

Grid Diversity Operator for Some Population-Based Optimization Algorithms

Ahmed Salah

Faculty of Science, Mathematics Department
Mansoura University
Mansoura, Egypt
a_salah@mans.edu.eg

Emma Hart

Institute for Informatics and Digital Innovation
Edinburgh Napier University
Edinburgh, UK
e.hart@napier.ac.uk

ABSTRACT

We present a novel diversity method named Grid Diversity Operator (GDO) that can be incorporated into multiple population-based optimization algorithms that guides the containing algorithm in creating new individuals in sparsely visited areas of the search space. Experimental tests on a set of unimodal and multimodal benchmark functions from the literature using GDO in conjunction with opt-aiNet algorithm show that GDO maintains better diversity in most cases, leading to an order-of-magnitude reduction in the number of objective function evaluations needed to converge while finding similar numbers of peaks in the majority of benchmarks.

Categories and Subject Descriptors

I.2 [Artificial Intelligence]: Problem Solving, Search

Keywords

Artificial Immune Systems; Evolutionary Algorithms; Optimization; Diversity; Grid

1. INTRODUCTION

Managing the diversity of a population has been recognized as one of the most influential factors within an Evolutionary Algorithms (EAs) right from their inception. From the exploration and exploitation perspective, an increase in diversity correlates with exploration phase of an optimization algorithm whilst a decrease correlates with the exploitation phase. Maintaining a diverse population through the use of exploration operators is key to achieving a balance between the two phases.

In this work we extend research within the field of diversity *maintenance* [1], proposing a novel genotypic diversity learning method named Grid Diversity Operator (GDO) that make use of the long-term history of all populations in order to suggest a biased distribution for new individuals. GDO is not specific to any particular algorithm, but can be

used with any optimization algorithm that supports the use of *infusion* techniques, that is, through insertion of new individuals after a certain number of generations or through special initialization techniques are used.

2. GRID DIVERSITY OPERATOR (GDO)

The Grid Diversity Operator (GDO) can be defined as a hybrid, non-niching, population-based, genotype diversity maintaining and learning technique. The basic idea is to split the feasible space (the domain) into smaller sub-spaces using the parameter $R \in \mathbb{R}^n$, which defines the number of intervals per dimension, where n is the number of dimensions for the problem. This process will form a 2D grid for 2-dimensional problem etc. The GDO attempts to distribute new individuals to the grid slots that have received fewer visits over time, thus increasing the explorative power of the algorithm.

First, A memory archive is created as an empty dictionary that has n components *key*, where each key matches a single value. The *key* components refer to the indices of a slot within the grid, while the *value* represents the number of individuals that have previously been placed in this slot.

The memory archive is updated every generation and after processing all individuals, the updated archive is used to initialize new individuals. For each new required, we pick a slot S at random and calculate its *distribution probability* P_S according to equation: $P_S = e^{(-N_S)}$ where N_S is the value matching the slot key in the archive or zero if the slot does not yet belong to the archive. Finally, the calculated probability P_S is compared to a uniformly distributed random number, r and if $P_S > r$ then a new individual is initialized randomly in this specific slot. If not, another slot is picked at random, and the steps are repeated until the individual is initialized successfully. The Grid Diversity Operator (GDO) is described in algorithm 1.

The process of updating the memory archive and distributing new individuals continues until the algorithm terminates, at which point the final population is expected to be more diverse than simply using a random initialisation procedure.

3. EXPERIMENTAL RESULTS

GDO has been injected into Opt-aiNet algorithm [2] and both versions (with and without GDO) were run 25 times on a set of 10 unimodal optimization functions from the literature [3] (Ackley, Sphere, Rosenbrock function, Goldstein-Price function, Booth, Bukin6, Matyas, Lévi13, Three-hump camel, Easom). The GDO variant was able to find the best

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO'15 Companion, July 11–15, 2015, Madrid, Spain.

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3488-4/15/07.

DOI: <http://dx.doi.org/10.1145/2739482.2764664>

Algorithm 1 Pseudocode for Grid Diversity Operator

Input: MemoryArchive, Resolution, N_{new} **Output:** S_{new}

```
 $S_{\text{new}} \leftarrow \emptyset$ 
for  $i = 1$  to  $N_{\text{new}}$  do
  distributed  $\leftarrow$  FALSE
  while  $\neg$  distributed do
    Key  $\leftarrow$  PickSlotAtRandom(Resolution)
    if KeyExist(MemoryArchive, Key) then
      V  $\leftarrow$  GetValueOfKey(MemoryArchive, Key)
      P  $\leftarrow e^{-V}$ 
    else
      P  $\leftarrow 1$ 
    end if
    r  $\leftarrow$  UniformRandom()
    if P > r then
      Individual  $\leftarrow$  CreateNewIndividualInSlot(Key)
      InsertIndividual( $S_{\text{new}}$ , Individual)
      distributed  $\leftarrow$  TRUE
    end if
  end while
end for
Return ( $S_{\text{new}}$ )
```

solution in 7 out of the 10 functions in the test and with significantly less number of function evaluations as shown in figure 1 and it was confirmed by a Wilcoxon rank sum test.

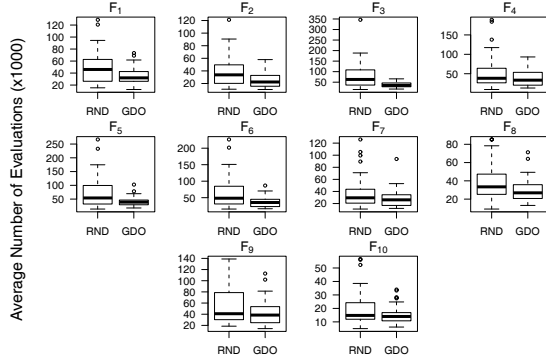


Figure 1: Graph comparing number of function evaluations between the variant with random initialization (RND) and GDO version

A similar experiment were performed on 12 multimodal optimization functions from the literature [4] (Bird, Vincent, Roots, Hilly, Rastrigin, Himmmlblau, Foxholes, Guichif4, HolderTable, Rastrigin49, Schwefel) and the results are shown in figures 2 and 3. The GDO version was able to find more optima than normal opt-aiNet in eight functions. The average number of function evaluations however was significantly lower with GDO than without it for all functions, according to a Wilcoxon rank sum test, showing improved performance on 11 functions out of the 12.

4. CONCLUSION

The GDO operator was shown to achieve effective exploration through testing on both unimodal and multimodal benchmarks when incorporated within opt-aiNet algorithm. When compared to the original algorithm it was shown to be an order of magnitude faster at converging, and at least as good (and occasionally better) at finding peaks on the majority of functions. Although we have only provided results obtained by incorporating the operator into op-aiNet, the

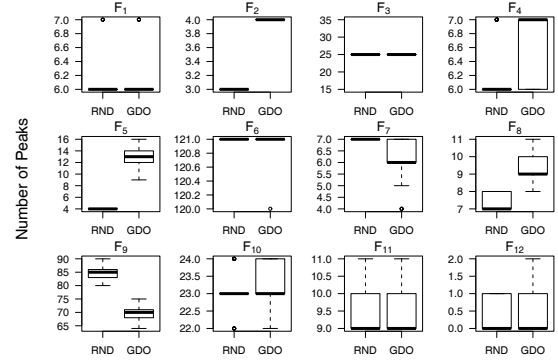


Figure 2: Graph comparing number of peaks located between the variant with random initialization (RND) and GDO version

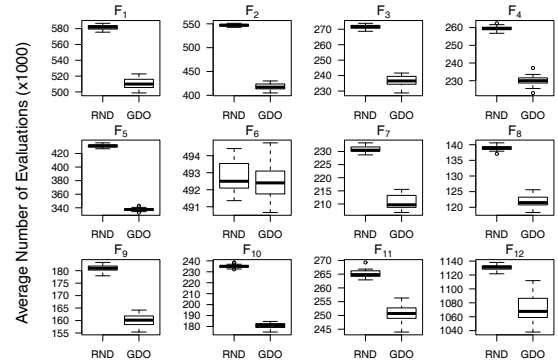


Figure 3: Graph comparing number of function evaluations between the variant with random initialization (RND) and GDO version

operator is by no means restricted to this function. It can be incorporated into any algorithm that includes an *infusion* step, e.g. the Saw-Tooth algorithm of [5].

5. REFERENCES

- [1] M. Črepinšek, S. Liu, and M. Mernik, “Exploration and exploitation in evolutionary algorithms,” *ACM Computing Surveys*, vol. 45, pp. 1–33, June 2013.
- [2] L. N. de Castro and J. Timmis, “An artificial immune network for multimodal function optimization,” *Evolutionary Computation, 2002. CEC’02. Proceedings of the 2002 Congress on*, vol. 1, pp. 699–704, 2002.
- [3] E. Cuevas and A. Reyna-orta, “A Cuckoo Search Algorithm for Multimodal Optimization,” *The Scientific World Journal*, vol. 2014, 2014.
- [4] Z. K. Silagadze, “Finding two-dimensional peaks,” *Physics of Particles and Nuclei Letters*, vol. 4, pp. 73–80, Feb. 2007.
- [5] V. K. Koumoussis and C. P. Katsaras, “A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 19–28, 2006.