

Extended Multi-objective fast messy Genetic Algorithm Solving Deception Problems

Richard O. Day and Gary B. Lamont

*Air Force Institute of Technology,
Dept of Electrical and Computer Engineering,
Graduate School of Engineering & Management,
Wright-Patterson AFB (Dayton) OH, 45433, USA
{Richard.Day, Gary.Lamont}@afit.edu

Abstract. Deception problems are among the hardest problems to solve using ordinary genetic algorithms. Designed to simulate a high degree of epistasis, these deception problems imitate extremely difficult real world problems. [1]. Studies show that Bayesian optimization and explicit building block manipulation algorithms, like the fast messy genetic algorithm (fmGA), can help in solving these problems. This paper compares the results acquired from an *extended* multiobjective fast messy genetic algorithm (MOMGA-IIa), ordinary multiobjective fast messy genetic algorithm (MOMGA-II), multiobjective Bayesian optimization algorithm (mBOA), and the non-dominated sorting genetic algorithm-II (NSGA-II) when applied to three different deception problems. The *extended* MOMGA-II is enhanced with a new technique exploiting the fmGA's basis function to improve partitioned searching in both the genotype and phenotype domain. The three deceptive problems studied are: interleaved minimal deceptive problem, interleaved 5-bit trap function, and interleaved 6-bit bipolar function. The *unmodified* MOMGA-II, by design, explicitly learns building block linkages, a requirement if an algorithm is to solve these hard deception problems. Results using the MOMGA-IIa are excellent when compared to the non-explicit building block algorithm results of both the mBOA and NSGA-II.

1 Introduction

Algorithms that solve problems by realizing good building blocks (BBs) are useful in solving extremely difficult problems: Protein Structure Prediction [2], 0/1 Modified Knapsack [3], Multiple Objective Quadratic Assignment Problem [4, 5], Digital Amplitude-Phase Keying Signal Sets with M-ary Alphabets and many academic problems [6, 7]. MOMGA-IIa originated as a single objective

* The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

messy GA (mGA). It evolved from being a single objective mGA into a multi-objective mGA called the MOMGA [8]. Many different MultiObjective Evolutionary Algorithms (MOEAs) were produced during this time period; however, the MOMGA is the only MOEA explicitly using good BBs to solve problems – even the Bayesian optimization algorithm (BOA) uses a probabilistic model to find good building blocks. The MOMGA has a population size limitation: as the BB size increases so does the population size during the Partially Enumerative Initialization (PEI) phase. This renders the MOMGA less useful on large problems. To overcome this problem, the MOMGA-II, based on the single objective fmGA, is designed. The fmGA is similar to the mGA in that it specifically uses BBs to find solutions; however, it has a reasonable population size and lower run time complexity (See Table 1) when compared to the mGA. MOMGA-II includes many different repair, selection, and crowding mechanisms. Unfortunately, the MOMGA-II is found to be limited when solving large deception problems [6]. This called for the development of basis function diversity measures in the MOMGA-IIa which are designed for smart BB searching in both the geno- and pheno-type domains. Also discussed in this investigation is the mBOA and the NSGA-II [7] neither of which compares well to MOMGA-IIa results.

The next section discusses in detail the MOMGA-II and MOMGA-IIa algorithm domains. In addition, a short description of the mBOA and the NSGA-II is provided. The mBOA and NSGA-II have been used to solve these three multiobjective problems (MOPs) in previous research [10, 6]. The three MOPs are then described in detail in Section 3. Next, experimental design, resources, parameter settings, and algorithm efficiency are discussed briefly in Section 4. Finally, in the results section, the mBOA, NSGA-II, MOMGA-II and MOMGA-IIa results are compared and analyzed.

Table 1. Complexity Estimates for serial GAs

	Single Objective Algorithm				Multiple Objective Algorithm		
<i>Phase</i>	<i>sGA</i> ^a	<i>ssGA</i> ^b	<i>mGA</i>	<i>fmGA</i>	<i>NSGA-II</i>	<i>mo-BOA</i>	<i>MOMGA-IIa</i>
Initialization	$O(l^n)$	$O(l^n)$	$O(l^k)$	$O(l)$			
Recombination	$O(gnq)$	$O(g)$					
Primordial	$O(\emptyset)$		$O(\emptyset)$	$O(l^2)^c$			
Juxtapositional			$O(l \log l)$	$O(l \log l)$			
Overall	$O(l^n)$	$O(l^n)$	$O(l^k)$	$O(l^2)$	$O(mn^3)^d$	$O(n^{3.5})^e$	$O(megn^2)^f$

^a l is the length of chromosome, n is the size of population, q is group size for tournament selection, g is the number of generations.

^b l is the length of chromosome, n is the size of population, g is the number of generations of reproduction.

^c Building Block Filtering

^d m is the number of objectives

^e This complexity is problem specific and in this case has been taken from the spin glass problem.[9]

^f e = number of eras, g = max number of generations