The Effect of Pseudo-Meiosis Genetic Algorithm on Bit-Coding Stationary Genetic Search

Bara'a Ali Attea
Department of Computer Science, College of Science, University of Baghdad, Baghdad-Iraq.

Received: 10/5/2004 ***Accepted: 2/11/2004***

Abstract

Yoshida and Adachi's diploid genetic algorithm- the so-called pseudo-Meiosis genetic Algorithm (psM GA)- was focused on preserving population diversity and adapting the population quickly to problem changes. The large population diversity provided by psM GA with good adaptation to the problem changes were effective and sufficient to tackle non-stationary traveling salesman problem. They did not apply this genetic algorithm to other problem domains, like bit-coding problems. Hence, in this paper we revisit the psM GA, apply it on bit-coding problems so as to see the effect and search power of this genetic search algorithm on this problem domain.

الخلاصة

الخوارزمية الجينية ثنائية المجموعة الكروموسومية والتي صممت من قبل الباحثين يوشيدا واداجي− والتي تسمى الخوارزميات الجينية ذات الانشطار الكاذب− ركزت على المحافظة على تنوع وتكيف المجتمع بصورة سريعة لتتلاءم مع تغير محيط المشكلة المراد حلها. التنوع الكبير للمجتمع الذي توفره الخوارزمية والتكيف الجيد مع التغير في محيط المشكلة كان كافيا وفعالا في معالجة التغير في محيط مشكلة البائع المتجول. الباحثان لم يطبقوا هذه الخوارزمية على مجالات أخرى من المشاكل مثل المشاكل التي تستخدم التمثيل الثنائي. على هذا الأساس هذا البحث سوف يطبق هذه الخوارزمية على المشاكل التي تستخدم التمثيل الثنائي, وذلك لدراسة فعالية وقوة الخوارزمية الجينية ذات الانشطار الكاذب على هكذا مجال من المشاكل.

Motivation

In nature most species utilize a dual or diploid chromosome structure in the non-gametic phase of their life cycle. Attempts to exploit diploidy with genetic algorithms are provided by Goldberg and Smith [1] [2], Yoshida and Adachi [3], and Greene [4]. These diploid GAs were mainly focused on preserving population diversity that is crucial particularly with multi-modal or non-stationary problems.

Some diploid genetic algorithms used dominance to generate a phenotype (or intermediate haploid chromosome) from diploid chromosomes. In methods based on dominance, chromosome position (or locus), and dominant values are chosen for each position to generate a haploid chromosome.

This kind of dominance method cannot be always used for various problems. Previous studies on diploid and dominance GAs have focused mainly on bit-coding problems, a solution is coded as a bit string and loci have no strong inter-dependency. But, order-coding problems often have strong constraints among loci. Consider a traveling salesman problem (TSP) and assume that each chromosome represents a tour in the form of a permutation of cities. The TSP has the constraint that every city must appear only once in a tour. If dominant cities are chosen independently of each other from a pair of chromosomes (tours) to form a new tour, most resulting tours would violate the constraint. Also, it is often difficult to define dominant-and-recessive relations among cities.

Another dominance method for diploid GAs, which differs from nature, is to select one of a pair of chromosomes according to some criterion, for example, choosing one randomly, or the better (or higher-fitness) one. This method never generates illegal (or lethal) haploid chromosomes, and can therefore be used for order-coding problems. But, it has the drawback that one of two chromosomes does not contribute to the GA search.

Yoshida and Adachi which proposed a diploid GA – the so called psM GA- studied its effect on ordering-problem (non-stationary traveling salesman problem). There simulation showed that the psM GA quickly adapted the population to the environment changes of TSP as it preserves population diversity. However, they did not address whether the mechanism of psM GA is also suffice in other problem domains like bit-coding problems. Hence, this paper revisits the psM GA using bit-coding problem domain.

The next section presents details of psM GA. Section 3 compares the performance of psM GA on bit-coding problem with haploid genetic algorithm. Finally, section 4 presents conclusions.

The pseudo-Meiosis Genetic Algorithm

1. Lingo used in psM GA

Below is some consideration that should be presented behind the description of psM GA [3].

a. Producing a single chromosome from a pair of chromosomes is analogous to meiosis in biology. Meiosis involves genetic recombination between homologous chromosomes and cell division producing haploid gametes. As the metaphor from nature, diploid-to-haploid mapping named pseudo-meiosis (psM). By “pseudo-“ they mean that throughout mapping the chromosome pair is kept intact, unlike real meiosis.

b. To avoid lethal haploids[1] and to make both homologous chromosomes contribute to a (post-meiosis) haploid, they use an ordinary crossover operator as psM mapping. Pseudo-meiosis mapping and crossover are the same from an algorithmic point of view. They distinguish between psM mapping and crossover, because psM occurs inside an individual whereas crossover occurs between individuals. Some crossover operators over two parent chromosomes yield two offspring chromosomes. In psM mapping, they assume that one of the two is selected according to some criterion (i.e. the one inheriting more properties of the chief chromosome, or the higher-fitness one).

c. One chromosome of a diploid does not undergo selection. If, during selection, both chromosomes of a diploid become extinct, and a clone offspring of a diploid is reproduced, the population diversity decreases, like with haploid GAs. Because one of the two chromosomes bypasses selection, the population retains its diversity.

d. Haploid-to-diploid re-pairing procedure builds a diploid from the one chromosome that survived selection, and another chromosome which bypassed selection. If both chromosomes of a diploid are processed as one body, repairing is not necessary.

2. Steps of psM GA

The psM GA has several steps. Below is a detailed explanation of each step (figure 1).

1. **(Initialization)** All chromosomes are created according to some rules, as in other GAs. In psM GA, each individual has two slots to hold a pair of chromosomes, a chief slot and an assistant slot. The chromosome in a chief slot, called a chief chromosome, and the one in an assistant slot is called an assistant chromosome. Chief and assistant properties do not change during generation.
2. **(Pseudo-meiosis)** Pseudo-meiosis mapping is applied to each diploid with probability P_{meio} in order to generate a post-meiosis (haploid) chromosome. For individuals that bypass pseudo-meiosis, a copy of the chief chromosome is treated as a post-meiosis one.
3. **(Selection, crossover and mutation)** This step is very similar to that of ordinary haploid GAs, except that GA operations are applied only to the post-meiosis chromosomes’ population. The post-meiosis chromosomes are first evaluated and then selected and reproduced based on their fitness values. The offspring chromosome undergoes crossover, with probability P_c, and mutation, with probability P_m.

---

[1] A lethal haploid chromosome means a haploid chromosome which violate the constraints of the problem. For TSP, the constraints is that every city must appears only once in a tour.

4. (Re-pairing) Each offspring chromosome is pulled back to the chief slot of its parent individual. The chief slot of an individual whose post-meiosis chromosome became extinct because of selection is filled with reproduced, and therefore promising, offspring. Each assistant chromosome undergoes mutation with probability P_{ma}. Note that, apart from mutation, there is no destructive operation for assistant chromosomes.
5. **(Generation-cycle)** steps 2 to 4 form one-generation cycle. The generation is repeated until some terminating condition satisfied.

The names, chief and assistant, come from their functions. The chief chromosome is related to the principal search cycle of the GA, whereas the assistant one works only for its chief pair inside the individual.

One of the unusual features of the psM GA is that assistant chromosomes vary independently of chief ones, and therefore continuously maintain population diversity. Assistant chromosomes work as follows during a search: copies of the better (or higher-fitness) post-meiosis chromosome will migrate to many individuals, and be psM-mapped with different assistant chromosomes in the next generation cycle. In other words, random local searches are performed around the better chief chromosomes. Because a variety of assistant chromosomes are retained independent of chief ones, local searches function continuously. In contrast, variation in population of typical haploid GAs tends to decrease as generation proceeds. The chances of having better chromosomes for crossover with different chromosomes also decrease.

Experimental Comparison with Haploid GA

In this section, we examine the behavior of the psM GA on bit-coding problems and compare its results with haploid genetic algorithm. Hence, before presenting experiments, we highlight below a brief overview of the haploid genetic algorithm.

1. Haploid Genetic Algorithm: A Brief Overview

Haploid Genetic Algorithms (HGAs) or simple genetic algorithms are a class of stochastic search algorithms. They are motivated by the computational process in natural evolution. The HGA works from a population of samples defined using some representation searches by selection, crossover and mutation [2][5].

i. Representation: HGA sometimes uses a sequence representation. Binary representation and gray coding are some examples often used in GAs. Strings are sometimes called chromosomes.

ii. Selection: the selection operator is responsible for detecting better regions of the search space. The fitness of a member is its objective function value. Selection computes an ordering among all the members of the population and gives more copies to the better strings at the expense of less fit members. Some widely used selection operators are roulette wheel selection and tournament selection.

iii. Perturbation operators:

a. Crossover: crossover works by swapping portions between two strings. Single point crossover is often used in haploid GAs. It works by first randomly picking a point between 0 and l. The participating strings are then split at that point, followed by a swapping of the split halves. Crossover is often applied with high probability.

b. Mutation: mutation randomly changes the entries of a string. Mutation is usually treated as a low profile in GAs because of its random nature of perturbation.

2. Algorithms Parameters Setup

Minimization experiments on test functions, described in table 1, were carried out in order to determine the performance of the psM GA with binary representation. Comparison was made with haploid genetic algorithm (HGA).

Simulation conditions were as follows: population size was 75 for the psM GA, and 150 for the HGA, so that the total number of chromosomes was 150 for both GAs. The tournament selection procedure with size two was used. Two-point crossover was used for both psM mapping and recombination. For both algorithms, crossover probability P_c=0.75. For the psM, pseudo-meiosis probability P_{meio}=0.001, mutation probability over (post-meiosis) haploid chromosomes is P_m=0.001, and mutation over assistant chromosomes P_{ma}=0.01.

The test functions include a range of function types that takes important characteristics (e.g., unimodal and multimodal, position of the optimal solution). The Sphere function (F1) is a continuous, strictly convex, unimodal and its minimum is at (0,…,0) in the center of the search space. It is the simplest, imaginable of all quadratic functions, nevertheless, is a standard problem. F4 is a multimodal Rastrigin's function which is made from the sphere function by modulating it with $10*\cos(2\pi*x_i)$. It characterized by having 11^n local minima, and by only one global minimum at (0,…,0). This function is considered as difficult for most methods. The Ackely's path function (F3) is a conventional generalized multimodal with global optimum at (0,…,0) the center of the search space. The Griewangk's (F4) function is another multimodal function, in which the summation term induces a parabolic "meta-structure" distributed by "waves". The waves are created by the cosine function in the product term. The product term induces a non-seprable feature for F4. Due to nonseparability of F4, it is difficult to optimize and search algorithm has to climb a hill to reach the next valley and the function has strong epistasis with small dimension.

3. Results

Tables 2 to 5 provide a summery of applying psM GA and Haploid GA to the test functions F1, F2, F3, and F4 respectively with moderate dimension (i.e. $n=10$). Each value in table computed is from the average of fifteen runs of three hundred generations.

Conclusions

Results presented assess the degradation of the psM GA's performance over that of the Haploid GA counterpart. In all these four problems, the psM GA failed completely to approach the global minimum while the Haploid GA succeeds in providing the reliable solutions in two of them. Both in F1 and F4, the Haploid GA achieves the desired solutions in almost all runs (i.e. in 15 different runs). This in turn emphasizes that diversity provided by the mechanism of psM GA couldn't accommodate the bit-coding domain. Another point here to be highlighted is that the inadequate performance of Haploid GA in F2 and F3 does not indicate the inability of Haploid GA in tracking these problems, but its main cause is the use of bit-coding (genotypic level) recombination process for these real-valued parametric optimization problems.

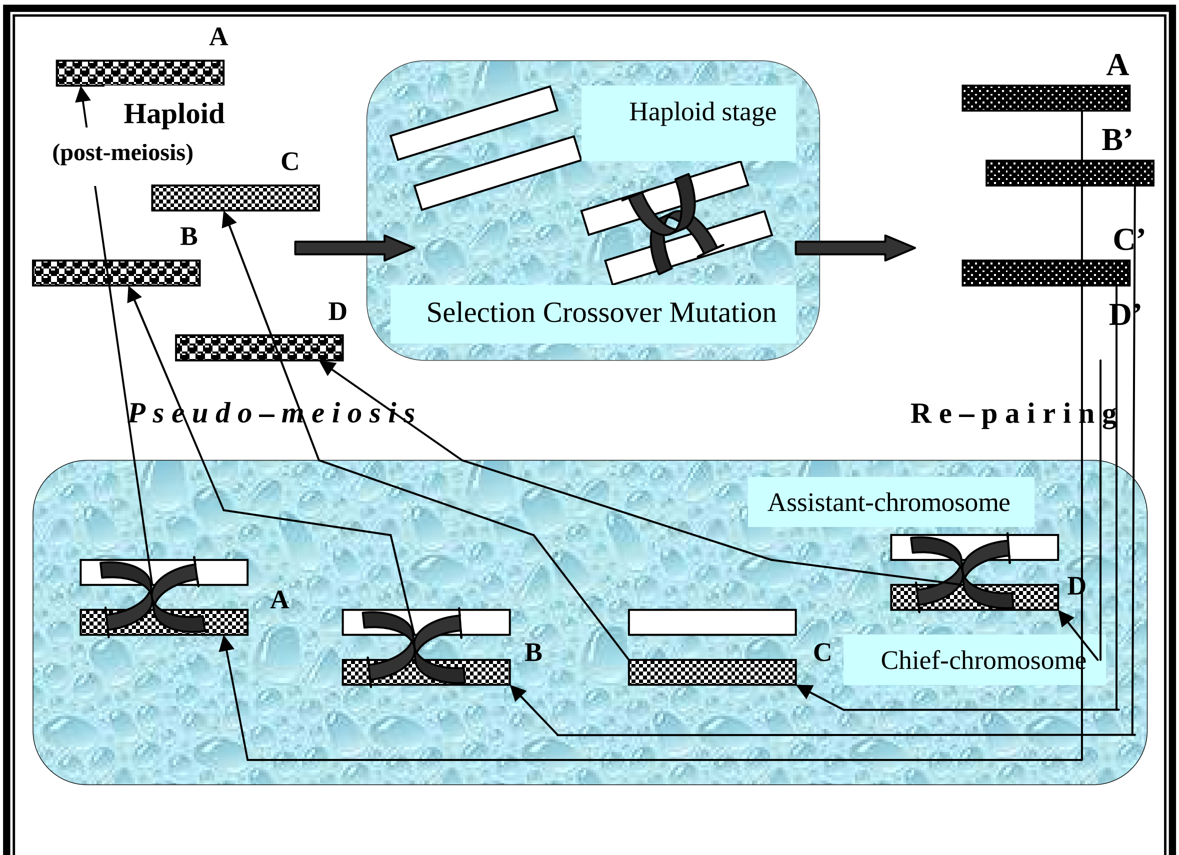

Figure 1 Genetic operations of the pseudo-Meiosis GA.

Table 1: The Four Test Functions

Problem No.	Numerical Description	Range of x_i	Reliable objective function value
F1	$f(\vec{x}) = \sum_{i=1}^{n} x_i^2$	[-5.12,5.12]	$\leq 2.500E-04$
F2	$f(\vec{x}) = 10*n + \sum_{i=1}^{n} (x_i^2 - 10*\cos(2\pi*x_i))$	[-5.12,5.12]	$\leq 4.9594E-02$
F3	$f(\vec{x}) = -20*\exp(-0.2*\sqrt{\frac{1}{n}*\sum_{i=1}^{n} x_i^2} - \exp(\frac{1}{n}*\sum_{i=1}^{n}\cos(2\pi*x_i)) + 20 + e$	[-32.768,32.768]	$\leq 2.0133E-03$
F4	$f(\vec{x}) = 1 + \sum_{i=1}^{n} x_i^2/4000 - \prod_{i=1}^{n}\cos(x_i/\sqrt{i})$	[-600.0,600.0]	$\leq 3.1309E-01$

Table 2: Test result of psM GA against Haploid GA on F1 where n=10

Gen. No.	psM GA	Haploid GA
20	1.3708E+01	8.2516E-01
40	1.0626E+01	1.9992E-01
60	9.2152E+00	6.3285E-02
80	7.3458E+00	1.4422E-02
100	6.5912E+00	4.4202E-03
120	5.7239E+00	9.8779E-04
140	4.9889E+00	2.5356E-04
160	4.2637E+00	5.1887E-05
180	3.9188E+00	9.2514E-06
200	3.7751E+00	4.5526E-06
220	3.6644E+00	6.3000E-07
240	3.5488E+00	1.5062E-07
260	3.3794E+00	5.8362E-08
280	3.3638E+00	2.6731E-08
300	3.3676E+00	9.9127E-09

Table 3: Test result of psM GA against Haploid GA on F2 where n=10

Gen. No.	psM GA	Haploid GA
20	5.4154E+01	1.1339E+01
40	3.7543E+01	6.2934E+00
60	3.0395E+01	5.3663E+00
80	2.8241E+01	5.0769E+00
100	2.7481E+01	4.9821E+00
120	2.5963E+01	4.9137E+00
140	2.4559E+01	4.8956E+00
160	2.4786E+01	4.8927E+00
180	2.4268E+01	4.8908E+00
200	2.4629E+01	4.8889E+00
220	2.4793E+01	4.8885E+00
240	2.3120E+01	4.8885E+00
260	2.5043E+01	4.8837E+00
280	2.2319E+01	4.8821E+00
300	2.3226E+01	4.8574E+00

Table 4: Test result of psM GA against Haploid GA on F3 where n=10

Gen. No.	psM GA	Haploid GA
20	1.3708E+01	7.4234E+00
40	1.0626E+01	4.3606E+00
60	9.2152E+00	3.5096E+00
80	7.3458E+00	3.1903E+00
100	6.5912E+00	3.0600E+00
120	5.7239E+00	3.0317E+00
140	4.9889E+00	3.0202E+00
160	4.2637E+00	3.0002E+00
180	3.9188E+00	2.9620E+00
200	3.7751E+00	2.9609E+00
220	3.6644E+00	2.9384E+00
240	3.5488E+00	2.9220E+00
260	3.3794E+00	2.9154E+00
280	3.3638E+00	2.8935E+00
300	3.3676E+00	2.8835E+00

Table 5: Test result of psM GA against Haploid GA on F4 where n=10

Gen. No.	psM GA	*Haploid GA*
20	2.1601E+01	3.7946E+00
40	1.0168E+01	1.6980E+00
60	5.3866E+00	1.1352E+00
80	3.6071E+00	8.8072E-01
100	2.6943E+00	5.6805E-01
120	2.1765E+00	3.3673E-01
140	1.7945E+00	1.7083E-01
160	1.5701E+00	1.3668E-01
180	1.4346E+00	1.2844E-01
200	1.3928E+00	1.2481E-01
220	1.3316E+00	1.2240E-01
240	1.2799E+00	1.2094E-01
260	1.2601E+00	1.2083E-01
280	1.1935E+00	1.2081E-01
300	1.1861E+00	1.2081E-01

References

1. Smith, R. E. and Goldberg, D. E. (**1992**): *Diploid and Dominance in Artificial Genetic Search*, *Complex Systems*, Vol. 6, pp. 251-285.
2. Goldberg, D. E. (**1989**): *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley.
3. Yoshida, Y. and Adachi, N. (**1994**): *A Diploid Genetic Algorithm for Preserving Population Diversity-pseudo-Meiosis GA, Parallel Problem Solving from Nature-PPSN III*, eds. Y. Davidor, H. Schwefel, and R. Manner, Springer-Verlage, October, pp.36-45.
4. Sima, A. (**1999**): *New Operators and Dominance Scheme for a Diploid GA*, Technical Report No.80626 Maslak Istanbul, Computer Engineering Department. Istanbul Technical University, Turkey.
5. Born, J. (**1996**): *An Evolution Strategy with Adaptation of the Step Sizes by a Variances Function. Parallel Problem Solving from Nature-PPSN IV, International Conference on Evolutionary Computation*, the 4th Conference on Parallel Problem Solving from Nature, eds. H. M. Voigt. W. Ebeling,I. Rechenberg, and H-P. Schwefel, Berlin, Germany, Seb. pp.389-398.