

Improved evolutionary optimization from genetically adaptive multimethod search

Jasper A. Vrugt* and Bruce A. Robinson

Los Alamos National Laboratory, EES-6, Mail Stop T003, Los Alamos, NM 87545

Communicated by Stirling A. Colgate, Los Alamos National Laboratory, Los Alamos, NM, November 27, 2006 (received for review March 29, 2006)

In the last few decades, evolutionary algorithms have emerged as a revolutionary approach for solving search and optimization problems involving multiple conflicting objectives. Beyond their ability to search intractably large spaces for multiple solutions, these algorithms are able to maintain a diverse population of solutions and exploit similarities of solutions by recombination. However, existing theory and numerical experiments have demonstrated that it is impossible to develop a single algorithm for population evolution that is always efficient for a diverse set of optimization problems. Here we show that significant improvements in the efficiency of evolutionary search can be achieved by running multiple optimization algorithms simultaneously using new concepts of global information sharing and genetically adaptive offspring creation. We call this approach a multialgorithm, genetically adaptive multiobjective, or AMALGAM, method, to evoke the image of a procedure that merges the strengths of different optimization algorithms. Benchmark results using a set of well known multiobjective test problems show that AMALGAM approaches a factor of 10 improvement over current optimization algorithms for the more complex, higher dimensional problems. The AMALGAM method provides new opportunities for solving previously intractable optimization problems.

evolutionary search | multiple objectives | optimization problems | Pareto front

Evolutionary optimization is a subject of intense interest in many fields of study, including computational chemistry, biology, bioinformatics, economics, computational science, geophysics, and environmental science (1–8). The goal is to determine values for model parameters or state variables that provide the best possible solution to a predefined cost or objective function, or a set of optimal tradeoff values in the case of two or more conflicting objectives. However, locating optimal solutions often turns out to be painstakingly tedious, or even completely beyond current or projected computational capacity (9).

Here, we consider a multiobjective minimization problem, with n decision variables (parameters) and m objectives: $\mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$, where \mathbf{x} denotes the decision vector, and \mathbf{y} is the objective space. We restrict attention to optimization problems in which the parameter search space X , although perhaps quite large, is bounded: $\mathbf{x} = (x_1, \dots, x_n) \in X$. The presence of multiple objectives in an optimization problem gives rise to a set of Pareto-optimal solutions, instead of a single optimal solution (10, 11). A Pareto-optimal solution is one in which one objective cannot be further improved without causing a simultaneous degradation in at least one other objective. As such, they represent globally optimal solutions to the tradeoff problem.

Numerous approaches have been proposed to efficiently find Pareto-optimal solutions for complex multiobjective optimization problems (12–15). In particular, evolutionary algorithms have emerged as the most powerful approach for solving search and optimization problems involving multiple conflicting objectives. Beyond their ability to search intractably large spaces for multiple Pareto-optimal solutions, these algorithms are able to maintain a diverse set of solutions and exploit similarities of

solutions by recombination. These attributes lead to efficient convergence to the Pareto-optimal front in a single optimization run (13). Of these, the nondominated sorted genetic algorithm II (NSGA-II) (14) has received the most attention because of its simplicity and demonstrated superiority over other methods.

Although the multiobjective optimization problem has been studied quite extensively, current available evolutionary algorithms typically implement a single algorithm for population evolution. Reliance on a single biological model of natural selection and adaptation presumes that a single method exists that efficiently evolves a population of potential solutions through the parameter space. However, existing theory and numerical experiments have demonstrated that it is impossible to develop a single algorithm for population evolution that is always efficient for a diverse set of optimization problems (16).

In recent years, memetic algorithms (also called hybrid genetic algorithms) have been proposed to increase the search efficiency of population based optimization algorithms (17). These methods are inspired by models of adaptation in natural systems, and use a genetic algorithm for global exploration of the search space, combined with a local search heuristic for exploitation. Memetic algorithms have shown to significantly speed up the evolution toward the global optimal solution for a variety of real-world optimization problems. However, our conjecture is that a search procedure that adaptively changes the way it generates offspring, based on the shape and local peculiarities of the fitness landscape, will further improve the efficiency of evolutionary search. This approach is likely to be productive because the nature of the fitness landscape (objective functions mapped out in the parameter space, also called the response surface) often varies considerably between different optimization problems, and dynamically changes en route to the global optimal solutions.

Drawing inspiration from the field of ensemble weather forecasting (18), we present an innovative procedure employing genetically adaptive evolutionary optimization. The method combines two concepts, simultaneous multimethod search, and self-adaptive offspring creation, to ensure a fast, reliable, and computationally efficient solution to multiobjective optimization problems. We call this approach a multi-algorithm, genetically adaptive multiobjective, or AMALGAM, method, to evoke the image of a procedure that blends the attributes of the best available individual optimization algorithms.

To successfully implement the AMALGAM method, three questions need to be addressed. First, how can we best make multiple algorithms meaningfully communicate with one another, and share their information? Second, what is the most effective method for adaptive offspring creation? Finally, which

Author contributions: J.A.V. designed research; J.A.V. performed research; J.A.V. and B.A.R. analyzed data; and J.A.V. and B.A.R. wrote the paper.

The authors declare no conflict of interest.

Freely available online through the PNAS open access option.

*To whom correspondence should be addressed. E-mail: vrugt@lanl.gov.

This article contains supporting information online at www.pnas.org/cgi/content/full/0610471104/DC1.

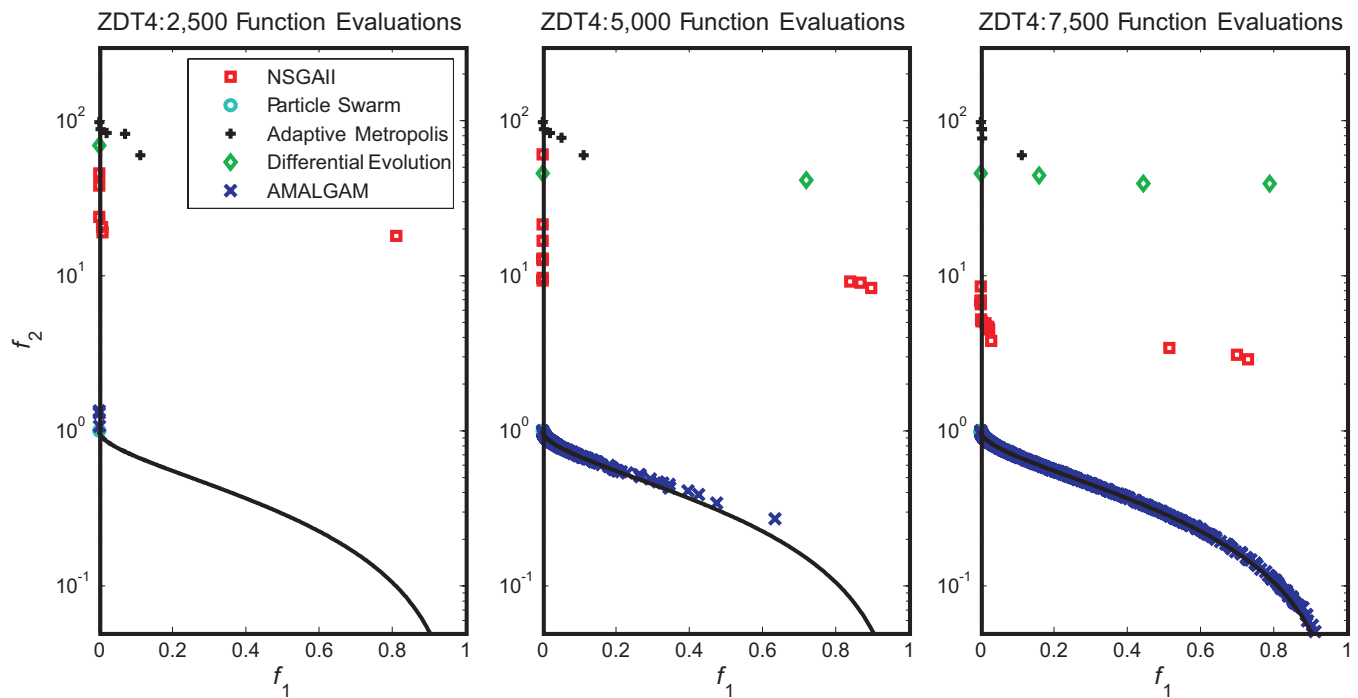


Fig. 1. Generated Pareto-optimal fronts after 25, 50, and 75 generations with the NSGA-II (squares), PSO (circles), AMS (+), DE (diamonds), and AMALGAM (x) optimization algorithms for test problem ZDT4. This benchmark problem has 21^9 different local Pareto-optimal fronts in the search space, of which only one corresponds to the global Pareto-optimal front. Combining the individual algorithms into a simultaneous multimethod search algorithm ensures a faster and more reliable solution to multiobjective optimization problems.

individual algorithms should be included? These issues will be confronted below.

Materials and Methods

Our multimethod evolutionary optimization implements a population-based elitism search procedure to find a well distributed set of Pareto solutions within a single optimization run. The basic algorithm is presented in supporting information (SI) Fig. 4 (see *SI Text*), and is described below.

The algorithm is initiated by using a random initial population P_0 of size N , generated by using Latin hypercube sampling. Then,

each parent is assigned a rank using the fast nondominated sorting (FNS) algorithm (14). A population of offspring Q_0 , of size N , is subsequently created by using the multimethod search concept that lies at the heart of the AMALGAM method. Instead of implementing a single operator for reproduction, we simultaneously use k individual algorithms to generate the offspring, $Q_0 = \{Q_0^1, \dots, Q_0^k\}$. These algorithms each create a prespecified number of offspring points, $N = \{N_i^1, \dots, N_i^k\}$, from P_0 using different adaptive procedures. After creation of the offspring, a combined population $R_0 = P_0 \cup Q_0$ of size $2N$ is created and R_0 ranked using FNS. By comparing the current

Table 1. Number of function evaluations needed to achieve convergence, and values of the convergence metric Υ , and diversity measure Δ after 150 generations for the 10 different test functions considered in this study

Function	Number of function evaluations*		After 150 generations			
	NSGA-II	AMALGAM	Convergence metric: Υ		Diversity metric: Δ	
			NSGA-II	AMALGAM	NSGA-II	AMALGAM
ZDT1	5,920	840	0.0053	0.0011	0.34	0.33
ZDT2	7,680	1,240	0.0068	0.0009	0.36	0.35
ZDT3	5,100	1,090	0.0027	0.0010	0.56	0.55
ZDT4	13,050 (24) [†]	6,320	0.0523	0.0022	0.73	0.32
ZDT6	8,940	950	0.0504	0.0011	0.53	0.40
SCH	6,350 (25)	3,610	0.0036	0.0032	0.49	0.37
SCH2	12,000 (22)	3,320	0.0052	0.0041	0.69	0.52
FON	4,930	1,590	0.0026	0.0017	0.38	0.33
KUR	6,544	6,122	0.0108	0.0099	0.48	0.47
ROT	N/C [‡]	9,440	1.50	0.35	1.14	0.55

Results for the NSGA-II and AMALGAM methods are presented. Statistics represent averages over 30 optimization runs. A detailed description of the various test problems and performance metrics appears in *SI Text*.

*Average number of function evaluations needed to result in a relative hypervolume smaller than 0.005.

[†]Number of runs that have failed to converge are shown in parentheses.

[‡]None of the 30 optimization runs have converged after 150 generations.

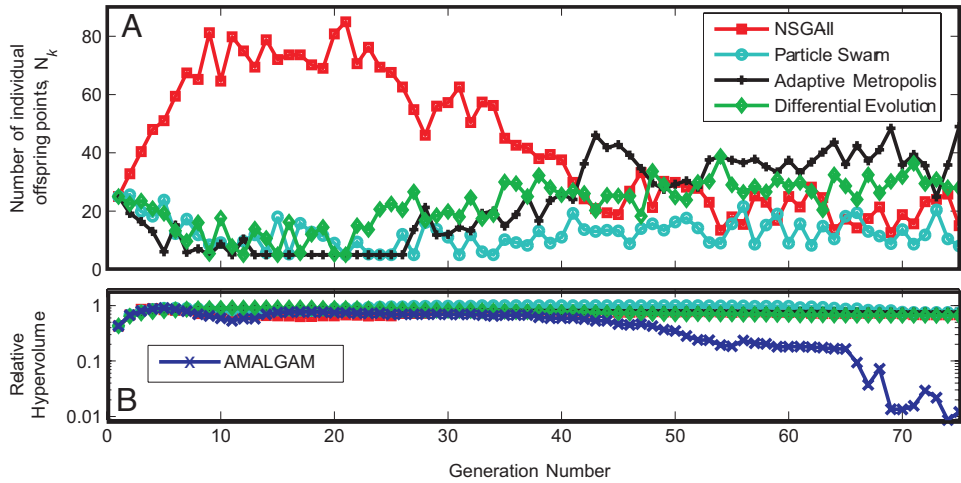


Fig. 2. Illustration of the concept of self-adaptive offspring creation. (A) Evolution of the number of offspring points generated with the NSGA-II (squares), PSO (circles), AMS (+), and DE (diamonds) algorithms within AMALGAM’s multimethod search as function of generation number for test problem ZDT4. (B) The hypervolume convergence metric for AMALGAM and each algorithm used individually. These results illustrate the utility of individual search algorithms during different stages of the optimization, and provide numerical evidence of Wolpert and Macready’s “No Free Lunch” theorem, showing that it is impossible to develop a single search algorithm that will always be superior to any other algorithm (16).

offspring with the previous generation, elitism is ensured because all previous nondominated members will always be included in R (12–14). Finally, members for the next population P_1 are chosen from subsequent nondominated fronts of R_0 based on their rank and crowding distance (14). The new population P_1 is then used to create offspring using the method described below, and the aforementioned algorithmic steps are repeated until convergence is achieved.

Our method for adaptive offspring creation is designed to favor individual algorithms that exhibit the highest reproductive success. To ensure that the “best” algorithms are weighted so that they contribute the most offspring to the new population, we update $\{N_t^1, \dots, N_t^k\}$ according to $N_t^i = N \cdot (P_t^i / N_{t-1}^i) / \sum_{i=1}^k (P_t^i / N_{t-1}^i)$. The term P_t^i / N_{t-1}^i is the ratio of the number of offspring points an algorithm contributes to the new population, P_t^i , and the corresponding number the algorithm created in the previous generation (N_{t-1}^i). The rest of the expression scales the reproductive success of an individual algorithm to the combined success of all of the algorithms. In this study, the minimum values for $\{N_t^1, \dots, N_t^k\}$ were

set to 5, to avoid the possibility of inactivating algorithms that may contribute to convergence in future generations.

The final issue is the decision of which individual algorithms to include in the search. In principle, the AMALGAM method is very flexible, and could accommodate any biological model for population evolution. Here we implement the NSGA-II (14), particle swarm optimization (PSO) (19), adaptive metropolis search (AMS) (20), and differential evolution (DE) (21) algorithms. These choices are based on the outcome of numerical experiments demonstrating that these four commonly used optimization methods are mutually consistent and complementary. A detailed description of the individual algorithms and their algorithmic parameters is presented in *SI Text*.

We anticipate two advantages of the AMALGAM method. First, by facilitating direct information exchange between individual algorithms, the method merges the strengths of different search strategies to increase the speed of evolution toward the Pareto-optimal solutions. Second, by adaptively changing preference to individual search algorithms during the course of the

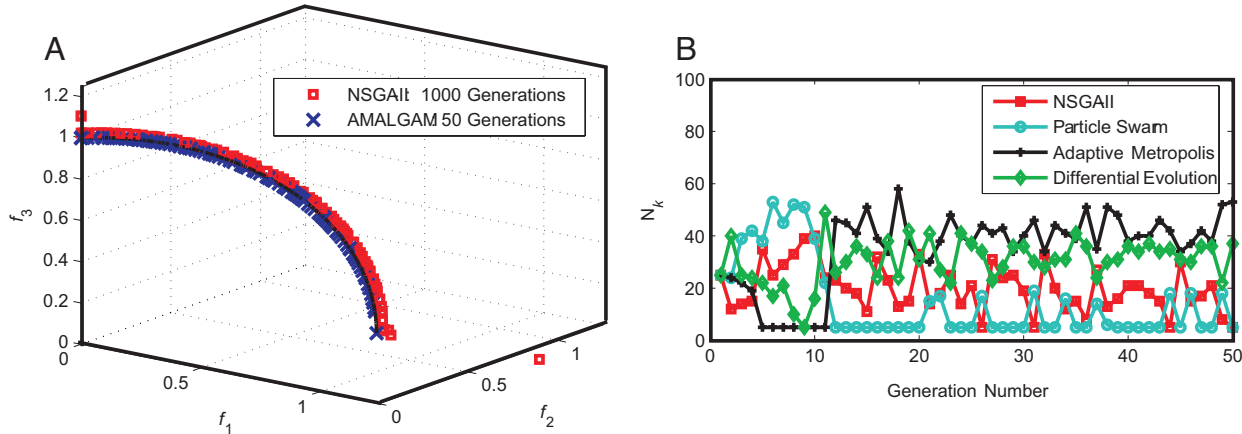


Fig. 3. Nondominated solutions found for test problem DTLZ6 (23) with NSGA-II and AMALGAM after 100,000 and 5,000 function evaluations, respectively. (A) The dark line marks the true Pareto-optimal front. Although classical multiobjective methods have difficulty finding solutions on the Pareto front, the AMALGAM method perfectly converges to the true solution set in far fewer function evaluations. (B) Self-adaptive offspring creation by varying the relative importance of the individual algorithms during the optimization is the breakthrough that enables this improvement.

optimization, the method should adapt quickly to the specific difficulties and peculiarities of the optimization problem at hand.

We conducted a wide range of numerical experiments using a set of well known multiobjective benchmark problems. SI Table 2 provides a detailed description and definition of the selected test functions. These functions cover a diverse set of problem features, including high-dimensionality, convexity, nonconvexity, multimodality, isolated optima, nonuniformity, and interdependence. For a given test, each optimization run was repeated 30 times using a population size of 100 points in combination with 150 generations, and average results are reported.

Results and Discussion

To demonstrate the advantages of multimethod optimization, consider Fig. 1, which shows the evolution of the nondominated fronts generated with the individual NSGA-II (squares), PSO (circles), AMS (+) and DE (diamonds) algorithms, and AMALGAM (x) method for test problem ZDT4. SI Movies 1 and 2 show this and another test problem (ROT). In each snapshot during the evolution, the dark line depicts the location of the true Pareto-optimal front. The results highlight the advantages of adaptive multimethod evolutionary search. After only 7,500 function evaluations, AMALGAM has progressed toward the true Pareto-optimal front, and has generated solutions that are far more evenly distributed along the Pareto front than any of the individual algorithms.

This improved performance is quantified in Table 1, which compares convergence statistics over 30 different optimization runs for the NSGA-II and AMALGAM methods for all benchmark problems considered. The metrics, described in detail in *SI Text*, measure the extent of convergence to a known set of Pareto-optimal solutions (Y), and the uniformity (or spread) of the solutions within this distribution (Δ). Smaller values for both metrics indicate better performance.

The improvement of the AMALGAM method over the NSGA-II algorithm is significant for all of the benchmark studies considered. The results in Table 1 demonstrate that AMALGAM is significantly more efficient in locating Pareto-optimal solutions than the current state-of-the-art NSGA-II algorithm, with efficiency gains approaching a factor 10 for the more complex, higher dimensional problems (ZDT1-ZDT6 and ROT). The new method even converges for the rotational problem (ROT) within 150 generations, indicating that our multimethod search can deal with correlated decision variables that classical genetic mutation and selection operators such as the NSGA-II algorithm have difficulty handling.

Fig. 2 depicts AMALGAM's evolution of the number of offspring points of the individual algorithms for test problem ZDT4 (Fig. 2A). This plot illustrates why the multimethod optimization exhibits superior performance. During the first part

of the optimization, the NSGA-II algorithm (squares) exhibits the highest reproductive success, owing to the proficiency of its classical genetic operators for crossover and mutation for global optimization. However, after ≈ 20 generations, the utility of the NSGA-II algorithm abruptly decreases in favor of (in sequential order) the DE (diamonds), AMS (+), and PSO (circles) algorithms. This combination of methods proves to be extremely effective at increasing the diversity of solutions along the Pareto front once the NSGA-II method does its initial work. This result confirms our conjecture that an adaptive strategy of switching algorithms to maintain efficiency during all stages of the optimization will outperform any individual algorithm, and provides strong support for the use of multimethod evolutionary search. The performance of AMALGAM on the other benchmark problems provides further justification for this conclusion.

Although two-objective problems provide a demonstration of the advantages of multimethod evolutionary optimization, it is desirable to investigate the performance of the AMALGAM method for higher dimensional problems. To this end, we examine the three objective, 12 parameter test problem DTLZ6 described in ref. 22, for which it is reported that existing evolutionary algorithms fail to locate solutions on the true Pareto front. Fig. 3A presents the theoretical Pareto-optimal curve and the optimization results for the NSGA-II and AMALGAM methods after 50 generations. The AMALGAM method locates the Pareto-optimal solution set in $\approx 5,000$ function evaluations, whereas the NSGA-II algorithm is unable to exactly find the Pareto set, even after 100,000 iterations. The evolution of the number of offspring points for the four individual algorithms in AMALGAM (Fig. 3B) again illustrates the virtue of genetically adaptive multimethod search.

The results presented herein illustrate that multimethod evolutionary optimization with adaptive offspring creation is a powerful new approach for solving complex optimization problems. This finding has some wider implications that go beyond the multiobjective test problems studied here. First, within the optimization and biological realm, our self-adaptive multimethod search provides important new ways to study evolutionary processes. Second, our results demonstrate that competition between individual algorithms and adaptive offspring creation dramatically improves the efficiency of evolutionary search. Combined with anticipated increases in computational power, the AMALGAM method should provide new opportunities for solving previously intractable optimization problems. Our next step is to apply AMALGAM to real world search and optimization problems.

We thank Dr. Steen Rasmussen, Velimir Vesselinov (Los Alamos National Laboratory; LANL), Dr. Patrick Reed (Penn State University, University Park, PA), and the three anonymous reviewers for comments and suggestions. J.A.V. is supported by the LANL Director's Funded Postdoctoral program.

1. Wales DJ, Scheraga HA (1999) *Science* 285:1368–1372.
2. Nowak MA, Sigmund K (2004) *Science* 303:793–799.
3. Lemmon AR, Milinkovitch MC (2002) *Proc Natl Acad Sci USA* 99:10516–10521.
4. Glick M, Rayan A, Goldblum A (2002) *Proc Natl Acad Sci USA* 99:703–708.
5. Bounds DG (1987) *Nature* 329:215–219.
6. Barhen J, Protopoulos V, Reister D (1997) *Science* 276:1094–1097.
7. Schoups GH, Hopmans JW, Young CA, Vrugt JA, Wallender WW, Tanji KK, Panday S (2005) *Proc Natl Acad Sci USA* 102:15352–15356.
8. Holland J (1975) *Adaptation in Natural and Artificial Systems* (MIT Press, Cambridge, MA).
9. Achlioptas D, Naor A, Peres Y (2005) *Nature* 435:759–763.
10. Goldberg DE (1989) *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, Reading, MA).
11. Deb K (2001) *Multi-Objective Optimization Using Evolutionary Algorithms* (Wiley, New York).
12. Zitzler E, Deb K, Thiele L (2000) *Evol Comp* 8:173–195.
13. Zitzler E, Thiele L (1999) *IEEE Trans Evol Comp* 3:257–271.
14. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) *IEEE Trans Evol Comp* 6:182–197.
15. Knowles J, Corne D (1999) *Proc 1999 Conf Evol Comp* (IEEE Press, New York).
16. Wolpert DH, Macready WG (1997) *IEEE Trans Evol Comp* 1:67–82.
17. Hart WE, Krasnogor N, Smith JE (2005) *Recent Advances in Memetic Algorithms* (Springer, Berlin).
18. Gneiting T, Raftery AE (2005) *Science* 310:248–249.
19. Kennedy J, Eberhart RC, Shi Y (2001) *Swarm Intelligence* (Morgan Kaufmann, San Francisco).
20. Haario H, Saksman E, Tamminen J (2001) *Bernoulli* 7:223–242.
21. Storn R, Price K (1997) *J Global Optimization* 11:341–359.
22. Deb K, Thiele L, Laumanns M, Zitzler E (2001) *KanGAL Rep*, 2001001.