Hybridization of the Multi-Objective Evolutionary Algorithms and the Gradient-based Algorithms

Xiaolin Hu, *Student Member,* IEEE
Automotive Engineering Institute
Wuhan Univ. Of Tech.
Wuhan, Hubei 430070, P.R. China
x.l.hu@163.com

Zhangcan Huang
School of Science
Wuhan Univ. Of Tech.
Wuhan, Hubei 430070, P.R. China
huangzc@mail.whut.edu.cn

Zhongfan Wang
Automotive Engineering Institute
Wuhan Univ. Of Tech.
Wuhan, Hubei 430070, P.R. China
wangzf@mail.whut.edu.cn

Abstract- It is known from single-objective optimization that hybrid variants of local search algorithms and evolutionary algorithms can outperform their pure counterparts. This holds, in particular, in continuous search spaces and for differentiable fitness functions. The same should be true in multi-objective optimization. This approach is started in this paper. An efficient gradient-based local algorithm, sequential quadratic programming (SQP) is combined with two well-known multi-objective evolutionary algorithms, strength Pareto evolutionary algorithm (SPEA) and non-dominated sorting genetic algorithm (NSGA-II) respectively, by means of a modified ε -constraint method. The resulting two hybrid algorithms demonstrate great success over two sets of well-chosen functions regarding the convergence rate. In addition, from the simulation studies, the hybridization approach also enhances, at least does not ruin, the diversity of the solutions.

1 Introduction

Real-world optimization problems often involve multiple, competing objectives in a highly complex search space. Multi-objective optimization problems (MOOPs) distinguish themselves from single-objective optimization problems (SOOPs) in that when preference information is absent no optimal solution is clearly defined but rather a set of alternative trade-off solutions exist. At present, the evolutionary algorithm (EA) is assumed to be one of the most promising approaches to solve the MOOPs. Among various characteristics of the EAs, the obvious one is that these algorithms operate on a set of solutions (or population), which caters for the need of finding a set of 'optima' for the MOOPs, i.e. Pareto solutions. Much work has been done in this field since the work of Schaffer in 1984 [1], which is recognized as the first real implementation of a multi-objective evolutionary algorithm, named vector evaluated genetic algorithm or VEGA. After nearly a decade, Goldberg suggested a revolutionary 10-line sketch of a new non-dominated sorting procedure [2]. Followed his suggestion, at least three different versions of multi-objective evolutionary algorithms (MOEAs), MOGA [3], NPGA [4], and NSGA [5], were derived. During the last two yeas of last century, by eliminating the fitness sharing factor and introducing the elitism, more effective MOEAs were formed, such as Zitzler and Thiele's SPEA [6], Knowles and Corne's PAES [7], and Deb et al's NSGA-II [8][9]. More recent development including Corne et al's PESA [10], Deb and Goel's NSGA-II with controlled elitism [11], and Zitzler et al's SPEA2 [12].

Though these MOEAs have shown great success on many complex problems, there exists some problem, one of which is that they often require a lot of objective evaluations. As the computational cost for evaluating a solution becomes surprisingly high, it will be tedious or even impossible to apply them to such problems. It is the inherent weakness of EAs that results in the embarrassment. EAs are stochastic algorithms, and a small number of samplings in the decision space are insufficient to assure the proper results. In order to overcome such a deficiency, EA workers have suggested various approaches to accelerate the algorithms' convergence rate, among which one of the most common and effective techniques is to incorporate local search approaches into EAs. It is based on the conception that EAs will guide the search direction into the right region (global area) and local search approaches then will find the accurate optimum there quickly because of their speedy convergences. Such a conception may be adopted by the MOEAs, too. But most local algorithms can only optimize SOOPs, while the *optimum* for multi-objective optimization problems is a set of solutions, which is termed *Pareto-optimal set* or *Pareto-optimal front.* Hence, some specifications must be made before the idea is applied to the MOEAs.

A direct and simple technique is to combine the MOEAs and ε -constraint method, using the local algorithms to optimize one of the objectives, while restricting other objectives within specified values. Ritzel et al discussed a naive approach to combine the ε -constraint method with GA [13]. Through a process of running the GA numerous times with different values of the constrained objectives, say, with different $\boldsymbol{\varepsilon}$ values, a trade-off surface can be developed. It is the fundamental conception of the CMEA invented by Ranjithan et al [14]. But this approach has insufficient *intelligence* in searching for the suitable $\boldsymbol{\varepsilon}$ vector. What's more, it is easy to imagine the high computational complexity. Ritzel et al rejected the approach in their paper [13]. In this study, we use a modified ε -constraint method to

incorporate an efficient gradient-based local search method, sequential quadratic programming (SQP), into two well known MOEAs, SPEA and NSGA-II. The focus will be placed on the convergence rate of the resultant hybrid MOEAs and the diversity as well as globality of the obtained solutions.

2 Mathematical Prelude

Some key definitions in relation to the multi-objective optimization used in this paper are stated below. For detailed information about them, please refer to [15].

Definition 1 (Multi-Objective Optimization Problem)

$$\left.\begin{array}{lll} \min & f_m(\mathbf{x}), & m = 1,2,\cdots,M; \\ \text{s.t.} & g_j(\mathbf{x}) \geq 0, & j = 1,2,\cdots,J; \\ & h_k(\mathbf{x}) = 0, & k = 1,2,\cdots,K; \\ & x_i \in [x_i^L, x_i^U], & i = 1,2,\cdots,n \end{array}\right\} \quad (1)$$

Without loss of generality, all the objectives are assumed to be minimized in this paper.

Definition 2 (Domination) Solution $\mathbf{x}^{(1)}$ is said to dominate the other solution $\mathbf{x}^{(2)}$ if both conditions below are true:

- $\mathbf{x}^{(1)}$ is no worse than $\mathbf{x}^{(2)}$ in all the objectives, or $f_j(\mathbf{x}^{(1)}) \leq f_j(\mathbf{x}^{(2)})$ *for all* $j = 1, 2, \ldots, M$.
- $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ in at least one objective, or $f_j(\mathbf{x}^{(1)}) < f_j(\mathbf{x}^{(2)})$ for at least one $j \in \{1, 2, \ldots, M\}$.

Definition 3 (Non-dominated Set and Pareto-optimal set) Among a set of solutions P, the non-dominated set of solutions P' are those that are not dominated by any other member of the set P.

P' is also called the *non-dominated front* of P.

If set P is the entire feasible search space, the non-dominated set P' is called the Pareto-optimal set.

Definition 4 (Global Pareto-optimal set) The non-dominated set of the entire feasible search space is the global Pareto-optimal set.

Definition 5 (Local Pareto-optimal set) If for every member $\mathbf{x}$ in a set P there exists no solution $\mathbf{y}$ (in the neighborhood of $\mathbf{x}$ such that $\|\mathbf{y} - \mathbf{x}\|_\infty \leq \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is a small positive number) dominating any member of the set P, then solutions belonging to the set P constitute a local Pareto-optimal set.

And the two main goals in a multi-objective optimization are:

- To find a set of solutions as close as possible to the true Pareto-optimal front.
- To find a set of solutions as diverse as possible.

3 SPEA and NSGA-II

It has been indicated at the very beginning of section 1 that evolutionary algorithms are leading methods in the multi-objective optimization field nowadays. Many salient MOEAs have emerged during last decades. Among those, Zitzler et al's SPEA [6] and Deb et al's NSGA-II [8][9] are most famous. This section will describe them briefly in that they are to be used as the frameworks of the hybrid MOEAs discussed in the next section.

The algorithm, strength Pareto EA (SPEA), introduces elitism by explicitly maintaining an external population E_t. This population stores a fixed number of the non-dominated solutions that are found until the beginning of a simulation. At every generation, newly found non-dominated solutions are compared with the existing external population and the resulting non-dominated solutions are preserved. The size of E_t is bounded to a limit $\overline{N}$. When the size exceeds $\overline{N}$, elites which are less crowded in the non-dominated front are kept. (It is to attain the second goal of multi-objective optimization.) In addition, the SPEA does more than just preserving the elites; it also uses these elites to participate in the genetic operations along with the current population in the hope of influencing the population to steer towards good regions in the search space. It combines the main population P_t of size N and E_t of size $\overline{N}$ together (into an R_t population) and performs a fitness assignment procedure, which emphases the elite members. Then, the conventional binary tournament selection operator as well as the crossover and mutation operators is implemented orderly to create a new population P_{t+1} of size N.

In NSGA-II, the offspring population Q_t is first created by using the parent population P_t. The two populations are then combined together to form R_t of size $2N$. A non-dominated sorting is used to classify the entire population R_t. Once the non-dominated sorting is over, the new population P_{t+1} is filled by solutions of different non-dominated fronts, one front at a time. The different non-dominated fronts of a set of solutions can be obtained by gradually disregarding the fronts previously found and identifying the new front of the resulting set of solutions. The filling starts with the best non-dominated front and continues with solutions of the second non-dominated front, followed by the third non-dominated front, and so on. Since the overall population size of R_t is $2N$, not all fronts may be accommodated in N slots available in the new population. All fronts which could not be accommodated are simply deleted. When the last allowed front is being considered, there may exist more solutions in the last front than the remaining slots in the new population. Instead of arbitrarily discarding some members from the last front, it would be wise to use the niching strategy to choose the more crowded members of the last front. Then the selection, crossover and mutation operators are used to create an offspring population Q_{t+1} of size N from P_{t+1}. The authors define a new selection operator, termed *crowded tournament operator*, which considers the solutions' dominance and diversity simultaneously.

It should be mentioned that both of the two algorithms attempt to attain the second main goal of multi-objective optimization indicated in section 2 by techniques

requiring no additional parameters compared to a more often used technique, *sharing function model* [15][16]. The NSGA-II defines a *crowding distance* of each solution to do the work and the SPEA suggests a clustering method.

More detailed descriptions of the two algorithms can be found in the papers mentioned in this section, however, the flow charts of the hybrid MOEAs presented in section 4.4 also serve for help to understand the two original MOEAs.

4 Hybrid MOEAs

4.1 Motivation for Hybridization

It is well known that EAs are probability-based algorithms. Though they have strong ability to converge to the true global optimum, the process is always slow compared to some local search algorithms. It is often a promising direction to combine the local algorithms and global algorithms considering the merits and deficiencies of each. Much work has been done in terms of single objective optimization. But for MOOPs, still much effort needs to be made.

It is worth arguing that the incorporation of EAs and local approaches for multi-objective optimization has an additional advantage than for single objective optimization, in that if there exist no local traps in the MOOP search space, the local technique will dramatically accelerate the MOEAs' convergence rate. Furthermore, there is no doubt for the necessity of the evolutionary algorithms in such a case. As it has been mentioned earlier, EAs have the ability to return a set of solutions, which is essential for MOOPs. Nevertheless, if such condition is true for single objective optimization, the EAs are nearly unnecessary.

But how to incorporate the local search procedure into the MOEAs is a problem since most local algorithms can only handle SOOPs. It may be wise to combine it with the ε-constraint method. So, let us discuss the ε-constraint method.

4.2 Modified ε-Constraint Method

Haimes et al. [17] suggested reformulating the MOOP by just keeping one of the objectives, say, *main objective*, and restricting the rest objectives within user-specified values. The problem stated in Eq. (1) is modified as follows:

$$\left.\begin{array}{ll} \text{Min} & f_\mu(\mathbf{x}) \\ \text{s.t.} & f_m(\mathbf{x}) \le \varepsilon_m,\ m = 1,2,\cdots,M;\ m \ne \mu \\ & g_j(\mathbf{x}) \ge 0, \quad j = 1,2,\cdots,J; \\ & h_k(\mathbf{x}) = 0, \quad k = 1,2,\cdots,K; \\ & x_i \in [x_i^L, x_i^U],\ i = 1,2,\cdots,n \end{array}\right\} \qquad (2)$$

The parameter ε_m represents an upper bound of the value of f_m. In terms of a two objective problem, assume that we retain f_2 as an objective and treat f_1 as a constraint: $f_1(\mathbf{x}) \le \varepsilon_1$. Fig. 1 shows three scenarios with different ε_1 values. If $\varepsilon_1 = \varepsilon_1^b$, the resulting problem with this constraint divides the original feasible objective space into two portions: $f_1 \le \varepsilon_1^b$ and $f_1 > \varepsilon_1^b$. The left portion becomes the feasible solutions of the resulting problem stated in Eq. (2). From fig. 1, it is clear that the minimum solution is 'B'. In this way, intermediate Pareto-optimal solutions can be obtained in the case of nonconvex objective space problems. This is the conspicuous advantage of the ε-constraint method distinguishing it from the classical weighted methods.

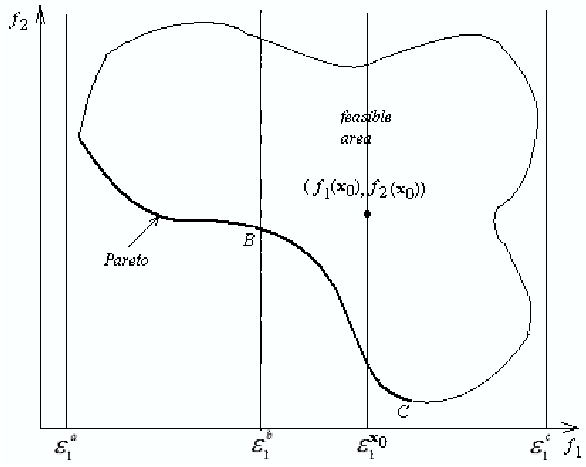

Figure 1. The ε-constraint method

But the results largely depend on the chosen $\boldsymbol{\varepsilon}$ vector. It must be chosen so that it lies within the minimum or maximum values of the (M-1) objective function. If $\varepsilon_1 = \varepsilon_1^a$ in fig. 1, there exists no feasible solution to the stated problem. On the other hand, if ε_1^c is chosen, the entire search space is feasible and the method will always find the unique optimum 'C'. Moreover, as the number of objectives increases, there exist more elements in the $\boldsymbol{\varepsilon}$ vector, thereby requiring more information from the user. Even it is combined with EAs, leaving the choice to be decided by the computer, how to avoid such a dilemma is confusing. Maybe, through a number of simulation runs the suitable $\boldsymbol{\varepsilon}$ vector would not be achieved.

In this study, instead of changing ε values for a fixed solution with a fixed main objective for different simulation runs, we change the solutions as well as their main objectives continuously in one simulation run. The process is as follows:

1) Several members of the current EA population are randomly selected.

2) For each selected solution, randomly chose a main objective to optimize, while restricting the remaining (M-1) objectives less than or equal to the solution's current corresponding objective values.

It is to say, if that selected solution from the population is $\mathbf{x}_0$, then in Eq. (2), $\varepsilon_m = f_m(\mathbf{x}_0)$ ($m = 1,2,\cdots,M;\ m \ne \mu$). In this manner, the $\boldsymbol{\varepsilon}$ vector is constant and can be easily decided. This approach is

termed a *modified* ε *-constraint method*. After the execution of this method on the current population, the resulting several optimized solutions substitute the original solutions in the population, and serve as *good* seeds in the population to accelerate the process towards the Pareto-optimal front.

This has at least two merits. The first one is obvious that the $\boldsymbol{\varepsilon}$ vector will never be false. In fig. 1, the $\varepsilon_1^{\mathbf{x}0}$, say, $f_1(\mathbf{x}_0)$, will never got the value ε_1^a or ε_1^c since the solution $\mathbf{x}_0$ is always in the feasible space. The other merit is that, since the solutions and their main objectives are selected randomly, no objective of the problem is biased. Therefore it maintains, or even improves the diversity of the solutions considering the local algorithms' fast convergence to each objective's optimum.

4.3 Sequential Quadratic Programming

In terms of the continuous problems, gradient-based algorithms are always the fastest method for searching for the optimum though they may only find the local optimum. And among those, Sequential Quadratic Programming (SQP) represents state-of-the-art in nonlinear programming methods. Schittowski [18], for example, has implemented and tested a version that out performs every other tested method in terms of efficiency, accuracy, and percentage of successful solutions, over a large number of test problems. At each major iteration an approximation is made of the Hessian of the Lagrangian function using a quasi-Newton updating method (in this study, BFGS method is used). This is then used to generate a QP sub-problem whose solution is used to form a search direction for a line search procedure. Since this method has been fully developed in some sense, the paper does not intend to discuss the details of SQP. An overview of SQP is found in Fletcher [19], Gill et al [20].

4.4 Hybrid NSGA and Hybrid SPEA

Based on the discussion above, two hybrid MOEAs named hybrid SPEA (or SPEA-SQP) and hybrid NSGA (or NSGA-SQP) are derived from the original SPEA and NSGA[1] outlined in section 3. The SQP algorithm is used to accelerate the convergence rate by means of the modified ε -constraint method. Fig. 2 shows the positions where the SQP procedures are inserted into the processes of SPEA and NSGA. The SQP procedures are both performed on the current populations.

It should be noted that in order to reduce the calculation, SQP is started with several individuals instead of the whole population, and at the *interval* generation instead of each generation. For each randomly selected solution, randomly select an objective to optimize using SQP.

Fig. 2 also briefly illustrates the processes of the original SPEA and NSGA.

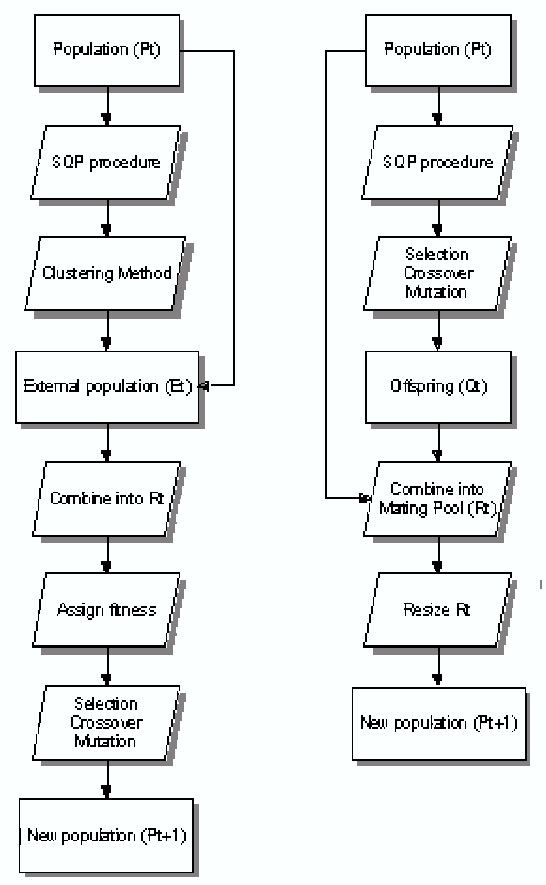

Figure 2. Two hybrid MOEAs (left: SPEA-SQP, right: NSGA-SQP)

5 Simulation Studies

In this section, two sets of unconstraint problems are used to test the new algorithms' performances.

5.1 The First set of Test Problems

Zitzler et al. framed six problems (ZDT1 to ZDT6) [23], which have been further studied by other researchers. The problems have the same structure as follows:

$$\begin{aligned} &\min \quad f_1(\mathbf{x}) \\ &\min \quad f_2(\mathbf{x}) = g(\mathbf{x})h(f_1(\mathbf{x}), g(\mathbf{x})) \end{aligned} \tag{3}$$

All the problems are presented here except the ZDT5 since its variables are discrete and hence it is not suitable for the SQP procedure.

ZDT1 :

$$\left.\begin{aligned} &f_1(\mathbf{x}) = x_1 \\ &g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\ &h(f_1, g) = 1 - \sqrt{f_1 / g} \end{aligned}\right\} \tag{4}$$

[1] It means NSGA-II. From now on, we will substitute 'NSGA' for 'NSGA-II' for convenience.

Here, n=30 and all variables lie in the range [0, 1]. The Pareto-optimal region corresponds to $0 \le x_1^* \le 1$ and $x_i^* = 0$, for i=2, 3, ⋯, 30.

ZDT2:

$$\left.\begin{aligned} f_1(\mathbf{x}) &= x_1 \\ g(\mathbf{x}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\ h(f_1, g) &= 1 - (f_1 / g)^2 \end{aligned}\right\} \quad (5)$$

n=30. All variables lie in the range [0, 1]. The Pareto-optimal region corresponds to $0 \le x_1^* \le 1$ and $x_i^* = 0$, for i=2, 3, ⋯, 30. This Pareto-optimal region is nonconvex.

ZDT3:

$$\left.\begin{aligned} f_1(\mathbf{x}) &= x_1 \\ g(\mathbf{x}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\ h(f_1, g) &= 1 - \sqrt{f_1 / g} - (f_1 / g)\sin(10\pi f_1) \end{aligned}\right\} \quad (6)$$

n=30. All variables lie in the range [0, 1]. The Pareto-optimal region corresponds to $0 \le x_1^* \le 1$ and $x_i^* = 0$, for i=2, 3, ⋯, 30, and not all points satisfying $0 \le x_1^* \le 1$ lie on the Pareto-optimal front, which is discontinuous.

ZDT4:

$$\left.\begin{aligned} f_1(\mathbf{x}) &= x_1 \\ g(\mathbf{x}) &= 1 + 10(n-1) + \sum_{i=2}^{n}\left(x_i^2 - 10\cos(4\pi x_i)\right) \\ h(f_1, g) &= 1 - \sqrt{f_1 / g} \end{aligned}\right\} \quad (7)$$

n=10. The variable x_1 lies in the range [0, 1] but all others lie in [-5, 5]. The global Pareto-optimal front corresponds to $0 \le x_1^* \le 1$ and $x_i^* = 0$, for i=2, 3, ⋯, 10. There exists 21^9 local Pareto-optimal fronts.

ZDT6:

$$\left.\begin{aligned} f_1(\mathbf{x}) &= 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\ g(\mathbf{x}) &= 1 + 9\left(\sum_{i=2}^{n} x_i / (n-1)\right)^{0.25} \\ h(f_1, g) &= 1 - (f_1 / g)^2 \end{aligned}\right\} \quad (8)$$

n=10. All variables lie in the range [0, 1]. The Pareto-optimal region corresponds to $0 \le x_1^* \le 1$ and $x_i^* = 0$, for i=2, 3, ⋯, 10. The Pareto-optimal front is nonconvex. But the most complexity is that the density of solutions across the Pareto-optimal region is highly non-uniform and the density towards the Pareto-optimal front is also thin.

For all algorithms, a real-coded genetic algorithm with *arithmetic crossover* operator and *non-uniform mutation* operator [15][21] is used. And the following parameter values are the same:

Population size: 50
Crossover rate: 0.6
Mutation rate: 0.1

For the SPEA and SPEA-SQP, the regular EA population size is 40 and the external population is 10. Both SPEA and NSGA run maximum number of generation 500 for all the problems but 2000 for ZDT4. ZDT4 is extremely difficult for its numerous local traps. The two hybrid algorithms, SPEA-SQP and NSGA-SQP, each runs total 20 generations and performs the SQP procedure every 5 generations for ZDT1, ZDT2 and ZDT6. For ZDT3, they total number of generations is 50 and the SQP interval number of EA generations is 5. While for ZDT4, the two parameters are set to 500 and 10, respectively. The proportion of the solutions to perform the SQP procedure to the whole EA regular population is 1/5.

For all studies the outputs are the non-dominated solutions in the final population (NSGA or NSGA-SQP) or the final EA population combined with the external population (SPEA or SPEA-SQP). Fig. 3 to Fig. 7 illustrate the results of the algorithms in the objective spaces, and Table 1 lists the detailed number of objective evaluations[2] and the CPU time for each the test problem.

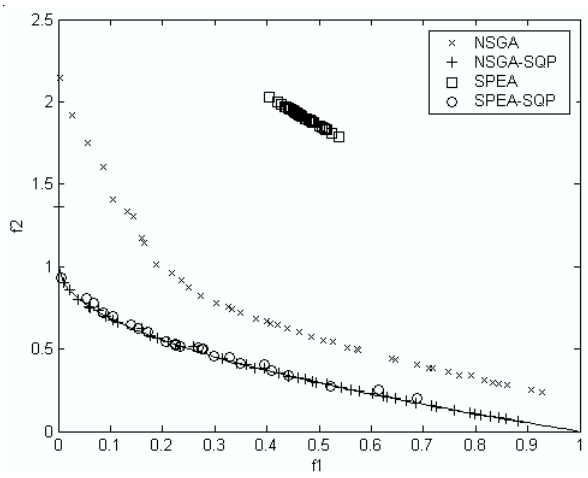

Figure 3. Non-dominated solutions of ZDT1

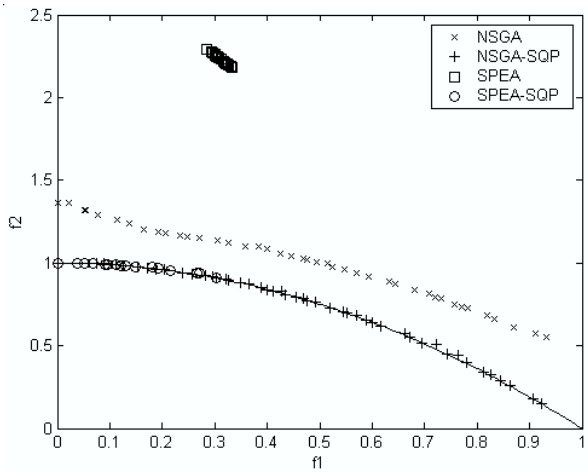

Figure 4. Non-dominated solutions of ZDT2

[2] 1 *evaluation* means one evaluation for each objecitve.

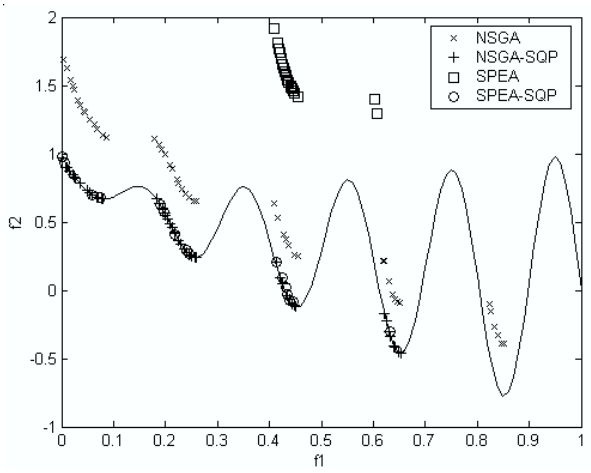

Figure 5. Non-dominated solutions of ZDT3

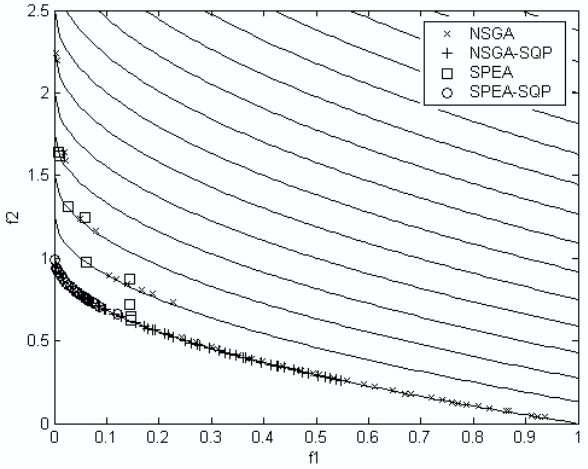

Figure 6. Non-dominated solutions of ZDT4

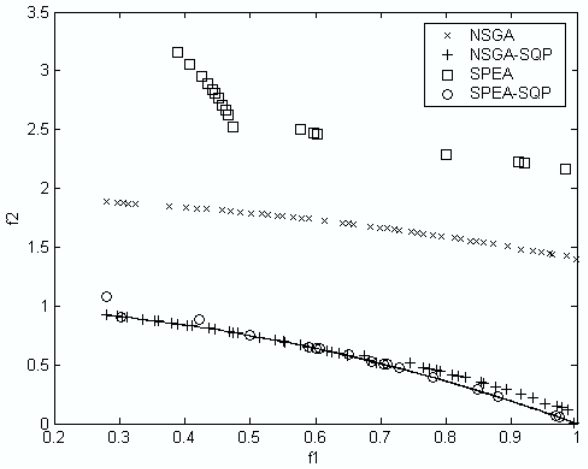

Figure 7. Non-dominated solutions of ZDT6

From Fig. 3 to Fig 7, we can see that the hybrid MOEAs have a better convergence to the Pareto-optimal front than the original MOEAs for all of the five problems. This point can be easily got from Fig. 3, Fig. 4, Fig. 5 and Fig. 7. Let's discuss Fig. 6 in detail. The ZDT4 has 21^9 local Pareto-optimal solutions and consequently makes a total of 100 distinct Pareto-optimal fronts including the global one. In Fig. 6, each continuous curve represents a local Pareto-optimal front and the lowest one represents the global front. This figure only shows a partial search region in the objective space. It can be seen that a very few of solutions of SPEA get to the true front, and some solutions of NSGA reside in the local fronts; while most solutions of SPEA-SQP and NSGA-SQP are global Pareto-optimal solutions. However, the diversity of the NSGA-SQP solutions is worse than that of the NSGA solutions.

Table 1. Objective evaluations and CPU time for ZDT1-ZDT6 (evaluations/CPU time)

	SPEA	NSGA	SPEA-SQP	NSGA-SQP
ZDT1	20000/447	25050/170	4063/19	4290/21
ZDT2	20000/259	25050/162	3296/12	3746/16
ZDT3	20000/256	25050/163	11483/53	11794/68
ZDT4	80000/268	100050/546	76778/216	93643/254
ZDT6	20000/141	25050/161	2042/8	2115/9

Table 1 remembers us that the results of the two hybrid MOEAs are obtained through far fewer objective evaluations as well as far less CPU time than the original MOEAs do for most problems. ZDT6 is an example. For this problem, SPEA-SQP evaluates the objectives nearly one tenth times as the SPEA does, and NSGA-SQP runs for nearly one-eighteenth CPU time as NSGA does, but both SPEA-SQP and NSGA-SQP get better solutions than SPEA and NSGA.

It can be also observed that the solutions of the SPEA-SQP represent a better diversity than that of the pure SPEA for all the five problems through Fig. 3 to Fig.7. And it should be mentioned that for ZDT6, which is designed to cause non-uniform density difficulty, neither blocks the NSGA nor blocks the NSGA-SQP in terms of the solutions' diversity..

5.2 The Second set of Test Problems

Although Zitzler et al's five functions represent different features and levels of difficulty, the first objective f_1 is always a simple function solely of the first decision parameter x_1. Consequently optimization chiefly consists of minimizing f_2, rather than the combined minimization of both objectives. For the purpose of searching for the extent of the Pareto front, five new test problems named HU1 to HU5 are introduced. All five are combinations of the following five base functions.

$$B1 = \sum_{i=1}^{n} \left(x_i - \frac{1}{3} \exp(\cdots / n^{\,2}) \right)^2 \tag{9}$$

$$B2 = \sum_{i=1}^{n} \left(x_i - \frac{1}{2} (\cos(\cdots) 0\pi(\cdots / n \;\; +1) \right)^2 \tag{10}$$

$$B3 = \sum_{i=1}^{n} (x_i - \sin^2(i-1)\cos^2(i-1)^{\;2} \tag{11}$$

$$B4 = \sum_{i=1}^{n} \left(x_i - \frac{1}{4}(\cos(i-1)\cos(2(i-1)) + 2) \right)^2 \qquad (12)$$

$$B5 = \sum_{i=1}^{n} \left(x_i - \frac{1}{2}(\sin(1000\pi(i/n + 1 \right)^2 \qquad (13)$$

Where n=10, and $x_i \in [0,1]$. These base functions are the modified versions of Fieldsend et al's [22]. This set of test functions are presented below:

$HU1 = \{B1, B2\}$

$HU2 = \{B3, B4\}$

$HU3 = \{B2, B3, B5\}$

$HU4 = \{B1, B4, B5\}$

$HU5 = \{B1, B3, B4, B5\}$

The EA parameter settings such as *Population size*, *Crossover rate* and *Mutation rate*, are the same as before. The SPEA-SQP and NSGA-SQP run both 20 generations and perform the SQP procedure every 5 generations. Record down the number of objective evaluations respectively for the two hybrid MOEAs and then run SPEA and NSGA some generations ensuring that the number of evaluations of SPEA and SPEA-SQP are close, and so are the number of evaluations of NSGA and NSGA-SQP. Such an approach guarantees the fairness to some extent for comparison.

Two metrics are adopted to evaluate the convergence and diversity: C metric for the former and J_1 metric for the latter. C metric is defined as follows [15]:

$$C(A,B) = \frac{|\{b \in B \mid a \in A, a \le b\}|}{|B|} \qquad (14)$$

It calculates the proportion of solutions in B, which are weakly dominated by solutions of A. $C(A,B)$ =1 means all members of B are weakly dominated by A. On the other hand, $C(A,B)$ =0 means no member of B is weakly dominated by A.

The J_1 metric [22]:

$$\mathcal{J}_1(A,B) = \frac{1}{m}\sum_{i=1}^{m} \frac{\max\{|a_i - a_i'|, a_i, a_i' \in A\}}{\max\{|b_i - b_i'|, b_i, b_i' \in B\}} \qquad (15)$$

This metric is an average ratio of the extents of two fronts in each objective dimension, where each objective is normalized by the extent of the front in B. If $\mathcal{J}_1(A,B)$ =1, the two fronts are on average proportionately equal in extent, whereas $\mathcal{J}_1(A,B)$ >1 means that on average the extent of A is greater than the extent of B. We use this metric because that the extent of the front is one of the important factors for evaluating the solutions' diversity and that it is a simple measure, though it does not provide all the information about the diversity.

The comparison results are shown in Table 2 and Table 3 (S and HS represent the non-dominated solutions obtained by SPEA and hybrid SPEA (SPEA-SQP), while N and HN represent the non-dominated solutions by NSGA and hybrid NSGA (NSGA-SQP), respectively). Since neither the C metric nor the J_1 metric is a symmetric operator, it is necessary to calculate all the values: $C(A,B)$, $C(B,A)$, $\mathcal{J}_1(A,B)$ and $\mathcal{J}_1(B,A)$. From Table 2, again, we can see that the solutions obtained by the hybrid SPEA have a better convergence to the Pareto-optimal front than the solutions by the pure SPEA on all the problems except HU5. This superiority of the hybrid NSGA to pure NSGA is demonstrated on the HU1, HU2, and HU3. As far as the average extent of the front in the objective space is concerned (shown in Table 3), the hybrid SPEA outperforms the pure SPEA on all problems; though there is no obvious proof that whether the hybrid NSGA is better than the NSGA or not.

Table 2. C metric values of the algorithms for HU1-HU5

	HU1	HU2	HU3	HU4	HU5
$C(S,HS)$	0	0	0	0	0
$C(HS,S)$	1	0.44	1	0.49	0
$C(N,HN)$	0	0.02	0	0	0
$C(HN,N)$	0.82	0.58	0.34	0.02	0.02

Table 3. $\mathcal{J}_1$ metric values of the algorithms for HU1-HU5

	HU1	HU2	HU3	HU4	HU5
$\mathcal{J}_1(S,HS)$	0.29	0.17	0.13	0.15	0.17
$\mathcal{J}_1(HS,S)$	3.48	5.96	9.93	7.67	8.92
$\mathcal{J}_1(N,HN)$	0.83	1.36	0.87	1.26	1.04
$\mathcal{J}_1(HN,N)$	1.23	0.83	1.18	0.80	1.18

6 Conclusions

In this study, the local search algorithm, sequential quadratic programming, is incorporated into two well-known multi-objective optimization algorithms by means of a modified ε-constraint method. The idea is simple but successful. To summarize the conclusions:

- If there are no local traps in the search space, the newly developed hybrid MOEAs have much faster convergence towards the Pareto front than the original MOEAs, either the objective evaluations or the CPU time is concerned.
- If there are local optima, the hybrid algorithms also have more powerful ability than the original MOEAs to identify the true Pareto-optimal front.
- The hybridization technique does not decrease the solutions' diversity evidently from our simulations. For some problems, it even enhances the extent of the Pareto front.
- The hybridization technique can be applied to other MOEAs besides SPEA and NSGA, and can be easily extended to solve the constraint MOOPs.

However, the local search algorithm adopted here is a gradient-based algorithm, which limits the applications of the hybrid MOEAs. If non-gradient-based algorithms will also do the work and still have so well effects is our next consideration.

Acknowledgements

This work is supported by the Research Subject of Ministry of Education of China (No. 02175). The first author would like to thank Dr. Carlos A. Coello Coello for providing some useful references and reviewing this paper carefully.

Bibliography

1. Schaffer, J. D., *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*. Ph. D. Thesis, 1984.
2. Goldberg, D.E., *Genetic Algorithms for Search, Optimization, and Machine Learning.* Reading, MA: Addison-Wesley, 1989.
3. Fonseca, C. M. and Fleming, P.J., "Genetic Algorithms for Multi-objective Optimization Formulation, Discussion, and Generalization," Forrest, S. (ed), *Proceedings of the Fifth International Conference of Genetic Algorithms,* Morgan Kaufmann, Los Altos, CA, 1993, Morgan Kaufmann Publishers, pp. 416-423.
4. Horn, J., Nafploitis, N. and Goldberg, D., "A Niched Pareto Genetic Algorithm for Multi-objective Optimization," Michalewicz, Z. et al. (eds), *Proceedings of the First IEEE Conference on Evolutionary Computation,* Orlando, Florida. 1994, IEEE Press, pp. 82-87.
5. Srinivas, N. and Deb, K., "Multi-objective Function Optimization using Non-dominated Sorting Genetic Algorithms," *Evolutionary Computation,* 1994, 2(3), pp. 221-248.
6. Zitzler, E. and Thiele, L., *An Evolutionary Algorithm for Multi-objective Optimization: The Strength Pareto Approach*. Technical Report 43, Zürich, Switzerland: Computer Engineering and Network Laboratory, Swiss Federal Institute of Technology.
7. Knowles, J. D. and Corne, D. W., "Approximating the Non-dominated Front using the Pareto Archived Evolution Strategy," *Evolutionary Computation,* 2000, 8(2), pp. 149-172.
8. Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T., *A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II*. Technical Report 200001, Indian Institute of Technology, Kanpur: Kanpur Genetic Algorithms Laboratory, 2000.
9. Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T., "A Fast Elitist Non-dominated Sorting Genetic Algorithm For Multi-objective Optimization: NSGA-II," Schoenauer, M. et al. (eds), *Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI),* Paris, France, 2000, Springer Verlag, pp. 849-858.
10. Corne, D., Knowles, J. and Oates, M., "The Pareto Envelope-based Selection Algorithm for Multi-objective Optimization," Schoenauer, M. et al. (eds), *Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI),* Paris, France, 2000, Springer Verlag, pp. 839-848.
11. Deb, K. and Goel, T., "Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence", Zitzler, et al. (eds), *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*. Zurich, Switzerland, March 2001: Springer-Verlag, 2001, 67-81.
12. Zitzler, E., Laumanns, M. and Thiele, L., *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, TIK-Report 103, May 2001, Swiss Federal Institute of Technology (ETH) Zurich.
13. Brian, J., Ritzel, J., Waylandm E. and Ranjithan, S., "Using Genetic Algorithms to Solve a Multiple objective Groundwater Pollution Containment Problem," *Water Resources Research,* 1994, 30(5), pp. 1589-1603.
14. Ranjithan, S. R., Chetan, S. K. and Dakshima, H. K., "Evaluation of the Constraint Method-Based Evolutionary Algorithm (CMEA) for a Multiobjective Optimization Problem," Zitzler, E. et al. (eds), *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, Zurich, Switzerland, March 2001. Springer-Verlag, pp. 299-313.
15. Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms,* Chichester: John Wiley & Sons, Ltd., 2001.
16. Goldberg, D.E. and Richardson, J., "Genetic Algorithms with Sharing for Multi-modal function optimization," Grefenstette, J. (ed), *Proceedings of the First International Conference on Genetic Algorithms and Their Applications,* Lawrence Erlbaum Associates, 1987, pp. 41-49.
17. Haimes, Y. Y., Lasdon, L.S. and Wismer, D.A., "On a Bicriterion Formulation of the Problems of Intergrated System Identification and System Optimization," *IEEE Transactions on Systems, Man, and Cybernetics.* 1971, 1(3), pp. 296-297.
18. Schittowski, K., "NLQPL: A FORTRAN-Subroutine Solving Constrained Nonlinear Programming Problems," *Annals of Operations Research*, 1985, Vol. 5, pp. 485-500.
19. Fletcher, R., "Practical Methods of Optimization," Vol. 1, *Unconstrained Optimization*, and Vol. 2, *Constrained Optimization*, John Wiley and Sons. 1980.
20. Gill, P. E., Murray W., and Wright M.H., "Practical Optimization," Academic Press, London, 1981.
21. Gen, M., and Cheng, R., *Genetic Algorithms and Engineering Design*. John Wiley & Sons, 1997.
22. Fieldsend, J. E., Everson, R. M. and Singh, S., "Using Unconstrained Elite Archives for Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, 2003, 7(3), pp. 305—323.
23. Zitzler, E., Deb, K. and Thiele, L., "Comparison of multi-objective evolutionary algorithms: Empirical results," *Evolutionary Computation,* 2000, 8(2), 125-148.