

Evolutionary Algorithm with Dynamic Population Size for Multi-Objective Optimization

E. F. Khor, K. C. Tan[†], M. L. Wang and T. H. Lee

Department of Electrical and Computer Engineering

National University of Singapore

10 Kent Ridge Crescent Singapore 119260

[†]Email: eletankc@nus.edu.sg

Abstract

This paper presents a novel "incremental" multi-objective evolutionary algorithm with dynamic population size that is adaptively computed according to the on-line discovered trade-off surface and its desired population distribution density. It incorporates the method of fuzzy boundary local perturbation with interactive local fine-tuning for broader neighborhood exploration to achieve better convergence as well as discovering any gaps or missing trade-off regions at each generation. The effectiveness of the proposed methodology is validated upon a benchmark multi-objective optimization problem.

1. Introduction

Evolutionary techniques for multi-objective (MO) optimization are currently gaining significant attention from researchers in various fields [1-5]. Unlike conventional methods that linearly combine multiple attributes to form a composite scalar objective function, a multi-objective evolutionary algorithm (MOEA) incorporates the concept of Pareto's optimality or modified selection schemes to evolve a family of solutions at multiple points along the trade-off surface. Based on different implemented strategies in cost assignments and selection methods, the study of evolutionary algorithms for MO optimization can be generally classified into five major groups, namely the objective reduction approaches, classified population approaches, weights randomizing approaches, preference relationship approaches and Pareto-based approaches.

If these evolutionary techniques for MO optimization, however, require a large and constant population size in order to discover the usually sophisticated trade-off surface. As addressed by [6], evolutionary optimization process may evolve too quickly and suffers from premature convergence if the population size is too small.

If the population is too large, however, undesired computational resources may be incurred and the waiting time for a fitness improvement may be too long in practice. Therefore the selection of an appropriate population size in evolutionary optimization is important and could greatly affect the effectiveness and efficiency of the optimization performance.

This paper proposes an incremental multi-objective evolutionary algorithm (IMOE) with dynamic population size for effective MO optimization. Instead of having a large and constant population to explore the solution surface, IMOE adaptively computes an appropriate population size according to the on-line evolved trade-offs and its desired population distribution density. In this way, the evolution could begin with a small population size initially, which is increased/decreased adaptively based upon the discovered Pareto front at each generation. This approach reduces the overhead computational effort and avoids any possible pre-mature convergence or incomplete trade-offs resulting from insufficient number of individuals.

In addition, the IMOE incorporates a fuzzy boundary local perturbation technique to encourage and reproduce the "incremented" individuals for better MO optimization. While maintaining the global search capability, the scheme enhances the local exploration and fine-tuning of the evolution at each generation so as to fill-up any discovered gaps or discontinuities among the non-dominated individuals that are loosely located or far away from each other along the trade-off surface. Details of the IMOE with fuzzy boundary local perturbation and other advanced methods are described in Section 2. Validation of the IMOE against a benchmark problem is given in Section 3. Conclusions are drawn in Section 4.

2. Incremented Multi-Objective Evolutionary Algorithm

The issue of dynamic population size in MO optimization currently remains an open problem for researchers in the field of evolutionary computation [7]. Extending from our earlier work of MOEA [8], an incremental multi-objective evolutionary algorithm is proposed to deal with this problem by adaptively computing an appropriate population size at each generation. The population size is thus dynamic based upon the on-line discovered Pareto front and its desired population distribution density along the trade-offs.

2.1 Dynamic Population Size

Consider an m -dimensional objective space, the desired population size, $dps^{(n)}$, with the desired population size per unit volume, ppv , and the approximated trade-off hyper-area of $A_{to}^{(n)}$ [8] discovered by the population at generation n can be defined as,

$$lowbps \leq dps^{(n)} = ppv \times A_{to}^{(n)} \leq upbps \quad (1)$$

where $lowbps$ and $upbps$ is the lower and upper bound for the desired population size $dps^{(n)}$, respectively, which can be treated as hard bounds that are optionally defined by the user. Note that since ppv is defined as a finite positive integer and $A_{to}^{(n)}$ is bounded if the objective space is bounded, $dps^{(n)}$ will be bounded within the limit of $lowbps$ and $upbps$ as given in eqn. 1.

The trade-offs for an m -objective optimization problem is in the form of an $(m-1)$ dimensional hyper-volume, which can be approximated by the hyper-surface $A_{to}^{(m)}$ of a hyper-sphere as given by [8],

$$A_{to}^{(m)} = \frac{\pi^{(m-1)/2}}{(\pi/2)!} \times \frac{(d^{(n)})^{m-1}}{2^{m-1}} \quad (2)$$

For simple approximation, $d^{(n)}$, can take the average of the shortest and longest possible diameter represented by $d_{min}^{(n)}$ and $d_{max}^{(n)}$ respectively. Let F_x and F_y denotes the non-dominated objective vectors that have the maximum distance between each other in the population, the computation of $d_{min}^{(n)}$ and $d_{max}^{(n)}$ can be directly illustrated in Fig. 1 where $d_{max}^{(n)} = d_1^{(n)} + d_2^{(n)}$. Note that as $A_{to}^{(m)}$ is always bounded if the space is bounded and ppv is always defined in finite value, $upbps^{(n)}$ will also be bounded.

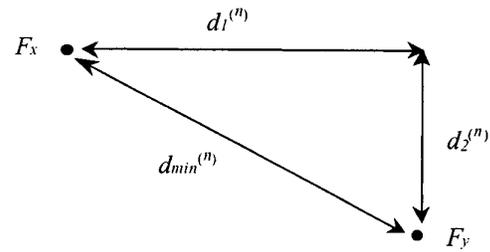


Fig.1 Computation for $d_{min}^{(n)}$, $d_1^{(n)}$ and $d_2^{(n)}$

2.2 Fuzzy Boundary Local Perturbation

In this section, a fuzzy boundary local perturbation (FBLP) scheme that perturbs the set of non-dominated individuals to produce the necessary "incremented" individuals for the desired population size as given by eqn. 1 in IMOEA is proposed. In brief, the FBLP is implemented for the following objectives: (1) Produce additional "good" individuals in filling up the gaps or discontinuities among existing non-dominated individuals for better representation of the Pareto front as shown in Fig. 2; (2) Perform interactive fine-learning to overcome weakness of local exploration in an evolutionary algorithm [9,10] and to achieve better convergence for evolutionary MO optimization; (3) Provide the possibility of exploration beyond the neighborhood perturbation to avoid pre-mature convergence or local saturation.

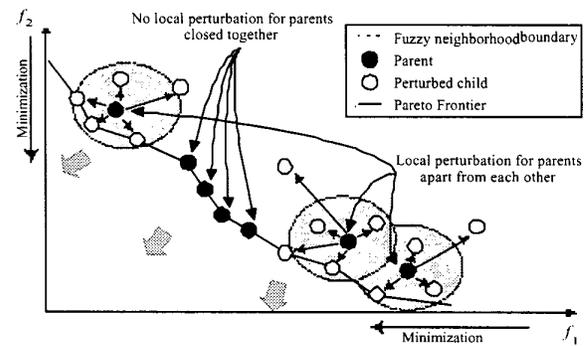


Fig. 2 FBLP for non-dominated parents with low niched cost (apart from other parents)

A simple and effective way to achieve the above objectives is to generate more Pareto optimum points within the evolution itself without going through the tedious two-stage neural network learning process. These additional Pareto points can be effectively obtained via the method of fuzzy boundary local perturbation at each generation. For this, only parent individuals that are being selected for reproduction from the tournament selection

scheme will be perturbed with the FBLP. Note that the selection criteria for the tournament is solely based upon the individuals' niched cost in the objective domain [11], instead of the cost of objective functions. Therefore parents with lower niched cost (located apart from other parents) will be given higher probability to be perturbed as compared to those with a high niched cost (located closed to other parents).

As shown in Fig. 2, the neighborhood boundary for the parents to be perturbed is fuzzy in such a way that the probability of perturbation is higher within the neighborhood region than those outside the neighborhood. Without loss of generality, consider a decimal coding scheme [8] with only one parameter being coded in a n -digit chromosome $X = \{x_i: i = 1, 2, \dots, n\}$, where i is the coding index such that $i = 1$ represents the most significant index and $i = n$ is the least significant index of the chromosome. A probability set $P = \{p_i: i = 1, 2, \dots, n\}$ that indicates the perturbation probability for each element of the chromosome X , with a value of "1" represents "always perturb", a value of "0" implies "never perturb" and their intermediate values, can be defined and represented by the following sigmoid function [12], $\forall i = 1, 2, \dots, n; n > 1$,

$$p_i = \begin{cases} b \left[2 \left(\frac{i-1}{n-1} \right)^2 + a \right] & , 1 \leq i \leq \beta \\ b \left[1 - 2 \left(\frac{i-n}{n-1} \right)^2 + a \right] & , \beta < i \leq n \end{cases} \quad (3)$$

As given in eqn. 3, the probability values of p_i have a lower bound of 'ab' and an upper bound of 'ab + b'. The coefficients 'a' and 'b' are chosen in the range of $0 \leq b \leq 0.7$ and $0 \leq ab \leq 0.02$, which are set according to the desired perturbation probability for the upper and the lower bounds. The value of β is defined based on the desired boundary of the digit x_β in the parameter space, which is selected as $(n/2)$ in this paper.

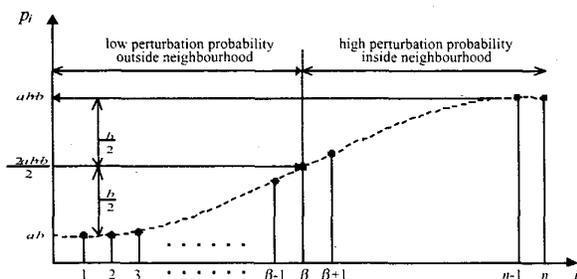


Fig. 3 The perturbation probability for a one-parameter chromosome in FBLP

As shown in Fig. 3, the more significant of a digit in the parameter X , the lower its perturbation probability in FBLP. This makes the possibility of the generated offspring to be lied within their parents' neighborhood higher than that of outside the neighborhood.

2.3 Program Flowchart of IMOEA

The overall program flowchart of the proposed algorithm including both the fixed (MOEA) [13] and the proposed dynamic (IMOEA) population size is shown in Fig. 4. The dynamic sharing method for niched cost estimation [8] is applied here to provide a simple and effective computation of α_{share} at each generation, which is capable of distributing the population uniformly along the Pareto front without the need of any *a-priori* knowledge in setting the α_{share} .

A switching preserved strategy (SPS) that preserves and allows the non-dominated individuals to be evolved with the population concurrently is proposed in this paper. As shown in Fig. 4, SPS performs population preservation based upon the on-line population distribution to ensure stability and diversity of the evolution as highlighted in the shaded region. Any non-dominated individuals resulted from current generation that are different from the evolved population will be added to the population at next generation to ensure stability of the Pareto optimal set. This combined population often exceeds the original population size, filtering is thus necessary in SPS to eliminate any extra individuals and maintain the desired population size. For this, if the number of non-dominated individuals in the combined population is less or equal to the desired population size, filtering is performed according to the individuals' Pareto rank values (cost in objective functions) in order to encourage stability of the evolution at the Pareto optimal frontier. However, if the number of non-dominated individuals exceeds the desired population size, filtering is done based on the shared costs (cost in niched function) in order to spread the individuals equally distributing along the Pareto front.

3. Validation on Benchmark Problem

The two-objective minimization problem given in [14] is studied here. This test function is chosen since it has a trade-off surface that is easy for visualization and comparison. The two-objective functions, f_1 and f_2 , to be minimized are given as:

$$f_1(x_1, \dots, x_8) = 1 - \exp\left(-\sum_{i=1}^8 \left(x_i - \frac{1}{\sqrt{8}}\right)^2\right)$$

$$f_2(x_1, \dots, x_8) = 1 - \exp\left(-\sum_{i=1}^8 \left(x_i + \frac{1}{\sqrt{8}}\right)^2\right)$$
(4)

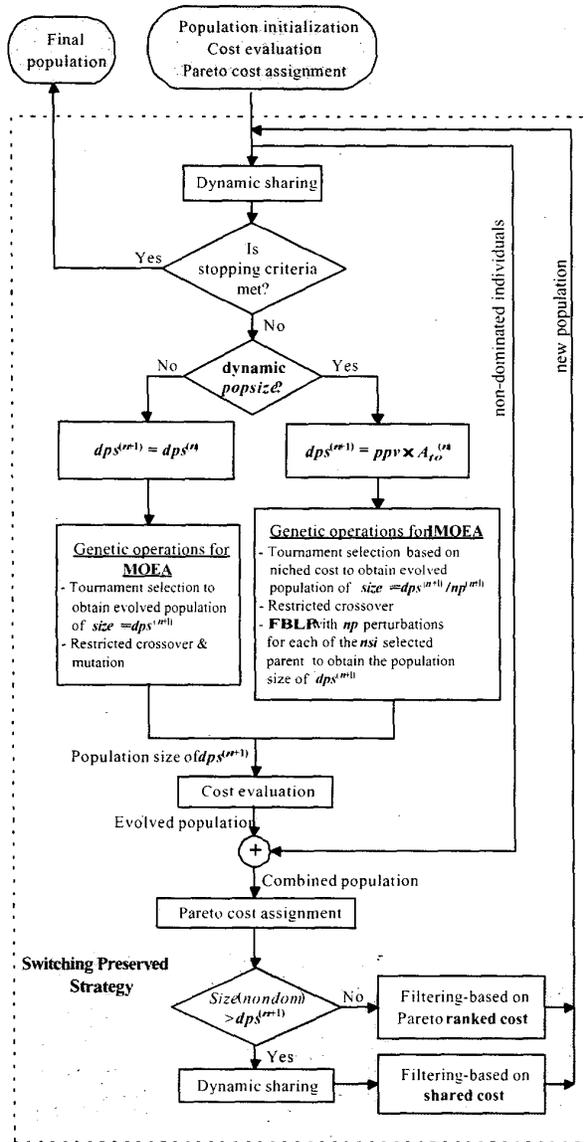


Fig. 4 Overall program flowchart with fixed (MOEA) and dynamic (IMOEA) population size

The problem of determining the appropriate population size to generate non-dominated individuals with desired distribution density can be overcome by adapting a dynamic population size for on-line adaptation as proposed in IMOEA, which estimates an appropriate and bounded population size at each generation to cover the entire trade-offs based on the desired population distribution density. The IMOEA was applied with a small initial population size of 10 and a pre-defined population size per unit volume of 200. As shown in Fig. 5a, the final population at the end of the evolution is uniformly distributed along the entire trade-off curve. As can be seen in Fig. 5b, the population size starts with a small initial population size of 10 and increases the population size adaptively based on the on-line growing Pareto front according to the surface of A_{to} in eqn. 2. The population size of IMOEA saturates at 302 in which the hyper-surface of A_{to} cannot be further increased since the entire trade-offs has already been discovered.

Fig. 6 illustrates performances of the IMOEA with a smaller setting of population size per unit volume by reducing the value of ppv from 200 to 100. The population distribution at the end of the evolution and the trace of population size at each generation are depicted in Fig. 6a and 6b, respectively. The final population size is now reduced to 146 which is smaller than that of 302, as desired.

Subsequently, a goal setting of $(f_1, f_2) = (0.9, 0.5)$ is incorporated in the optimization problem. Fig. 7 shows the distribution of non-dominated individuals at 4 different stages along the evolution. As can be seen, the population size starts to grow from an initial population of 10, and reaches 70 at the end of the evolution based on the limited size of the focused tradeoff curve and the desired population distribution density in the objective domain. Along the evolution, the population size of IMOEA varies and increases according to the on-line discovered Pareto front before satisfying the goal setting (stages 1-3). This shows that the IMOEA is capable of evolving an appropriate population size, starting from a small initial population with less computational overhead, to effectively represent the entire final trade-offs with or without any goal settings.

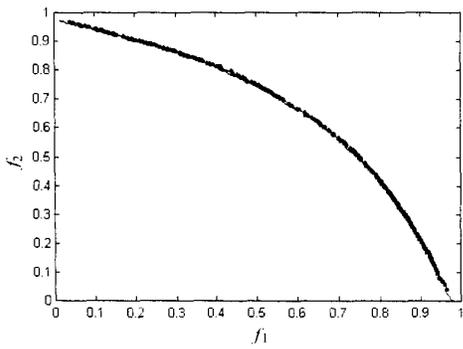
4. Conclusion

This paper has developed an Incremented Multi-Objective Evolutionary Algorithm (IMOEA). Unlike the conventional fixed-population-size evolutionary technique, this IMOEA is able to practice the dynamic population size based on the discovered trade-off hyper-

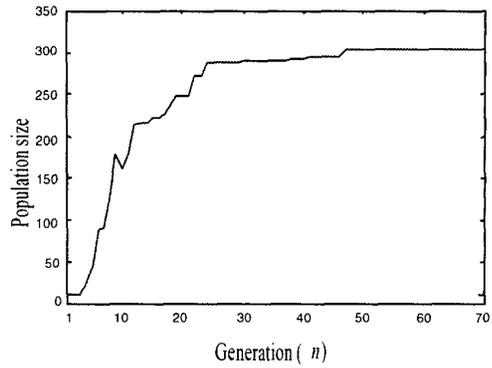
surface by the population and the desired population distribution density. This algorithm has solved the difficulty to decide the appropriate population size to sufficiently explore the tradeoffs that is often unknown initially. Besides, the new proposed Fuzzy Boundary Local Perturbation (FBLP) can create additional individuals within the fuzzy neighborhood of their parents and improve the local searching ability. The advantages of the above methodologies have been illustrated via a benchmark two-objective minimization problem with extensive comparisons and investigations.

References

- [1] R. Allenson, "Genetic Algorithms with Gender for Multi-function Optimization," Technical Report: EPCC-SS92-01, Edinburgh Parallel Computing Center, Edinburgh, Scotland, 1992.
- [2] D. Beasley, D. R. Bull, and R. R. Martin, "A Sequential Niche Technique for Multimodal Function Optimization," *Evolutionary Computation*, MIT Press Journals, vol. 1, no. 2, pp. 101-125, 1993.
- [3] C. M. Fonseca, and P. J. Fleming, "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part I: A Unified Formulation," *IEEE trans. on System, Man, and Cybernetics-Part A: System and Humans*, vol. 28, no. 1, pp. 26-37, 1998.
- [4] D. E. Goldberg, and J. Richardson, "Genetic Algorithms with Sharing for Multimodal Function Optimization," *Proc. of Second Int. Conf. on Genetic Algorithms*, Lawrence Erlbaum, pp. 41-49, 1987.
- [5] J. D. Schaffer, "Multi-objective Optimization with Vector Evaluated Genetic Algorithms," *Genetic Algorithms and Their Applications: Proc. of the 1st Int. Conf. on Genetic Algorithms*, Lawrence Erlbaum, pp. 93-100, 1985.
- [6] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS-A Genetic Algorithm with Varying Population Size," *Proc. of the 1st Conf. on Evolutionary Computation*, vol. 1, pp. 73-74, 1994.
- [7] C. A. Coello, "An Updated Survey of GA-based Multiobjective Optimization Techniques," Technical Report: Lania-RD-98-08, Laboratorio Nacional de Informatica Avanzada (LANIA), Veracruz, Mexico, 1998.
- [8] K. C. Tan, T. H. Lee, and E. F. Khor, "Evolutionary Algorithms with Goal and Priority Information for Multi-objective Optimization," *IEEE Proc. of the Congress on Evolutionary Computation*, Washington, D.C, USA, vol. 1, pp. 106-113, 1999.
- [9] B. Dengiz, F. Altiparmak, and A. E. Smith, "Local Search Genetic Algorithm for Optimal Design of Reliable Networks," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 3, pp. 179-188, 1997.
- [10] M. Hagiwara, "Pseudo-hill Climbing Genetic Algorithm (PHGA) for Function Optimization," *Proc. of the Int. Conf. on Neural Networks*, vol. 1, pp. 713-716, 1993.
- [11] C. M. Fonseca, "Multiobjective genetic algorithms with application to control engineering problems," Ph.D. Thesis, Dept. Automatic Control and Systems Eng., University of Sheffield, Sheffield, UK, 1995.
- [12] E. Cox, *The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems*, Academic Press, Inc., London, 1994.
- [13] K. C. Tan, Q. G. Wang, T. H. Lee, and D. Khoo, "A Multi-objective Evolutionary Algorithm (MOEA) Toolbox for Matlab," (Freely available on Internet at <http://vlab.ee.nus.edu.sg/~kctan/moea.htm>), 2000.
- [14] C. M. Fonseca, and P. J. Fleming, "Genetic Algorithm for Multiobjective Optimization, Formulation, Discussion and Generalization," In Forrest, S., ed. *Genetic Algorithms: Proc. of the 5th Int. Conf.* Morgan Kaufmann, San Mateo, CA, pp. 416-423, 1993.

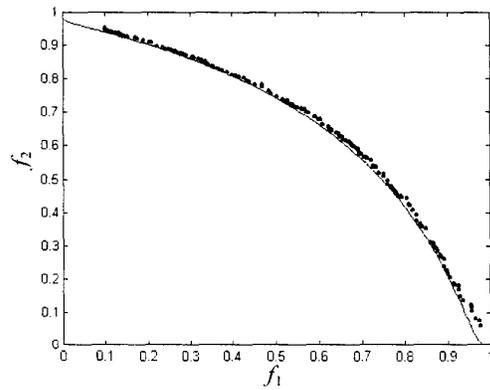


(a) Population distribution in the objective domain

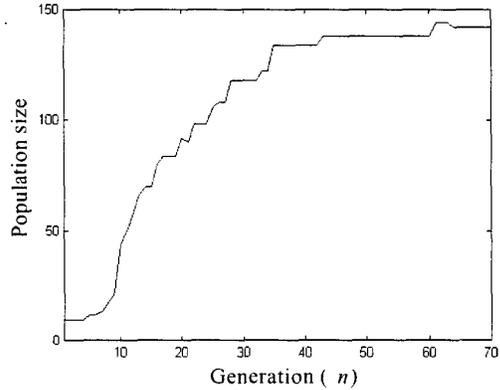


(b) Population size versus generation

Fig. 5 Simulation results for $ppv = 200$

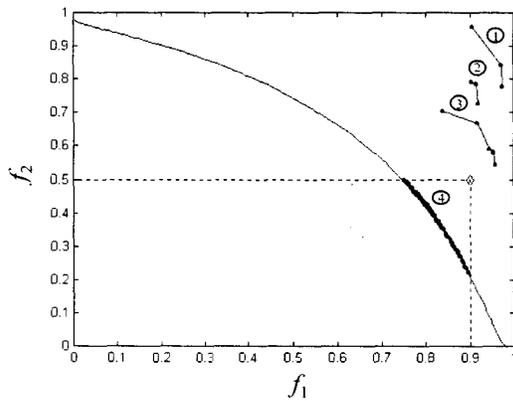


(a) Population distribution in the objective domain

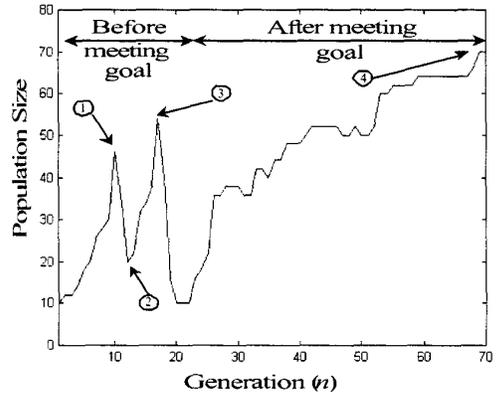


(b) Population size versus generation

Fig. 6 Simulation results for $ppv = 100$



(a) Population distribution in the objective domain



(b) Population size versus generation

Fig. 7 Simulation results for goal specification