Exploiting the Marginal Profits of Constraints with Evolutionary Multi-objective Optimization Techniques

Yan Zhenyu
Laboratory of Software Engineering
Wuhan University
Wuhan 430072, P.R. China

Zhi Wei
Department of Computer Science
Rutgers University
Piscataway, NJ 08854, USA

Kang Lishan
Laboratory of Software Engineering
Wuhan University
Wuhan 430072, P.R. China

Abstract

Many real-world search and optimization problems naturally involve constraint handling. Recently, quite a few heuristic methods were proposed to solve the nonlinear constrained optimization problems. However, the constraint-handling approaches in these methods have some drawbacks. In this paper, we gave a Multi-objective optimization problem based (MOP-based) formula for constrained single-objective optimization problems. We proposed a way to solve them by using multi-objective evolutionary algorithms (MOEAs). By simulation experiments, we find this approach for constraint handling not only can find the constrained optimality, but also can provide the decision maker (DM) with a group of trade-off solutions with slightly constraint violation and meanwhile with substantial gain in the objective function. This can enable the DM to have more freedom to choose his preferred solution and therefore exploit more profits in the margin of constraint violations, where the constraint violations are small or acceptable.

Keywords: *constraint handling, constrained optimization problems, evolutionary multi-objective optimization, decision making.*

1. Introduction

Many real-world search and optimization problems naturally involve constraint handling. By judging whether the objective functions and constraints are linear or nonlinear, we can classify the constrained optimization problems into two categories: linear constraint programming and nonlinear constraint programming. Simplex algorithm [1] can handle the former efficiently, while it faces difficulties when dealing with the latter, since the nonlinear objective functions and nonlinear constraints make the problem harder. Therefore, many heuristic methods were proposed to solve the nonlinear ones. In these heuristic methods, the classical ways to handle constraints is to convert the objective function and constraints into a weighted sum of objectives (penalty-function approach) [2] and then try to find the feasible and optimal solutions by optimizing the weighted sum function.

However, this constraint-handling method has some drawbacks. It is difficult to fix a weight vector for successful working and improper weight vector may lead the search process to local optimality rather than the global ones. Furthermore, in many real-world problems, some constraint can be 'soft' [3], that is, a solution with a permissible constraint violation can still be considered if there is a substantial gain in the objective function, which are not taken into account by the penalty-function approach. Thus, we need more flexible methods to provide the decision makers (DM) with more candidate solutions, including the solutions with slightly constraint violation but meanwhile with substantial gain in the objective function. In addition, in some difficult real world problems, the nonlinear constraints make most of, even all of the search space infeasible. We call this phenomenon over-constrained. To solve the over-constrained problems, there should be some compromises in one or several constraints based on the DM's experience and other conditions. To solve this kind of problems, we must release or compromise some constraints. It is the DM knowing the background and the real mean of the problem who should determine which constraints compromise and how much they need compromise. Therefore we should provide the DM a group of trade-off solutions rather than just one to make efficient decision.

We find that the objective handling for multi-objective optimization problems (MOPs) is somewhat similar to the constraint handling for constrained single-objective optimization problems (SOPs). The traditional ways of objective handling in MOP generally convert a MOP into a SOP by sum approach and solve it with classical optimization techniques. This sum approach in solving MOPs faces the similar difficulties as that faced with the penalty-function approach we described above. With the development of MOP optimization techniques, especially evolutionary multi-objective optimization (EMO), we may consider: can we convert the constraint SOPs into MOPs and solve them with the state-of-the-art EMO techniques.

In this paper, we gave a MOP-based formula of constrained single-objective optimization problems. Then we proposed the way to solve them by using one of the current multi-objective evolutionary algorithms. By simulation experiments, we find this approach for constraint handling not only can find the constrained optimality, but also can present the decision maker a group of trade-off solutions with slightly constraint violation and meanwhile with substantial gain in the objective function. This can enable the DM to have more freedom to choose his preferred solution and therefore exploit more profits in the margin of constraint violations, where the constraint violations are small or acceptable.

2. Constrained single-objective optimization

In a Constrained SOP, there exist a single objective function and a number of constraints [1]:

$$\begin{aligned} &Minimize \quad f(x) \\ &Subject\ to: \quad g_j(x) \le 0, \quad j = 1,2,...,J \\ &\qquad h_k(x) = 0, \quad k = J+1, J+2,...,K \\ &\qquad x_i^{(L)} \le x_i \le x_i^{(U)} \quad i = 1,2,...,n \end{aligned} \tag{1}$$

Without loss of generality, we assume that the objective function f (x) is minimized. For a maximization problem, the duality principle can be used to convert the problem into a minimization problem. In most difficult and real-world problems, the constraints $g_j(x)$ and $h_k(x)$ are nonlinear and make most of the search space infeasible. This causes difficulty even in finding a single feasible solution.

Therefore, many traditional methods [4], including conjugate gradient method, Newton iteration method, and modern heuristic methods, including simulated annealing [5], Tabu search, genetic algorithms [6] … etc., were proposed to solve the nonlinear constrained optimization problems. In these methods, the classical ways to handle constraints is to convert the objective function and constraints into a weighted sum of objectives and then try to find the feasible and optimal solutions by optimizing it. This method is largely known as the penalty-function approach, where the original objective function *f (x)* and all constraints are added together with a weighted vector consisting of penalty parameters, as follows [4]:

$$\begin{aligned} Minimize \ \ P(X,R,r) = f(x) + \sum_{i=1}^{j} R_i < g_i(X) > \\ + \sum_{i=j+1}^{k} r_i |h_i(X)| \end{aligned} \tag{2}$$

$$X = (x_1, x_2, ..., x_n)\ , x_i \in D_i\ , i = 1,2,...,n$$

$$R, r \quad are\ penalty \quad parameters$$

Where

$$< g_i(X) > = \begin{cases} g_i(X) & g_i(X) > 0 \\ 0 & g_i(X) \le 0 \end{cases}$$

$$|h_i(X)| = \begin{cases} h_i(X) & h_i(X) > 0 \\ -h_i(X) & h_i(X) \le 0 \end{cases}$$

However, this constraint handling method has some drawbacks including, but not limited to, the following:

(a) The quality of the solutions is heavily dependent on a weight vector (also called penalty parameters). The components of the weight vector determine a fixed path from anywhere in the search space towards the constrained minimum. Sometimes, instead of converging to the true constrained minimum, the path terminates to a local minimum. Thus, for sufficiently nonlinear problems, not all weight vectors will allow a smooth convergence towards the true constrained minimum. Often, the user has to experiment with various weight vectors to solve the constrained optimization problem.

(b) In many real-world problems, some constraints can be 'soft', that is, a solution with a permissible constraint violation can still be considered if there is a substantial gain in the objective function, which is not taken into account by the sum approach.

In some difficult real world problems, the nonlinear constraints make all of the search space infeasible. To solve the over-constrained problems, the DM should compromise in one or several constraints based on his experience and other conditions. Therefore we should provide the decision makers a group of trade-off solutions rather than just one to make efficient decision. However, the penalty-function constraint handling method cannot tackle this kind of problems.

3. Solving constrained SOPs with EMO

3.1. The MOP-based description of constrained SOPs

A constrained SOP defined in (1) can be posed as a MOP of minimizing the objective function and minimizing all constraint violations.

$$\begin{cases} Minimize & f(X) \\ Minimize & < g_i(X) > \quad i = 1,2,...,j \\ Minimize & |h_i(X)| \quad i = j+1, j+2,...,k \end{cases} \tag{3}$$

$$X = (x_1, x_2, ..., x_n),\ x_i \in D_i\ , i = 1,2,...,n$$

We can define:

$$f_i(X) = \begin{cases} f(X) & i = 0 \\ < g_i(x) > & i = 1,2,...,j \\ |h_i(x)| & i = j+1, j+2,...,k \end{cases}$$

Then (3) is equivalent to (4):

$$\begin{aligned} &Minimize \quad F(X) \equiv (f_0(X), f_1(X),...,f_k(X)) \\ &X = (x_1, x_2,...,x_n),\ x_i \in D_i\ ,\ i = 1,2,...,n \end{aligned} \quad (4)$$

Using this way, problem (1) is converted into problem (4), which is a typical MOP without constraints.

3.2. Evolutionary multi-objective optimization (EMO) for Solving constrained SOPs

Multi-objective optimization (MO) methods, as the name suggests, deal with finding optimal solutions to multiple objective optimization problems (MOPs). In a MOP, the presences of conflicting objectives give rise to a set of optimal solutions (called Pareto optimal Solutions [3]), instead of a single optimal solution. Thus, it becomes essential that a Multi-objective optimization algorithm find a wide variety of Pareto optimal solutions, instead of just one of them.

Evolutionary algorithms (EAs) are a natural choice for solving MOPs because of their populations approach. A number of Pareto-optimal solutions can be captured in an EA population, thereby allowing the DM to find diverse multiple Pareto-optimal solutions in one simulation run. In addition, the good search abilities of EAs can guarantee their good performance when dealing with problems having nonconvex search spaces and other difficult search spaces [7]. A lot of successful Multi-objective evolutionary algorithms (MOEAs), such as VEGA [8], NPGA [9], MOGA [10], NSGA [11], SPEA [12] have been proposed for EMO.

To solve the constrained optimization problem defined by (1), we first need to convert it to (4) by the way we have described above. We then try to find the good uniform solutions to approximate the Pareto front [7] (Pareto front is the image of the set of optimal solutions) of this MOP by using MOEAs. After we present these solutions, including the optimal solution without constraint violation and the solutions that violate one or some constraints marginally or by a large extent, to the DM, he will choose the BEST one from these candidate solutions based on his experience, his preference or other subjective or objective factors.

It is easy to prove that the Pareto front of (4) definitely includes the optimal solution of (1). Besides this solution, it also comprises the solutions with constraint violations and meanwhile with some gains in the objective function.

3.3. Advantages of using EMO in solving constrained optimization problems

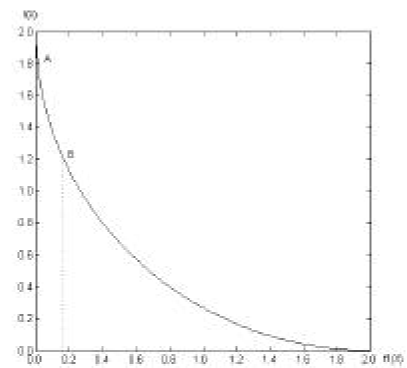
Constraint handling with EMO has some advantages comparing with other constraint handling methods:

(a) The Pareto front of (4) includes the optimal solution of (1), if (1) has a feasible optimal solution. Therefore, we can conclude that the optimal solution set we get by penalty-function approach is a subset of the solution set we get by EMO approach.

(b) By EMO, the constraint-handling problem can be solved in a natural way. There is no need of any penalty parameters and penalized objective function.

(c) Besides the optimal solution of (1), This method can provide the DM with the trade-off solutions that violate one or some "soft" constraints marginally or by a large extent but have substantial gain in the objective function. In figure 1, if the constraint is "hard", that is, no violation is permitted; then the only solution is on point A, where the constraint violation is zero. If the constraint is

f(x)



Constraint violation

Figure 1. An example for the Pareto front (trade-off) of objective function and constraint violation

"soft", then the DM can choose his preferable solution, which can make the most profits for him, among the whole Pareto front. The DM will carry out this posteriori decision making, based on his experience, his preference or other subjective or objective factors. With the visualization of the candidate solutions as in figure 1, the trade-off of the objective function and the "soft" constraints can by easily carried out by the DM. If we use penalty-function approach for constraint handling, then the decision process is priori. For the priori decision making process, we must decide the weight parameters before optimization, which is very difficult for both decision makers and optimizators.

(d) The good search abilities of EAs can guarantee their good performance when tackling the problems with nonconvex search spaces and other difficult search spaces.

(e) It can handle over-constrained problems. In figure 1, if the minimum of the constraint violation is above zero, then the whole search space is infeasible (this more likely happens, if the problem has more constraints and there are no points where every constraints equals zero). To solve this kind of problems, we must release or compromise some constraints. This compromise decision process is problem-dependent. With the group of nondominated solutions presented by this method, the DM, who has more knowledge of the real mean of the problem than the computer scientists can determine which constraints compromise and how much they need to compromise and therefore choose the best solution.

When practical considering, we are usually interested in the solutions, which are biased towards the region where all constraint violations are small. In figure 1, although the DM can choose his favorite solution from the whole Pareto front, he may be especially interested in the solutions within the region AB, where the constraint violation are small. Thus, we can use an algorithm, which is bias in finding Pareto optimal solutions, namely, which can find dense solutions towards the region with small constraint violations and lank solutions towards the region with large constraint violations.

4. Experiments and analyses

To help gain further insight on the effectiveness of this EMO constraint handling approach, a series of experimental simulations were run.

4.1. Test functions

Test function 1 (TF1) is an extremely hard and famous constrained optimization problem, known as Bump problem [2].

$$\begin{aligned} &Maximize\ f(x)=\frac{\left|\sum_{i=1}^{n}\cos^4(x_i)-2\prod_{i=1}^{n}\cos^2(x_i)\right|}{\sqrt{\sum_{i=1}^{n} ix_i^{\,2}}} \\ &s.t.\quad \prod_{i=1}^{n}x_i \ge 0.75\ ,\sum_{i=1}^{n}x_i \le 7.5n \\ &where\quad 0<x_i<10,\quad 1\le i\le n \end{aligned} \tag{5}$$

Test function 2 (TF2) is a constrained optimization problem with two "soft" constraints from [5,6].

$$\begin{aligned} &Minimize\quad f(X)=(x_1-2)^2+(x_2-1)^2 \\ &s.t.\quad h(X)=x_1-2x_2+1=0 \\ &\qquad g(X)=-0.25x_1^2-x_2^2+1\ge 0 \\ &where\quad x_1\in[-1.82,0.82],\ x_2\in[-0.41,0.92] \end{aligned} \tag{6}$$

4.2. Implementation details

Here, we adopted SEEA [13] as the MOEA for EMO.

In TF1, we notice that although it has two constraints, the second constraint is ineffective, that is, it has no effect on the feasible search space. We converted the first constraint into an objective; meanwhile, we converted the problem into a minimization problem by the duality principle. Therefore, TF1 was converted into (7) by the means we described in section 3.1:

$$\begin{cases} Min\ f_1(x)=-\dfrac{\left|\sum_{i=1}^{n}\cos^4(x_i)-2\prod_{i=1}^{n}\cos^2(x_i)\right|}{\sqrt{\sum_{i=1}^{n} ix_i^{\,2}}} \\ Min\ f_2(x)=\begin{cases} 0.75-\prod_{i=1}^{n}x_i & if\quad (0.75-\prod_{i=1}^{n}x_i)>0 \\ 0 & otherwise. \end{cases} \end{cases} \tag{7}$$
$$where\quad 0<x_i<10,\quad 1\le i\le n.$$

As we discussed in section 3.3, here we integrated the bias techniques into SEEA. Since it is easy for this integration, we will not explain it in detail. Please refer to [3] for the details. The other parameters are set as following:

n (problem's dimension) =2, Max generation=10000; Population=50; Number of parents for multi-parent crossover = 6; s_{share}=0.041.

TF2 has two "soft" constraints, both of which are effective. Therefore we converted TF2 into a three objective problem:

$$\begin{cases} Minimize\quad f_1(X)=(x_1-2)^2+(x_2-1)^2 \\ Minimize\quad f_2(X)=|h(X)| \\ Minimize\quad f_3(X)=\begin{cases} 0 & g(X)\ge 0 \\ -g(X) & otherwise \end{cases} \end{cases} \tag{8}$$
$$where\ x_1\in[-1.82,0.82],\ x_2\in[-0.41,0.92]$$

For TF2, we used SEEA directly without bias techniques. The other parameters are set as following:

Max generation=10000; Population=80; Number of parents for multi-parent crossover = 6; s_{share}=0.038.

4.3. Experimental results and discussions

Figure 2 is the Pareto front of (7) we got with bias SEEA in a simulation run.

In Figure 2, $f_1(x)$ is TF1's objective function and $f_2(x)$ is the first constraint violation of TF1. If the DM is especially interested in the solutions, whose constraint violation is less than $e = 0.03$, there will be five candidate solutions, whose objective function values and constraint violations are as following, for DM to choose.

Solution 1: f1= -0.36491219978220,
f2= 0.00000000000000
Solution 2: f1= -0.36565642706854,
f2= 0.00135722933260
Solution 3: f1= -0.37878744522919,
f2= 0.02577871038941
Solution 4: f1= -0.37933409043999,
f2= 0.02703805632788
Solution 5: f1= -0.37987044678696,
f2= 0.02786564881056

In current literatures [2,14], when n=2, the best solution of Bump problem (5) found with penalty-function method is 0.36497974587066. Although the DM can use this non-constraint-violated solution, he will not know how much he can gain if slight constraint violations to a certain extend are permitted. While our constraint handling method can find a group of candidate solutions. We can find that although solution 2-5 have constraint violations (since f2>0), they meanwhile have substantial gains in the objective functions. Basing on the real mean of this problem, the DM can make trade-off between constraint violations and objective function gains and therefore choose his preferable solutions freely to gain more profits. The main difference between TF1 and TF2 is that TF2 has two "soft" constraints. Therefore we converted TF2 into (8), a three objective problem. Consequently, The Pareto front of (8) we got with SEEA is of three dimensions, so we plot it with three two-dimension graphs.

Our constraint handling method can give the vivid relationship between the objective function and the constraints. Figure 3 is the relationship between the objective function and the first constraint of TF2, figure 4 is the relationship between the objective function and the second constraint and figure 5 is the relationship between the first and the second constraint. From figure 3-5, we notice visually that at point A, although the first constraint(since $f_2>0$) is violated, there is a substantial

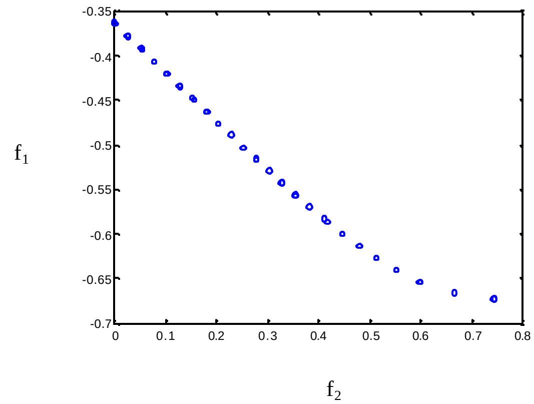


Figure 2. Pareto front of (7)

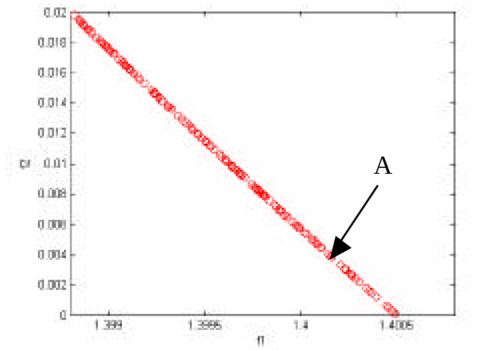


Figure 3. Trade-off front between f_1 and f_2

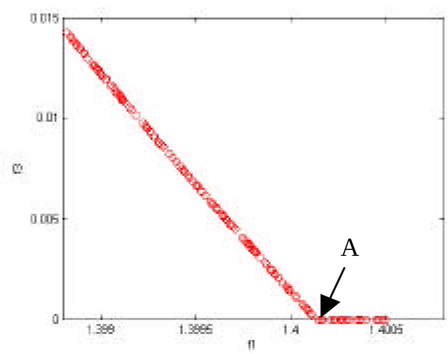


Figure 4. Trade-off front between f_1 and f_3

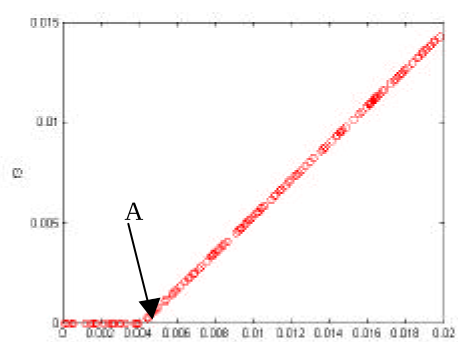


Figure 5. Trade-off front between f_2 and f_3

gain in the objective function, at the same time the second constraint (f_3=0) is not violated. Therefore, we consider point A may be very alluring to the DM. Besides point A, the other solutions with different extent constraint violations and different extent objective gains also can be presented to the DM. All these cannot be realized with penalty-function constraint handling method.

5. Conclusion and future work

In this paper, we first analyzed the drawbacks of penalty-function constraint handling method for nonlinear constraint programming. Then we presented a new constraint handling method based on EMO. Experimental tests demonstrated that this constraint handling method not only can find the constrained optimality, but also can provide the decision maker with a group of trade-off solutions with slightly constraint violation and meanwhile with substantial gain in the objective function, This enables the DM to have more freedom to choose his preferred solution and therefore exploit more profits in the margin of constraint violations, where the constraint violations are small or acceptable. In addition, the good search abilities of EAs can guarantee their good performance when tackling the problems with nonconvex search spaces and other difficult search spaces.

In the near future, we intend to integrate our EMO-based constraint handling into a decision support system, which can offer DM visualization decision support for linear and nonlinear constraint programming. We also intend to do some experiments of tackling over-constrained problems with this method and therefore extend our system in order to manage over-constrained problems.

References

[1] LU Kaicheng. Single-objective, multi-objective and integer programming. Tsinghua University Press,1999(ch).

[2] Michalewiz Z. Genetic Algorithms +Data Structures = Evolution Programs. Springer-Verlag, Berlin. 1992

[3] Deb,K. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley&Sons,Ltd Baffins Lane,Chichester,West Sussex,PO19,IUD,England. 2001

[4] YUAN Yaxiang, Sun Wenyu. Theory and method of optimization. Science Press,1999(ch).

[5] WU Zhiyuan, SHAO Huihe,WU Xinyu. Annealing Accuracy Penalty-function Based Nonlinear Constrained Optimization Method with Genetic Algorithms. *Control and Decision*, 1998,Vol.13 No.2,pp136-140(ch).

[6] Homaifar A, Charlene X Q, Lai SH. Constrained optimization via genetic algorithms. Simulation. 1994,62 (4) :242-254.

[7] David A.VanVeldhuizen, Gary B. Lamont. Multi-objective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evolutionary Computation*, 2000, 8(2):125-147.

[8] Schaffer,J. Multiple objective optimization with vector evaluated genetic algorithms. *Proceeding of the First International Conference Genetic Algorithms*, 1985, page 93-1000, Lawrence Erlbaum, Hillsdale, New Jersey.

[9] Horn,j, Nafpliotis,N., Goldberg,D.E. A Niche Pareto Genetic Algorithm for Multi-objective Optimization. *Proceedings of the first IEEE Conference Evolutionary Computation*, 1994, pages 82-87, IEEE Press, Piscataway, New Jersey.

[10] Carlos Fonseca, Peter J. Fleming Multi-objective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part 1:A Unified Formulation. *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, 1998, 28(1): 26-37.

[11]N.Srinivas, Kalyanmoy Deb. Mult-objective Optimization Using Nondominated Sorting In Genetic Algorithms, *Evolutionary Computation*, 1995, 2(3) 221-248.

[12] Zitzler,E., Thiele, L. Multi-objective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4), 257-271.

[13] Yan Zhenyu, Kang Lishan, Bob (R I) Mckay. SEEA For Multi-Objective Optimization: Reinforcing Elitist MOEA Through Multi-Parent Crossover, Steady Elimination and Swarm Hill Climbing, *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning*, 2002, Vol. I, pages 21-26, Singapore.

[14] GUO Tao,KANG Li-shan,LI Yan. A New Algorithm for Solving Function Optimization Problems with Inequality Constraints. *J Wuhan Univ(Nat Sci ED)*,1999,45(5B):771-775(Ch).