



Towards generalized expression for schemata count

Rajneesh Pawar^{a,*}, J.S. Saini^b, M. Gopal^c, A.P. Mittal^d

^a Electrical Eng. Dept., Deenbandhu Chhotu Ram University of Science & Technology, Murthal, Sonapat, Haryana 131039, India

^b Deenbandhu Chhotu Ram University of Science & Technology, Murthal, Sonapat, Haryana 131039, India

^c Electrical Eng. Dept., Indian institute of Technology, Hauz Khas, New Delhi 110 016, India

^d Instrumentation & Control Eng. Divn., Netaji Subhas Institute of Technology, Sector 3, Dwarka, New Delhi 110075, India

ARTICLE INFO

Article history:

Received 16 February 2008

Received in revised form 22 May 2009

Accepted 25 July 2010

Available online 11 August 2010

Keywords:

Breakup of similarity bits

Chromosomes

Genetic Algorithm (GA)

Schemata

ABSTRACT

The main goal of research reported in this paper is to propose a new technique for schemata count of Genetic Algorithms. Till recently, no methods existed for counting the schemata except the manual one that was obviously cursed by the tedium involved. A substantially generalized expression has been developed that directly computes the number of unique (different) schemata in a population, using breakup of similarity bits for non-repeated strings.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

The solution strategy of GAs gives prime importance to the structural representation of the solutions contained in the population using a fixed-length binary string and to the notion of schemas or schemata (the building blocks) [1–4]. GAs derive their processing strength from what is known as implicit parallelism. GAs operate on schemas and the processing leverage is well illustrated through Schema Theorem [2]. The idea of schemata gives us a powerful and compact way of talking about the similarities among strings. Since schemata provide the basic means of analyzing the net effect of reproduction and other genetic operators on building blocks contained within the population [1], counting them can be an enlightening exercise. However, the tedium of counting them can be reduced by the proposed generalized technique for different populations for non-repeated strings via a single line computation.

In trying to develop this new technique, we build on a fundamental premise made in [6] and then further proceed to uncover the correlation between the schemata and the breakup of 'similarity bits' of n -strings in the population. A complete illustration shows that the unique schemata count is independent of total similarity or total diversity bits count in a population. However, this

count is seen to depend upon the breakup of similarity bits count. A cue was, therefore, taken that a definitive relation may be developed to relate the breakup of similarity bits count with the number of unique schemata in a population. In our previous contribution [8], we developed a relation for directly computing the unique schemata count of a population for non-repeated strings depending upon the breakup of similarity bits count. However, the developed relation was restricted to a specific case of maximum similarity bits count, i.e., $s(b)$ being equal to $m - 1$ and further that the count of such $s(b)$'s is equal to $n - 1$ (where m = length of string and n = no. of strings) [8].

The above contribution is further extended in this paper to a more general case for obtaining the unique schemata count for non-repeated strings of different populations wherein the restriction imposed above was removed. This paper, therefore, advances the state of the art by developing a more generalized expression for directly computing the unique schemata count of different populations for non-repeated strings via a single computation.

The rest of the paper is organized in four sections: Section 2 provides the Rationale for computing schemata count for GAs. Section 3 provides theoretical background. It introduces the basics of schema theory. Relevant definitions from literature are put forth along with two new definitions framed by us in [8]. Although the aim of Section 4 is to present the development of a generalized case (removing restrictions in [8]), yet as a prelude and to aid the understanding, also presented are illustrations for a new technique for schemata count for the restricted case [8]. Conclusions and some future directions are put forth in Section 5.

* Corresponding author.

E-mail addresses: rajneeshdpawar@gmail.com (R. Pawar), jssain@rediffmail.com (J.S. Saini), mgopal@ee.iitd.ernet.in (M. Gopal), alok@nsit.ac.in (A.P. Mittal).

Nomenclature

Notations

n	number of strings
m	length of the string
$s(b)$	similarity bits count and $s(t)$ – total similarity bits count
$d(b)$	diversity bits count and $d(t)$ – total diversity bits count

2. Rationale for computing schemata count for GAs

Each individual in the population is an instance of 2^m schemata. In a population of n -strings, the number of unique schemata lies between 2^m and $n \cdot 2^m$. Therefore, it can be said that while the GA explicitly evaluates n -strings during any generation, it is implicitly evaluating a much larger number of schemata. This effect is known as intrinsic/implicit parallelism in GAs [1].

The usage of Schemata Count will help in a quantitative assessment of the convergence of GAs and will serve as a novel tool in the place of other convergence criteria. It is being envisioned Schemata Count shall make a significant dent into the research of operations which might result in newer applications.

3. Theoretical aspects of schemata

This section explains the notion of schemas and related concepts.

3.1. Basics of schema theory

Holland [5] invented the idea of *schema* (*schemas* or *schemata* for plural) to formally conceptualize the notion of 'building blocks'. The theoretical foundations of classical GAs rely on using a fixed-length binary string representation of solutions, and on the notion of a schemas.

A schema is a string over an extended alphabet $\{0, 1, *\}$ where the '0' and the '1' retain their normal meanings and the '*' is a wild card or don't care symbol. A schemata is a set of genes that are made up of a partial solution. A schemata with ' m ' defined elements and ' $n - m$ ' 'don't care' positions (such as an m -cell sub-placement in an n -cell placement problem) can be considered as an $(n - m)$ -dimensional hyperplane in the solution space [7]. All points on that hyperplane (i.e., all configurations that contain the given subplacement) are instances of the schema.

Example: The schema (010*11) matches the two strings (010011) and (010111).

If the don't care symbol appears in two places, there will be four strings that match it, and in general if there are ' n ' don't care symbols in a schema, then it matches exactly 2^n strings. On the other hand, a string can be represented by any of the 2^m schemas, where m is the length of the string.

Example: The string (0101) is matched by any of the following 2^4 (=16) schemata: (0101); (010*); (01*1); (0*01); (*101); (01**); ... (****).

3.2. Definitions

A few definitions already available in literature and required in the context of this paper are put forth here as under:

Definition 1 (Defined bits). The defined bits are the number of bits other than the don't care '*'s within the schema and their positions in the string are called the defining positions.

Example: There are 3 defined bits in the schema (01*1).

Definition 2 (Non-defined bits). The non-defined bits are the number of bits having don't care '*'s within the schema and their positions in the string are called the non-defining positions.

Example: There is only 1 non-defined bit in the schema (01*1).

Definition 3 (Schema Order). The order of a schema is simply the number of fixed positions present in the template [4]. All schemata do not have equal importance in directing the search of the GA. Some schemata are more specific than others. A schema that is more specific provides more information about the design space than the schema that is less specific.

Example: The schema (1*01*1*1**) is much more specific than the schema (1*****).

Definition 4 (Defining Length). The length of a schema is the distance between the first and the last specific string positions [4]. Certain schemas span more of the total string length than others.

Example: The schema (1*****0**) spans a larger portion of the string than the schema (1*0*****). In spite of the fact that both the schemata are of the same order, i.e., 2, the first schema (with defining length 7) gives information about different regions of the design space while the second one (with defining length 2) gives more information about a specific region of the design space but the information is restricted to that specific region and there is no information about other areas of the design space.

We now put forth two more definitions framed by us in [8] and needed in the context of the current work also:

Definition 5 (Similarity bits count and total similarity bits count). The number of matching alleles between any two strings is called the similarity bits count. The total of all such counts in a population of ' n '-strings constitutes the total similarity bits count.

Example: Consider a population of strings S1: (010), S2: (011) and S3: (001), the similarity bits count between strings S1 and S2, $s(b)=2$ and that between S2 and S3, $s(b)=2$ and that between S1 and S3, $s(b)=1$ and the total similarity bits count, $s(t)=2+2+1=5$.

Definition 6 (Diversity bits count and Total diversity bits count). The number of non-matching alleles between any two strings is called the diversity bits count. The total of all the diversity bits in a population of ' n '-strings constitutes the total diversity bits count.

Example: Considering the above example, the diversity bits count between strings S1 and S2, $d(b)=1$ and that between strings S2 and S3, $d(b)=1$ and that between strings S1 and S3, $d(b)=2$, and the total diversity bits count, $d(t)=1+1+2=4$.

4. Towards generalized expression

From the previous related research [6] that formed the starting point for our current research, the number of schemata is correlated to the total number of similarity or diversity bits. However, a difficulty arises when the total similarity bits count for two different populations is same but the breakup of similarity bits count is different.

The basis for evolving the Schemata Count has been through gaining empirical insights into the correlations between Schemata Count and number of strings, length of the string and variants of similarity bits count. We showed in [8] that unique schemata count is correlated with the breakup of similarity bits count. To that end, we took two illustrations that are reproduced here to aid the understanding of the further issues being probed in this paper. In the third illustration, we further extend the restricted case (of second illustration) to a more generalized one for obtaining the

Download English Version:

<https://daneshyari.com/en/article/497036>

Download Persian Version:

<https://daneshyari.com/article/497036>

[Daneshyari.com](https://daneshyari.com)