An Efficient Design Methodology for the Nondominated Sorted Genetic Algorithm-II

LATE BREAKING PAPER 2003 GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE (GECCO 03), CHICAGO, IL.

Venkat Devireddy [1], Patrick Reed [2]

[1] Department of Industrial and Manufacturing Engineering
The Pennsylvania State University
406B, Sackett Building
University Park, PA 16802
Telephone: 814-865-2342
Email: vkd106@psu.edu
[2] Department of Civil and Environmental Engineering
The Pennsylvania State University
212 Sackett Building
University Park, PA 16802
Telephone: 814-863-2940
Fax: 814-863-7304
Email: preed@engr.psu.edu

Abstract. Many real world problems require careful balancing of fiscal, technical, and social objectives. Informed negotiation and balancing of objectives can be greatly aided through the use of evolutionary multiobjective optimization (EMO) algorithms, which can evolve entire tradeoff (or Pareto) surfaces within a single run. The primary difficulty in using these methods lies in the large number of parameters that must be specified to ensure that these algorithms effectively quantify design tradeoffs. This paper addresses this difficulty by introducing a multi-population design methodology that automates parameter specification for the Nondominated Sorted Genetic Algorithm-II (NSGA-II). The NSGA-II design methodology is successfully demonstrated on four test problems. Using this methodology, multiobjective optimization problems can now be solved automatically with only a few simple user inputs.

1 Introduction

This paper seeks to eliminate the need for trial-and-error parameter analysis by introducing an automated design methodology for the Nondominated Sorted Genetic Algorithm-II [2]. The NSGA-II is the focus of this work because the algorithm has been shown to perform as well or better than other second generation Evolutionary Multiobjective Optimization (EMO) algorithms on difficult, high order problems [1,2]

The NSGA-II design methodology presented in this study builds on previous genetic algorithm (GA) design methodologies [5, 4, 6] to introduce a multi-population approach that automates parameter specification while significantly reducing the computational effort required to solve multiobjective applications. The design methodology fully exploits the efficiency of the NSGA-II to enable the automatic solution of a new class of high order multiobjective applications in which users can select, understand, and balance more than two performance criteria [7]. This paper summarizes the design methodology and verifies the approach on four EMO test problems. Additionally, this paper investigates the efficiency of using small initial populations to evolve Pareto fronts.

This paper is organized in the following manner. Section 2 discusses the design methodology through a series of 3 steps. In Section 3, we present four test problems that have a wide range of properties and have been chosen to demonstrate the efficiency of the design method. Section 4 discusses the results of the application of the design methodology to the test problems, while in Section 5 the scope for future work and the conclusions are discussed.

2 Three-Step Design Methodology

This paper demonstrates an efficient 3-step design methodology for the NSGA-II, a second generation EMO genetic algorithm [2]. The NSGA-II requires the user to specify the four following parameters

- population size
- probability of crossover
- probability of mutation
- run length (number of generations)

The 3-step NSGA-II design method tested in this study reduces the complexity of specifying these parameters, minimizes user interaction, and substantially reduces the computational effort required to solve multiobjective applications. The design methodology assumes that computationally intensive fitness functions for real-world applications preclude identifying parameter settings for a distribution of initial random number seeds and instead focuses on finding optimal parameter settings for a single random number seed.

Step 1 - Preliminary Problem Analysis

Following the approach of *Lobo* [4] the initial population N_0 is set to an arbitrarily small value. Small populations allow the NSGA-II to search for nondominated solutions using a minimum number of function evaluations. *Coello et al*. [1] have shown that EMO algorithms can be used effectively with small population sizes.

Successive increases in population size described in Step 3 will automatically adjust for an undersized population. Successive population increases occur until one of the two conditions are satisfied: (1) the nondominated front has been sufficiently quantified or (2) NSGA-II has used a user specified maximum number of function evaluations.

These stopping criteria are specified by requiring the user to answer 3 basic questions:

1. What is the minimum percentage change in the number of nondominated solutions, Δ_{ND}, for two successive runs to be considered identical?
2. What is the average time required to evaluate a design, T_{eval}?
3. What is the maximum acceptable run time, T_{tot}?

The run length t for each population size tested is estimated to be approximately equal to $2l$, where l is the length of the binary strings that represent the designs, as was recommended by *Thierens et al.* [10].

Step 2 - Probability of crossover and mutation

In step 1, initial population size and run termination criteria are specified. In this step, the probability of crossover P_c and the probability of mutation P_m are set. The value of P_c is set according to the disruption boundary relationship developed by *Thierens* [9],

$$P_c \leq \frac{s-1}{s} \quad . \tag{1}$$

where parameter s is the total number of individuals competing in the tournament selection. Equation (1) is intended to protect pertinent building blocks from being destroyed due to excessive crossover. Since the child populations in the NSGA-II are selected using binary tournaments (i.e., $s = 2$), the probability of crossover is equal to 0.5 or 50 percent. Based on the empirical studies by *De Jong* [3] and *Schaffer et al.* [8] the value of the probability of mutation P_m is set according

$$P_m \approx \frac{1}{N} \quad . \tag{2}$$

Step 3 -Multi population trial runs

Steps 1 and 2 of this design methodology provide all of the parameters required by the NSGA-II to perform trial runs. In a manner analogous to *Lobo* [4] and *Reed et al.* [5], the last step of the design methodology involves automatically initiating trial runs for successively doubled population sizes. A minimum of two runs using two successively doubled population sizes N_0 and $2\ N_0$ are required to determine if the nondominated set has been sufficiently captured (i.e., $\Delta_{ND} \leq$ 10 percent for this

case). If the second run with population size equal to $2\ N_0$ results in greater than a Δ_{ND} percent change in the number of nondominated solutions identified then the population size is again doubled and a third run will be initiated. Offline analysis is used to keep an archive of the nondominated solutions identified across all of the trial runs. The trial runs for successively doubled populations continue until either the nondominated set has been sufficiently captured or the maximum number of function evaluations has been reached.

3 Test Problems

The design methology developed in this paper was tested on four test problems denoted by Test Problem-1 (TP-1), Test Problem-2 (TP-2), Test Problem -3 (TP-3) and Test Poblem-4 (TP-4). The four test problems were chosen to demonstrate the methods efficiency on convex, nonconvex, and discontinuous solution spaces. TP-4 was chosen from Coello et al [1] while the rest of the problems were selected from Zitler et al. [11]. The test functions are given in Table 1.

Table 1. Test problems used to demonstrate the method

Problem	Objective Functions (all are of the minimization type)	Variable bounds
TP-1	$f_1 = x_1$, $f_2 = g(x)\ h(f1(x), g(x))$ *where* $g(x) = 1 + (9/n-1)\ ?\ x_i$ *for* $i = 2$ *to* n $h(f_1, g) = 1 - (f_1/g)^{1/2}$ $n = 30$	$0 = x_i = 1$
TP-2	$f_1 = x_1$, $f_2 = g(x)\ h(f1(x), g(x))$ *where* $g(x) = 1 + (9/n-1)\ ?\ x_i$ *for* $i = 2$ *to* n $h(f_1, g) = 1 - (f_1/g)^2$ $n = 30$	$0 = x_i = 1$
TP-3	$f_1 = x_1$, $f_2 = g(x)\ h(f1(x), g(x))$ *where* $g(x) = 1 + (9/n-1)\ ?\ x_i$ *for* $i = 2$ *to* n $h(f_1, g) = 1 - (f_1/g)^{1/2} - (x_1/g(x))\sin(10?\ x_1)$ $n = 30$	$0 = x_i = 1$

TP-4	$f_1 = x_1$, $f_2 = (1 + 10x_2)[1 - p^a - p\sin(2?qx_1)]$ *where,* $p = (x_1/1+10x_2)$, $q = 4$, $a = 2$	-? $= (x_1, x_2) = ?$

All the problems were solved using an initial population of 10. Problems TP-1, TP-2 and TP-3 had a string length of 240 while TP-4 had a string length of 30. Using the empirical formula that the run length should approximately be equal to twice the string length, the run length for TP-1, TP-2 and TP-3 is set to 480 generations while in TP-4 it is set to 60. The value of $?_{ND}$ was set to 10 %. This implies that the search could be stopped if the difference between the number of nondominated points in two successive runs of the algorithm is less than 10%. The value of $?_{ND}$ was set uniformly for all the test problems though in reality its value may vary depending on the application and the user's discretion.

5 Results and Discussions

The four test problems were successfully solved using the 3-step method discussed in section 2. The results are summarized in Table 2.

The efficacy of the method was verified by comparing the results obtained with those obtained using the parameter setting for NSGA-II recommended by Deb [2]. The runs were carried out for 250 generations and for a fixed population size of 100. The resultant nondominated set was quantified using offline analysis. A probability of crossover P_c equal to 0.9 and a mutation probability P_m equal to 0.01 was used in this test. Deb's parameter settings yielded 256, 240, 71 and 238 nondominated points for test problems TP-1, TP-2 ,TP-3 and TP-4 ,respectively. A total of 25,000 functional evaluations were required in each of the problems in generating these solutions.

Table 2. Results showing the total number of nondominated points and fitness evaluations from implementing our proposed 3-step methodology.

	Number of nondominated points for each population						Total fitness Calculations
Population	10	20	40	80	160	320	
TP-1	253	256	X				14908
TP-2	238	240	X				14878
TP-3	69	71	X				14616
TP-4	4	25	38	213	234	X	19297

In TP-1, 253 non dominated points were obtained using an initial population of 10. Doubling the population size from 10 to 20 resulted in only a 0.79% increase in the number of non dominated solutions evolved, hence the search was terminated and the resulant Pareto curve shown in Figure 1.

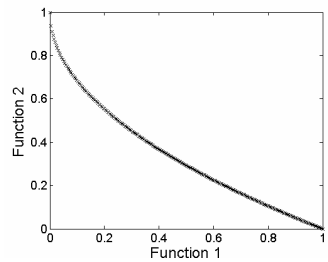

Fig. 1. Test Problem TP-1

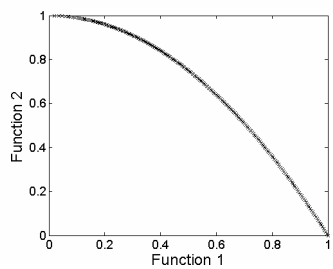

Fig. 2. Test Problem TP-2

For the nonconvex test problem TP-2, 238 and 240 nondominated points were obtained using population sizes of 10 and 20 respectively. Again, we notice that the tradeoff was sufficiently quantified using a very small population. The value of Δ_{ND} in this case is found to be 0.84% . Figure 2 shows the final Pareto curve obtained for TP-2 using the 3-step design method.

The discontinuous Pareto curve for TP-3 is shown in Figure-3. It was evolved using a maximum population size of 20 because doubling the initial population resulted in only a 3% increase in the number of nondominated solutions. (i.e. an increase from 69 to 71 nondominated solutions).

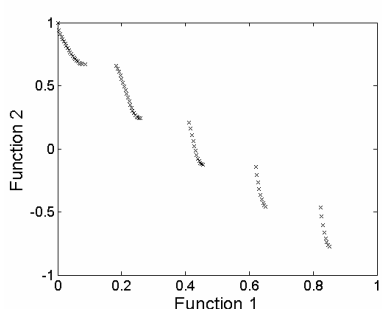

Fig. 3. Test Problem TP-3

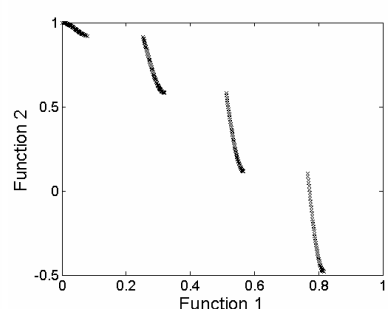

Fig. 4. Test Problem TP-4

Pre-convergence of small populations was prevented for TP-1, TP-2 and TP-3 because of the dynamic mutation rate which this method incorporates. Recall that mutation rate in the method is set as the inverse of the population and when the population is doubled the mutation rate is reduced by half. By starting with a small population, the mutation rate is set to a high value (i.e., $P_m = 0.1$) and the problem of pre-convergence is avoided.

Test problem TP-4 is also a discontinuous function. It can be seen from Table 2 that the initial population is too small, yielding only 4 nondominated solutions. The method automatically corrects the problem with subsequent population increases. As the population was increased to 80 the search yielded 213 points, representing 5 times the number of nondominated solutions obtained using a population size of 40. The search for TP-4 terminated with a population of 160 and a 9.9 percent increase in the number of nondominated solutions.

These results demonstrate the efficiency of our proposed 3-step method in evolving close approximations to the Pareto front. The method was able to sufficiently quantify the tradeoffs for the 4 test problems using significantly fewer than the 25,000 function evaluations required for Deb's recommended parameter settings. Test problems TP-1, TP-2 and TP-3 were solved using less than 15,000 fitness evaluations indicating a relative savings of at least 40% in computational time. Though TP-4 required more fitness evaluations than the other problems, it should be noted that it still saved more than 20% in computational time. Additionally, the automated design methodology proposed in this paper eliminates the need for trial-and-error analysis in applications where Deb's recommended population size of 100 may be too small (e.g., in applications with greater than 2 objectives).

5 Conclusions and Future Work

The method described in this approach builds on the previous parameterless design methodologies [5, 4, 6] by introducing a multipopulation approach that uses explicitly small populations to sufficiently quantify tradeoffs. The simple 3-step approach reduces user interaction and eliminates the need for trial-and-error parameter analysis. The method has been successfully demonstrated on four test problems.

The method utilizes offline analysis, which is a special case of the archiving methodologies currently being recommended. Further work will investigate interactive archives. Also, we will investigate the method's sensitivity to random seed effects.

References

1. Coello Coello, C. A., D. A. Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, New York, NY, 2002.

2. Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons LTD, New York, NY, 2001.

3. DeJong, K. A., *An analysis of the behavior of a class of genetic adaptive systems*, Doctoral dissertation, University of Michigan, Ann Arbor, MI, 1975.

4. Lobo, F., *The Parameter-less Genetic Algorithm: Rational and Automated Parameter Selection for Simplified Genetic Algorithm Operation*, Doctoral Dissertation, Universidad Nova de Lisboa, Lisboa, Portugal, 2000.

5. Reed, P., B. Minsker, and D. E. Goldberg, Designing a competent simple genetic algorithm for search and optimization, *Water Resources Research*, 36(12), 3757-3761, 2000.

6. Reed, P., B. Minsker , and D. E. Goldberg, A multiobjective approach to cost effective long-term groundwater monitoring using an Elitist Nondominated Sorted Genetic Algorithm with historical data, *Journal of Hydroinformatics*, 3(2), 71-90, 2001.

7. Reed, P., *Striking the Balance: Long-term Groundwater Monitoring Design for Multiple Conflicting Objectives*, Doctoral Dissertation, University of Illinois, Urbana, IL, 2002.

8. Schaffer, J. D., R. A. Caruana, L. J. Eshelman, and R. Das, A study of control parameters affecting online performance of genetic algorithms for function optimization, In Schaffer, J. D. (Ed.), Proceedings of the Third International Conference on Genetic Algorithms, p. 51-60, Morgan Kaufmann, San Mateo, CA, 1989.

9. Thierens, D., *Analysis and design of genetic algorithms*, Doctoral dissertation, Katholieke Universiteit Leuven, Leuven, Belgium, 1995.

10. Thierens, D., D. E. Goldberg, and A. G. Pereira, Domino Convergence, Drift, and the Temporal-Salience Structure of Problems, In The 1998 IEEE International Conference on Evolutionary Computation Proceedings, pp. 535-540, IEEE Press, New York, NY, 1998.

11. Zitzler ,E., K. Deb ., and L. Thiele, Comparison of Multiobjective Evolutionary Algorithms: Empirical Results (Revised Version), TIK-Report No. 70, Swiss Federal Institute of Technology (ETH) Zurich , Switzerland, 1999