

A NEW METHOD TO CONSTRUCT THE NON-DOMINATED SET IN MULTI-OBJECTIVE GENETIC ALGORITHMS

Jinhua Zheng ¹⁾, Zhongzhi Shi ²⁾, Charles X. Ling³⁾ and Yong Xie ¹⁾

1)College of Information Engineering, Xiangtan University, Hunan, China (411105)

2) Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China (100080)

3) Department of Computer Science, University of Western Ontario, London, Canada (N6A 5B7)

jhzheng@xtu.edu.cn shizz@ics.ict.ac.cn cling@csd.uwo.ca

Abstract: There have been widespread applications for Multi Objective Genetic Algorithm (MOGA) on highly complicated optimization tasks in discontinuous, multi-modal, and noisy domains. Because the convergence of MOGA can be reached with the non-dominated set approximating the Pareto Optimal front, it is very important to construct the non-dominated set of MOGA efficiently. This paper proposes a new method called Dealer's Principle to construct non-dominated sets of MOGA, and the time complexity is analyzed. Then we design a new MOGA with the Dealer's Principle and a clustering algorithm based on the core distance of clusters to keep the diversity of solutions. We show that our algorithm is more efficient than the previous algorithms, and that it produces a wide variety of solutions. We also discuss the convergence and the diversity of our MOGA in experiments with benchmark optimization problems of three objectives.

Key words: Multi-objective Genetic Algorithm, Multi-objective Optimum, Dominated relationship, Non-dominated set

1. INTRODUCTION

The Multi Objective Genetic Algorithm (MOGA), which is one of machine learning algorithms, is praised for its ability to solve high complex problems. Moreover, the real world tasks always involve simultaneous

optimization of multiple objectives, and MOGA can find a wide spread of Pareto optimal solutions in a single simulation run. Hence, MOGA is getting very popularity in the recent years.

It is known that genetic algorithms can be used to solve single objective optimization, and the best solution is usually the global minimum or the global maximum in this case. However, multi objective optimization is very different from the optimization with only a single objective. There may not exist one solution that is best with respect to all objectives in multi objective optimization. There always exists a set of solutions that are superior to the rest of solutions in the search space when all objectives are considered, but are inferior to other solutions in one or more objectives. These solutions are called Pareto optimal solutions or non-dominated solutions. The rest of the solutions are known as dominated solutions. In MOGAs, the key problem is to construct a set of candidate solutions that are non-dominated for an evolutionary population. While the construction procedure is repeated, the set of candidate solutions is made go closer to the true Pareto optimal solutions continually, and reach the true Pareto optimal solutions in the end.

There exist several kinds of MOGAs with different approaches of constructing the non-dominated set. In this paper, we propose a new method called Dealer's Principle to construct non-dominated sets of MOGA (in section 3), and the time complexity of the new algorithm is analyzed (in section 3), which indicates that it is more efficient to use Dealer's Principle to construct a non-dominated set than the previous methods. To keep the diversity of the solutions, we discuss a clustering algorithm based on the core distance of clusters (in section 4). Then we design a new MOGA based on the clustering procedure, in which the Dealer's Principle is used to construct the non-dominated set (in section 5). We discuss the convergence and the diversity of MOGA, and we test our algorithm in the experiment with a benchmark optimization problem of three objectives (in section 6). It is shown that our algorithm is more efficient than previous algorithms, and that it produces a wide spread of solutions (in section 6).

2. A BRIEFLY REVIEW OF MOGAS

The popular MOGAs include Aggregating Function, Schaffer's VEGA, Fonseca and Fleming's MOGA, Horn and Nafpliotis's NPGA, Zitzler and Thiele's SPEA, and Srinivas and Deb's NSGA.

In Aggregating Function, multi objectives are fit together linearly into a single objective, and each sub-objective is assigned a coefficient. Thus, the optimization of multi objectives is transformed into the optimization of a

single objective [1]. The main merit of this method is of high performing efficiency, but its disadvantages are also distinct. For example, the coefficient of each sub-objective must be changed continually in the evolution process, and it is difficult to find the Pareto optimal solutions no matter how to change the coefficients when the search space is concave [2].

In 1984, Schaffer proposed Vector Evaluated Genetic Algorithm (VEGA) in his doctor's dissertation. It seems that only the extreme points in Pareto optimal front can be found when applying VEGA to solve some problems [3], because it cannot be compromised in light of the attributes of each sub-objective. The VEGA has been improved by Ritzel, Wayland [2] and Surry [4], but it is difficult to find the Pareto optimal solutions when the search space is concave [5].

Fonseca and Fleming suggested another kind of MOGA [6]. Each individual is given a rank calculated respectively. The rank of all non-dominated individuals is 1, and the rank of the dominated individual is 1 plus the number of individuals that dominate it. The selection operation is implemented with the mechanism of the fitness function sharing. The main merits of the method are highly efficient and easy to implement [7]. Similar to the other MOGA with parameters, one of its disadvantages is that the method depends upon the choice of the parameters excessively. The other disadvantage is the premature convergence because of the selection pressure [8]. Moreover, the multi Pareto Optimal points cannot be found when different points correspond to the same function value [9].

Horn and Nafpliotis brought forward A Niche Pareto Genetic Algorithm for Multi objective Optimization (NPGA) based on Pareto dominated relationship [10,11]. At first, a comparison set called CS is selected randomly from the population, and then two individuals are also selected randomly. If one of the two individuals is dominated by CS, then the other is selected into the next evolutionary population; otherwise one of the two individuals is selected by the Niche shared methods into the future evolution [11,12]. The main merit of NPGA is highly efficient, and the Pareto Optimal front can be obtained by this method. Its disadvantage is that there is no general principle to select the sharing parameters and the size of comparison set.

In 1999, Zitzler and Thiele described Strength Pareto Evolutionary Algorithm (SPEA) [13]. The fitness of individual is called the strength. The strength of individuals in the non-dominated set is defined as the ratio of the individuals dominated by it over the population size. The strength of the others (dominated individual) is defined as 1 plus the number of individuals dominating it. Here, the low strength individuals have a high probability of

reproduction. The time complexity is $O(n^3)$. Zitzler [14] improved SPEA, and advanced it to SPEA2 in which the time complexity is $O(n^2 \log n)$.

The Non-dominated Sorting in Genetic Algorithm (NSGA [15,16,17]) was proposed by Srinivas and Deb. First, for each individual two entities are calculated: (1) n_i , the number of individuals which dominate the individual i , and (2) s_i , a set of individuals which the individual i dominates. The individual i is non-dominated if $n_i = 0$; otherwise individual i is dominated. The non-dominated sorting procedure of NSGA has a computational complexity of $O(n^2)$.

The main merits of SPEA2 [14] and NSGA-II [17] are good performances in convergence and diversity. Moreover, MOGAs with no parameter (such as SPEA and NSGA) do not have the problems of selecting and adjusting parameters, so they have widespread applications. But their disadvantage is lower performing efficiency than MOGAs with parameters. To make MOGAs with no parameters more efficient, we propose a new MOGA in which the non-dominated set is constructed by using a new method called Dealer's Principle in this paper.

3. DEALER'S PRINCIPLE TO CONSTRUCT A NON-DOMINATED SET

In the MOGAs based on Pareto optimum, it is very important to construct a non-dominated set efficiently because the speed of the construction affects convergent speed of an algorithm directly. Because the new method has no backtracking when constructing non-dominated sets, a new non-dominated individual needs not to be compared with non-dominated individuals that already exist. Firstly, a dealer, a reference individual, is selected from the candidate solutions in each round of comparison (the dealer is usually the first individual in the current evolutionary population). Then the dealer is compared with the other members of the current candidate solutions one by one. The individual dominated by the dealer must be removed immediately. After a round of comparison, the dealer is joined into the non-dominated set if it is not dominated by any other individuals; otherwise the dealer must be removed. This process is repeated until there is no candidate solution. Therefore, we call our new method the Dealer's Principle. Before we discuss our algorithm, some definitions are needed.

Definition 1: Assume the size of set P is n , every individual in P has r attributes, and $f_k(\cdot)$ is an evaluation function ($k=1,2,\dots,r$). The relationship between individuals in P is defined as follows: (1) Dominated relationship: $\forall X, Y \in P$, if $f_k(X) \leq f_k(Y)$, ($k=1,2, \dots, r$), and

$\exists l \in \{1, 2, \dots, r\}$ such that $f_l(X) < f_l(Y)$, then we have X dominates Y , or $X \succ Y$. Here “ \succ ” represents dominated relationship, and we say X is a non-dominated individual, and Y is a dominated individual.
 (2) Irrelevance: $\forall X, Y \in P$, if X and Y do not dominate each other, or there is no relations between X and Y , then we say that X and Y are irrelevant.

Definition 2: There exist $X \in P$, if not $\exists Y \in P, Y \succ X$, the X is a non-dominated individual of P . The set consisting of non-dominated individuals of P is called the non-dominated set of P .

Definition 3: Assume Nds is a non-dominated set of $P, Nds \subseteq P. \forall X \in P$, if X is a non-dominated individual of P , then $X \in Nds$. Nds is called the biggest non-dominated set of P .

Now we discuss the Dealer’s Principle to construct the non-dominated set from a set of candidate solutions. Assume P is an evolutionary population, and Q is a set of candidate solutions. Starting with an empty Nds of the non-dominated set, let $Q=P$. One individual X is picked at random from Q (in fact, X is removed from Q in this case) and compared with other individuals in Q one by one, and then the individuals dominated by X are deleted. If X is not dominated by any other members of Q , then it is a non-dominated individual, and it is joined into Nds . We continue this process until Q is empty.

The Algorithm 1 shows how to implement the method to construct a non-dominated set of an evolutionary population.

Algorithm 1: Construct a non-dominated set with Dealer’s Principle
 Function Nds (Pop: population) { $Q=Pop$; while ($Q \neq \phi$) do { $X \in Q, Q=Q-\{X\}$; x -is-non-dominated=.T.; for ($Y \in Q$) { if ($X \succ Y$) then $Q=Q-\{Y\}$ else if ($Y \succ X$) then x -is-non-dominated=.F. } if (x -is-non-dominated) then $Nds=Nds \cup \{X\}$; } } It can be proved that the Nds is a non-dominated set of P , and is the biggest non-dominated set of P . Now we analyze the computational complexity of the algorithm with the Dealer’s Principle to construct a non-dominated set. Suppose that n is the size of set P , and there are m non-dominated individuals in P . Firstly we discuss three particular cases: (i) there exist $(m-1)$ non-dominated individuals after $(m-1)$ rounds of comparison, but $(m-1)$ individuals do not remove any dominated individual from Q . After the m^{th} round of comparison, the m^{th} non-dominated individual removes $(n-m)$ dominated individuals from Q . The computational complexity in this situation is: $(n-1)+((n-2)+\dots+(n-m))=(2n-m-1)m/2 < nm$. (ii) While the early $(n-m)$ rounds of comparison, all the $(n-m)$ dealers are dominated individuals, and no other individual is removed from the current candidate solutions of Q by them, and the m non-dominated individuals are generated in the latter m rounds of comparison. This is the worst situation. The computational complexity under this situation is: $(n-1)+ (n-2)+\dots+1=n(n-1)/2=O(n^2)$. (iii) In the first round, there is an individual that is

not dominated by any members of Q , and this non-dominated individual removes $(n-m)$ individuals dominated by it from Q . In the later $(m-1)$ rounds of comparison, there are $(m-1)$ non-dominated individuals generated. This is the best situation. The computational complexity is: $(n-1)+[(m-2)+(m-3)+\dots+1]=(n-1)+(m-1)(m-2)/2=O(n+m^2)$.

As a general rule, there have been k rounds of comparison altogether, $m \leq k \leq n$. There are k individuals are removed naturally for k rounds of comparison, one at a time, here there are m non-dominated individuals and $(k-m)$ dominated individuals. There are $(n-k)$ dominated individuals removed from the set of candidate solutions Q for k rounds of comparison. Suppose that the probability of removing $(n-k)$ dominated individuals is equal: $(n-k)/k$, the computational complexity of k rounds of comparison is: $(n-1)+ (n-2-(n-k)/k) + (n-3-2(n-k)/k) + \dots + (n-k-(k-1)(n-k)/k) = [(n-1)+ (n-2)+ \dots + (n-k)] - [(n-k)/k + 2(n-k)/k + \dots + k(n-k)/k] + k (n-k)/k = [(n-1)+ (n-k)]k/2 - [(n-k)/k + k(n-k)/k]k/2 + (n-k) = k(n-1)/2 + (n-k)/2 = O(kn) < n^2$

From the analysis above, the time complexity of the algorithm to construct a non-dominated set with the Dealer's Principle is better than that in SPEA2 and NSGA-II. Especially in the early stage of algorithm execution, m is always much less than n (or $m \ll n$), so the algorithm is more efficient.

4. THE CLUSTERING METHOD BASED ON CORE DISTANCE

In the research of MOGAs, the diversity of solutions is one of the most important issues. MOGAs with parameters employ niche technique [11] and fitness sharing [12] to keep or maintain the diversity of solutions. MOGAs with no parameters employ a crowding procedure [16] or a clustering procedure [14,19] to maintain the diversity of solutions. It is shown from some studying [14,19] that the MOGAs that employ the clustering procedure have better diversity than these that employ the crowding procedure. While the early stage of a MOGA execution, a non-dominated set Nds is small, but it increases gradually. When $|Nds| > n$, how choose n individuals from Nds and keep the diversity of solutions at the same time? Therefore, a clustering procedure based on core distance is proposed in this paper.

The clustering procedure based on core distance employs a bottom-up approach to cluster individuals. Initially, we regard each of n individuals as a subclass and calculate the distance between their cores (centers), then combine two subclasses that they have the smallest core distance. The procedure of calculation and combination will be repeated until the stop

condition is satisfied ($|Nds| \leq n$). The clustering procedure is depicted in algorithm 2.

Algorithm 2: clustering procedure based on core distance

(1) initialize cluster set C , let every subclass of C includes one individual in Nds : $C = \bigcup_p \{ \{p\} \}$ $p \in Nds$ The core (o_i) of every subclass is the only individual in the subclass.

(2) if $|C| \leq \text{bound}$, then goto (8). Here, bound is the maximal limit size of non-dominated set.

(3) calculate the core distance between any two subclasses:
 $d(o_i, o_j) = \|o_i - o_j\|$, $o_i, o_j \in C$ Where $\|o_i - o_j\|$ is distance between o_i and o_j .

(4) choose two new subclass c_i and c_j which have minimal distance between their cores: $c_i, c_j: \min\{d(o_i, o_j) \mid o_i, o_j \in C\}$ merge c_i and c_j into subclass $c_k: C = C \setminus \{c_i, c_j\} \cup \{c_i \cup c_j\}$

(5) calculate the core of subclass c_k ,
 $d(p) = \min\{d(p, q) \mid p, q \in c_k, p \neq q\}$ $o_k = \min\{d(p) \mid p \in c_k\}$

(6) if $|C| \leq \text{bound}$, then goto (8).

(7) calculate the distance from c_k to any other subclass of C between their cores. $d(o_k, o_i) = \|o_k - o_i\|$ where o_k, o_i is the core of c_k and c_i respectively, $c_k, c_i \in C$ and $c_k \neq c_i$. then goto (4).

(8) return $Nds = \bigcup \{ o_i \}$, o_i is the core of c_i , $c_i \in C$.

The time complexity of the clustering procedure is $O(n^2)$.

5. MOGAS BASED ON CLUSTERING PROCEDURE

Assume that the size of the evolutionary population is n . Pop_t represents the population in t^{th} generation, Q_t denotes the new evolutionary population after the genetic operators (selection, crossover, and mutation) are applied to Pop_t , and the size of Q_t is also n . Set $R_t = Pop_t \cup Q_t$, and the size of R_t is $2n$. By constructing the non-dominated set Nds of R_t continually, the individuals in Nds are going more and more close to Pareto optimal front in order to accomplish multi-objective optimization.

If $|Nds| < n$, by means of a clustering procedure or a randomization procedure, $(n - |Nds|)$ individuals are generated and are combined with Nds to make a new population Pop_{t+1} . In this algorithm, the clustering procedure is evoked one time while the randomization procedure performing four times. This would achieve the best trade-off between the speed of the algorithm and the diversity of the solutions. If $|Nds| > n$, the clustering procedure is used to reduce the size of the non-dominated set. In this case, the clustering procedure is repeated until Nds include only n individuals or $|Nds| \leq n$. The

MOGA based on the clustering procedure, in which the Nds is constructed with the Dealer's Principle, is briefly described in Algorithm 3.

Algorithm 3: Multi-objective genetic algorithm based on the clustering procedure Multi-objective-GA(Pop₀) { Q_t = ψ (Pop_t); // ψ is a genetic operator including selection, crossover and mutation R_t=Pop_t ∪ Q_t; //combine parent generation and new generation Nds=Construct-Nds(R_t); //construct Nds with Dealer's Principle if (|Nds|<n) then if (count ≤ 3) then { Pop_{t+1}=Nds ∪ {Random-procedure(n-|Nds|)}; //a randomization procedure is used to generated (n-|Nds|) individuals count=count+1 } else {Pop_{t+1}= Nds ∪ Cluster (R_t-Nds, n-|Nds|); count=0 } //the clustering procedure is used to generate (n-|Nds|) individuals else if (| Nds|>n) then Pop_{t+1} =Cluster(Nds, n); //the clustering procedure is used to reduce the size of Nds t=t+1; }

Here Cluster(P, m) is the clustering procedure, P is a population on which the clustering operator applies, and m is the size of the clustering result. Initially, count=0.

6. EXPERIMENT

6.1 Convergence and Diversity

In this paper, we use a simple metric for evaluating convergence. A reference set is needed while evaluating convergence. A reference set P* is either a set of Pareto-optimal points (if known) or a non-dominated set of points in a combined pool of all generation-wise populations obtained from a MOGA run. This means P*=Non-dominated($\cup_{t=0}^T Nds^{(t)}$), where Nds(t) is the non-dominated set in the tth generation, (t=0, 1, ..., T). From each individual i in the current non-dominated set, calculate the smallest normalized Euclidean distance to P* as follows [19]:

$$pd_i = \min_{j=1}^{|\rho^*|} \sqrt{\sum_{k=1}^m \left(\frac{f_k(i) - f_k(j)}{f_k^{\max} - f_k^{\min}} \right)^2} \quad (1)$$

Here, f_k^{\max} and f_k^{\min} are the maximum and the minimum function values of kth objective function in P*, and m is the number of the objective function. Calculate the average of pd_i :

$$C(P^{(t)}) = \sum_{i=1}^{|Nds^{(t)}|} pd_i / |Nds^{(t)}| \quad (2)$$

In order to satisfy $C(P^{(t)}) \in [0,1]$, we normalize the $C(P^{(t)})$ values by its maximum value:

$$\bar{C}(P^{(t)}) = C(P^{(t)}) / C(P^{(0)}) \quad (3)$$

Fig. 5 shows the convergence of three kind of MOGAs. It is clear that the $\bar{C}(P^{(t)})$ will become less and less when t increases. It also means that the set of candidate solutions will approach the Pareto optimal solutions gradually when $\bar{C}(P^{(t)})$ decreases continually.

Besides the convergence, we hope that the final solutions should have a good diversity. The obtained set of the first non-dominated solutions is compared with a uniform distribution and the deviation is computed as follows:

$$\Delta(d_i, \bar{d}) = \frac{1}{|Nds|} \sum_{i=1}^{|Nds|} (|d_i - \bar{d}|) / \bar{d} \quad (4)$$

$$\Delta_v = \sum_{i=1}^{10} \frac{(\Delta_i - \bar{\Delta})^2}{10} \quad (5)$$

Here, $d_i = \text{Min}\{s(i, j) \mid j \in Nds\}$, $s(i, j)$ is the Euclidean distance between individual i and j , \bar{d} is the average of d_i , the deviation measure Δ_i is the value of $\Delta(d_i, \bar{d})$ in one run, the measure $\bar{\Delta}$ is the average of Δ_i for 10 runs, and Nds is the non-dominated set.

$\Delta(d_i, \bar{d})$ represents the variation degree of d_i . Thus, it is clear that an algorithm having a smaller Δ maintains better distribution of solutions. Here Δ_v is the variance of Δ in 10 runs, so the smaller the value of Δ_v , the better the consistency of the solutions. It is shown in Fig. 6 and in Table 1.

6.2 Simulation Results

The test problem DLTZ3 is identical to the problem described as follows:

$$\text{Minimize } f_1(X) = (1 + g(X_k)) \cos(x_1\pi/2) \cos(x_2\pi/2) \cdots \cos(x_{m-2}\pi/2) \cos(x_{m-1}\pi/2),$$

$$\text{Minimize } f_2(X) = (1 + g(X_k)) \cos(x_1\pi/2) \cos(x_2\pi/2) \cdots \cos(x_{m-2}\pi/2) \sin(x_{m-1}\pi/2),$$

$$\text{Minimize } f_3(X) = (1 + g(X_k)) \cos(x_1\pi/2) \cos(x_2\pi/2) \cdots \sin(x_{m-2}\pi/2),$$

.....

Minimize $f_{m-1}(X) = (1 + g(X_K)) \cos(x_1\pi / 2) \sin(x_2\pi / 2),$

Minimize $f_m(X) = (1 + g(X_K)) \sin(x_1\pi / 2),$

$0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, n$

Where $g(X_K) = |X_K| + \sum_{i=m}^n ((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))), (x_m, \dots, x_n) \in X_K$

Here $|X_K|$ is the size of X_K set, $X = \{(x_1, x_2, \dots, x_{m-1}, x_m, \dots, x_n)\}, X_K = \{(x_m, \dots, x_n)\}$. The global Pareto-optimal front corresponds to $x_i = 0.5, (i = m, \dots, n)$.

In experiment, the population size is 200, the crossover probability is 0.8, and the mutation probability is 1/len (where len is the number of variables). We use the binary coding and an equal length of genes in solving corresponding problems. We use FMOGA to denote our MOGA discussing in this paper.

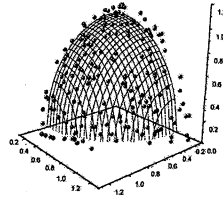


Figure 1. The NSGAI population on test problem DTLZ3

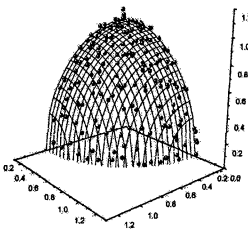


Figure 2. The FMOGA population on test problem DTLZ3

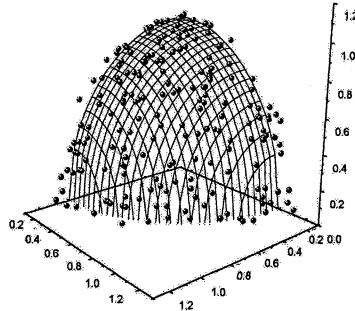


Figure 3. The SPEA2 population on test problem DTLZ3

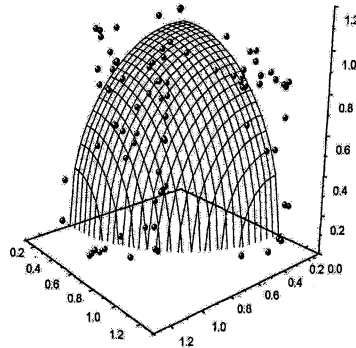


Figure 4. Fig 4. the NPGA population on test problem DTLZ3

We compare FMOGA with NSGA-II, SPEA2 and NPGA. Under the same environment, each algorithm runs for 500 generations. Figure 1 to Figure 4 show the distribution metrics for FMOGA, NSGA-II, SPEA2 and NPGA. In terms of the distribution of solutions, from the simulation results, it is shown that FMOGA, NSGA-II and SPEA2 do better than NPGA; furthermore, it is remarkable that FMOGA performs well than NSGA-II and SPEA2. Figure 6 shows the diversity at every generation of a MOEA run. The best performance is provided by FMOGA and the worst is NPGA. Figure 5 shows the convergence that FMOGA, NSGA-II and SPEA2 have reached closely to the true Pareto-optimal front, but NPGA cannot be shown in Figure 5 because $\overline{C}(P^{(t)}) > 0.3$.

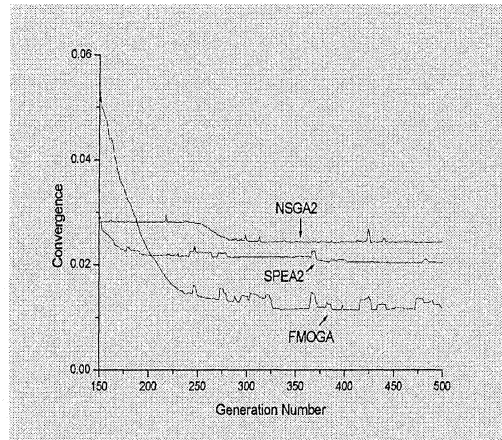


Figure 5. comparison of convergence metrics

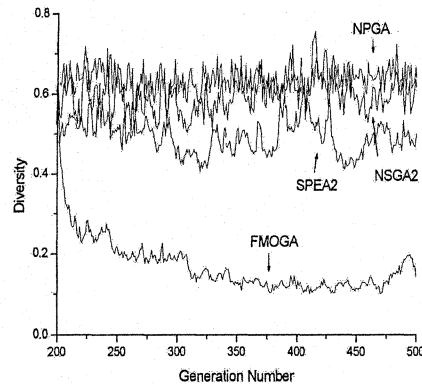


Figure 6. Fig 6. comparison of the diversity metrics

Table 1 shows the average value in 10 independent runs for FMOGA, NSGA-II, SPEA2 and NPGA. It is clear from Table 1 that NPGA is the fastest of all, and FMOGA is more efficient than NSGA-II and SPEA2. It is also shown in Table 1 that the FMOGA is more consistent than the other MOGA because its deviation Δ_V is the least of all.

Table 1. comparison of time, diversity and variance of diversity using four algorithms

Algorithm	FMOGA	NPGA	SPEA2	NSGA-II
Time	6.245	5.365	7.856	7.564

Diversity(Δ)	0.2306	0.6539	0.4093	0.6162
Variance of diversity(Δ_{ν})	0.0528	0.2642	0.0754	0.0862

7. CONCLUSIONS

This paper proposes a new method to construct the non-dominated set of MOGA based on the Dealer's Principle. To keep the diversity of candidate solutions, a clustering procedure based on the core distance is discussed. To test the efficiency of the Dealer's Principle to construct the non-dominated set and the performance of the clustering procedure, we design a MOGA namely FMOGA. By the experimental results, it is shown that our new MOGA using the Dealer's Principle to construct non-dominated set and using clustering procedure to keep the diversity of solutions is of better performance on speed, convergence and diversity of solutions than previous approaches.

REFERENCES

1. Hisashi Tamaki. Generation of a set of Pareto-optimal solutions by genetic algorithm. T SICE, 1995, 31(8): 1185-1192.
2. Brian J. Ritzel, J. Wayland Eheart, and S. Ranjithan. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. Water Resources Research, 30(5): 1589-1603, may 1994.
3. Schaffer, J. D., Multi objective optimization with vector evaluated genetic algorithms. In J.Grefenstette, ed., Proceedings of an International Conference on Genetic Algorithms and their Applications, 93-100, 1985.
4. Patrick D. Surry, Nicholas J. Radcliffe, and Ian D. Boyd. A MultiObjective Approach to Constrained Optimisation of Gas Supply Networks : The COMOGA Method. In Terence C. Fogarty, editor, Evolutionary Computing. AISB Workshop. Selected Papers, Lecture Notes in Computer Science, pages 166--180. Springer-Verlag, Sheffield, U.K., 1995.
5. Richardson, J. T., Palmer, M. R., Liepins, G., & Hilliard, M. Some guidelines for genetic algorithms with penalty function. Proceeding of the third International Conference on Genetic Algorithms. Morgan-Kauffman, 191-197,1989.
6. Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, discussion and Generalization, In Stephanie Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pages 416-423, San Mateo, California. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers, 1993.
7. Carlos Artemio Coello Coello. *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design*. PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA, April 1996.

8. Goldberg, D. E. and Deb, K. A comparison of selection schemes used in genetic algorithms, *Foundations of Genetic Algorithms*, 69-93, 1991.
9. Kalyanmoy Deb. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design, In Kaisa Miettinen, Marko M. Mäkelä, Pekka Neittaanmäki, and Jacques Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, chapter 8, pages 135-161. John Wiley & Sons, Ltd, Chichester, UK, 1999.
10. Jeffrey Horn and Nicholas Nafpliotis. Multiobjective Optimization using the Niche Pareto Genetic Algorithm, Technical Report IlliGAL Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.
11. Horn J., Nafpliotis N., & Goldberg D. E. A Niche Pareto genetic Algorithm for Multiobjective Optimization. *Proceeding of the first IEEE Conference on Evolutionary Computation*, 82-87, 1994.
12. Goldberg, D. E., & Richardson, J. J. Genetic Algorithms with sharing for multi-modal function optimization. *Genetic Algorithms and Their Applications: Proceedings of the second ICGA*, Lawrence Erlbaum Associates, Hillsdale, NJ, 41-49, 1987.
13. Zitzler, E. and L. Thiele (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4): 257-271, November 1999.
14. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. *EUROGEN 2001 - Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems*, September 2001.
15. N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. Technical report, Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, India, 1993.
16. Deb K., Agrawal S., Pratap A., & Meyarivan T. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. *KanGAL Report No.200001*, 2000.
17. Kalyanmoy Deb, Amrit Pratap, Sameer Agrawal and T. Meyrivan. A Fast and Elitist Multi-objective Genetic Algorithm : NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182-197, April 2002.
18. Deb, K, Mohan, M. and Mishra, S. A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions. *KanGAL Report No. 2003002*, February, 2003.
19. Deb, K and Jain, S. Running performance metrics for evolutionary multi-objective optimization. *KanGAL Report No. 2002004*, May 2002.