrete}$ $d_a\in\{80,85,90,95\},\quad d_b\in\{75,80,85,90\}$	$g_1(X)=p_1d_o-d_b\le0$ $g_2(X)=p_2d_b-d_a\le0$ $g_3(X)=\left\lvert\Delta_a+(\Delta_a-\Delta_b)\frac{a}{l}\right\rvert-\Delta\le0$
I-Beam	**4**	$f_1(X)=2x_2x_4+x_3(x_1-2x_4)$ $f_2(X)=\frac{6*10^4}{\sigma(X)}$ $\sigma(X)=x_3(x_1-2x_4)^3+2x_2x_4(4x_4^2+3x_1(x_1-2x_4))$ $10\le x_1\le80,\quad 10\le x_2\le50,\quad 0.9\le x_3,x_4\le5$	$g_1(X)=16-\frac{1.8*10^5x_1}{\sigma(X)}-\frac{1.5*10^4x_2}{(x_1-2x_4)x_3^3+2x_4x_2^3}\ge0$

Table 3 Engineering design problems in Group 3.

Problem	n	Objective functions	Constraints
TNK	**2**	$f_1(X)=x_1$ $f_2(X)=x_2\qquad x_1,x_2\in[0,\pi]$	$g_1(X)=-x_1^2-x_2^2+1+0.1\cos(16\arctan(x_1/x_2))\le0$ $g_2(X)=(x_1-0.5)^2+(x_2-0.5)^2\le0.5$
Hole	**2**	$f_1(X)=(t+1)^2+a+b\bullet e^{-c(t-d)^2}$ $f_2(X)=(t-1)^2+a+b\bullet e^{-c(t+d)^2}$ $a=2p\bullet\sin^{\frac{2}{h}}(\frac{\pi}{2}(-(x+1-\frac{\sqrt{2}}{2})\bullet\sin\frac{\pi}{4}+(y-1+\frac{\sqrt{2}}{2})\bullet\cos\frac{\pi}{4}))$ $c=q/d_0^2,\qquad d=q/2a+d_0$ $\begin{cases}b=(p-a)\bullet e^q & a\le p\\ b=0 & a>p\end{cases}$	$u=\sin(\frac{\pi}{2}((x+1-\frac{\sqrt{2}}{2})\bullet\cos\frac{\pi}{4}+(y-1+\frac{\sqrt{2}}{2})\bullet\sin\frac{\pi}{4}))$ $\begin{cases}t=u^h & u\ge0\\ t=-(-u)^h & u<0\end{cases}$ $h=6,\quad p=2,\quad q=0.2,\quad d_0=0.02$ $x,y\in[-1,1]$
Water	**3**	$f_1(X)=106780.37(x_2+x_3)+61704.67$ $f_2(X)=3000x_1$ $f_3(X)=(305700)2289x_2/(0.06*2289)^{0.65}$ $f_4(X)=(250)2289\exp(-39.75x_2+9.9x_3+2.74)$ $f_5(X)=25(1.39/(x_1x_2)+4940x_3-80)$ $0.01\le x_1\le0.45,\quad 0.01\le x_2,x_3\le0.10$	$g_1(X)=0.00139/(x_1x_2)+4.94x_3-0.08\le1$ $g_2(X)=0.000306/(x_1x_2)+1.082x_3-0.0986\le1$ $g_3(X)=12.307/(x_1x_2)+49408.24x_3+4051.02\le5*10^4$ $g_4(X)=2.098/(x_1x_2)+8046.33x_3-696.71\le1.6*10^4$ $g_5(X)=2.138/(x_1x_2)+7883.39x_3-705.04\le10^4$ $g_6(X)=0.417/(x_1x_2)+1721.26x_3-136.54\le2000$ $g_7(X)=0.164/(x_1x_2)+631.13x_3-54.48\le550$