A New Multiobjective Evolutionary Algorithm: OMOEA

SanYou Zeng
Dept. of Computer Science, China University of GeoSciences,
430074 Wuhan, Hubei, P.R.China
Dept. of Computer Science, Zhuzhou Institute of Technology,
412008 Zhuzhou, Hunan, P.R.China
sanyou-zeng@263.net
Yuping Chen
Dept. of Computer Science, China University of GeoSciences,
430074 Wuhan, Hubei, P.R.China
kang@whu.edu.cn

LiXin Ding
State Key Laboratory of Software Engineering, Wuhan University
430072 Wuhan, Hubei, P.R.China
yujh@wipm.ac.cn
LiShan Kang
Dept. of Computer Science, China University of GeoSciences,
430074 Wuhan, Hubei, P.R.China
State Key Laboratory of Software Engineering, Wuhan University
430072 Wuhan, Hubei, P.R.China
kang@whu.edu.cn

Abstract- A new algorithm is proposed to solve constrained multi-objective problems in this paper. The constraints of the MOPs are taken account of in determining Pareto dominance. As a result, the feasibility of solutions is not an issue. At the same time, it takes advantage of both the orthogonal design method to search evenly, and the statistical optimal method to speed up the computation. The output of the technique is a large set of solutions with high precision and even distribution. Notably, for an engineering problem WATER, it finds the Pareto-optimal set, which was previously unknown.
Keywords evolutionary algorithms, orthogonal design, multi-objective optimization, Pareto-optimal set

1 Introduction

Evolutionary algorithms can both handle large search spaces and generate multiple alternative trade-offs in a single optimization run for multiobjective optimization problems. Representative evolutionary techniques include vector evaluated genetic algorithm (VEGA)[3, 4], Hajela and Lins genetic algorithm(HLGA)[5], Pareto-based ranking procedure(FFGA)[6], niched Pareto genetic algorithm (NPGA)[7, 8], Nondominated Sorting Genetic Algorithm (NSGA)[9] and a new effective approach to multi-objective optimization, the strength Pareto evolutionary algorithm (SPEA)[10]. Recently, some of the proposed methods have made further progress, for instance, NSGAII[12], SPEA2[13], rMOGAxs[14], and some new MOEAs come about, such as Generalised Regression GA (GRGA)[11].

In this paper, an orthogonal multi-objective evolutionary algorithm (OMOEA) is designed for multi-objective optimization problems (MOPs). The constraints of MOPs is first incorporated in Pareto dominance, to modify Pareto dominance(strict partial ordered relation), so that the feasibility is not an issue. Then the orthogonal design and the statistical optimal method is generalized suitable for MOPs with discrete variables. Applying the generalized design method, a niche evolution procedure is constructed, which is the kernel of OMOEA. OMOEA not only applies the orthogonal array to produce the initial niche-population, but also a statistical optimal method to generate offspring for crossover in the niche evolution procedure. The evolutionary process of the new technique is that an original niche evolves first, and splits into a group of sub-niches; then every sub-niche iterates the above operations. Due to the uniformity of the search, without blindness and randomness, and to the optimality of the statistics, so the niche evolution procedure can quickly yield an evenly distributed niche-population which is very close to the non-dominated set for the niche. At the same time, because of the exponential increase of the splitting frequency of niches, the new algorithm can yield a large set of solutions. Therefore, within a few generations, the OMOEA will quickly yield a large, constraint-satisfying, close-to-Pareto-optimal set with even distribution and high precision.

In the remainder of the paper, we briefly mention the definition of multiobjective optimization problem in Section 2. Thereafter, we describe the constrained Pareto dominance as the improvement of the traditional Pareto dominance in Section 3, in OMOEA, the search of Pareto-optimal set (non-dominated set) is based on constrained Pareto dominance, not traditional Pareto dominance. And in Section 4, we suggest a generalized orthogonal design method for MOPs on the purpose to locate as small region as possible where non-dominated set stays. Section 5 presents niche evolving and splitting which constructs the main components of OMOEA, and Section 6 describes the proposed OMOEA. The next section presents numerical experiments and discussion. Finally, we outline the conclusions of this paper.

2 Problem Definition

Definition 1 *(Multi-objective Optimization Problem(MOP)) A general MOP includes a set of N parameters (decision variables), a set of K objective functions, and a set of L constraints. Objective functions and constraints are functions of the decision variables. The optimization goal is to*

$$\begin{array}{ll} minimize & y = f(x) = (f_1(x), f_2(x), ..., f_K(x)) \\ subject\ to & e(x) = (e_1(x), e_2(x), ..., e_L(x)) \leq 0 \\ where & x = (x_1, x_2, ..., x_N) \in \mathbf{X} \\ & \mathbf{X} = \{(x_1, x_2, ..., x_N) | l_i \leq x_i \leq u_i\} \\ & l = (l_1, l_2, ..., l_N) \\ & u = (u_1, u_2, ..., u_N) \\ & y = (y_1, y_2, ..., y_K) \in \mathbf{Y} \end{array} \quad (1)$$

where x is the decision vector, y is the objective vector, $\mathbf{X}$ denotes the decision space,l and u are the upper bound and lower bound of the decision space, and $\mathbf{Y}$ is called the objective space.

3 Pareto-dominance with Constraints

The Pareto dominance (strict partial ordered relation) is based on the decision space $\mathbf{X}$ without consideration of the constraints, while the Pareto-optimal set $\mathbf{M}(\mathbf{X}_f, \prec)$ is based on the feasible set. Thus it is inconvenient to get the Pareto-optimal set from traditional Pareto dominance, because we care whether an individual is feasible when evaluating its fitness. So the Pareto dominance will be modified. First a concept of constraint objective is introduced.

Definition 2 *Let*

$$\begin{array}{ll} & f_0(\mathbf{x}) = \sum_{j=1}^{L} \beta_j < e_j(\mathbf{x}) >_+ \\ where & < e_j(\mathbf{x}) >_+ = max\{e_j(\mathbf{x}), 0\} \\ & \beta_j > 0 \end{array} \quad (2)$$

$f_0(\mathbf{x})$ is called constraint objective.

Then a new relation "$\prec$" is defined in consideration of both objectives and constraints as follows.

Definition 3 *(Constrained Pareto dominance) For any two decision vectors, $\mathbf{a}$ and $\mathbf{b}$*

$$\begin{array}{ll} \mathbf{a} \prec \mathbf{b}\ iff & \left\{ \begin{array}{lll} f_0(\mathbf{a}) < f_0(\mathbf{b}) & or & \\ f_0(\mathbf{a}) = f_0(\mathbf{b}) & and & \mathbf{f}(\mathbf{a}) < \mathbf{f}(\mathbf{b}) \end{array} \right\} \\ where & \\ & \mathbf{a}, \mathbf{b} \in \mathbf{X} \end{array} \quad (3)$$

In definition 3, the constraint objective has been added to the objectives, but the constraint objective has the highest priority. By the constrained Pareto dominance "$\prec$", OMOEA will search the Pareto-optimal set $\mathbf{M}(\mathbf{X}, \prec)$ from decision space $\mathbf{X}$ without worrying about the feasibility of solutions.

Remark: There are some similar constraint handling ideas ,(cf. Fonseca 1998[17] and Deb 2000[12]).

4 Generalization of Orthogonal Design Method

In a discrete single objective optimization problem, when there are N factors (variables) and each factor has Q levels, the search space consists of Q^N combinations of levels. When N and Q are large, it may not be possible to do all Q^N experiments to obtain optimal solutions. Therefore, it is desirable to sample a small, but representative set of combinations for experimentation, and based on the sample, the optimal may be estimated. The orthogonal design was developed for the purpose [15]. The selected combinations are scattered uniformly over the space of all possible combinations Q^N. In this paper a kind of orthogonal array is employed, we denote it $[a_{i,j}]_{M\times P}$ where $M = Q^2$, $P = Q + 1$; and where the jth factor in the ith combination has level $a_{i,j}$ and $a_{i,j} \in \{1, 2, ..., Q\}$. We denote the jth column of the orthogonal array $[a_{i,j}]_{M\times P}$ by $\mathbf{a}_j$. The details to construct the orthogonal array $[a_{i,j}]_{M\times P}$ are as follows [16].

Algorithm 1 *Construction of orthogonal array $[a_{i,j}]_{M\times P}$ where $M = Q^2$, $P = Q + 1$*

`for(i=1; i`$\leq Q^2$`; i++)`$a_{i,1} = \lfloor \frac{i-1}{Q} \rfloor\ mod\ Q$;

`for(i=1; i`$\leq Q^2$`; i++)`$a_{i,2} = (i-1)\ mod\ Q$;

`for(t=1; t `$\leq$`Q-1; t++)` $a_{2+t} = (a_1 \times t + a_2)\ mod\ Q$;

`Increment` $a_{i,j}$ `by one for` $1 \leq i \leq M\ and\ 1 \leq j \leq P$;

For a problem with N factors, we choose the former N columns of orthogonal array $[a_{i,j}]_{M\times P}$, and denote it $[a_{i,j}]_{M\times N}$, Denote the corresponding yields of the M combinations by $[y_i]_{M\times 1}$, where the ith combination (experiment) has yield y_i. From the yields of the selected combinations, we can get optimal solution by statistical methods. That is , calculate the mean value of the yield for each factor at each level, and evaluate the effect of each factor at each level by the mean value. We choose the best combination, which is composed of the best levels of the N factors, as the optimal solution. Denote the mean values by $[\Delta_{k,j}]_{Q\times N}$ where the objective has the mean value $\Delta_{k,j}$ at the kth level of the jth factor; and

$$\Delta_{k,j} = \frac{Q}{M} \sum_{a_{i,j}=k} y_i \quad (4)$$

where the jth factor has the level $a_{i,j}$ in the ith combination(experiment). The objective has the value y_i at the ith combination, and $\sum_{a_{i,j}=k} y_i$ implies the sum of y_i where $\forall i$ which satisfy $a_{i,j} = k$. In the jth column of $[\Delta_{k,j}]_{Q\times N}$, i.e., $\{\Delta_{k,j}|k = 1, 2, ..., Q\}$, suppose the index k of the min element in the column is $k_j^{(0)}$, for a minimization problem, the optimal solution is estimated as $(k_1^{(0)}, k_2^{(0)}, ..., k_N^{(0)})$.

The orthogonal design method was designed for single-objective optimization problems(**SOPs**). Here it is generalized for MOPs to estimate non-dominated set. The main difference between SOPs and MOPs is that SOPs have only simple-objective while MOPs have more objectives than one. In this section, the objective vector is denoted by $\mathbf{y} = (y_1, y_2, ..., y_K)$, and the decision vector (factors) by $(x_1, x_2, ..., x_N)$, where the factor x_j is quantized into a level set $L_j = \{x_{1,j}, x_{2,j}, , ..., x_{Q,j}\}, j = 1, 2, ..., N$. $\mathbf{D}$ denotes the set of all possible combinations of levels. Statistically, if Q is large enough, then the discrete non-dominated set regarding $\mathbf{D}$ will be very close to the continuous non-dominated set regarding $\mathbf{X}$. So when Q is large enough, we may regard the discrete one as a satisfactory approximation to the continuous one. Applying the orthogonal design method, there is a best level for each factor in SOPs. However, in MOPs, this can not be concluded since there are more objectives than one. But there may be a non-dominated set relative to the level set for each factor.

To get the non-dominated set relative to the level set for each factor, denote the values of objectives by $[y_{i,k}]_{M\times K}$, where the kth objective in the ith combination has the value $y_{i,k}$; and the mean values of the K objectives at the Q levels of the N factors by $[\Delta_{q,j,k}]_{Q\times N\times K}$, where the kth objective at the qth level of the jth factor has mean value $\Delta_{q,j,k}$. Denote

$$\mathbf{\Delta}_{q,j} = (\Delta_{q,j,1}, \Delta_{q,j,2}, ..., \Delta_{q,j,K}) \tag{5}$$

Now we define dominance relation "$\prec_j$" on the level set for the jth factor, j=1, 2, ..., N.

Definition 4 *for any two levels $x_{u,j}, x_{v,j} \in L_j$ of the jth factor, j=1, 2, ..., N.*

$$\begin{array}{ll} x_{u,j} \prec_j x_{v,j} & iff\ \mathbf{\Delta}_{u,j} < \mathbf{\Delta}_{v,j} \\ & \mathbf{\Delta}_{u,j}\ and\ \mathbf{\Delta}_{v,j}\ are\ defined\ in\ (5) \end{array} \tag{6}$$

It is easy to verify that "$\prec_j$" is a strict partial ordered relation on L_j. $\mathbf{M}(L_\mathbf{j}, \prec_\mathbf{j})$ denotes the non-dominated set regarding L_j. Statistically, we have the following conclusion:

The non-dominated set of the Cartesian product of the N non-dominated sets relatively evenly distributes over the non-dominated set relative to $\mathbf{D}$, and the size of the Cartesian product is probably much smaller than that of $\mathbf{D}$.

5 Niche Evolving and Splitting

The concept of niche here means following.

Definition 5 *A niche is a hyper-rectangle $\mathbf{X}_n$ in the decision space $\mathbf{X}$ of MOP, that is, a hyper-rectangle $\mathbf{X}_n \subseteq \mathbf{X}$*

Since OMOEA is mainly the iteration of niche evolving and splitting, we let a section (Section 5) state Niche Evolving and Splitting. Niche evolution consists of producing initial niche-population (cf. subsection 5.1) evenly scattered over the niche by using orthogonal array, executing crossover operator (cf. subsection 5.2) with the initial niche-population as the parents, where a generalized orthogonal design method and statistical optimal method is used, including the offspring into niche-population, from the niche-population, searching the non-dominated set which is the output of the niche evolution procedure.

5.1 Production of the Initial Niche-population

An orthogonal array $[a_{i,j}]_{M\times N}$ actually corresponds to a population with size M, so that $[a_{i,j}]_{M\times N}$ determines a population, then let $[a_{i,j}]_{M\times N}$ represent a population. The production of a uniformly scattered niche-population over a niche is as follows

Algorithm 2 *Construction of initial niche-population*

1. *Execute algorithm 1 to construct orthogonal array $[a_{i,j}]_{Q^2\times(Q+1)}$;*
2. *Delete the last $Q + 1 - N$ columns of $[a_{i,j}]_{Q^2\times(Q+1)}$ to get $[a_{i,j}]_{Q^2\times N}$.*
3. *Get initial niche-population $\mathbf{P}_n(0)$: $\mathbf{P}_n(0) \leftarrow [a_{i,j}]_{Q^2\times N}$*

5.2 Crossover Operator

In most evolutionary algorithms, the parents are stochastically chosen from the population. Here, in the evolution of the niche, all individuals in the population are picked as parents for the crossover operator. The details are as follows

Algorithm 3 *Crossover operator*

1. *Input initial niche-population $\mathbf{P}_n(0)$ as parents.*
2. *Using $\mathbf{P}_n(0)(i.e.\ [a_{i,j}]_{Q^2\times N}))$, construct three dimensional array $[\Delta_{q,j,k}]_{Q\times N\times K}$*
3. *For each Factor j, j=1, 2, ..., N, construct non-dominated set $\mathbf{M}(L_j, \prec_j)$*
4. *Construct the Cartesian product*

   $$\mathbf{M}(L_1, \prec_1) \times \mathbf{M}(L_2, \prec_2) \times ... \times \mathbf{M}(L_N, \prec_N)$$

5. *Get offspring $\mathbf{S}$: $\mathbf{S} \leftarrow \mathbf{M}(L_1, \prec_1) \times \mathbf{M}(L_2, \prec_2) \times ... \times \mathbf{M}(L_N, \prec_N)$*

5.3 Procedure of Niche Evolution

Denote $\mathbf{P}_n(0)' = \mathbf{P}_n(0) \bigcup \mathbf{S}$. Statistically, if Q is large enough, the new technique searches the non-dominated set relative to $\mathbf{P}_n(0)'$ and regard it as representative of the non-dominated set relative to $\mathbf{X}_n$. Now, we describe the algorithm for evolving niche $\mathbf{X}_n$.

Algorithm 4 *Evolution of niche*

1. *Initiate paramter Q*
2. *Execute algorithm 2 to produce initial niche-population: $\mathbf{P}_n(0)$*
3. *Execute crossover operator (algorithm 3) to breed offspring $\mathbf{S}$*
4. *Unite initial niche-population and offspring: $\mathbf{P}_n(0)' = \mathbf{P}_n(0) \bigcup \mathbf{S}$*
5. *Construct non-dominated set $\mathbf{M}(\mathbf{P}_n(0)', \prec)$ from $\mathbf{P}_n(0)'$ as the output niche-population $\mathbf{P}_n(1)$. This is representative of the non-dominated set relative to niche $\mathbf{X}_n$*

5.4 Procedure for Splitting Niche

If the precision of the output niche-population $\mathbf{P}_n(1)$ is not satisfactory, then the niche $\mathbf{X}_n$ splits into sub-niches, where each individual in $\mathbf{P}_n(1)$ has corresponding sub-niche, the range length of each variable (factor) in each sub-niche is the difference between two successive levels of the factor. The procedure for splitting niche $\mathbf{X}_n$ from $\mathbf{P}_n(1)$ is as follows.

Algorithm 5 *Niche splitting*

1. *For each $\mathbf{s} \in \mathbf{P}_n(1)$, construct a new niche*

$$\begin{array}{ll} \mathbf{X}_n^{(\mathbf{s})} = & \{\mathbf{x} | \, |x_j - s_j| \leq \delta_j/2 \; j = 1, 2, ..., N\} \\ where & \mathbf{x} = (x_1, x_2, ..., x_N) \\ & \mathbf{s} = (s_1, s_2, ..., s_N) \end{array}$$

   where δ_j is the difference of two successive levels.

2. *Get a group of sub-niches Ψ_n: $\Psi_n \leftarrow \{\mathbf{X}_n^{(\mathbf{s})} | \mathbf{s} \in \mathbf{P}_n(1)\}$*

6 The Orthogonal Multiobjective Evolutionary Algorithm

The evolutionary process of OMOEA is that an original niche (the decision space $\mathbf{X}$) evolves first (cf. subsection 5.3), and splits into a group of sub-niches (cf. subsection 5.4); then every sub-niche iterates the above operations,until stopping criterion is satisfied.

Let $\mathbf{P}$ denote the global-population, Ψ the set of all sub-niches. After each evolution generation, $\mathbf{P}$ and Ψ will be updated. The details of the OMEA are as follows.

Algorithm 6 *orthogonal multi-objective evolutionary algorithm*

1. *Input decision space $\mathbf{X}$ as initial niche.*
2. *Execute algorithm 4 to let the niche evolve into $\mathbf{P}_n(1)$*
3. *For the output $\mathbf{P_n}(1)$ of initial niche $\mathbf{X}$, execute algorithm 5 to split the niche into a group Ψ_n of niches.*
4. *Initiate $\mathbf{P}$ and Ψ: $\mathbf{P} \leftarrow \mathbf{P}_n(1)$; $\Psi \leftarrow \Psi_n$.*
5. *gen=1*
6. *If current $\mathbf{P}$ does not reach the required precision, and the solution number of $\mathbf{P}$ is not more than a critical value, then goto step 7, else goto step 11.*
7. *For each niche $\mathbf{X}_n^{(\mathbf{s})} \in \Psi$, execute algorithm 4 to let niches evolve and yield $\mathbf{P}_n^{(\mathbf{s})}(1)$*
8. *For the output $\mathbf{P}_n^{(\mathbf{s})}(1)$ of each niche $\mathbf{X}_n^{(\mathbf{s})}$, execute algorithm 5 to split the niche into a group of niches Ψ_n.*
9. *$\Psi \leftarrow \bigcup \Psi_n$; $\mathbf{P}$: $\mathbf{P} \leftarrow \bigcup \mathbf{P}_n^{(\mathbf{s})}(1)$*
10. *gen=gen+1; goto step 6*
11. *Output $\mathbf{P}$ as the satisfying close-to-Pareto-optimal set of MOP*

7 Numerical Experiments and Discussion

7.1 Test functions

Five MOPs are taken to test OMOEA. The first four are the benchmark problems: ZDT_1, ZDT_2, ZDT_3 and ZDT_4 [19] with a little modified. The final one is W [18] which is an engineering problem with constraints, and its Pareto-optimal set is unknown. The OMOEA will yield high precision solutions when level point stays at Pareto-optimal point, in this, the performance of OMOEA would'nt be well tested. Therefore, we modify the first four problems by randomly changing a little both bounds of variables, and denote the modified ones by $ZDT_1', ZDT_2', ZDT_3', ZDT_4'$.

7.2 Results and discussion

7.2.1 Generation

Since the crossover operator is very efficient in the niche evolution, fewer evolution generations of the algorithm will yield excellent results. OMOEA has to locate the global Pareto-optimal solutions roughly and not trapped in local Pareto-optimal in the first generation. In the rest generations, it enhances the precision of the solutions. Figure 1 shows the fact. There are two generations taken, in the first

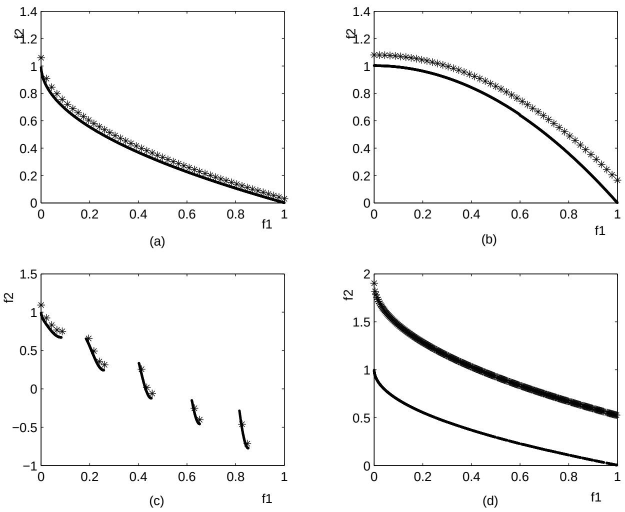

Figure 1: Result comparisons on the first four problems between the first two generations. (a) For ZDT_1'. (b) For ZDT_2'. (c) ForZDT_3'. (d) For ZDT_4'.

generation where the numbers of level $Q_1 = 41$ for ZDT_1', ZDT_2' and ZDT_3', $Q_1 = 263$ for ZDT_4' to locate global Pareto-optimal roughly, in the second where $Q_2 = 29$ for all to enhance precision. Figure 1 also distinctly illustrates that the populations of the generation 1 reach higher precision for the problems ZDT_1', ZDT_2', ZDT_3', while for ZDT_4', this reaches lower and the second generation largely improves the precision of the populations. Generally, we recommend evolving two generations, in the first generation, a larger Q is taken so as to roughly locate the global Pareto-optimal solutions, in the second, a smaller Q will ensure improving the precision of solutions and avoid large computation. For the problem W, OMOEA only uses one generation where $Q_1 = 89$, since the number of solutions in the close-to-Pareto-optimal set we obtain is very large, if we continued to improve the close-to-Pareto-optimal set, the computation would be virtually endless. Therefore, once the number of solutions in the close-to-Pareto-optimal set of MOPs exceeds a given maximum, the algorithm will return with the current close-to-Pareto-optimal set of MOPs as output. Figure 2 shows the resulting Pareto-optimal set of W and its the projections on the planes $x_1 - x_2$, $x_1 - x_3$ and $x_2 - x_3$.

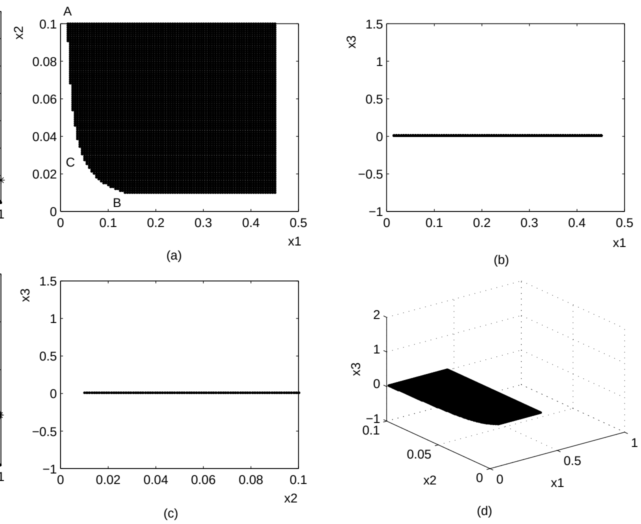

Figure 2: The graph of the close-to-Pareto-optimal set of $\mathbf{W}$ and its projections of on the three planes. (a)The projection on the plane $x_1 - x_2$. (b) The projection on $x_1 - x_3$. (c) The projection on $x_2 - x_3$. (d)The graph of the close-to-Pareto-optimal set

7.2.2 Quantization

By quantization, a continuous niche yields discrete one. Statistically, a large number of quantization level Q provides a large number of sample for a good summing and helps not trapping in local Pareto-optimal, and if Q is large enough, then the non-dominated set regarding the discrete niche will be very close to that regarding the continuous one. So when Q is large enough, we regard the discrete non-dominated set as a satisfactory approximation to the continuous non-dominated set. At the same time, experiment results show that a problem with large number of local Pareto-optimal fronts needs a large Q ,while one with fewer local Pareto-optimal a smaller Q enough, to locate global Pareto-optimal solutions in the first generation. For example, for ZDT_1', ZDT_2' and ZDT_3', OMOEA can locate the global Pareto-optimal solutions with $Q = 31$, however, for ZDT_4', our technique is likely trapped in local Pareto-optimal until $Q \geq 211$. Figure 3 show the resulting solutions of ZDT_4' with different Q in the first generation, where in the second, $Q = 29$. Theoretically, that how large Q is selected so as not to be trapped in local Pareto-optimal for OMOEA is not very clear. For ZDT_4', the Pareto-optimal front is formed with $g(\mathbf{x}) = 1$ where $x_2 = 0, ..., X_{10} = 0$, the best local Pareto-optimal fronts with $g(\mathbf{x}) = 1.25$. We know $g(\mathbf{x}) \approx 1.25$ at $x_2 = \pm 0.008, ..., x_{10} = \pm 0.008$, in this, the difference of two successive levels should be less

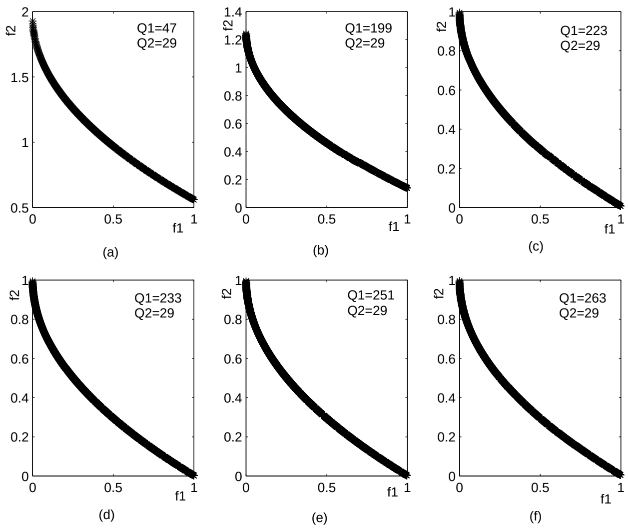

Figure 3: the resulting solutions of ZDT_4' with different Q_1 in the first generation, while $Q_2 = 29$ in the second generation. (a)$Q_1 = 47$, and results trapped in local optimal. (b)$Q_1 = 199$, results trapped in local optimal. (c)$Q_1 = 223$, and results approximate global optimal. (d)$Q_1 = 233$, and results approximate global optimal. (e)$Q_1 = 251$, and results approximate global optimal. (f)$Q_1 = 263$, and results approximate global optimal.

than 0.016, Therefore, $Q > 650$ will ensure OMOEA not trapped in local Pareto-optimal. However, none is trapped in local Pareto-optimal with $Q = 263$ in the first generation among 100 runs performed on ZDT_4'. Therefore, for a problem with large local Pareto-optimal, a large Q, unnecessary much large, should be taken to locate global Pareto-optimal in the first generation.

On the other hand, the number of the quantization levels Q should be prime, since a prime Q is prerequisite to ensure the orthogonality of the array constructed by algorithm 1, and therefore, to guarantee uniform search. Theoretically, whether there exists orthogonal array for any integer $Q > 0$ is yet unknown. So a prime Q is recommended.

7.2.3 Comparison

We downloaded the kernel of the SPEA, and programmed SPEA by ourselves which is likely poorer than Zitzler's SPEA, and find OMOEA runs much faster than SPEA made by us for the chosen test problems except W. In some way, we may conclude that OMOEA converges to Pareto-optimal solutions faster than SPEA for the chosen problems except W, though OMOEA expends much time to get huge non-dominated solutions for W. We chose the results of the SPEA in solving the test functions $\mathbf{ZDT}_1 - \mathbf{ZDT}_4$ as a comparison with those of OMOEA. In our algorithm the taken parameters $Q_1 = 41$, $Q_2 = 29$ for ZDT_1', ZDT_2' and ZDT_3', $Q_1 = 263$, $Q_2 = 29$ for ZDT_4'. The results are shown in figure 4, and assessed by both coverage [10] and spread [1] (PP314-316) in tables 1, 2. Both the figures and the tables show that not only the quality, but also the spread of the optimal solutions, from our algorithm is superior to SPEA. In particular, since $\mathbf{ZDT}_4$ has 21^9 local Pareto-optimal fronts, previously known algorithms including VEGA , HLGA , FFGA, NPGA, NSGA, rMOGAxs and SPEA are all trapped in local Pareto-optimal solutions. As to OMOEA, when the number of quantization level is smaller than 200, our algorithm is usually also trapped in local Pareto-optimal solutions unless the level points stay at the global Pareto-optimal points by chance. Once the number of quantization levels is larger than 200, OMOEA eliminates local Pareto-optimal solutions and approximates global Pareto-optimal solutions (cf. Figure 3). Regarding function W, the convergence of OMOEA to close-to-Pareto-optimal solutions is shown in figures 2. The right-bottom graph is the three dimensional graph of the close-to-Pareto-optimal set, and other three graphs in figure 2 are the projections of the graph on the planes $x_1 - x_2$, $x_1 - x_3$ and $x_2 - x_3$. We find that the third coordinate x_3 of our close-to-Pareto-optimal solutions is 0.01. The left-top graph shows that the projections of the graph on the x_1x_2-plane is a rectangle: $0.01 \le x_1 \le 0.45$, $0.01 \le x_2 \le 0.1$ with the left-bottom corner cut off by a curve ACB. After checking the resulting value of the constraint functions at our close-to-Pareto-optimal solutions, we find that only the first constraint boundary $g_1 = 1$ has nearby close-to-Pareto-optimal solutions. We infer that the curve ACB is an approximation to a portion of the first constraint boundary $g_1 = 1$. In fact, the curve ACB is very close to the first constraint boundary $g_1 = 1$. So may say that the Pareto-optimal set of $\mathbf{W}$ is the region enclosed by

$$\begin{array}{ll} x_1 = 0.01 & x_1 = 0.45 \\ x_2 = 0.01 & x_2 = 0.10 \\ x_3 = 0.01 & \\ \multicolumn{2}{l}{0.00139/(x_1x_2) + 4.94 \times 0.01 - 0.08 = 1} \end{array} \tag{7}$$

Table 1: The $\mathcal{C}$ values of resulting solutions between SPEA and OMOEA found for the former four test functions.

Coverage	ZDT_1'	ZDT_2'	ZDT_3'	ZDT_4'
$\mathcal{C}$(OMOEA,SPEA)	1	1	1	1
$\mathcal{C}$(SPEA,OMOEA)	0	0	0	0

Table 2: The Δ values of resulting solutions of both SPEA and OMOEA found for the test functions.

Spread	ZDT_1'	ZDT_2'	ZDT_3'	ZDT_4'
Δ_{OMOEA}	0.002	0.004	0.004	0.010
Δ_{SPEA}	0.055	0.198	0.079	0.618

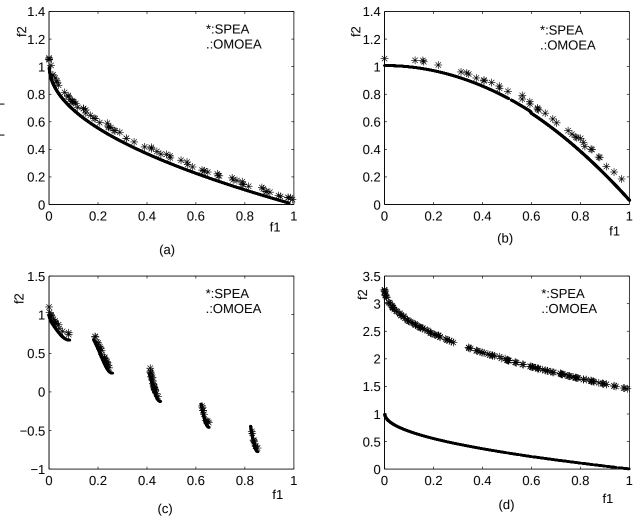

Figure 4: Result comparison between SPEA and OMOEA on test function $\mathbf{ZDT}_1', \mathbf{ZDT}_2', \mathbf{ZDT}_3', \mathbf{ZDT}_4'$ with SPEA. (a) For $\mathbf{ZDT}_1'$. (b) For $\mathbf{ZDT}_2'$. (c) For $\mathbf{ZDT}_3'$. (d) For $\mathbf{ZDT}_4'$.

Generally, the type of the optimal solutions is dot for one objective problems, line for two, surface for three, and so on. Such information indicates that cardinality of optimal solutions exponentially increases with the increase of number of the objectives in general. It seems difficult to represent a high dimensional Pareto-optimal set by finite optimal solutions; and it is the difficulty that causes OMOEA's exponential computation for high dimensional Pareto-optimal set. A wise way to overcome the difficulty to an extent might be to model the structure of the Pareto-optimal set.

7.2.4 Limitation

Since the technique is absolutely new, there exists some limitations. If the model is additive and quadratic, the optimal statistics in crossover operator will be very valid, however, if not, then the statistics may introduce error which may cause that a fewer non-dominated levels are eliminated from non-dominated level set(cf. Definition 4) and some dominated levels near a non-dominated level are interlarded into the non-dominated level set. Although it is both impossible and unnecessary to pick up all non-dominated solutions, a lost non-dominated solution may destroy a little the diversity of the solutions. A way not losing non-dominated levels is to modify definition 4 with consideration of noise [20]. A false non-dominated solutions will increase the computations for eliminating it. At the same time, although the Cartesian product of the non-dominated level sets is excellently good, and the size is probably much smaller than that of the corresponding discrete niche, it will be large when Pareto-optimal set is high dimensional, which, in addition to the false non-dominated solutions, will result in exponential computation complexity. A possible way to overcome this limitation is to execute orthogonal search on the Cartesian product and decrease the exponential computation into polynomial in the future work. Therefore, with overcoming the limitations, the 'linked' problems would be easy to solve.

8 Conclusion

The constraints of the MOPs are take account of in determining Pareto dominance, as a result, the feasibility of solutions is not an issue. The orthogonal design and the statistical optimal method are generalized to MOPs, and the generalized design method is applied to construct a new framework for multi-objective evolutionary algorithm (MOEA). Due to the uniformity of the search, optimality of the statistics, OMOEA yields large set of solutions with high precision and even distribution. Notably, for the engineering problem WATER, it finds the Pareto-optimal set, which was previously unknown.

Acknowledgment This work was supported by National Natural Science Foundation of China (No.s: 60204001, 70071042, 60073043, 60133010), A Project Supported by Scientific Research Fund of Hunan Provincial Education Department(No. 02C640), and Youth Chengguang Project of Science and Technology of Wuhan City(No.:20025001002). We thank Bob McKay for assistance with the form of expression of this paper.

Bibliography

[1] Deb, K.(2001). Multi-objective optimization using evolutionary algorithms. Chichester, UK: Wiley

[2] Horn, J. (1997). F1.9 multicriteria decision making. In T. Bäck,D.B.Fogel, and Z. Michalewicz (Eds.),*Handbook of Evolutionary Computation*. Bristol (UK): Institute of Physics Publishing.

[3] Schaffer, J. D. (1984).*Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. Ph. D. thesis, Vanderbilt University. Unpublished.

[4] Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette (Ed.),*Proceedings of an International Conference on Genetic Algorithms and Their Applications*, Pitts-burgh, PA, pp. 93C100. sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).

[5] Hajela, P. and C. Y. Lin (1992). Genetic search strategies in multicriterion optimal design. Structural Optimization 4, 99C107.

[6] Fonseca, C. M. and P. J. Fleming (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo, California, pp. 416C423. Morgan Kaufmann.

[7] Horn, J. and N. Nafpliotis (1993). Multiobjective optimization using the niched pareto genetic algorithm. IlliGAL Report 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Champaign.

[8] Horn, J., Nafpliotis, N. and Goldberg, D. E. (1994). A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation*, Volume 1, Piscataway, NJ, pp. 82C87. IEEE.

[9] Srinivas, N. and Deb, K. (1994). Multiobjective optimization using non-dominated sorting in genetic algorithms. Evolutionary Computation 2(3), 221C248.

[10] Zitzler, E.(1999). Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph. D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland. TIK-Schriftenreihe Nr. 30, Diss ETH No. 13398, Shaker Verlag, Aachen, Germany.

[11] Tiwari, A., Roy, R.(2002). Generalised Regression GA for Handling Inseparable Function Interaction: Algorithm and Applications. Proceedings of the seventh international conference on parallel problem solving from nature. (PPSN VII). Granada, Spain

[12] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan(2001). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II, IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182–197

[13] Zitzler, E., Laumanns, M. and Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-Report 103. ETH Zentrum, Gloriastrasse 35, CH-8092 Zurich, Switxerland.

[14] Purshouse, R. C., Fleming, P. J. (2001). The Multiobjective Genetic Algorithm Applied to Benchmark Problems–an Analysis. Research Report No. 796. Department of Automatic Control and Systems Engineering University of Sheffield, Sheffield, S1 3JD, UK.

[15] Montgomery, D. C.(1991). Design and Analysis of Experiments. 3rd ed. New York: Wiley.

[16] XiWen Ma. (1981) Mathmetical theory for orthogonal design. the Peaple's Education Publisher: Beijing, China.(In chinese)

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

[18] Ray, T., TRai, K. and Seow, K. C. (2001). An evolutioanry algorithm for multiobjecvtive optimization. Engineering Optimization 33(3), 399-424.

[19] Zitzler, E., Deb, K. and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: empirical results. Evolutionary Computation 8(2): 173-195.

[20] Rudolph, G. (2001). A Partial Order Approach to Noisy Fitness Functions. Proceedings of the 2001 Congress on Evolutionary Com-putation (CEC 2001), pp. 318C325. Seoul Korea.