Dynamic Optimization of Migration Topology in Internet-based Distributed Genetic Algorithms

Johan Berntsson
School of Software Engineering and Data Communications
Queensland University of Technology
QLD 4001, Australia
j.berntsson@qut.edu.au

Maolin Tang
School of Software Engineering and Data Communications
Queensland University of Technology
QLD 4001, Australia
m.tang@qut.edu.au

ABSTRACT

Distributed Genetic Algorithms (DGAs) designed for the Internet have to take its high communication cost into consideration. For island model GAs, the migration topology has a major impact on DGA performance. This paper describes and evaluates an adaptive migration topology optimizer that keeps the communication load low while maintaining high solution quality. Experiments on benchmark problems show that the optimized topology outperforms static or random topologies of the same degree of connectivity. The applicability of the method on real-world problems is demonstrated on a hard optimization problem in VLSI design.

Categories and Subject Descriptors: D.2 Software Engineering: Miscellaneous

General Terms: Performance Algorithms

Keywords: genetic algorithms, migration topology, adaptation, Internet computing

1. INTRODUCTION

The Internet is the most powerful parallel and distributed computation environment in the world and the idle cycles and memories of computers on the Internet have been increasingly recognized as a huge untapped source of computation power. As a result, the research and practice on developing Internet-based parallel and distributed GAs have attracted a great deal of attention recently.

Developing parallel computation applications on the Internet is quite different from in traditional parallel computation environments. The communication overhead (narrow bandwidth and high latency) makes massive data transfer and tight synchronization difficult in island Internet-based GAs, and it is desirable to minimize data transfer between the participating computers without compromising the performance of the DGA. The communication load is determined by the migration policy, and the migration topology is of special interest since it is known that dense topologies are known to find good solutions faster [2]. However, the connectivity may grow exponentionally with the number of islands, limiting the scalability of an Internet GA. This paper investigates several strategies for dynamic migration topology optimization.

GECCO'05, June 25–29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

2. DYNAMIC TOPOLOGY OPTIMIZATION

The optimization methods have been developed and tested on a hybrid peer-to-peer framework for island-model DGAs, which uses island nodes to run GA processes, and a supervisor that performs monitoring and adaptation. The framework uses clustering in the supervisor on the feedback data (elite solutions received from the islands) to find groups of islands that work in similar partitions of the search space, and to optimize the migration topology with the goal of reducing the connectivity while maintaining good performance. The main steps are outlined in the pseudo-code below. Several strategies for *MakeTopology* are described in Section 2.1. If any parameter is to be changed, *UpdateIslandTopology* will send the new migration policy to the islands.

```
procedure Adapt()
  cluster_set=MakeClusters(data_set)
  new_topology=MakeTopology(cluster_set)
  UpdateIslandTopology(new_topology)
```

Clustering is an unsupervised learning method which divides data into natural groups automatically based on similarity. In the current application the problem is simplified by the fact that only a small subset of individuals is evaluated. The computational expense grows with the number of islands, which is much smaller than the total population. Furthermore, it is not necessary to find optimal clusters. A heuristic method that finds useful clusters for efficient adaptation is enough.

The clustering algorithm used in this work is K-medoid [3], which can be applied to all genomes, including those that can only provide nominal data, given a distance function that compares two genomes, e.g. Hamming distance for bit genomes. The aim of K-medoid is to partition the data set of n data points into K groups so as to minimize the total within-group sum of distances about K representative points, or *medoids*, among the data points. The stored number of individuals from each island is limited, and a newly arrived data point replaces the oldest when the buffer is full.

The setting of the number of clusters parameter K is a non-trivial problem. The Minimum Description Length (MDL) criterion from information theory is used to find to find a good K. MDL which is a measure of how efficiently a given cluster model encodes the data set, and the system uses a bootstrap function to try $K \in [1 \dots K_{max}]$ and selecting the k with minimal MDL. The optimal K tends to be small compared to n, and $K_{max} = \sqrt{n}$ has empirically been found to be a reasonable setting.

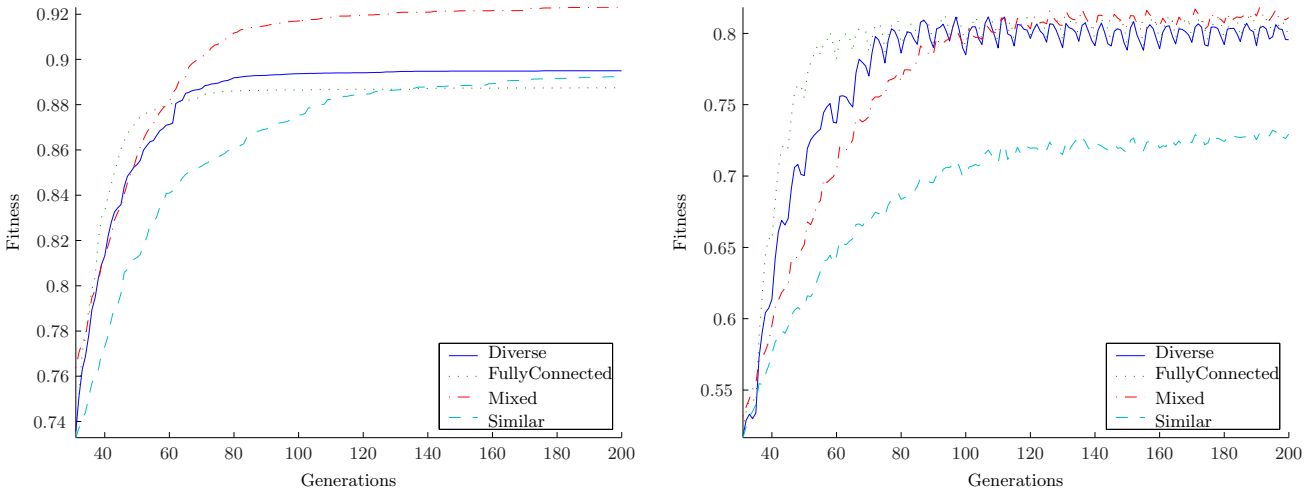

Figure 1: Continuous adaptation; optimal (left) and mean fitness (right)

2.1 Experimental Results

The clusters created by the clustering algorithm are used to create new migration topologies. The topologies tested are:

- Similar: Ring topology between the islands in each cluster, no communication between clusters.
- Diverse: Each island in a cluster communicates with all islands in the other clusters.
- Mixed: Ring topology between islands in each cluster, and one link to each other clusters.
- Fully: Each island communicates with all other islands, regardless of the cluster.

Many experiments have been conducted to evaluate each alternatives over a wide range of test problems. Figure 1 displays the running of F101 from the Whitley test suite [4] with the parameter settings pc=0.7, pm=0.005, 2-tournament, generational GA, 16 islands, 30 individuals/island, elite=1, 1-point crossover, 10 variable, 10 bits per variable, migrant rate=1 individual (best replaces worst), and migration interval=5 generations, averaged over 50 runs. *Connectivity* is also measured, defined as the number of one-way migration paths between the islands. For n islands, this means that a fully connected topology has connectivity $n(n-1)$, and a uni-directional ring topology has connectivity n.

Densely connected topologies have better average fitness and worse optimal performance. The similar topology has 7% of the connectivity of the fully connected, and performs worst in both categories. However, only a 3% increase in connectivity allows the mixed topology to improve the average fitness significantly, and to achieve the best optimal performance. The improved results can be explained thus: the clusters concentrate on exploring promising partitions in search space. The added connectivity in the mixed model is not big enough to force premature convergence, and the mixed topology keeps the number of clusters up compared to other dynamic topologies, dividing the islands into smaller groups and slowing down information exchange between the clusters. This seems to provide good balance between exploration and exploitation, which is key to good GA design.

3. DISCUSSION AND CONCLUSIONS

This paper has presented an adaptive migration topology optimizer for Internet-based island model genetic algorithms. The same set of experiments reported on F101 have also been carried out with the F102 and F8F2 functions. The results suggest that clustering gives a big boost in performance (especially optimal) for a small increase in connectivity. This is especially evident in the mixed topology. Furthermore the method has been applied to a challenging real-world VLSI floorplanning problem [1]. The adaptive DGA has significantly better optimal performance, and the best found solution is better than the best solution reported in [1], even though the total population is smaller.

In future research, we hope to extend on the current system and investigate the applicability of the proposed approach on optimization of other parameters that have an impact on communication overhead, such as migration rate and interval.

4. REFERENCES

[1] J. Berntsson and M. Tang. A slicing structure representation for the multi-layer floorplan layout problem. In *Applications of Evolutionary Computing: Proceedings of EvoWorkshops 2004*, volume 3005 of *Lecture Notes in Computer Science*, pages 188–197. Springer-Verlag, 2004.

[2] E. Cantú-Paz and M. Mejia-Olvera. Experimental results in distributed genetic algorithms. In *International Symposium on Applied Corporate Computing*, pages 99–108. Texas A&M University, Monterrey, Mexico, 1997.

[3] L. Kaufman and P. J. Rousseeuw. *Finding groups in data : an introduction to cluster analysis*. Wiley, New York, 1990.

[4] D. Whitley, K. Mathias, S. Rana, and J. Dzubera. Evaluating evolutionary algorithms. *Artificial Intelligence*, volume 85(1-2), pages 245–276, 1996.