Messy Genetic Algorithm Based Multi-Objective Optimization: A Comparative Statistical Analysis

Jesse B. Zydallis [1], Gary B. Lamont [1], David A. Van Veldhuizen [2]

[1] Dept of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433, USA
{jesse.zydallis, gary.lamont}@afit.af.mil
[2] Air Force Research Laboratory, Optical Radiation Branch, Brooks AFB, TX 78235, USA
david.vanveldhuizen@brooks.af.mil

Abstract. Many real-world scientific and engineering applications involve finding solutions to "hard" Multiobjective Optimization Problems (MOPs). Genetic Algorithms (GAs) can be extended to find acceptable MOP Pareto solutions. The intent of this discussion is to illustrate that modifications made to the Multi-Objective messy GA (MOMGA) have further improved the efficiency of the algorithm. The MOMGA is a Multiobjective Evolutionary Algorithm (MOEA) extension of the existing single-objective building block (BB) based messy Genetic Algorithm (mGA). The modified MOMGA uses a probabilistic BB approach to initializing the population referred to as Probabilistically Complete Initialization (PCI). This does have the effect of improving the efficiency of the MOMGA through the reduction of the computational bottleneck encountered with the mGA. This paper presents statistical results obtained from the modified MOMGA compared to the results of the original MOMGA as well as those obtained by other MOEAs as employed for a generic test suite.

1 Introduction

We recently developed a Multiobjective Evolutionary Algorithm (MOEA), the Multi-Objective messy GA (MOMGA) that takes a novel approach to solving Multiobjective Optimization Problems (MOPs). The MOMGA extends the notion of building block-based EAs to the MOP domain. Building Blocks (BBs) define chromosomes and contain the information that the EA is attempting to evaluate and move towards the Pareto Front. The BB approach is used in the MOMGA to increase the number of "good" building blocks that are present in each subsequent generation. These "good" building blocks in the current population are exploited. Part of the novel approach of the MOMGA is its extension of the existing single-objective BB-based messy Genetic Algorithm (mGA) to the MOEA domain [3]. The associated test suite indicated that the MOMGA was as good, if not better, than other MOEA approaches for unconstrained problems in a generic test suite [4].

2 Modified MOMGA

A modification to the MOMGA is presented in this paper along with comparative statistical testing results. The previous MOMGA implements a deterministic process to produce the enumeration of all possible BBs, of a specified size, for the initial population referred to as Partially Enumerative Initialization (PEI). This approach is usually not efficient since utilizing the full enumeration is computationally expensive. Thus we modified our MOMGA to use a probabilistic approach to initializing the population referred to as Probabilistically Complete Initialization (PCI) [1]. The probabilistic BB approach initializes the population by creating a controlled number of BB clones of a specified size. These BBs then are filtered to probabilistically ensure that all of the desired BBs are in the initial population. This approach should effectively reduce the computational bottleneck encountered with PEI.

3 Pareto Front Analysis and Results

This paper compares the results that the modified MOMGA obtains with the results that other well-known MOEAs and the original MOMGA achieve for numerous unconstrained test functions. In our previous research on the MOMGA, this MOEA test suite was utilized to allow for absolute comparisons of different MOEA approaches. Thus, this MOEA test suite is also utilized in the comparison of the performance of the modified MOMGA to other popular MOEAs [2,4]. We also evaluate MOPs that contain side constraints for inclusion into the suite based upon an extensive classification of constrained MOPs. Utilization of the test suite is advantageous to the community in the fact that it presents data that is base lined from a standard test suite [4].

These test problems all attempt to find the global maximum or minimum curve or surface, depending on the optimization criteria, through the use of a Pareto Front (PF) analysis. The true optimum values, to the Multiobjective Problem being solved, lie on the true Pareto Front denoted PF_{True}. All MOEAs attempt to find this true Pareto Front but in actuality may not. The values that the MOEA determines to be optimum are referred to as PF_{Known}. In many cases the true global optimum is not found by any EA, however, a “good” analysis of the PF_{Known} values are necessary to determine if they are close enough to the PF_{True} values to be satisfactory solutions to a MOP [2].

Statistical analysis of the experimental results, and observations made are presented. We also quantitatively compare our MOEA performance with others. Since no single metric can represent total performance of an MOEA, a series of appropriate metrics is used to measure the performance in the phenotype domain [4]. The metrics used consist of comparisons of the PF_{Known} values to the PF_{True} values to calculate error ratios, generational distances, maximum PF error, and spacing along the Pareto Front [3]. Genotype metrics are also considered including error ratio, generational distance, spacing, and overall nondominated vector generation measures [2]. Another philosophical development of a test suite reflects similar functionality [5].

The work presented also focuses on complex optimization problems much like those faced by design engineers. Through testing and analysis it is shown that the PCI MOMGA performs as well as or better than the PEI MOMGA and many of the well-known MOEA implementations. These other MOEA implementations include the Multiobjective Genetic Algorithm (MOGA), the Vector Evaluated Genetic Algorithm (VEGA), the Nondominated Sorting GA (NSGA), and the Strength Pareto EA (SPEA) [5]. The modified MOMGA tends to be as effective as the original MOMGA in regard to the fore mentioned unconstrained numerical functions as well as constrained MOPs using the above metrics. An efficiency improvement was anticipated in fewer individual fitness calculations due to the PCI modification. We are continuing to modify the MOMGA to address more complex problems, including discrete optimization NP Complete (NPC) problems, i.e. knapsack, TSP, etc.

4 Conclusions

Through initial testing, the modified MOMGA has shown to be more efficient then its predecessor using a test suite of MOPs [2]. This is as expected since we have moved from a PEI implementation to a PCI initialization of the population and have reduced the computation bottleneck imposed by PEI. Further analysis of constrained test functions for inclusion into our suite is currently being completed. Additional focus on input parameters may yield better performance results as this testing was completed utilizing generic test parameter values. Further testing will center on performance evaluations of the MOMGA on NPC MOPs and real world applicability by increasing the size of the MOPs tested. Preliminary results show that our modifications have increased the efficiency of the MOMGA. Future work will also include parallelizing the PCI MOMGA and completing a statistical analysis of the results.

References

1. Goldberg, D. E., Deb, K., Kargupta, H., and Harik, G. (1993). Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms. Technical Report 93004, University of Illinois at Urbana-Champaign.
2. Van Veldhuizen, D. A. (1999). Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. PhD Dissertation, AFIT/DS/ENG/99-01, Air Force Institute of Technology, Wright-Patterson AFB.
3. Van Veldhuizen, D. A. and Lamont, G. B. (2000). Multiobjective Optimization with Messy Genetic Algorithms. In Proceedings of the 2000 Symposium on Applied Computing, pp. 470-476. ACM.
4. Van Veldhuizen, D. A. and Lamont, G. B. (2000). On Measuring Multiobjective Evolutionary Algorithm Performance. In Proceedings of the 2000 Congress on Evolutionary Computation, pp. 204-211.
5. Zitzler, E. Deb, K., and Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation, 8(2) pp. 173-195.