DYNAMIC POPULATION SIZE IN MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

Haiming Lu   Gary G. Yen
School of Electrical and Computer Engineering
Oklahoma State University, Stillwater, OK, 74078

Abstract - In this paper, the authors propose a new evolutionary approach to multiobjective optimization problems; the Dynamic Multiobjective Evolutionary Algorithm (DMOEA). In DMOEA, a population growing and a population decline strategies are designed, and several important indicators are defined in order to determine the adaptive individual "killing" scheme. By examining the selected performance indicators of a test function, DMOEA is found to be effective in directing the population into an optimal population size, keeping the diversity of the individuals along the trade-off surface, tending to extend the Pareto front to new areas, and finding a well-approximated Pareto optimal front.

I. INTRODUCTION

During the past decade, several Multiobjective Evolutionary Algorithms (MOEAs) have been proposed and applied in Multiobjective Optimization Problems (MOPs) [1]. These algorithms share the unique purpose—searching for a uniformly distributed, *near-optimal* and *near-complete* Pareto front for a given MOP. However, this ultimate goal is far from being accomplished by the existing MOEAs as described in literature [1]. In one respect, most of the MOPs are very complicated and require the computational resources to be homogenously distributed in a high dimensional search space. On the other hand, those better-fit individuals generally have strong tendencies to restrict searching efforts within local areas because of the "genetic drift" effect [2], which results into the loss of diversity due to stochastic sampling. Additionally, most of the existing MOEAs adopt a heuristically chosen population size to fulfill the computation. However, as addressed in [3], evolutionary algorithm may suffer from *premature* convergence if the population size if too small, whereas a too large population size will result in a heavy burden of undesired computation and a long waiting time for fitness improvement.

In the case of Single Objective (SO) optimization, several methods of determining an optimal population size from different perspectives have been proposed [3-5]. Since the purpose of solving an SO problem is to search for a single optimal solution at the final generation, which makes the preservation of population diversity is not an important issue to be concerned. However, in order to solve MOPs, an MOEA needs to uniformly distribute its computation effort in all the explored and unexplored areas and locate reasonable number of possible nondominated points to sketch a *near-complete* Pareto front. In general, the size of final Pareto set resulted by most MOEAs is equivalent to the size of initial population. As indicated in [6], the exact trade-off surface of an MOP is often unknown in *a priori*, it is difficult to estimate an optimal number of individuals necessary for effective exploration of the solution space as well as a good representation of the trade-off surface. This difficulty implies that a "guessed" size of the initial population is not appropriate in a real world application. Therefore, a dynamic population size autonomously adjusted by the on-line characteristics of population trade-off and density distribution information will be more efficient and effective than a constant population size in terms of avoiding *premature* convergence and unnecessary computational complexity.

As pointed out in [6], the issue of dynamic population in MOEAs has not been well attended yet. Although in some elitism based MOEAs, main population and elitist archive are separated and updated by exchanging elitists between them, the size of the main population or the sum of the main population and the archive is still fixed [7-8]. Therefore, a "guessed" size of initial population is still needed in these algorithms. Tan, Lee and Khor proposed an Growinging MOEA (IMOEA) [6], which devises a fuzzy boundary local perturbation technique and a dynamic local fine-tuning method in order to achieve broader neighborhood explorations and eliminate gaps and discontinuities along the Pareto front. However, this algorithm adopts a heuristic method to estimate the desired population size $dps^{(n)}$ for next generation according to the approximated trade-off hyper-area of current generation, but not to the trade-off and density information of the entire objective space. Therefore, the computation load may be wrongly determined if the approximation $dps^{(n)}$ value is incorrect in some situation (e.g. MOPs with local Pareto front), which makes IMOEA result in a *premature* solution set in some cases. Moreover, IMOEA is relatively complicated and its robustness needs to be further examined by different initial populations.

In this paper, the authors propose a novel Dynamic Multiobjective Evolutionary Algorithm (DMOEA). In DMOEA, an MOP will be converted into a bi-objective optimization problem in terms of individual's rank and density values and a population growing strategy is designed based on the converted fitness. Moreover, three types of qualitative indicators—age, health and environment— are associated with each individual in order to determine the probability of the elimination of an individual. By using DMOEA, the optimal population size can be autonomously determined and is not sensitive to the initial population size.

0-7803-7282-4/02/$10.00 ©2002 IEEE

The remainder of this paper is organized as follows. Section 2 describes the proposed dynamic population-sized evolutionary algorithm and introduces a "heating" schedule—population growing strategy and a "cooling" schedule—population declining strategy in order to control the population size within a reasonable range during the evolutionary process. In Section 3, the proposed DMOEA is tested by a MOP benchmark function with different initial population sizes; three performance indicators are inspected to examine the effectiveness and robustness of the proposed algorithm. Finally, Section 4 provides some concluding remarks along with pertinent observations.

II. DYNAMIC MULTIOBJECTIVE EVOLUTIONARY ALGORITHM

Generally, the approximation of the Pareto-optimal set involves two objectives: the distance to the true Pareto front is to be minimized while the diversity of the generated solutions is to be maximized [8]. For the first objective, a Pareto-based fitness assignment (ranking scheme) is usually designed in many existing MOEAs [1] in order to guide the search towards the ideal Pareto optimal front. For the second objective, some MOEAs provide a density estimation method to preserve the population diversity. Unfortunately, these two objectives are conflicting since the diversity preservation process will slow down the convergence speed, or even degrade the quality of the resulting Pareto front. In one respect, as general GA, MOEA exploits the "genetic drift" effect to converge the solution to each of the optimal point. On the other hand, the "genetic drift" phenomenon must be avoided in order to sketch a uniformly sampled trade-off surface for the final Pareto front. This contradicted issue is very difficult to be solved by MOEAs with fixed population size, since they have to homogenously distribute the predetermined computation resource to all the possible directions in the objective space. Therefore, to cope with this contradiction a Dynamic Multiobjective Evolutionary Algorithm (DMOEA) is proposed herein.

Similar to the other advanced MOEAs [7-9], DMOEA also converts the original MOP into a bi-objective optimization problem—minimizing individual rank value and maintaining individual density value. Furthermore, based on these two objectives, we introduce two types of qualitative indicators—health and environment indicators.

A. Pareto Rank and Health Indicator

First introduced by Goldberg [10], population-ranking schemes have been widely applied in MOEAs. The general ranking method is shown in Figure 1, it ensures that all the non-dominated individuals in the population will be assigned rank 1 and removed from temporary assertion, then a new set of non-dominated individuals will be assigned rank 2, and so forth. By being combined with population density information, some other ranking schemes have also been designed and applied in different MOEAs [7-9]. However, in DOMGA, we measure pure Pareto rank value since it represents a very important indicator—"*health*", of an individual. Assume at generation n, individual y has a rank value $rank(y,n)$, the *health* value of individual y at generation t is given as

$$H(y,n) = \frac{1}{rank(y,n)}. \tag{1}$$

From Equation (1), a nondominated individual, who is the healthiest, will have a H value equal to 1 and an individual with higher rank value will have a lower H value that is close to 0. Therefore, H value indicates the pure Pareto rank information of an individual and is independent with the density value of this individual. Moreover, the relationship between rank value and H value is not linear. In one aspect, the H value drops very fast if an individual's rank value is greater than 1, which results in a significant difference between dominated and nondominated individuals in terms of health condition. On the other hand, the H value also saturates very fast, which assigned all the dominated individuals very low H value (near zero) if their rank values are greater than 10. This characteristic can be used by the individual elimination scheme that will be discussed later.

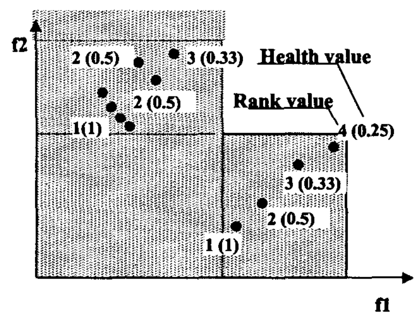


Figure 1 Individuals' rank values and the corresponding H values

B. Density Value and Environmental Indicators

Referring to [6], consider an m-dimension objective space, the desired population size, $dps(n)$, with the desired population size per unit volume, ppv, and the explored trade-off hyper-area of $A_{to}(n)$ discovered by the population at generation n can be defined as:

$$dps(n) = ppv \times A_{to}(n). \tag{2}$$

Therefore, with given population size per unit volume, ppv, the optimal population size can be obtained if an MOEA can correctly discover all the trade-off hyper-areas for an MOP. Instead of estimate the trade-off hyper-area $A_{to}(n)$ for each generation, DMOEA concentrates on searching for a *near-complete* final set of trade-off hyper-areas and ensure each of these areas contains ppv number of nondominated individuals. Therefore, by using DMOEA, the optimal population size and final Pareto front will be found simultaneously at the final generation.

To calculate the density value of each individual, a prior known objective space needs to be divided into several cells. The cell width in each objective dimension can be formed as

$$d_i = \frac{\max F_i - \min F_i}{K_i}, i = 1, \ldots, n, \tag{3}$$

where d_i is the width of the cell in the ith dimension, K_i denotes the number of cells designated for the ith dimension (i.e., in Figure 2, $K_1 = 12$ and $K_2 = 8$), and the upper and lower boundaries of the objective space in each dimension are prior known and denoted as $\max F_i$ and $\min F_i$. The density value of an individual is defined as the number of the individuals located in the same cell. In general, the finer the resolution of the cell is (i.e., smaller K_1 and K_2), the better performance the DMOEA will provide. Two environmental indicators—local and global densities—are associated with each individual to show current density information of the concerned individuals. A global density indicator tells how crowded the population distributed in the objective space, and a local density value indicates how crowded a concerned cell is. Assume at generation n, the density value of individual y is $density(y,n)$, and the average density of the population is D_{avg}, then the global density indicator of the population is defined as

$$DG(n) = \frac{D_{avg}}{ppv}, \tag{4}$$

and the local density indicator of individual y is defined as

$$DL(y,n) = \frac{density(y,n)}{ppv}. \tag{5}$$

Therefore, by these two density indicators, we can obtain the information about how crowded each cell and the entire population are, comparing to the desired ppv value.

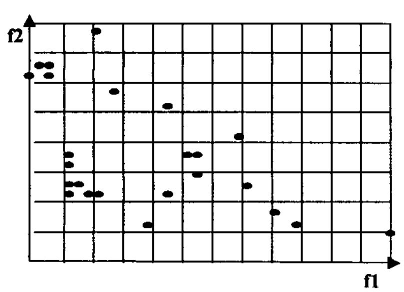


Figure 2 Objective space and density cells

C. Population Growing Strategy

In general, if an MOEA has a fixed population size, a "replacement" scheme is always applied. In this scheme, in order to keep the population size unchanged, a newborn offspring will replace one of its parents if its fitness value is higher than that of the parent. However, this scheme brings up a problem that some of the replaced parents may still be very valuable and have not been well exploited yet before they are replaced. Although some MOEAs adopt an elitist archive in addition to the main population in order to store some of the nondominated individuals generated during the evolutionary process, this problem is still not completely resolved. Therefore, DMOEA applies two independent strategies—population growing strategy and population declining strategy. The first strategy is a pure population growth strategy that ensures each individual survives enough generations so that it can contribute its valuable schemas. Meanwhile a population declining strategy is also designed to prevent the population size growing to excessively. The second strategy will be discussed in the next Subsection.

Since minimizing individual's rank value and maintaining each cell's density value are two converted objectives of DMOEA, crossover and mutation operations need to be devised to fit both of the purposes. For crossover, a reproduction pool with fixed number of selected parents is setup; a Cellular GA [11] is then applied to explore the new search area by "diffusion"—each selected parent performs crossover with the best individual (the one with the lowest rank value) within the same cell and the nearest neighboring cells that contain individuals. If one offspring produces better fitness (a lower rank value or a lower population density value) than its selected parents, it will be kept to the next generation; otherwise, it will not survive. The mutation operation is analogous. As a result, this strategy will guarantee that a newborn individual will have better fitness value than at least one of its parents, which helps DMOEA to cover all the unexplored cells in the objective space.

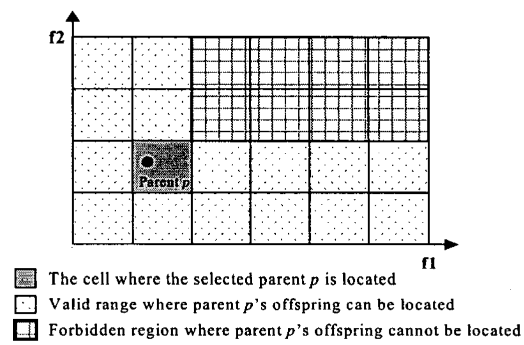


Figure 3 Illustration of the valid range and the forbidden region

However, as DMOEA encourages an offspring to exist in a sparse area thus all the unexplored cells will be visited, it is expected that some offspring will tend to move toward an opposite direction to the Pareto front when the cells close to the Pareto front are crowded. Obviously, these movements are harmful to the population to converge to the Pareto front. To prevent "harmful" offspring surviving and affecting the evolutionary direction and speed, a *forbidden region* concept is proposed in the offspring-generating scheme for the density subpopulation, thereby preventing the "backward" effect. The

0-7803-7282-4/02/$10.00 ©2002 IEEE

forbidden region includes all the cells dominated by the selected parent. The offspring located in the *forbidden region* will not survive in the next generation, and thus the selected parent will not be replaced. As shown in Figure 3, suppose our goal is to minimize objectives f_1 and f_2, and a resulting offspring of the selected parent p is located in the forbidden region. By DMOEA, this offspring will be eliminated even if it is located in a sparse area because this kind of offspring has the tendency to push the entire population away from the desired evolutionary direction.

D. Population Declining Strategy

As discussed in Subsection 2C, a population declining strategy is necessary to prevent the population size growing unbounded. In DMOEA, an individual will be retired a probability, which depends on several important indicators we mentioned in Subsection 2A and 2B. Moreover, to ensure that each appeared individual has enough time to provide its valuable schemas, an "age" indictor is also designed in DMOEA. For an individual in the initial population, its age value is assigned to be one, and its age will increase by one if the individual survives at the next generation. Similarly, the age of a newborn offspring is one and grows generation by generation as long as it lives. Assume at generation n, an individual y has an age value $age(y,n)$, its age indicator $A(y,n)$ is given by

$$A(y,n) = \frac{age(y,n) - A_{th}}{n}, \tag{6}$$

where A_{th} is a prior specified age threshold, which means that an individual will not be eliminated if its age is less than A_{th}.

To ensure that an eliminated individual has low fitness value, DMOEA moderately remove three types of individuals with different probabilities:

1) At generation n, find a set $Y_r = \{y_1, \cdots, y_k\}$ that contains all the individuals with the highest rank value $r_{\max}$. Therefore, if $r_{\max}$ is larger than 1, the probability of the death of an individual y_i to be eliminated is given by

$$p_1 = (1 - H(y_i, n))^2 \times A(y_i, n), \quad i = 1, \ldots, k. \tag{7}$$

2) At generation n, find a set $Y_d = \{y_1, \cdots, y_m\}$ that contains all the individuals with the highest density value, and then find a set $Y_{dr} = \{y_1, \cdots, y_{mr}\}$ that includes all the individuals with highest rank value $r_{d\max}$ among Y_d. Therefore, if $r_{d\max}$ is greater than 1, the probability of y_i to be eliminated is given by

$$p_2 = (1 - H(y_i, n))^2 \times DG(n) \times (DL(y_i, n) - 1) \times A(y_i, n)$$
$$i = 1, \ldots, mr. \tag{8}$$

3) At generation n, if $r_{\max}$ is equal to 1, find a set $Y_{rc} = \{y_1, \cdots, y_s\}$ that contains all the individuals with the highest density value. Therefore, the probability of individual y_i to be eliminated is given by

$$p_3 = (DL(y_i, n) - 1) \times A(y_i, n), i = 1, \ldots, s. \tag{9}$$

To determine if an individual will be eliminated, a random number between [0, 1] is generated to compare with a concerned probability according to the situation of the given individual. Therefore, from Equations (7)-(9), we can make several interpretations and conclusions as follows.

1) Since the age indicator $A(y_i, n)$ influences all of three probabilities, p_1, p_2 and p_3 will be negative numbers if the age of the concerned individual is smaller than the age threshold A_{th}. This implies that if an individual is not old enough, it will not be eliminated from population no matter how high its rank and density value is.

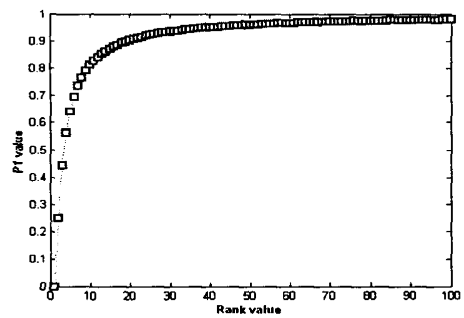


Figure 4 Relationship between rank values and p_1 values

2) At each generation, DMOEA will remove those most "unhealthy" individuals with different probabilities p_1, based on their rank values and ages. Assume the age indicator of an individual y is $A(y,n) \approx 1$, the relationship between its rank value and p_1 value is illustrated as Figure 4. Without considering the effects of other indicators, when an "unhealthy" individual in the set Y_r has a very high $r_{\max}$ value, it will have a very high probability (p_1) to be eliminated since it is too far away from current Pareto front. Moreover, as $r_{\max}$ drops and becomes more and more close to 1, p_1 will decrease very fast and the concerned individual will not die easily because it is very likely to be evolved into an elitist in the future. Therefore, this "shell removing" strategy will keep eliminating the individuals located on the outside layer with an adaptive probability until the entire population converges into a nondominated set.

3) As all the individuals in the same cell share the fixed computation resource (or "living resource"), the individuals located in a crowded area have to compete much harder for the limited resource than those situate in a sparse area. Therefore, another elimination scheme based on global and local density values is designed in DMOEA in order to remove some "unhealthy" individuals that stay in the most

crowded areas. From Equation (8), at each generation, if an individual belongs to the set Y_{dr}, it will have a probability p_2 to be eliminated based on its age, health and density condition. From this scheme, the population tends to be homogeneously distributed by eliminating the redundant crowdness.

4) After every individual has converged into a Pareto point, another elimination scheme is implemented based on p_3 values. Therefore, the resulting trade-off hyper-areas $A_{to}(n)$ are counted and the final population is truncated to ensure each cell contains ppv number of individuals; thus the optimal population size can be calculated by Equation (2).

III. EXPERIMENTAL RESULTS

In this paper, we used a modified MOP designed by Deb [12] as the preliminary test function that has a discontinuous Pareto front. Figures 5 shows the objective space and true Pareto front for this problem. More sophisticated benchmark test function will be evaluated later.

Minimize $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$, where

$$f_1(x_1, x_2) = x_1,$$

$$f_2(x_1, x_2) = (1 + x_2) \times (1 - \frac{x_1}{1 + x_2})^2 - \frac{x_1}{1 + x_2} \times \sin(10\pi x_1) \quad (10)$$

subject to $0 \le x_1, x_2 \le 1$.

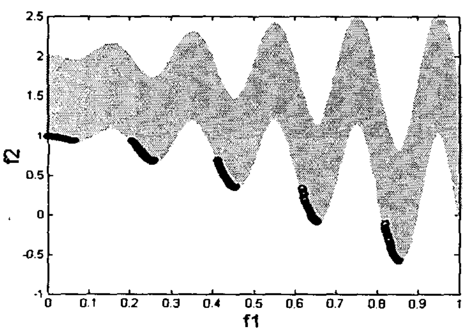
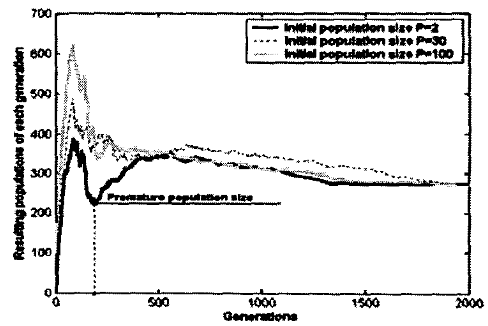
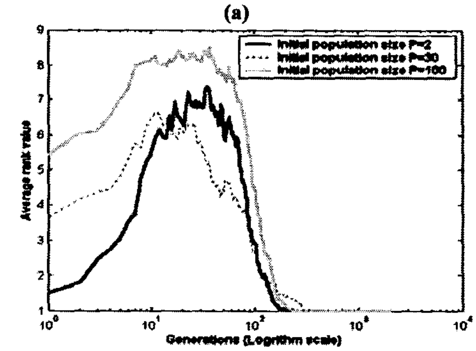


Figure 5 Objective on space and Pareto front of test function

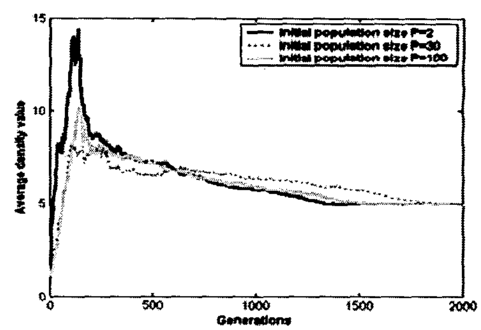
For the given test function, we select the boundary of the feasible objective space to be [0, 1] and [-1, 2.5] and the number of cells of each dimension to be $K_1 = 50$ and $K_2 = 100$. The desired population size per cell is predefined as $ppv = 5$. Three specified initial population sizes—2, 30 and 100—are chosen to test the robustness of DMOEA. The age threshold, the stopping generation, the chromosome length of each decision variable, the crossover rate and the mutation rate are chosen to be 10, 2000, 15, 0.7, and 0.1, respectively in the simulation. We run DMOEA 30 times for each selected population size to obtain the average results and for each run, a new initial population with the specified number of individuals is randomly generated and evolved by DMOEA. Moreover, we use three indicators derived from each generation to quantitatively measure the performance: average population rank value, average population density value and average generational distance value. The final average population rank value, final average population density value and final average generational distance are derived from the last generation and illustrated via Box plots for the test function considered.



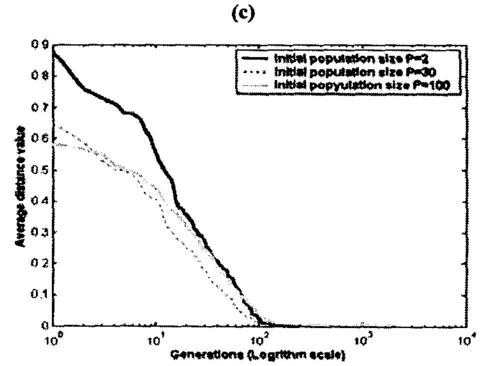
(a)



(b)



(c)



(d)

Figure 6 Evolutionary trajectories of (a) population size, (b) average rank, (c) density and (d) distance values of three initial population sizes ($A_{th} = 10$)

The evolutionary trajectories for the average sizes of populations and the values of three indicators over 30 runs are illustrated in Figure 6 (a), (b), (c) and (d) respectively. The corresponding Box plots of the final indicator values are shown in Figure 7. From Figures 6 and 7, we can observe that for the given MOP test function, chosen grid of cells and predefined *ppv* value, 275 individuals are determined as the final optimal population size (Pareto set). This implies that there are 55 trade-off cells (hyper-areas $A_{to}(n)$) that contain nondominated individuals discovered by DMOEA at the final generation.

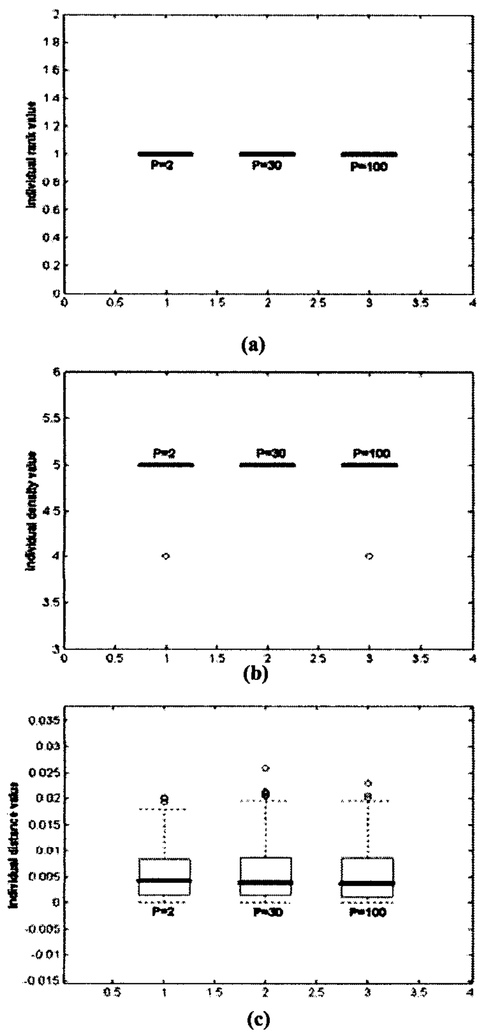


Figure 7 Box plots of (a) population rank, (b) density and (c) distance values of test function

Moreover, from Figure 7, it is obviously that the resulting population size and the qualities of the final Pareto fronts by DMOEA are not sensitive to the selection of the initial population size P, even in an extreme case when $P = 2$, which is the minimum population size that an evolutionary algorithm can be initiated. It is worthy to note that, when $P = 2$, the population growing strategy has a difficulty in balancing the population declining strategy, which may result in a minimum population size when the population converges to a nondominated set (Figure 6(a)). However, according to DMOEA, after the entire population converges to a nondominated set, the population growing is mainly based on the density information, of the new individuals. This means an offspring will survive mostly because it is a nondominated point and located in a sparse cell. Meanwhile, at this stage, only p_3 is valid to be considered in population declining strategy, which is designed to be very conservative to eliminate any Pareto points. Therefore, DMOEA can overcome this difficulty by keeping growing population size after this stage.

IV. CONCLUSION

In this paper, a new multiobjective evolutionary algorithm—Dynamic Multiobjective Evolutionary Algorithm (DMOEA) is proposed. Based on two converted objectives—minimizing rank and density value, several quantitative indicators are implemented and a population growing and a population declining strategy are designed to adaptively update population size. From the simulation results, DMOEA can effectively exploit an optimal population size by locating all the trade-off hyper-areas and approximate a *near-optimal, near-complete* Pareto front. Therefore, DMOEA shows its potential in solving complicated MOPs with different characteristics. This part will be studied in the future work.

REFERENCES

[1] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evol. Comput.*, vol. 3, pp. 1-16, 1995.

[2] S. W. Mahfoud, "Genetic drift in sharing methods," in *Proc. 1st IEEE Cong. Evolutionary Computation*, vol. 1, pp. 67-72, 1994.

[3] J. Arabas, Z. Michalewicz, and J. Mulawka, " GAVaPS-A genetic algorithm with varying population size," in *Proc. 1st IEEE Cong. Evolutionary Computation,* pp. 73-74, 1994.

[4] N. Zhuang, M. Benten and P. Cheung, "Improved variable ordering of BDDS with novel genetic algorithm," in *Proc. IEEE Symp. Circuits and Systems*, pp. 414-417, 1996.

[5] J. Grefenstte, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Systems, Man and Cybernetics*, vol. 16, pp. 122-128, 1986.

[6] K. Tan, T Lee and E. Khor, "Growinging multi-objective evolutionary algorithms: performance studies and comparisons," in *Proc. 1st Evolutionary Multi-Criterion Optimization*, pp. 111-125, 2001.

[7] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proc. Parallel Problem Solving from Nature—PPSN VI*, pp. 849-858, 2000.

[8] E. Zitzler, M. Laumanns, and L. Thiele, *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, Technical Report TIK-Report 103, Swiss Federal Institute of Technology, 2001.

[9] H. Lu and G. G. Yen, "Multiobjective optimization design via genetic algorithm," in *Proc. 40th IEEE Conf. Control Applications,* pp. 1190-1195, 2001.

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning,* Reading, MA: Addison-Wesley, 1989.

[11] T. Krink and R. K. Ursem, "Parameter control using agent based patchwork model," in *Proc. 7th IEEE Cong. Evolutionary Computation,* pp. 77-83, 2000.

[12] K. Deb, *Multiobjective Genetic Algorithms: Problem Difficulties and Construction of Test Problems*, Technical Report TR CI-49/98, University of Dortmund, Germany, 1998.