Testing Parallelization Paradigms for MOEAs

Sadeesha Gamhewa and Philip Hingston
Edith Cowan University
2 Bradford St
Mt Lawley
+61 8 9370 6427

sgamhewa@student.ecu.edu.au, p.hingston@ecu.edu.au

ABSTRACT
In this paper, we report on our investigation of factors affecting the performance of various parallelization paradigms for multi-objective evolutionary algorithms. Different parallelization paradigms emphasize separate development of sub-populations versus communication and coordination between sub-populations to greater or lesser degrees. We hypothesized that the characteristics of a particular problem will favour some paradigms over others. We tested this hypothesis by creating variations on test problems with different characteristics, and testing the performance of different paradigms in a cluster environment.

Categories and Subject Descriptors
G.1.6 [**Optimization**]: Global Optimization

General Terms
Algorithms, Measurement, Performance, Experimentation.

Keywords: Multi-objective evolutionary algorithms, parallelization, test problem characteristics

1. INTRODUCTION
In this paper, we report on our investigations into performance differences between various parallelization paradigms for multi-objective evolutionary algorithms (MOEAs).

A number of paradigms have been proposed and explored, but there currently exists little guidance as to when and where to use which parallelization paradigm. This is our aim in the work reported in this paper – to make a contribution towards understanding which parallelization paradigms are best suited to which kinds of optimization problems.

2. EXPERIMENTS
For these experiments, we implemented various (parallel or serial) versions of a popular MOEA, Deb's NSGA-II [1]. We used a crossover probability of 0.9, mutation probability of 0.033, distribution index for SBX crossover of 15, and distribution index for polynomial mutation of 20.

We used the Message Passing Interface (MPI) specification for parallelizing MOEAs. We used a homogenous cluster of 21 processors set up as follows: Processors: x86 architecture, single 2.6GHz Pentium 4 processor with 512MB of memory; Operating system: Rocks 3.3.0 (Makalu); Network: Ethernet based 10/100 LAN; MPI implementation: MPICH version 1.2.5.2.

Copyright is held by the author/owner(s).
GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

The models implemented were (see [4] for details):

Master-slave: We distributed the population members evenly across slave processors. Each slave performs objective function evaluations for all population members it receives, and sends the objective values back to the master processor.

Island-10 (IS10): For the two Island models, we chose a variation in which islands are connected in a ring structure and all islands execute identical MOEAs (NSGA-II) with identical parameters. We used a uniform elitist migration scheme. Migration and replacement was performed at fixed intervals of 10 generations.

Island-100 (IS100): This was the same as Island-10, with migration every 100 generations.

We constructed the test problems for this study using the WFG toolkit [2]. The test problems are all bi-objective problems with 20 real-valued parameters in the range [0...1]. All problems were based on the shape function of the I1 problem from [2], which has a connected concave Pareto front. Transformations were applied to create eight problems with all combinations of the binary characteristics listed below:

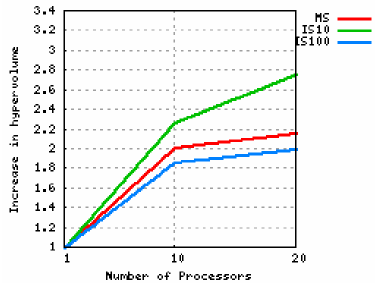
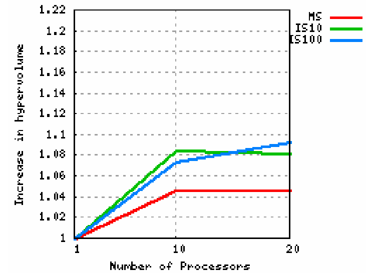
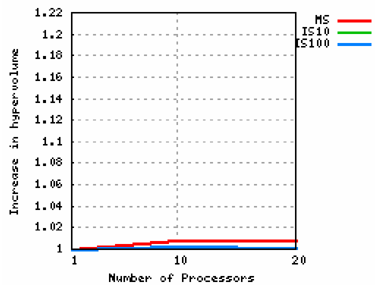
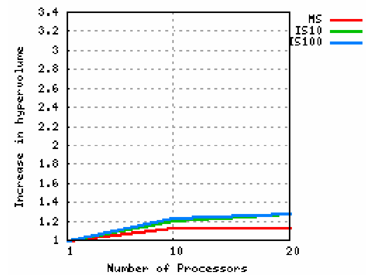
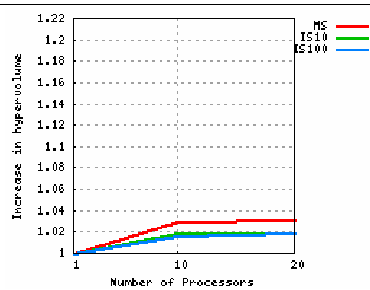
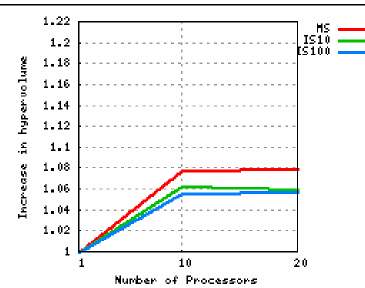
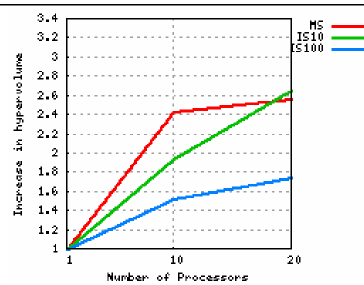
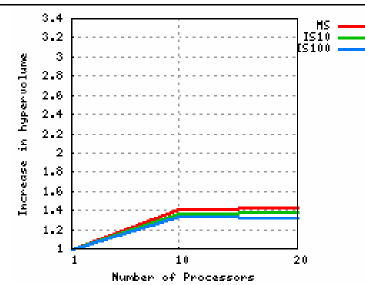
- Uni-modal problems (UM) were created by applying a linear shift to distance parameters. Multi-modal problems (MM) used a multi-modal shift of position and distance parameters.
- Non-separable problems (NS) used a non-separable reduction. Separable problems (SP) used a weighted sum reduction.
- Biased problems (BI) used a polynomial bias transformation. Non-biased problems (NB) used no bias transformation.

To judge the quality of the solutions found, we used the average hypervolume achieved over a given number of independent runs of the algorithm in a fixed period, normalized by dividing by the average hypervolume achieved using the standard serial MOEA.

The eight test problems were solved independently 25 times, with a 2 minute fixed time period for each problem-solving attempt, using total populations of 160 solutions.

3. RESULTS
- The two Island models performed similarly, except on the SP/BI problems, where IS100 was better than IS10;
- Island models do better than Master-slave on the UM/NS problems;
- Master-slave was better than the Island models on UM/SP/NB, MM/SP/BI, MM/SP/NB and MM/NS/NB, suggesting that Master-slave may do better on multi-modal, separable or non-biased problems.

Table 1 – Average relative hypervolumes achieved for each model on different problem types.

Uni-modal (UM)	separable (SP)	non-separable (NS)
non-biased (NB)		
biased (BI)		
multi-modal (MM)	**separable (SP)**	**non-separable (NS)**
non-biased (NB)		
biased (BI)		

4. REFERENCES

[1] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II [Electronic version]. IEEE Transactions on Evolutionary Computation, 6(2), 182-197.

[2] Huband, S., Barone, L., While, L., & Hingston, P. (2005). A Scalable Multi-objective Test Problem Toolkit. In Proceedings of Evolutionary Multi-Criterion Optimisation, Third International Conference (pp. 280-294). Berlin: Springer-Verlag.

[3] Huband, S., Hingston, P., Barone, L., & While, L. (2006). A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit [Electronic version]. IEEE Transactions on Evolutionary Computation, 10(5), 477-506.

[4] Van Veldhuizen, D. A., Zydallis, J. B., & Lamont, G. B. (2003). Considerations in Engineering Parallel Multiobjective Evolutionary Algorithms [Electronic version]. IEEE Transactions on Evolutionary Computation, 7(2), 144-173.