

On Extended Compact Genetic Algorithm

Kumara Sastry David E. Goldberg

IlliGAL Report No. 2000026
April, 2000

Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of General Engineering
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue, Urbana, IL 61801

On Extended Compact Genetic Algorithm

Kumara Sastry

David E. Goldberg

Illinois Genetic Algorithms Laboratory (IlliGAL)

Department of General Engineering
University of Illinois at Urbana-Champaign

Abstract

In this study we present a detailed analysis of the extended compact genetic algorithm (ECGA). Based on the analysis, empirical relations for population sizing and convergence time have been derived and are compared with the existing relations. We then apply ECGA to a non-azeotropic binary working fluid power cycle optimization problem. The optimal power cycle obtained improved the cycle efficiency by 2.5% over that existing cycles, thus illustrating the capabilities of ECGA in solving real-world problems.

1 Introduction

It has been shown in recent works [1, 2, 3] that ensuring effective building block (BB) mixing is an integral part of efficient GA design. These studies also showed that this could be achieved through a tight linkage of the set of alleles belonging to a BB. Based upon this concept many novel *competent* GA designs have been proposed which can be broadly classified into three groups; (1) Perturbation techniques like fast messy GA (FMGA)[4], gene expression messy GA(GEMGA)[5], Linkage identification by nonlinearity check/Linkage identification by detection GA(LINC/LIMD GA)[6], (2) Linkage adaptation techniques like linkage learning GA (LLGA)[7], and (3) Probabilistic model based techniques like extended compact GA (ECGA)[8] and Bayesian optimization algorithm (BOA)[9]. Though there have been extensive study on most of these algorithms, none have been reported for ECGA although preliminary studies[8] showed the technique to be effective. The purpose of this study is to perform a detailed analysis of ECGA and to test its ability in solving real world problems. The test case presented in this paper is that of a binary working fluid power cycle optimization(PCO), which is formulated as a nonlinear programming problem (NLP). The model equations are highly nonlinear and possesses many singular points leading to a failure of traditional optimization techniques.

The results obtained in the ECGA analysis enable us vital insights in determining tournament size, population sizing and estimating the convergence time and also its capabilities. Empirical relations for population sizing and convergence time have been evaluated based on the experiments. The results obtained on PCO illustrate the ability and effectiveness of ECGA in solving real-world problems. In this paper, we first present a brief overview of ECGA, followed by a discussion on results obtained, and a section on conclusion.

2 Extended Compact Genetic Algorithm (ECGA)

ECGA, proposed by Harik[8] is based on a key idea that the choice of a good probability distribution is equivalent to linkage learning. The measure of a good distribution is quantified based on minimum description length(MDL) models. The key concept behind MDL models is that given all things are equal, simpler distributions are better than the complex ones. The MDL restriction penalizes both inaccurate and complex models, thereby leading to an optimal probability distribution. Thus, MDL restriction reformulates the problem of finding a good distribution as an optimization problem that minimizes both the probability model as well as population representation. The probability distribution used in ECGA is a class of probability models known as marginal product models (MPMs). MPMs are formed as a product of marginal distributions

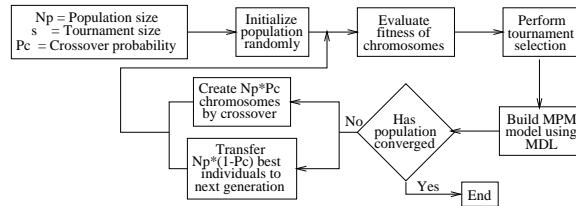


Figure 1: ECGA procedure

on a partition of the genes and are similar to those of CGA[10] and PBIL[11]. Unlike the models used in CGA and PBIL, MPMs can represent probability distributions for more than one gene at a time. MPMs also facilitate a direct linkage map with each partition separating tightly linked genes. Hence, in the current study each gene partition would refer to a BB.

A flowchart of ECGA procedure is given in fig. 1. Two things need further explanation, one is the identification of MPM using MDL and the other is the creation of a new population based on MPM. The identification of MPM in every generation is formulated as a constrained optimization problem,

$$\text{Minimize} \quad C_m + C_p \quad (1)$$

$$\text{Subject to} \quad 2^{l_{bb,i}} \leq N_p \quad \forall i \in [1, N_{bb}] \quad (2)$$

where C_m is the model complexity which represents the cost of a complex model and is given by

$$C_m = \log_2(N_p + 1) \sum_{i=1}^{N_{bb}} (2^{l_{bb,i}} - 1) \quad (3)$$

and C_p is the compressed population complexity which represents the cost of using a simple model as against a complex one and is evaluated as

$$C_p = \sum_{i=1}^{N_{bb}} \sum_{j=1}^{2^{l_{bb,i}}} N_{ij} \log_2 \left(\frac{N_p}{N_{ij}} \right) \quad (4)$$

N_{bb} in the equations represent the number of BBs, $l_{bb,i}$ is the length of BB $i \in [1, N_{bb}]$, N_{ij} is the number of chromosomes in the current population possessing bit-sequence $j \in [1, 2^{l_{bb,i}}]$ ¹ for BB i . The constraint (eqn. 2) arises due to finite population size.

A new population is generated based on the optimal MPM as follows, population of size $N_p(1 - P_c)$ where P_c is the crossover probability, is filled by the best individuals in the current population. This differs from the original ECGA procedure[12] in which $N_p(1 - P_c)$ individuals were taken to be the last $N_p(1 - P_c)$ individuals of the current generation irrespective of their fitness. The rest $N_p P_c$ individuals are generated by randomly choosing subsets from the current individuals. These subsets are the gene groups identified by the current MPM. This procedure is also different from the original ECGA procedure in which the subsets were chosen by probabilistic polling. The original procedure leads to a stochastic method were as the one used in the current study is a deterministic method in which the frequencies of the bit sequence of a particular subset remains constant. However, it has been observed that the outcome of both the methods are similar and in fact the deterministic method is computationally much faster[13].

3 Results and Discussion

Apart from the cooling system problem, four different test functions were used to analyze ECGA and are shown in fig. 2. The onemax problem which is just the summation of all ones in the binary string is known to be a GA-easy problem, while the others are GA-hard problems with the folded trap function being harder than the trap functions. Details of these test functions are given elsewhere[14, 2]. One common aspect about

¹Note that a BB of length k has 2^k possible sequences where the first sequence denotes be $00 \dots 0$ and the last sequence $11 \dots 1$

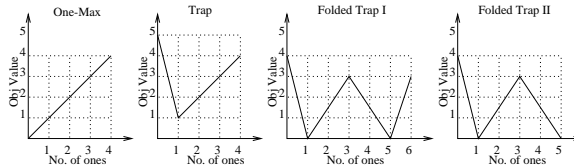


Figure 2: Functions used to analyze ECGA

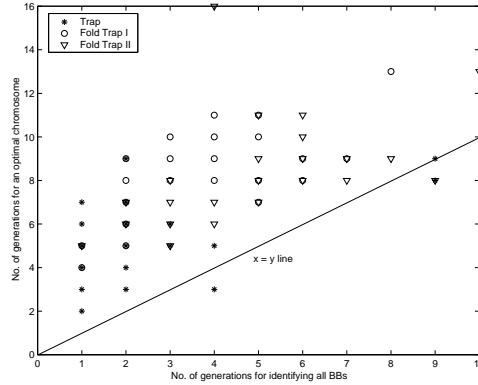


Figure 3: No. of generations taken to identify all the BBs vs no. of generations taken to find one optimal chromosome

these functions are that the BBs are known apriori and they are *static* in nature and have *uniform scaling*. By *static* we mean that the BBs do not change in time or over the generation. By *uniform scaling* we mean that all the BBs have equal weightage to fitness, for eg., in case of onemax problems a gene with a value one would contribute a unit value to the fitness irrespective of its position in the string.

For the analysis of ECGA, onemax problems of length 40, 50, and 60, trap functions with 10 BBs of length 4, 5, and 6, folded trap I with 10 BBs of length 6 and folded trap II with 10 BBs of length 5 were considered. The population size was varied from 50 to 25,000, the tournament size was varied from 2 to 20, and the crossover probability was taken to be 1.0. For all problems except the onemax problem, convergence to the optimal solution was achieved only in the case when all the BBs were identified correctly and is illustrated in fig. 3 in which the number of generations taken to find at least one individual having the optimal fitness is plotted against the generations taken to identify all the BBs. This exemplifies the importance of linkage learning in solving GA-hard problems.

Two modes of BB identification was observed during the study and are shown in fig. 4. The first mode is *flash identification* in which all the BBs are identified and learned together. The BBs identified for the trap functions and folded trap functions with high tournament size ($s \geq 10$) fall under this category. The second mode is *sequential identification* in which the BBs are identified and learned. This mode was observed for folded trap functions with small tournament size. By the term “learn” we mean that in every individual the genes corresponding to the identified BB have optimal values (for eg., for trap function in fig. 2 the optimal value for the genes in a BB would be zero). This suggests that as the problem becomes harder the BB identification switches from a flash mode to a sequential mode.

In the case of onemax problems, the BB size is one as each bit is independent of each other. However, the BB size identified by ECGA is more than one and the average BB size identified by ECGA in the first generation for various population sizes and different tournament sizes is plotted in fig. 5(a). Superficially, it seems as though the result obtained is wrong, but further investigation indicates that the results are correct and the MPM is indeed consistent with Pearson’s rank correlation test which shows that the genes in the BBs are correlated. The reason for the observed correlation is as follows: the randomly generated initial population is biased towards certain gene sequences resulting in *apparent-linkage* (hitchhiking) between those genes. This bias occurs due to finite population size and also due to the random number generator itself. The first fact can be verified from fig. 5(a), in which the average BB size reduces with the increase in population

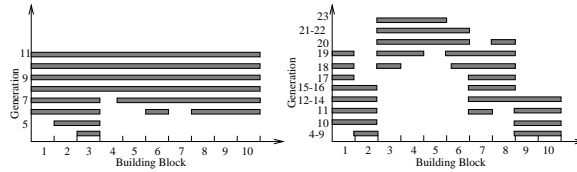


Figure 4: Building block identification modes, the first plot is the flash identification and the second sequential.

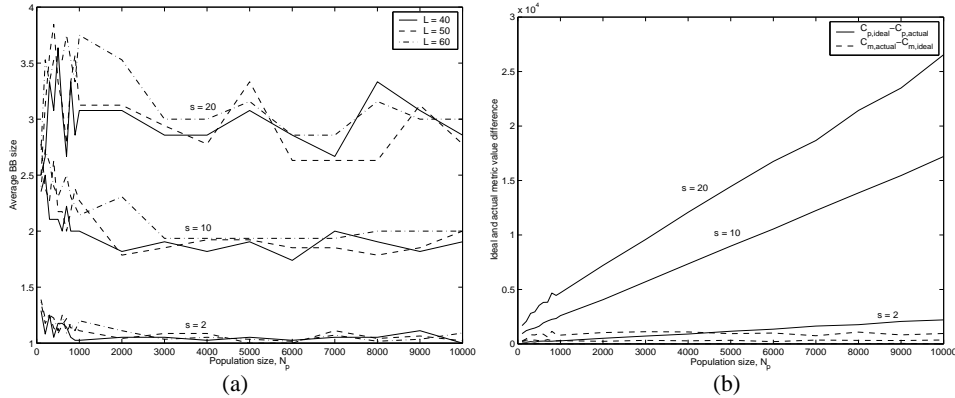


Figure 5: (a) Average BB size at the first generation. (b) Difference in the ideal and actual model complexity and compressed population complexity

size. This bias is then amplified by the selection procedure and the amplification factor increases with the tournament size making the apparent-linkage more pronounced. This fact is reflected in fig. 5(a) which clearly shows that the average BB size increases with the tournament size. Five different random number generators were tested in the current study and a generator that produced a least correlated data was used. Although the bias is very small for large population sizes, the reason for the average BB size being greater than one at large population sizes can be attributed to the MDL metric and in particular to compressed population complexity C_p . The differences between ideal C_p and the actual C_p ($C_{p,ideal} - C_{p,actual}$) is compared with the difference between the ideal and actual model complexities ($C_{m,actual} - C_{m,ideal}$) in fig. 5(b). The ideal case would refer to all the genes being independent of each other. The plot shows that the cost of having a complex model is very small when compared to the gains obtained by compressing the population and in fact this difference increases as the population increases. So, even a small bias gets amplified by the compressed population complexity metric causing BBs of size greater than one even for large population.

The convergence time was found to be dependent on the tournament size, string length and the problem type. The convergence time was found to be independent of the population size provided that it was above some threshold value. Even the average fitness at each generation was found to be relatively independent of the population size. Profiles of average fitness of the population at each generation, averaged over different population sizes and divided by the optimal fitness value for all the test functions in fig. 6 (a)-(c). The fitness profiles for onemax problems and trap functions are very similar and that of folded trap is similar to the trap/onemax of lower tournament size. The average of convergence time for different population sizes is plotted against the tournament size in fig. 6(d). The convergence time t_{conv} was observed to have the relation

$$t_{conv} = 1 + k_c \left(\frac{\log_2(L) + \log_2(\log_2(L))}{\log_2(s)} \right) \quad (5)$$

where L is the string length and k_c is a constant and has been empirically found to lie between 2.5-2.8571. This relation is also plotted in fig. 6(d) and fits very well with the results obtained from trap functions and onemax problems and slightly underestimates for the folded trap case. However, this relation does provide us with a lower bound estimate on the convergence time. This relation is of the same functional form as

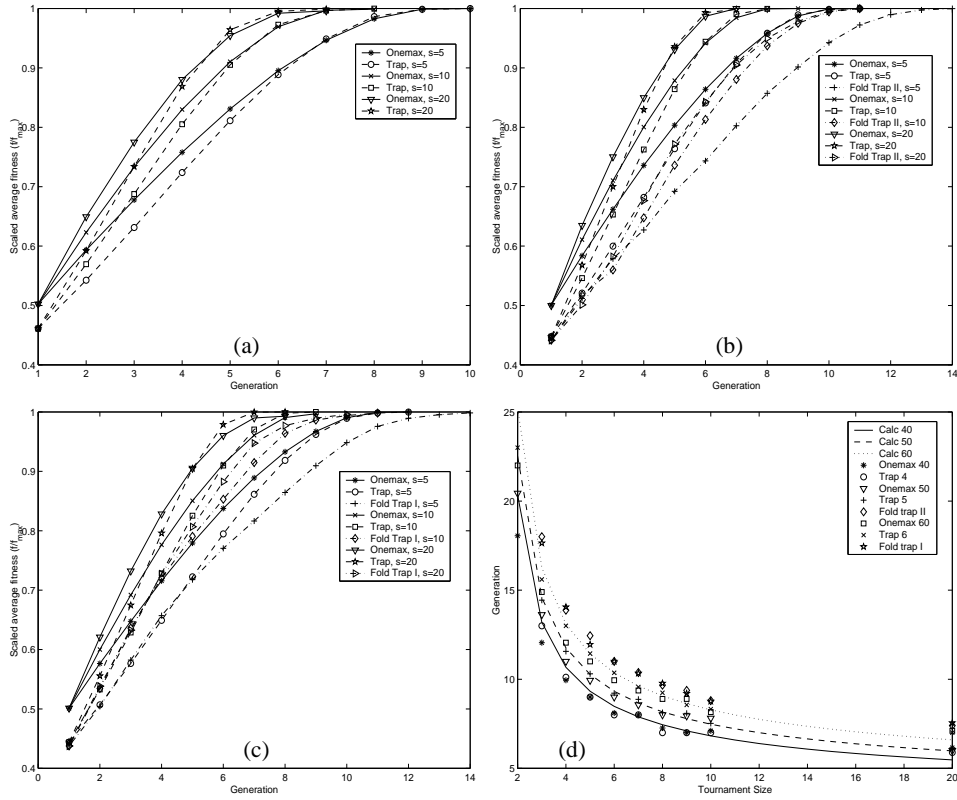


Figure 6: Scaled fitness values at different tournament sizes (a) Onemax, and trap-4, (b) Onemax, trap-5 and folded trap II, and (c) Onemax, trap-6, folded trap I. (d) Convergence time for the test problems

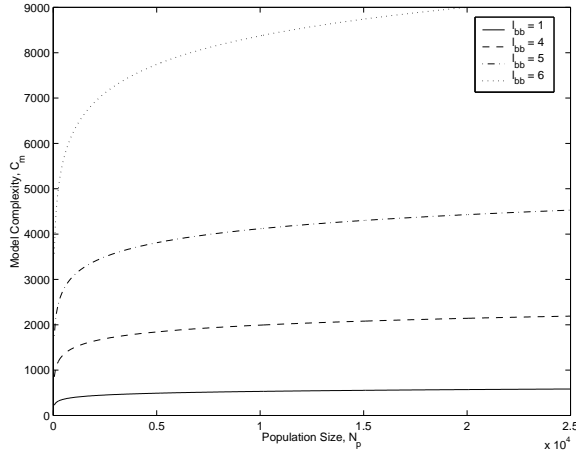


Figure 7: Model complexity

the convergence time derived for GAs[15] except that the population size is replaced by the string length. However, for larger problems ($L > 100$) the convergence time was observed to follow the relation given by Muhlenbein & Voosen[16]

$$t_{\text{conv}} = \frac{\pi\sqrt{L}}{2I} \quad (6)$$

where I is the selection intensity and further details are reported elsewhere[17].

ECGA requires a minimum population size above which the population size has negligible effect on its performance. The minimum population required was observed to be the value at which the model complexity, C_m , reaches its asymptotic value, or in other words the slope of C_m w.r.t N_p is below some threshold ϵ . The model complexity is plotted for different BB size in fig. 7 and is defined by eqn. (3). Assuming that the problem of interest has N_{bb} BBs of size l_{bb} and taking the derivative of C_m with respect to N_p

$$\frac{\partial C_m}{\partial N_p} = \frac{N_{bb}(2^{l_{bb}} - 1)}{\log(2)} \frac{1}{N_p + 1} \quad (7)$$

Equating $\frac{\partial C_m}{\partial N_p}$ to ϵ the population sizing equation is given by

$$N_p = \frac{N_{bb}(2^{l_{bb}} - 1)}{\epsilon \log(2)} \quad (8)$$

This equation is of the same form as the population sizing equations reported in studies on GAs[18] and BOA[19]. This is an interesting fact, considering that the above derivation is a totally different approach than the ones adopted in those studies. The constant ϵ was found to be a function of tournament size and BB size and was empirically determined to be

$$\epsilon = \frac{\min\{0.1515, c_1 s + c_2\}}{3.5 + 0.5(l_{bb} - 1)} \quad (9)$$

The above equation physically means that slope can be higher for higher tournament sizes and for problems with smaller BB size. The predictions of the population sizing equation is compared with that obtained from experiments in fig. 8 and it can be seen that the equation overestimates the population size required for smaller and simpler problems. However, the significant result is that the population size increases *linearly* with the number of BBs. Another important factor to be observed from fig. 8 is that for population sizing the folded trap functions behave as though their BB size is smaller by one, i.e., folded trap II behaves similar to trap 4 and folded trap I behaves similar to trap 5.

The results of the detailed analysis of ECGA can be summarized as follows: (1) For easy problems all the BBs are identified and learned together (flash identification), and for hard problems the BBs are identified

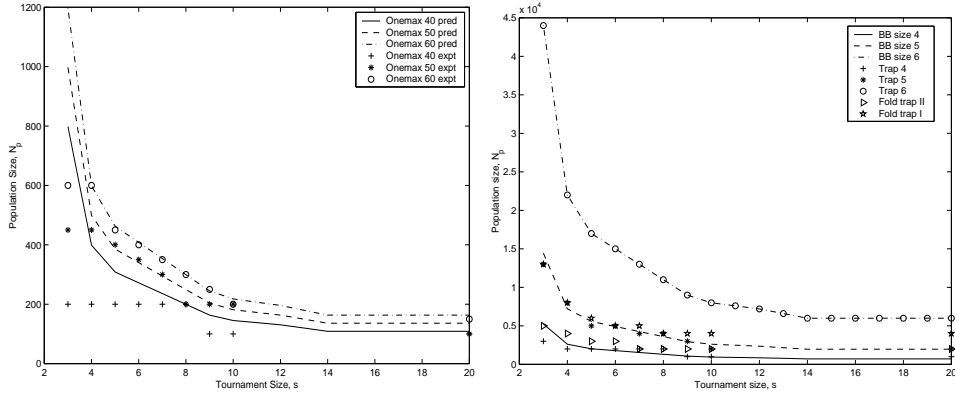


Figure 8: Minimum population size required: predicted vs experimental results

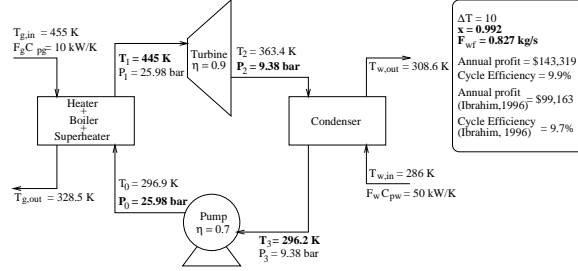


Figure 9: Optimal Power Cycle. Bold face indicates the decision variables

and learned sequentially (sequential identification). (2) The optimal solution is found only when all the BBs are identified. (3) The convergence time is similar to the convergence time derived for GAs with the population size replaced by the string length. (4) Hard problems with tournament size s , behave like easy problems with smaller tournament size s' , where $s' < s$, in terms of convergence time. This phenomenon is also reflected in the average fitness profile. (5) Convergence time for larger problems is proportional to the square root of the string length (6) A minimum population size is required for finding the optimum solution above which the population size has little or no effect in terms of convergence time and solution quality. (7) The population sizing equation obtained is similar to those reported for GAs and BOA and the population size increases linearly with the number of BBs. And (8) Hard problems with BB size l_{bb} behave like easy problems with BB size l'_{bb} , where $l'_{bb} < l_{bb}$, in terms of population sizing.

3.1 Power Cycle Optimization (PCO)

A simple Rankine cycle (fig. 9) similar to that used in [20] is considered. This cycle is chosen for illustration and studies on complicated cycles with more than 50 variables and 70 constraints are currently underway. The working fluid is a mixture of ammonia (NH_3) and water (H_2O). Heat is to be recovered from a geothermal source at temperature, $T_{g,in}$ of 455 K and heat capacity, $F_g C_{pg}$ of $10 \frac{kW}{K}$. A cold stream, used as heat sink, has a temperature, $T_{w,in}$ of 286 K and a heat capacity, $F_w C_{pw}$ of $50 \frac{kW}{K}$. The minimum temperature difference, ΔT , between the hot and cold source is taken to be 10 K. The turbine and pump are modeled as isentropic process with an efficiency of 0.9 and 0.7 respectively. The thermodynamic properties of $\text{NH}_3\text{-H}_2\text{O}$ mixtures were calculated as described in [21]. This correlation covers vapor-liquid equilibrium pressures of 0.2-110 bar and temperatures of 230-600 K. Analytical gradients of all the terms in the phase-equilibrium equations were computed to facilitate numerical stability of the nonlinear equation solver.

The PCO is formulated as an NLP and is given below.

$$\max \quad C_t \left| \dot{W}_t \right| - C_p \left| \dot{W}_p \right| - C_w \left| \dot{Q}_c \right| \quad (10)$$

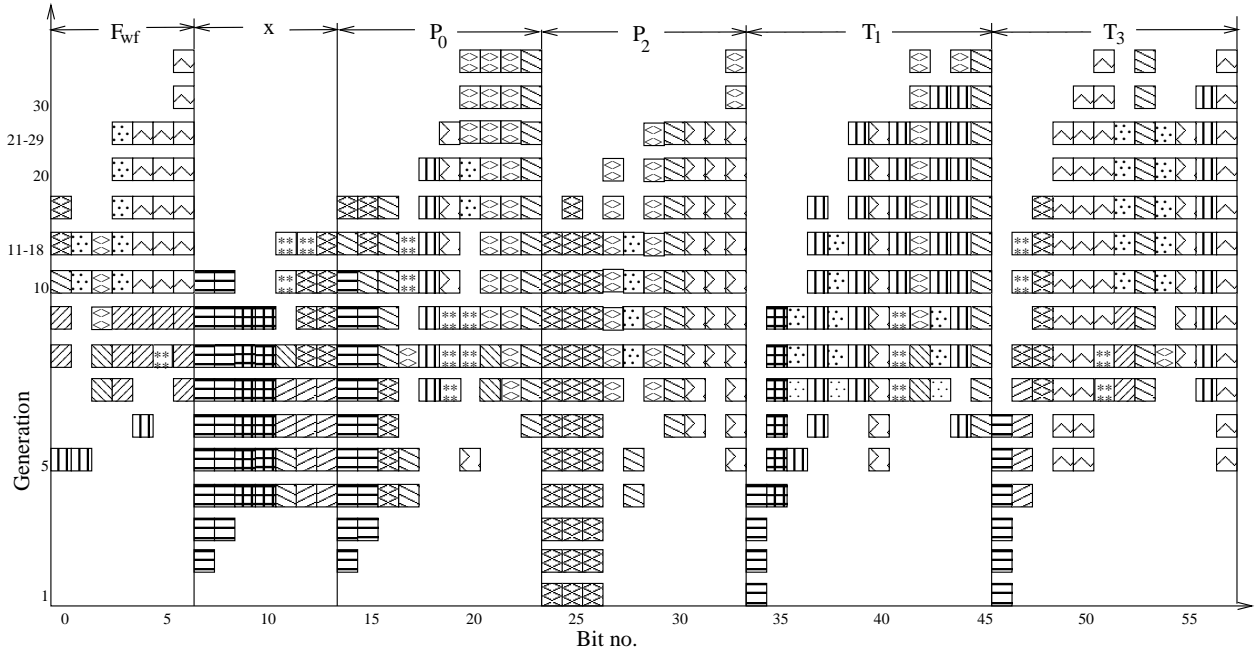


Figure 10: BB identification for the PCO problem. Same patterned box refers to the genes belonging to a BB. Vacant spots indicate BBs of size ≤ 2

$$\text{s.t.} \quad T_0 + \Delta T \leq T_{g,out} \leq T_{g,in} \quad (11)$$

$$T_{w,in} \leq T_{w,out} \leq T_2 - \Delta T \quad (12)$$

$$T_{dew}(P_0, x) \leq T_1 \leq T_{g,in} - \Delta T \quad (13)$$

$$T_{w,in} + \Delta T \leq T_3 \leq T_{bub}(P_2, x) \quad (14)$$

$$P_0 \geq P_2 \quad (15)$$

$$T_2 \geq T_{dew}(P_2, x) \quad (16)$$

$$T_0 \leq T_{bub}(P_0, x) \quad (17)$$

where \dot{W}_t is the work done by the turbine, \dot{W}_p is the work done on the pump, and \dot{Q}_c is the heat removed by the condenser. T_{bub} and T_{dew} refer to bubble and dew-point temperature respectively, and x is NH_3 mass fraction. Apart from the above constraints, two other constraints exist: (1) the global bounds on the decision variables, and (2) the minimum temperature between the hot stream and the cold stream is greater than ΔT at all points in the heat exchangers.

The constraints were handled through penalty method, in which the fitness value is penalized for constraint violation. For eg., a constraint $g(\underline{x}) \leq g^u$ is incorporated in the fitness function $f(\underline{x})$ as

$$f'(\underline{x}) = f(\underline{x}) - \phi \max[0, g(\underline{x}) - g^u]$$

The penalty parameter ϕ was chosen such that all the constraints were uniformly scaled and that the penalty for a constraint violation of even a small magnitude would be two or three orders of magnitude of the objective value. The latter condition not only ensures that a feasible solution is far better than the infeasible one, but also implicitly ensures that an infeasible solution close to a very good feasible solution to be better than a bad feasible solution. The uniform scaling is essential for eliminating unnecessary bias towards some constraints over others.

Unlike the test problems, the BBs are not known for the PCO and they are not static due to temporal-salience of BBs[22], and are not uniformly scaled. The time evolution of the BBs as identified by ECGA is shown in fig. 10. It can be seen from the figure that initially the most significant bits of x , P_0 , P_2 , T_1 , and T_3 form a BB. Initially F_{wf} does not form a BB with other variables as it does not affect the feasibility which is very important in the first few generations. This is readily illustrated in fig. 10, in which flowrate

forms BBs with other variables only in the 7th generation and also lower order BBs start forming in the same generation. Once the most significant bits converge, the lower significant bits form BBs, illustrating the domino convergence effect[22].

The optimal cycle given by ECGA is shown in fig. 9. The cycle efficiency, defined as the ratio of useful work W_t over the total heat absorbed, for the optimal cycle is 9.9% which is superior to that existing in literature, which is 9.7%[20]. The annual profit obtained by the optimal cycle is \$143,319 and that by the cycle in[20] is \$99,163 indicating the effectiveness of ECGA in solving highly nonlinear problems.

4 Conclusion

This paper presents a detailed analysis of ECGA using both GA-easy problems (onemax) and GA-hard problems (trap and folded trap). The analysis gives important insights to the working of ECGA and empirical results for convergence time and population sizing have been derived and compared with those existing in literature. The results on PCO problem exemplifies the capabilities of ECGA in solving real-world problems.

Acknowledgements

The work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grants F49620-94-1-0103, F49620-95-1-0338, F49620-97-1-0050, and F49620-00-0163. Research funding for this work was also provided by a grant from the National Science Foundation under grant DMI-9908252. Support was also provided by a grant from the U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0003. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are my own 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, the National Science Foundation, the U. S. Army, or the U.S. Government.

References

- [1] D. Goldberg, D. Thierens, and K. Deb, "Toward a Better Understanding of Mixing in Genetic Algorithms," *J. Soc. Instrument and Control Engineers*, vol. 32, no. 1, pp. 10–16, 1993.
- [2] D. Thierens, *Analysis and Design of Genetic Algorithms*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 1994.
- [3] D. Thierens and D. Goldberg, "Mixing in Genetic Algorithms," in *International Conference On Genetic Algorithms* (S. Forrest, ed.), (San Mateo, CA), pp. 38–45, Morgan Kaufmann, 1993.
- [4] D. Goldberg, K. Deb, H. Kargupta, and G. Harik, "Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms," in *International Conference on Genetic Algorithms*, (San Mateo, CA), pp. 56–64, Morgan Kaufmann, 1993.
- [5] H. Kargupta, "The Gene Expression Messy Genetic Algorithm," in *International Conference on Evolutionary Computation*, (Piscataway, NJ), pp. 814–819, IEEE Service Center, 1996.
- [6] M. Munetomo and D. Goldberg, "Linkage Identification by Non-Monotonicity Detection for Overlapping Functions," *Evolutionary Computation*, vol. 7, no. 4, pp. 377–398, 1999.
- [7] G. Harik and D. E. Goldberg, "Learning Linkage," in *Foundations of Genetic Algorithms* (R. Belew and M. Vose, eds.), vol. 4, (San Francisco, CA), pp. 247–262, Morgan Kaufmann Publishers, Inc, 1997.
- [8] G. Harik, "Linkage Learning via Probabilistic Modeling in the ECGA," IlliGAL Report No. 99010, University of Illinois at Urbana-Champaign, Urbana, IL, January 1999.

- [9] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz, “BOA: The Bayesian Optimization Algorithm,” in *Genetic and Evolutionary Computation Conference*, vol. 1, (San Francisco, CA), pp. 525–532, Morgan Kaufmann Publishers, Inc., 1999.
- [10] G. Harik, F. Lobo, and D. E. Goldberg, “The Compact Genetic Algorithm,” in *IEEE Conference on Evolutionary Computation*, (Piscataway, NJ), pp. 523–528, IEEE Service Center, 1998.
- [11] S. Baluja, “Population-Based Incremental Learning: A Method of Integrating Genetic Search Based Function Optimization and Competitive Learning,” Tech. Rep. CMU-CS-94-163, Carnegie Mellon University, 1994.
- [12] F. Lobo and G. Harik, “Extended Compact Genetic Algorithm in C++,” IlliGAL Report No. 99016, University of Illinois at Urbana-Champaign, Urbana, IL, January 1999.
- [13] F. Lobo and M. Pelikan. Personal Communication, 2000.
- [14] D. Goldberg, K. Deb, and J. Horn, “Massive Multimodality, Deception, and Genetic Algorithms,” in *Parallel Problem Solving from Nature* (R. Manner and B. Manderick, eds.), vol. 2, (UK), pp. 37–46, Elsevier Science Publishers, 1992.
- [15] D. Goldberg and K. Deb, “A Comparative Analysis of Selection Schemes used in Genetic Algorithms,” in *Foundations of Genetic Algorithms*, vol. 1, (San Mateo, CA), pp. 69–93, Morgan Kaufmann, 1991.
- [16] H. Muhlenbein and D. S. Voosen, “Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization,” *Evolutionary Computation*, vol. 1, no. 1, pp. 25–49, 1993.
- [17] B. L. Miller, *Noise, Sampling, and Efficient Genetic Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, May 1997.
- [18] D. Goldberg, K. Deb, and J. Clark, “Genetic Algorithms, Noise, and the Sizing of Populations,” *Complex Systems*, vol. 6, pp. 333–362, 1992.
- [19] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz, “Bayesian Optimization Algorithm, Population Sizing and Time to Convergence,” IlliGAL Report No. 2000001, University of Illinois at Urbana-Champaign, Urbana, IL, 2000.
- [20] M. Ibrahim, “Design Considerations of Ammonia-Water Rankine Cycle,” *Energy*, vol. 21, pp. 835–841, October 1996.
- [21] M. Ibrahim and S. Klein, “Thermodynamic Properties of Ammonia-Water Mixtures,” in *ASHRAE Transactions. Part 1*, vol. 99, (Alanta, GA), pp. 1495–1502, ASHRAE, 1993.
- [22] D. Thierens, D. E. Goldberg, and A. G. Pereira, “Domino Convergence, Drift, and the Temporal-Saliency Structure of Problems,” in *IEEE International Conference on Evolutionary Computation*, (Piscataway, NJ), pp. 535–540, IEEE Service Center, 1998.