ADAPTATION OF PARAMETRIC UNIFORM CROSSOVER IN GENETIC ALGORITHM

Farhad Nadi[1] and Ahamad Tajudin Khader[2]

[1] School of Computer Sciences, Universiti Sains Malaysia, Penang Malaysia
fnm.cod09@sudent.usm.my
[2] School of Computer Sciences, Universiti Sains Malaysia, Penang Malaysia
tajudin@cs.usm.my

ABSTRACT

Exploration of the search space occurs at the cost of destructing existing good solutions. This cost will grow as the search progresses. The parametric uniform crossover is a general form of the uniform crossover operator. Using this operator, it would be possible to control the swapping probability of each locus. An adaptive method proposed that control the value of the exchange probability of the parametric uniform crossover. The population will be diversified in case that the population's diversity decreases. The recombination of the solutions would be done with regards to their fitness distance to reduce the amount of destruction of good solutions. The experiments conducted show significant improvement in the performance of the parametric uniform crossover in comparison with to the state-of-the-art methods.

KEYWORDS

Genetic Algorithms; Parameter Control; Parametric uniform crossover; Exploration; Exploitation

1. INTRODUCTION

Recombination, which most of the time is equivalent to crossover in Genetic Algorithms (GAs), plays an important role in GAs [1]. In traditional GAs, n-point crossover operators with a low value for n are recommended based on theoretical and empirical foundations [2][3]. However, some later researches have shown that higher values for the number of crossing points (n) would be beneficial in some circumstances [4][5]. It has been shown that different crossover operators have different destructive effects [1].

The destruction of the sampling distribution could be translated to exploration of the search space, which is at the expense of exploitation. Mutation operator has some destructive effects as well; however its effect is negligible due to its low rate, most of the time. Parametric Uniform Crossover (PUC) is a general form of uniform crossover operator on which the exchange probability of the loci (p_0) could be controlled. Unlike the uniform crossover where its destructive effect is fixed, the destructive effect of the PUC could be controlled by the value of the p_0 [1].

Jan Zizka (Eds) : CCSIT, SIPP, AISC, PDCTA - 2013
pp. 443–450, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3650

Existence of a proper balance between exploration and exploitation of the search [2] is of importance in searching for high performance quality solutions. The amount of destruction in hyper planes could be seen as exploration of the search space, which will be done at the expense of exploitation. Thus, varying the destruction level of the hyper planes changes the balance between exploration and exploitation [1].

Uniform crossover has the most effect on destructing building blocks, in comparison to the *n*-point crossover operators. The destructive effect of the other operators is dependent to the hyperplane defining length. However, the uniform crossover has the advantage that its destructive effect is independent of the hyperplane defining length [1].

Lack of any adaptive methodology that controls the value of p_0 [6], as well as the promising potentiality that it could offer [1], motivated us for this study. An adaptive method is proposed for controlling the exchange rate of the parametric uniform crossover. The value of the p_0 would be adjusted based on the fitness distance of the solutions. The higher is the fitness distance of two solutions, the lower will be the value of p_0. The remainder of this paper is organized as follows. The proposed method will be introduced in section 2.The experimental setup will be presented in section 3. It will be followed by the results and discussion in section 4. The paper will be concluded in section 5.

2. ADAPTIVE PARAMETRIC UNIFORM CROSSOVER

In the last section the effectiveness of the PUC has been shown with the aid of variations of algorithms that adapt the p_0. In the following, an adaptive method namely, Adaptive Parametric Uniform Crossover (APUC) will be introduced. The proposed method, would control would utilize the same concept like the previous section for changing the value of p_0.

Regarding to the building blocks hypothesis Goldberg1989, good solutions will be constructed from building blocks. The better and better solutions will be constructed using the previous best partial solutions (building blocks). Therefore, maintaining the good solutions, is crucial in every step of the search. The destruction's cost of the good solutions would be higher in line with the convergence of the algorithm towards a solution. It is due to the high order of the constructed schemata (building block).

Recombination has a crucial role in combining the good low order schemata together and constructing schemata of higher order. Nevertheless, A source of the novel building blocks is needed is a priori for the recombination operator. The number of novel building blocks decreases gradually in line with the decrease of the population's diversity. In the proposed method, the search progresses as like in normal SGA until the population diversity decreases below a threshold. Algorithm 1 shows a pseudo code of the main loop of the proposed method. After the lose of diversity, the population will be diversified by increasing the mutation rate and changing the survival selection strategy to random replacement (See lines 15). The diversified population has both high order schemata as well as low order ones. In order to prevent destruction of the existing good solution in recombination with low order schemata the value of p_0 changes adaptively. While exploitation is needed exploration of the search space should be done as well. To this end, combination of the solutions that their fitness distance is more that 60% of the fitness distance of the best and worst solutions will be controlled to reduce the destruction of the good solutions. In the case where the fitness distance of the chosen solutions are less that 60%, the mixing will be done as usual (p_0=0.5).

If the difference of the fitness value between the chosen solutions for recombination is less than λ percent of the difference between the fitness value of the best and worst existing solutions. In this study, this value is fixed to λ=0.6. Selection of λ would effect the exploratory power of the search. The higher is the λ the higher is the exploratory power of the search. It is because the more the destruction would of the high order building blocks would be allowed. It has to be noted that the destruction is not bad all the times and it would allow the search to prevent stagnations and find new solutions. To this end, the value of λ is set such away to allow exploration as well as the survival of high order hyper planes. However, more priority is given to the exploration.

Genotypic Diversity Measure (GDM) of the population will be calculated using $GDM(.) = (DS(\rho) \cdot 1 - (\frac{X}{N}))$ Where X is the number of redundant individuals within the population (ϱ), N is the population size, and $DS(.)$ is (Eq. (1)) the measure of homogeneity of the population.

$$DS(\rho) = \frac{\sum_{i=1}^{N-1} \quad \sum_{j=i+1}^{N} \frac{h(I_{i,I_j})}{l}}{(\frac{N.(N-1)}{2})} \tag{1}$$

Algorithm 1: Pseudo code of the main loop of the APUC algorithm

```
boldiv = False
while (Termination criteria is not met) do
    $p_1$; $p_2$ ← Select($s_p$), $P_0$ ← 0.5
    if boldiv&&|f($I_1$) - f($I_2$)| < f($I_b$) - f($I_w$) * λ then
        $p_0$ ← 0.1
    end
    $p_0$ = 0.5 . (f($I_b$) - f($I_w$) * λ)
    $c_1$, $c_2$ ←    Recombine($p_1$, $p_2$, $p_0$)
    $p_m$ = 0.01, idx = $idx_b$
    if GDM(ρ) > 0.99 then //GDM(.) returns the population's diversity.
        $p_m$ = 0.1 + (0.05 - (1 - GDM(ρ)))
        boldiv = True
        idx = round(U(0, 1) * N)
    end
    $c_1$ ← mutate($c_1$, $p_m$)
    $c_2$ ← mutate($c_2$, $p_m$)
    $\rho_{idx}$    ((f($c_1$) > f($c_2$))?$c_1$ : $c_2$
end
```

Where $h(.,.)$ is the hamming distance function $\{0,1\}^n{:}\rightarrow Z^+$, which is defined as $h(x,y) = \sum_{i=1}^{l} x_i \oplus y_i$, where l is the length of the individuals, and ⊕ is the binary XOR operator.

3. EXPERIMENT SETUP

In all of the experiments the used algorithms have been steady state with binary tournament selection. Random deletion was used as survival strategy. A mutation rate of 0.01 and crossover rate of 1 has been used where it was applicable. The population size was set to 50 and 100 for epistatic and MPG problems respectively. maximum number of iterations was set to 10000, and 500 for MPG and other algorithms respectively.

The algorithms performance is measured over 50 independent runs for each of the problems. The mean of the best fitness (*MBF*) of the results are derived for all of the experiments. For the case of the epistasis problems, the standard deviation (*stdev.*) of the best results are also reported as well.

The proposed algorithm will be compared with some of the algorithms from literature including canonical GA with a randomly chosen constant mutation rate [7], PGA [8], SSRGA [7], SSRGA-II [9], *self-adaptive (SAGA)* [10], *adaptive (AGA)*[11], and the algorithms in [12] and [13].

Two classes of test functions are used for evaluating the proposed methods, including multi-modal boolean satisfiability and epistatic problems. A brief introduction to the test functions will be introduced in the following.

3.1 MPG Function

The multi-modal problem generator (MPG) is a useful and tunable benchmark proposed by Spears [14, 15]. In this benchmark the fitness value of a given individual will be calculated according to the Hamming distance between the individual and the nearest peak [16].

Following the steps in [13], 25 independent runs have been done for each benchmark and each algorithm. Stopping criteria for each experiment was 10,000 iterations or reaching the optimum, i.e. 1.

3.2 Epistatic Functions

Spears [15] has introduced a method for the creation of epistatic problems using boolean expressions. His proposed method has the capability for increasing the level of epistasis, thus making different epistasis problems with different levels of difficulty. The Spears method is based on conversion of the Hamiltonian Circuit (HC) problems into the SAT expressions. A more detailed definition of the this benchmark could be found in [15].

4. RESULTS AND DISCUSSION

Table 1 shows the MBF value of the compared algorithm over 10 different instances. The performance of the APUC method has shown to be good in the first 4 instances of the MPG test function. The results of the APUC method are comparable with the results of the meta-GA and REVAC.

Two paired-samples t-tests were conducted to compare the APUC with the two nearest algorithms which are REVAC and Meta-GA. There was no significant difference in the scores for APUC (M = 0.992, SD =.007) and REVAC (M = 0.989, SD = 0.012); $t(9)$ = 1.104, p = 0.298. Likewise no significant difference found between APUC and meta-GA (*M* = 0.992, *SD* = 0.002) ; *t*(9) = 0.169, *p* = 0.869. These results suggest that the APUC method has performed as good as the REVAC and meta-GA algorithms over the MPG test function.

Table 1: Comparing APUC with compared methods in [13, 17]

Peaks	GASAT	GAHSAT	hand-tuned	meta-GA	REVAC	APUC
	MBF	MBF	MBF	MBF	MBF	MBF
1	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	0.988	1.0	1.0
10	0.9956	0.9939	0.9961	0.993	0.996	1.0
25	0.9893	0.9879	0.9885	0.994	0.991	0.9903
50	0.9897	0.9891	0.9876	0.994	0.995	0.9871
100	0.9853	0.9847	0.9853	0.983	0.989	0.9870
250	0.9867	0.9850	0.9847	0.992	0.966	0.9882
500	0.9834	0.9876	0.9865	0.989	0.970	0.9853
1000	0.9838	0.9862	0.9891	0.987	0.985	0.9852

The results of the compared algorithms over different instances of epistasis problem are reported in Table 2. Except instances where *N*=6 and *N*=16, the APUC has obtained better than or equal results in comparison with the other algorithms.

A paired-samples t-test was also conducted to compare the APUC with PGA that has obtained the nearest results to the APUC. There was a significant difference in the results for APUC (*M*=0.959,*SD*=0.030) and PGA (*M*=0.924,*SD*=0.071);*t*(7)=2.402,*p*=0.047. These results suggest that APUC has been able to obtain significantly better results in comparison with the nearest algorithm.

The MPG benchmark is concerning problem with multiple-peaks, while the second benchmark is epistasis problem. Accordingly, the APUC has shown good results in the easier instances of MPG. Referring to the nature of the APUC, the diversification of the population will be delayed until the search converges to a solution. While diversification would result in exploration of the search space, however the APUC tend to keep the existing good solution found in the first convergence. In case of multiple peaks, if the search converges to a false peak it will finding the right peak would not be easy. A destructive diversification might be more useful in the case of convergence to false peaks.

Table 2: Comparison of the proposed method with benchmark methods of literature on epistatic problem with different levels of epistasis.

Degree of epistasis		N=6	N=11	N=16	N=21	N=26	N=31	N=36	N=41
APUC	avg.	0.975	0.990	0.989	0.980	0.962	0.944	0.924	0.91
	stdev.	0.011	0.003	0.002	0.003	0.004	0.006	0.004	0.005
PGA	avg.	0.991	0.990	0.994	0.967	0.909	0.869	0.842	0.827
	stdev.	0.019	0.004	0.001	0.005	0.007	0.007	0.006	0.006
SSRGA	avg.	1	0.968	0.931	0.893	0.866	0.848	0.833	0.823
	stdev.	0	0.004	0.006	0.006	0.005	0.007	0.004	0.005
AGA	avg.	1	0.96	0.922	0.888	0.865	0.847	0.836	0.826
	stdev.	0	0.007	0.008	0.007	0.006	0.005	0.004	0.004
SAGA	avg.	0.980	0.943	0.904	0.873	0.853	0.837	0.827	0.817
	stdev.	0.019	0.007	0.01	0.006	0.007	0.005	0.004	0.004
CGA	avg.	0.989	0.948	0.906	0.876	0.856	0.840	0.827	0.819
	stdev.	0.017	0.011	0.009	0.007	0.008	0.005	0.005	0.004

In the other words, the main assumption of the APUC is to find novel building blocks by diversification of the population. This would be beneficial where there is one basin of attraction in the search space. In the case of multi-peak problems there are several basins of attractions in the search space and the result of the algorithm would be very dependent on the first convergence of the search. In the case that the search converges to the vicinity if the global peak, the algorithm would perform well. Although, there would be a chance for the algorithm to find the a better solution than the best so far solution, but the chance of such finding is low.

As mentioned earlier, the APUC tries to reduce the destruction rate of the building blocks. This could turn to a disadvantage in the problems where multiple peaks exist in the search space. In the case where the search converges to a false peak, the APUC would hardly be able to find the global optima. Particularly, when the distance of the global optima is far from the local optima found so far. As the algorithm's tendency is to not destruct the existing building blocks, the search would be prevented from exploring the search space.

In summary it can be said that the APUC would restrict the exploration to the adjacency of the first solution that the search would converge to it. In other words, the probability of visiting new locations in the search space that are far from the converged solution would be low in this method.

5. CONCLUSION

Crossover operator in GA is usually a binary operator that takes two parents as input. The operator will create two off-springs by mixing the genetic materials of the input parents. Depending on the nature of the parents, the new off-springs could either diversify or intensify the search space. Looking from building block point of view, crossover could facilitate construction of larger building blocks or in other hand destruct existing building blocks. Uniform crossover exchanges approximately $l/2$ of the genetic materials of the given parents with each other. It has been shown that this level of exchange is the maximum feasible level of mixing for two parents, which is equivalent to the highest level of destruction. Parametric Uniform Crossover (PUC) is a general form of uniform crossover by which it would be possible to control the amount mixing of genetic material to off-springs. As the level of mixing could be controlled, therefore the diversification and intensification could be controlled as well. An adaptive method has been proposed where the level of destruction/mixing of the solutions will be controlled with regards to the fitness distance of the solutions with each other. The search proceeds as in simple GA with uniform crossover, until the population loses its diversity. The population then will be diversified and in order to not lose the existing good solutions due to their recombination with low fit solutions, the value of the p_0 will be tuned accordingly. A wide range of experiments have been conducted over different test functions. The comparison of the proposed method with the state-of-the-art (self) adaptive method of literature has shown suitability of the proposed method. The simplicity of the proposed method in the view of its performance would suggests the method as a suitable choice form many domains.

ACKNOWLEDGEMENT

This research has been supported partially by grant number 1001/PKOMP/817047 and PRGS Grant 1001/PKOMP/844037 from the Universiti Sains Malaysia.

REFERENCES

[1] W. Spears and K. De Jong, On the virtues of parameterized uniform crossover," in Proceedings of the Fourth International Conference on Genetic Algorithms (R. Belew and L. Booker, eds.), (San Mateo, CA), pp. 230{236, Morgan Kaufman, 1991.

[2] J. H. Holland, Adaptation in Natural and Arti_cial Systems. Ann Arbor, MI, USA: University of Michigan Press, 1975. K. A. De Jong, An analysis of the behavior of a class of genetic adaptive systems. PhD thesis, University of Michigan, Ann Arbor, MI, USA, 1975. AAI7609381.

[3] L. Eshelman, R. Caruana, and J. Scha_er, Biases in the crossover landscape," in Proceedings of the third international conference on Genetic algorithms, pp. 10{19, Morgan Kaufmann Publishers Inc., 1989.

[4] G. Syswerda, Uniform crossover in genetic algorithms," in Proceedings of the 3rd International Conference on Genetic Algorithms, (San Francisco, CA, USA), pp. 2{9, Morgan Kaufmann Publishers Inc., 1989.

[5] K. De Jong, Parameter setting in eas: a 30 year perspective," in Parameter Setting in Evolutionary Algorithms (F. Lobo, C. Lima, and Z. Michalewicz, eds.), pp. 1{18, Springer, 2007.

[6] F. Vafaee and P. C. Nelson, An explorative and exploitative mutation scheme," in IEEE Congress on Evolutionary Computation, pp. 1{8, IEEE, 2010.

[7] F. Nadi and A. T. A. Khader, A parameter-less genetic algorithm with customized crossover and mutation operators," in GECCO (N. Krasnogor and P. L. Lanzi,eds.), pp. 901{908, ACM, 2011.

[8] F. Vafaee, Controlling Genetic Operator Rates in Evolutionary Algorithms. PhD thesis, Dept. Computer Science, University of Illinois at Chicago, 2011.

[9] T. B ack and M. Sch utz, Intelligent mutation rate control in canonical genetic algorithms," in Proceedings of the 9th International Symposium on Foundations of Intelligent Systems, ISMIS '96, (London, UK, UK), pp. 158{167, Springer-Verlag, 1996.

[10] M. Srinivas and L. M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms," IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no. 4, pp. 656{667, 1994.

[11] Eiben, M. Schut, and A. de Wilde, Is self-adaptation of selection pressure and population size possible? - a case study," in Parallel Problem Solving from Nature PPSN IX, pp. 900{909, 2006.

[12] W. de Landgraaf, A. Eiben, and V. Nannen, Parameter calibration using metaalgorithms," in Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp. 71 {78, sept. 2007.

[13] K. A. D. Jong and W. M. Spears, A formal analysis of the role of multi-point crossover in genetic algorithms," Ann. Math. Artif. Intell., vol. 5, no. 1, pp. 1{26, 1992.

[14] W. M. Spears, Evolutionary Algorithms: The Role of Mutation and Recombination (Natural Computing Series). Springer, June 2000.

[15] P. Vajda, A. E. Eiben, and W. Hordijk, Parameter control methods for selection operators in genetic algorithms," in Proceedings of the 10th international conference on Parallel Problem Solving from Nature, (Berlin, Heidelberg), pp. 620{630, Springer-Verlag, 2008.

[16] A. Eiben, M. Schut, and A. de Wilde, Boosting genetic algorithms with (self-) adaptive selection," in Proceedings of the IEEE Conference on Evolutionary Computation (CEC 2006), pp. 1584{1589, IEEE, 2006.

Authors

Farhad Nadi is currently a Post-doctoral fellow in the School of Computer Science at the Universiti Sains Malaysia (USM), Penang, Malaysia. He received his M.Sc. and Ph.D. degrees in Computer Science from the same university back in years 2009 and 2013, respectively. His research interest focuses on Evolutionary Computation and neuroscience.

Ahamad Tajudin Khader is presently the Dean of Graduate Studies & Research at Universiti Sains Malaysia (USM), Penang, Malaysia. He holds a PhD degree from Strachclyde University, U.K and both BSc. (Hons) as well as M.Sc. (Hons) from Ohio, U.S.A. Assoc. Prof. Dr Ahamad Tajudin's research focuses on Genetic Algorithms, particularly in the areas of population modeling and parameter control. He is also actively involved in research on other evolutionary algorithms and problems in scheduling, timetabling as well as planning.