

Efficiency Updates for the Restricted Growth Function GA for Grouping Problems

Allan Tucker
School of Information
Systems, Computing and
Mathematics, Brunel
University, West London, UK
allan.tucker@brunel.ac.uk

Stephen Swift
School of Information
Systems, Computing and
Mathematics, Brunel
University, West London, UK
stephen.swift@brunel.ac.uk

Jason Crampton
Information Security Group,
Royal Holloway,
University of London, Egham,
Surrey, UK
jason.crampton@rhul.ac.uk

Categories and Subject Descriptors: I.2.8 [Computing Methodologies]: Heuristic methods; F.2.2 [Theory of Computation]: Sorting and searching

General Terms: Algorithms, Performance.

Keywords: Genetic Algorithms, Restricted Growth Function, Grouping.

1. ABSTRACT

Problems that require the partitioning of a set of variables in order to compute a solution such as bin packing or line balancing are typically NP-hard. Hence, researchers have focused on producing heuristic methods for finding appropriate partitions. Many of the representations used in optimisation algorithms including those in GA methods suffer from degeneracy [2]. Furthermore, Falkenauer has found that representations with less degeneracy result in more efficient GAs with respect to grouping problems [1]. Previously we developed a new representation for grouping genetic algorithms called the Restricted Growth Function GA (RGFGA) [3]. The RGFGA effectively removes all degeneracy, resulting in a more efficient search. However, one flaw of the RGFGA is that it converges too quickly resulting in a population with very little diversity. We exploit visualisation techniques, which can be used in conjunction with the Hamming distance, as well as introducing a novel population generator and a crossover operator which exploits the notion of extrema within grouping problems, to ensure diversity within the population. A *restricted growth function* is a function $f : [n] \rightarrow [n]$ such that $f(1) = 1, f(i+1) \leq \max\{f(1), \dots, f(i)\} + 1$. Note that there is a one-to-one correspondence between the set of RGFs and the set of partitions of $[n]$. In particular, the RGF represents a partition into $m \leq n$ groups, where 1 by convention belongs to the first group, i belongs to the $f(i)$ th group, and $\max\{f(1), \dots, f(n)\} = m$. The one-to-one correspondence means that there is no degeneracy in the representation of a partition using an RGF.

We introduce a new random RGF generator to create a better coverage of the search space, and a new crossover operator for the RGFGA in order to prevent premature convergence. The grouping problem search space contains two *extrema*: one occurs when all elements belong to a single group; the other when each group contains a single element. We choose two distinct RGFs as parents as in the original RGFGA, f and g . However, one child (rather than two) is

chosen using an existing path linking method [3] between f and g . The other child is generated from the path generated between one extremum chosen at random and one of the parents chosen at random. It is hoped that this modified crossover will ensure that a subset of children will be ‘pulled away’ from any local maxima toward the extrema in order to prevent premature convergence. A visualisation of this new crossover with the extrema points along with the old crossover used in [3] can be plotted within the search space using multidimensional scaling with Hamming distance between RGFs. The original RGF generator in [3] meant that at each iteration, the probability of a new group being generated decreases as the number of groups increases. Visualisation shows how the RGF generator biases individuals to be closer to the extrema with only one group. Therefore, we propose a new algorithm for generating RGFs that ensures an equal probability of creating a new group or using existing groups, whenever a new variable is assigned. Visualisation illustrates the resulting distribution of RGFs with individuals less clustered around one extremum than before.

We tested the old RGFGA crossover with different combinations of the new one and the new random population generator as well as testing strawmen approaches on a binpacking dataset and a multivariate time-series dataset that were outlined in [3]. Our hypothesis was that the two updates to our previous RGFGA would reduce the premature convergence of the algorithm. The results appear to suggest that either new update to the RGFGA improves upon the original in terms of controlling premature convergence, resulting in a more efficient search. However, the combination does not appear to add any further improvement. Future work will involve applying the RGFGA to consensus clustering algorithms for gene expression data which currently use simple heuristic search techniques in order to cluster data without the biases of standard clustering techniques. We also intend to explore the parallelisation of the RGFGA using different distributed GA architectures.

2. REFERENCES

- [1] E. Falkenauer. A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2:5–30, 1996.
- [2] N. Radcliffe and P. Surry. Fitness variance of formae and performance prediction. In D. Whitley and M. Vose, editors, *Foundations of genetic algorithms 3*, pages 51–72, 1995.
- [3] A. Tucker, J. Crampton, and S. Swift. Rgfga: An efficient representation and crossover for grouping genetic algorithms. *Evolutionary Computation*, 13(4):477–499, 2005.