Efficiency Enhancement of ECGA Through Population Size Management

Vinicius V. Melo
Inst. Math. and Comp. Science
University of Sao Paulo
Sao Carlos, SP, Brazil
vmelo@icmc.usp.br

Thyago S.P.C. Duque
Illinois Genetic Alg. Lab.
thyago@illigal.ge.uiuc.edu

Alexandre C.B. Delbem
Inst. Math. and Comp. Science
University of Sao Paulo
Sao Carlos, SP, Brazil
acbd@icmc.usp.br

Abstract

This paper describes and analyzes population size management, which can be used to enhance the efficiency of the extended compact genetic algorithm (ECGA). The ECGA is a selectorecombinative algorithm that requires an adequate sampling to generate a high-quality model of the problem. Population size management decreases the overall running time of the optimization process by splitting the algorithm into two phases: first, it builds a high-quality model of the problem using a large population; second, it generates a smaller population, sampled using the high-quality model, and performs the remaining of the optimization with a reduced population size. The paper shows that for decomposable optimization problems, population size management leads to a significant optimization speedup that decreases the number of evaluations for convergence in ECGA by a factor of 30% to 70% keeping the same accuracy and reliability. Furthermore, the ECGA using PSM presents the same scalability model as the ECGA.

1. Introduction

A Genetic Algorithm [2] that can solve hard problems quickly, accurately and reliably is called a Competent GA. Several Competent GAs have been proposed in the last years including the Gene Expression Messy GA (gemGA), the Linkage Learning GA (LLGA), the Extended Compact GA (ECGA), the Bayesian Optimization Algorithm (BOA) [2], the Factorized Distribution Algorithm (FDA) [6], among others. One key characteristic present on several competent GAs is the ability to identify Building Blocks (BBs) [2], which are of strongly interacting variables that should be processed simultaneously. This information is used to perform recombination in a non-disruptive way, thereby enabling these methods to solve deceptive problems using a polynomial number of function evaluations [2].
Particularly, the ECGA performs the BB identification via a probabilistic model building process. The quality of the probabilistic model is crucial for the performance of the ECGA, since a better model results in a better use of the genetic material of the population. The model quality can be enhanced using a large sample for the construction of the model. However, if the size of the population is inadequately large, the computational cost to evaluate the population becomes unnecessarily high. Therefore, it is important to properly size the population, by balancing Quality and Performance.

Population size theories [2] argues that using a larger population size, there is a higher guarantee of finding the global optimum than using of a smaller population size. Also, the use of a smaller population size implicates the needing of a higher number of generations to achieve the global optimum, while a larger population needs less generations. Thus, the number of evaluations of both strategies can be close.

This paper introduces an Efficiency-Enhancement Technique (EET) called Population Size Management (PSM), which can significantly speed up the optimization process of ECGA and other genetic and evolutionary algorithms by considerably reducing the total number of evaluations by a factor of 30% to 70% while keeping the same accuracy and reliability. Furthermore, the ECGA using PSM presents the same scalability model as the one presented in [10].

The paper is organized as follows. Section 2 introduces the ECGA, describing relevant modeling aspects. Section 3 proposes the PSM. Section 4 presents experimental results showing the reduction of the population size and evaluations required by the ECGA to solve the test problems. Finally, Section 5 summarizes and concludes the paper.

2. Extended Compact Genetic Algorithm

This section describes the ECGA and analyzes the population size and number of evaluations required for convergence. Section 2.1 presents the ECGA's probabilistic model. Section 2.2 shows the basic procedure of an ECGA.

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.250

Section 2.3 presents the theory related to the required population size and number of function evaluations required by the ECGA. Section 2.4 presents the Bisection Method used to find the Minimum Population Size. Finally, Section 2.5 presents EETs developed by other researchers.

2.1. Probabilistic Model

As a probabilistic model building genetic algorithm (PMBGA) [2], ECGA replaces the traditional variation operators of GAs by building a probabilistic model of promising solutions and sampling the model to generate new candidate solutions.

To build an accurate probabilistic model, it is important to detect the interaction among the variables. The detection process is called linkage learning [5]. In the ECGA's approach, the linkage learning corresponds to determination of an adequate Marginal Probability Model (MPM), which is a model with BB information and a probability distribution function of the bits in the population. The functions are evaluated using the minimum description length (MDL) metric. The key concept of this metric is that, given two distributions with very similar behavior, the simpler distributions is preferred over the more complex one. By penalizing both inaccurate and complex models, the MDL can be used to guide a search process to a proper probability distribution. Thus, finding a good distribution can be seen as an optimization problem with two objective functions: minimize the complexity of the probability model description and maximize the model accuracy. More details can be found in [5].

2.2. Basic ECGA Procedure

The ECGA evolves a population of candidate solutions represented by fixed-length strings over a finite alphabet (for example, binary strings). The initial population is randomly generated according to a uniform distribution over the set of all potential solutions. Each iteration (generation) starts by selecting promising solutions from the current population using any standard selection method of genetic or evolutionary algorithms as, for example, tournament selection [1]. From the promising solutions, the ECGA builds an MPM. Then, a new population is created by sampling from the MPM. A next iteration is executed, unless a predefined termination criterion is met. Algorithm 1 synthesizes the ECGA.

Algorithm 1 Algorithm of the ECGA.

```
1. Initialization: use a random uniform binary distribution;
2. Evaluation of the fitness value of the individuals;
3. Selection: the ECGA usually employs s-wise tournament selection;
4. Construction of the new probabilistic model: both the structure and the parameters of the MPM are determined using a greedy search heuristic;
5. Generation of new individuals: new individuals are created by sampling from the MPM;
6. Replacement of the parents by the offspring;
7. Repetition of Steps 2-6 until a termination criterion is met.
```

2.3. Population Size Theory

The success of populational optimization algorithms, like the GA, is highly influenced by the choice of an adequate population size. On the one hand, a small population size provides a small mapping of the search space, possibly resulting in a premature convergence. On the other hand, a large population size will considerably increase the computational effort. Therefore, the study of models for prediction of adequate population sizes for different problems is an important key in the development of a GA theory.

For problems in which the BB identification is necessary, Goldberg, Deb, and Clark [3] proposed population-sizing models for correctly deciding between competing BBs by incorporating noise information arising from other BBs. Using this approach, they have concluded that a failure in the correct identification of the BBs in the first generation could be detrimental to the overall solution quality, preventing the algorithm from converging to the correct solution.

Pelikan, Sastry, and Goldberg [7] showed that the population size (*n*) required for the BOA to build an accurate model scales polynomially with the problem size, with a degree *D* of the order between 1.05 and 2.1 ($\Theta(m^{1.05}) \leq n \leq \Theta(m^{2.1})$, where m is the number of BBs). These bounds also apply to many other model-building GAs. Empirical results show that *n* roughly scales as $\Theta(m^{1.4})$ [9]. As high is the problem size, the closer to 1 is the value of *D*.

However, based on empirical results that indicate the selection pressure affects the population size, Yu et. al. [10] lately demonstrated that the population size that the ECGA requires to solve a problem with m BBs of size k with a failure rate of $\alpha = \frac{1}{m}$ is given by

$$n \geqslant c\frac{s_{to}}{ln^2(s_{to}/1.6)}2^{2k}mlog_2(m)(\frac{\sigma_{BB}^2}{d^2}), \tag{1}$$

where c is a positive constant, s_{to} is the tournament size, σ_{BB} is fitness-variance of a BB and d is the signal difference between competing BBs. Thus, the models predict that ECGA's population size scales subquadratically with the number of BBs, estimated as $\Theta(mlog_2(m))$. It is important to note that this model is valid for additively-separable problems. To experimentally find the minimum population size needed to solve a problem, the bisection method can be used, as described in Section 2.4.

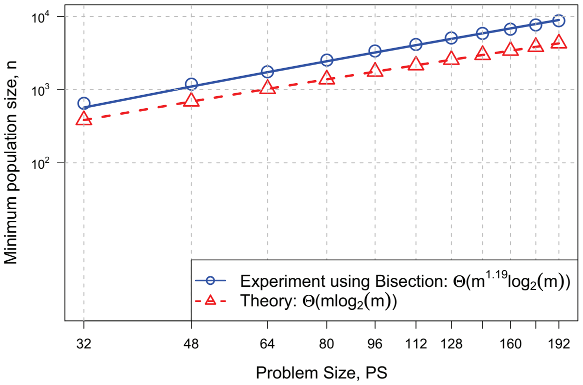

Figure 1. Population size: model versus experiment for the deceptive function from Equation 3 (Section 4.1), using $k = 4$

2.4. Finding the Minimum Population Size using the Bisection Method

The bisection method [8] can determine the minimum population size required for the ECGA such that the number of correctly converged BBs, with failure probability $\alpha = \frac{1}{m}$, is at least $m-1$. According to the population-sizing theory, the estimated size tends to $\Theta(m^D log_2(m))$, with $D = 1$. This result has been verified in experiments using up to a billion of variables [4]. However, for smaller problems, the population size predicted by the bisection method is close to $\Theta(m^{1.2} log_2(m))$ (see Figure 1).

3. Population Size Management

To understand the principle behind the PSM, it is important to understand how the ECGA processes the population and why they need large populations. Two decision-wise population sizing models have been studied by researchers [2]: The generation-wise model and the gambler's ruin model. The gambler's ruin is less conservative and results in a better agreement with empirical data for simple GAs. However, for multivariate PMBGAs, the generation-wise model results in a better agreement. This model assumes that the decisions have to be made correctly in the first generation. The decisions made in the first generation highly bias the search on subsequent generations. As a consequence, the population used on the first generation has to be large enough to support the correct decisions. The idea for PSM is derived from this observation.

As presented before, the quality of the probabilistic model is crucial for the performance of the ECGA [2]. Thus, the PSM must guarantee the generation of a high-quality model in the very first generation. The high-quality of the first model focuses the search process in the subsequent iterations on promising regions, since the search space is now reduced. A reduced population size can be used for the second generation. The procedure can go on, reducing the population even more in the following generations. The use of a reduced population size may affect the quality of the generated model; however, the new models tend to be close to the high-quality model found in the first generation. Thus, the model built with the reduced population will also be a high-quality model, consequence of the refinement of the search in the promising region.

Algorithm 2 ECGA with PSM.

```
Phase 1
1: Initialization: the population is usually initialized with random individuals – use a large population size;
2: Evaluation of the fitness value of the individuals;
3: Selection: the ECGA usually employs s-wise tournament selection;
4: Construction of the new probabilistic model: both the structure and the parameters of the MPM are determined using a greedy search heuristic;
Phase 2
5: Generation of new individuals: new individuals are created by sampling from the probabilistic model – now, use a reduced population size;
6: Replacement of the parents by the offspring;
7: Repetition of Steps 2–6 until a termination criterion is met.
```

To build the high-quality model in the first generation, we wisely provide a large enough sample of the search space. The minimal population size needed to obtain a high-quality model can be determined by the bisection method. The resulting approach from the combination of PSM with ECGA is synthesized in the Algorithm 2.

The PSM reduces the overall running time showing that is possible to build an accurate model with relatively low number of individuals. As the population size can be smaller, the number of evaluations will be also lower. The next Section presents and discusses a set of experiments with an ECGA using the PSM.

4. Experiments

This Section is organized as follows. Section 4.1 describes the test problems and the experimental methodology used. Section 4.2 presents the results.

4.1. Test Problems and Configuration

The deceptive functions used in the experiments are described below.

$$f_{trap}(u) = \begin{cases} 1, & if\ u = k \\ 1 - d - u * \frac{1-d}{k-1}, & otherwise, \end{cases} \quad (2)$$

where *u* is the number of 1s in the string, *k* is the size of the trap function and *d* ($d = \frac{1}{k}$) is the fitness signal;

$$f_{deception}(X) = \sum_{i=0}^{m-1} f_{trap(x_{ki}+x_{ki+1}+...+x_{ki+k-1})}, \qquad (3)$$

where X is a string and *x* is a subset of bits from *X*. Thus, this equation sums the values of the traps for *X*; and

$$f_{decnoise}(X) = f_{deception}(X) + G(0, \sigma_N^2), \qquad (4)$$

where $G(0, \sigma_N^2)$ is the noise which follows a Gaussian distribution with mean 0 and variance σ_N^2.

The test results are organized in tables presenting the following information:

1. Problem size (PS): the problem size in which the algorithm was tested;
2. Population Size using ECGA (ECGA): the population's size when using the original ECGA;
3. Population Size using ECGA with PSM (PSM): the population's size when using the PSM; PSM_F is the size of the First population, used to build the high quality-model. PSM_R is the size of the Remaining population;
4. Ratio of POPulation Size (RPOPS): $\frac{PSM_R}{ECGA}$. For instance, *RPOPS=0.4* means that the reduced population's size is *40%* the size of the original population;
5. Ratio of Function Evaluations (RFE): the number of function evaluations when using PSM divided by the number of function evaluations when not using PSM.

The population sizes were determined by the bisection method over 30 trials. For the original ECGA and PSM_F, we used the *average size* found by the bisection method. For the PSM_R, we used the *largest size* over the trials.

4.2. Results

Table 1 presents the results for k=4. The RPOPS and RFE show reductions in the population size and in the number of function evaluations larger than 50%. Thus, the PSM enables to double the computational performance while maintaining the same quality of the solutions and scalability. Another interesting point is that, for PSs larger than 96, RPOPS and RFE decrease, indicating that, for larger problems, the population needed to solve them can be even smaller. Tests using PSs of thousands of variables will be performed in future works to verify this statement.

Table 1. Results for *k=4* without noise.

PS	$ECGA$	PSM_F	PSM_R	$RPOPS$	RFE
32	652	736	304	0.46626	0.44165
48	1192	1344	604	0.50671	0.49336
64	1768	2112	822	0.46493	0.43631
80	2544	2944	1080	0.42453	0.40427
96	3376	3584	1735	0.51392	0.49193
112	4144	4352	2056	0.49614	0.48152
128	5090	5120	2411	0.47367	0.46725
144	5858	6304	2759	0.47098	0.45283
160	6751	7488	2986	0.44230	0.42322
176	7680	8608	3371	0.43893	0.41811
192	8801	9344	3856	0.43813	0.42724

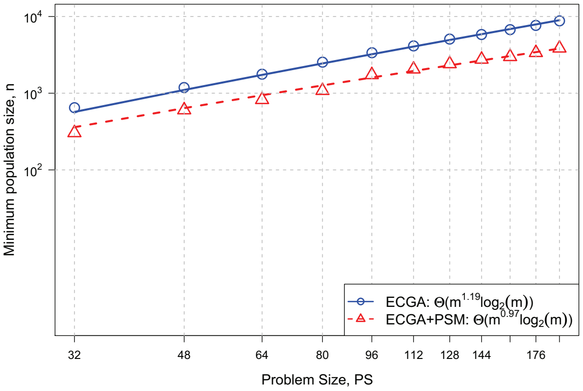

Figure 2. Population size for k=4 without noise: ECGA versus ECGA with PSM.

Figure 2 shows the improvement in the performance of the ECGA with PSM. The population size was $\Theta(m^{1.19}log_2(m))$, but with PSM it is considerably lower, $\Theta(m^{0.97}log_2(m))$, with *D* lower than the theoretical 1 (see Section 2.3). A similar improvement can be verified in the number of function evaluations, showed in Table 1 (see column RFE).

For the function k=4 with noise (Equation 4), the results (see Table 2) show more than 50% of reduction in population size and close to 60% in the number of function evaluations. The presence of noise makes the problem relatively harder than the problem of Equation 3, but the PSM provides an even higher improvement which can be further studied in future works.

Figure 3 illustrates the considerable reduction in the population size and in the number of evaluations, as presented in Table 2. The populational requirement of the ECGA with PSM grows slower, which leads to an important enhancement in the efficiency for the tested noisy problem.

Table 3 presents the results for tests without noise for k=5. The reductions of population size are about 50% can

Table 2. Results for k=4 with noise.

PS	$ECGA$	PSM_F	PSM_R	$RPOPS$	RFE
16	352	348	269	0.76420	0.75608
32	1046	1152	562	0.53728	0.49073
48	2022	2240	906	0.44807	0.39681
64	3053	3264	1443	0.47265	0.42321
80	4380	4608	1902	0.43425	0.38259
96	5615	5888	2643	0.47070	0.43573

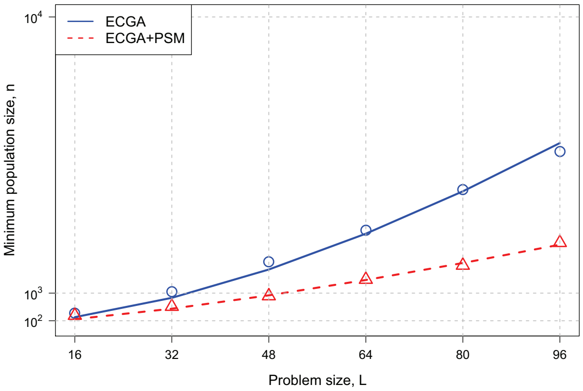

Figure 3. Population size for k=4 with noise: ECGA versus ECGA with PSM.

be seen where PS is 15, 30, 60 and 75. The benefit was smaller for PS 45 and 90, with a reduction of about 30%, which is also a considerable improvement over the ECGA without the PSM.

Furthermore, reductions in the population size from $\Theta(m^{1.34}log_2(m))$ to $\Theta(m^{1.13}log_2(m))$ (Figure 4) and in the number of evaluations from $\Theta(m^{1.83}log_2(m))$ to $\Theta(m^{1.64}log_2(m))$ (see Table 3) are also important evidences of the effect of the PSM approach. These results indicate that the PSM's contribution is even more significant for the larger problem.

Table 3. Results for k=5 without noise.

PS	$ECGA$	PSM_F	PSM_R	$RPOPS$	RFE
15	340	380	155	0.45588	0.49901
30	1080	1180	520	0.48148	0.48694
45	1996	2130	1413	0.70792	0.69431
60	3182	3500	1800	0.56568	0.56311
75	4529	5000	2310	0.51005	0.50335
90	5781	6020	3907	0.67583	0.67431

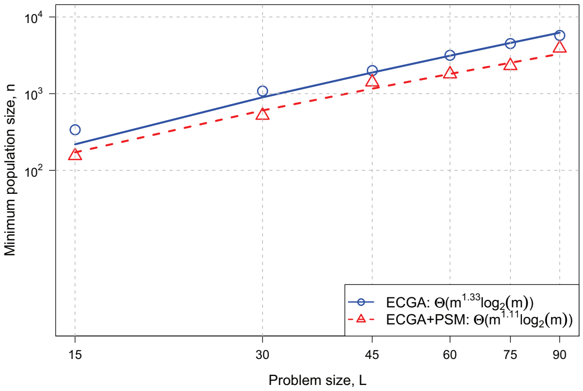

Figure 4. Population size for k=5 without noise: ECGA versus ECGA with PSM.

5. Summary and Conclusions

In order to reduce the running time of evolutionary algorithms for complex problems, various approaches have been developed. The algorithms with best performance found in the literature create probabilistic models used to generate offspring.

To solve decomposable problems, the ECGA, studied in this work, needs a large population size $\Theta(mlog_2(m))$ to generate a high-quality model of the problem. This population size is used by the entire evolution process. However, this $\Theta(mlog_2(m))$ behavior is actually shifted up by some degrees, being D one of them. Therefore, the behavior would change to $\Theta(m^D log_2(m))$. The value of D tends to 1 for very large problem sizes, with thousands of variables, not being accurate for smaller problems, which may require $D \approx 1.2$.

This paper presented an EET called PSM to speed up the optimization process of ECGA. In PSM, the ECGA's process is divided in two phases. In the first phase, it generates a high-quality model by using a large population size. This high-quality model leads the algorithm to a promising region. In the next phase, the algorithm starts a small population inside this promising region, by sampling from the high-quality model. By reducing the population size, the number of evaluations will be reduced; consequently, the overall running time. For the smaller problems tested in this work, the PSM technique achieved results corresponding to D lower than the theoretical value equal to 1.

By using this approach, the ECGA's running time is reduced by a factor of 30% to 70% while keeping the same accuracy and reliability. Furthermore, the ECGA using PSM presents the same scalability model as the ECGA, according to [10].

The results with the deceptive function show that the population size required by ECGA can be considerably lower than the estimated to the original ECGA, if a high-quality model is constructed for the first generation. In face of this, we provide more indicatives that a key point of the success of the ECGA, and probably, of other competent GAs, is the first probabilistic model. Thus, an EET to accurately build this model is an important research field.

5.1. Acknowledgment

This work was supported by CAPES (a Brazilian research agency) and sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grants FA9550-06-1-0370 and AF FA9550-06-1-0096. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government. The experiments were done using the ECGA software developed by Fernando Lobo, available at *http://www.illigal.uiuc.edu.*

References

[1] D. Goldberg and K. Deb. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In *Foundations of Genetic Algorithms*, pages 69–93, 1991.

[2] David E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[3] David E. Goldberg, Kalyanmoy Deb, and James H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362, 1992.

[4] D.E. Goldberg, K. Sastry, and X. Llorà. Toward routine billion-variable optimization using genetic algorithms: Short Communication. *Complexity*, 12(3):27–29, 2007.

[5] G. Harik. Linkage Learning via probabilistic modeling in the ECGA. Technical report, University of Illinois at Urbana Chapaign, Urbana, IL, 1999.

[6] Heinz Mühlenbein and Thilo Mahnig. The factorized distribution algorithm for additively decomposed functions, 1999.

[7] M. Pelikan, K. Sastry, and D. E. Goldberg. Scalability of the bayesian optimization algorithm. *International Journal of Approximate Reasoning*, 31(3):221–258, 2003.

[8] K. Sastry. *Evaluation-relaxation Schemes for Genetic and Evolutionary Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 2001.

[9] K. Sastry and D.E. Goldberg. Designing Competent Mutation Operators Via Probabilistic Model Building of Neighborhoods. *Genetic and Evolutionary Computation-Gecco 2004: Genetic And Evolutionary Computation Conference, Seattle, Wa, Usa, June 26-30, 2004, Proceedings*, 2004.

[10] Tian-Li Yu, Kumara Sastry, David E. Goldberg, and Martin Pelikan. Population sizing for entropy-based model building in discrete estimation of distribution algorithms. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 601–608, New York, NY, USA, 2007. ACM.