Genotypic Differences and Migration Policies in an Island Model

Lourdes Araujo
Dpto. Lenguajes y S.I.
UNED, Spain
lurdes@lsi.uned.es

Juan J. Merelo
Dpto. de Arq. y Tec. de C.
U. de Granada, Spain
jmerelo@geneura.ugr.es

Antonio Mora
Dpto. de Arq. y Tec. de C.
U. de Granada, Spain
amorag@geneura.ugr.es

Carlos Cotta
Dpto. de Lenguajes y C.C.
U. de Málaga, Spain
ccottap@lcc.uma.es

ABSTRACT

In this paper we compare different policies to select individuals to migrate in an island model. Our thesis is that choosing individuals in a way that exploits genotypic differences between populations can enhance diversity, and improve the system performance. This has lead us to propose a family of policies that we call *multikulti*, in which nodes exchange individuals different "enough" among them. In this paper we present a policy according to which the receiver node chooses the most different individual among the sample received from the sending node. This sample is randomly built but only using individuals with a fitness above a threshold. This threshold is previously established by the receiving node. We have tested our system in two problems previously used in the evaluation of parallel systems, presenting different degree of difficulty. The multikulti policy presented herein has been proved to be more robust than other usual migration policies, such as sending the best or a random individual.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*; I.2.8 [**Artificial Intelligence**]: Problem solving, control methods and search—*Heuristic methods*

General Terms

Algorithms, Performance

Keywords

Genetic algorithms, Parallelism, Island model, Migration policy, Diversity

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

1. INTRODUCTION AND STATE OF THE ART

Any easily decomposable algorithm can improve its performance by distribution in different computing nodes; Evolutionary algorithms (EAs) [10] are such kind of algorithms: since they are population-based algorithms, computation can be divided in many different ways: by farming out evaluation of members of the population to other computing nodes (in an approach usually called *farming* [24]), or by letting every node carry out its own evolutionary algorithm, with frequent interchange of information among them via *migration* of individuals from one node to the others; this is usually called an *island model* [29, 25] (a mention should also be done to *diffusion EAs* [9, 23, 26, 2], in which the population is endowed with a spatial structure that restricts selection, mating and replacement; some early works on memetic algorithms were actually conducted following this approach [20, 19]. Note that this model lends itself naturally to parallelization on massively parallel systems and –as the island model– can be also used from an algorithmic point of view without relying on physical parallelization. The focus of this work is on the island model though.)

In these initial papers, linear speedups[1] with the number of nodes were easily achieved. However, since the prevention of inbreeding is one of the keys in maintaining a healthily diverse population that can efficiently explore the search space, and distributed computing partially isolates populations from each other leading them to explore different portions of the search space, it has been frequently proved that distributed evolutionary algorithms can improve the results of the sequential version at the same computational effort measured in number of fitness evaluations (see, for instance, [3]).

In these parallel evolutionary algorithms, there is an additional factor that contributes to this increased performance: the effect that an incoming immigrant have on the population they are incorporated to. Different choices for the immigrant selection method will yield different performance. And this immigrant selection method presents different facets that can be studied:

1. the number of individuals undergoing migration,

[1] ratio of the sequential execution time to the parallel one

2. the frequency of migration, i.e. the number of generations or evaluations between migrations,

3. the policy for selecting immigrants,

4. the immigrant replacement policy,

5. the topology of the communication among subpopulations, and

6. the synchronous or asynchronous nature of the connection among subpopulations.

Some of them have been studied in the literature: for instance, Alba *et al.* [5] look at the last one, concluding that asynchrony does not have a negative effect on performance, and can even outperform synchronous ones; one of the authors [18] also looked at what would be the degree of asynchrony that would achieve the best algorithmic performance, applying also the above mentioned theory of intermediate disturbances.

The most exhaustive studies have been made on points 3 and 4 by Cantú-Paz: selection of outgoing immigrants and their incorporation into the population [7, 8]. Cantú-Paz studied the four possible combinations of random and fitness-based emigration and replacement of existing individuals. He found that the migration policy that causes the greatest reduction in work (measured as *takeover time*, that is, the number of generations required to converge to the best individual from the initial population, by applying selection only) is to choose both the immigrants and the replacements according to their fitness, because this policy increases the selection pressure and may cause the algorithm to converge significantly faster. However, if convergence is too fast it can lead to algorithm failure, as Cantú-Paz [8] states referring to parallel EAs:

> *Rapid convergence is desirable, but an excessively fast convergence may cause the EA to converge prematurely to a suboptimal solution.*

However, as stated by Denzinger and Kidney in [11], the effect of this high-fitness immigrant in the population will usually be to reduce diversity via combination with other high-fitness individuals, which will eventually lead to a diversity collapse, as high-fitness individuals are circulated from one node to others. In fact, it is usually observed that after a few rounds of immigration, populations no longer behave as isolated islands, and their composition is very similar. And, curiously enough, this is in accordance with the biological theory called the *intermediate disturbance hypothesis*, [28], which states that diversity is affected by disturbances depending on their magnitude: it will produce a collapse on diversity if it is too big, no effect if it is too small, and a maximum increase in diversity if it is in the intermediate region. In fact, this hypothesis explains some results [18] found in evolutionary algorithms.

There should, then, be a way of designing evolutionary algorithms so that diversity is preserved, if not enhanced. Some authors [1] have proposed a model that differs from the island model, and which follows an approach of segregation and reunification. In this case, subpopulations evolve independently until detecting local premature convergence, which is indicated by a selection pressure value computed in each subpopulation. If stagnation is detected the operations for this subpopulation are stopped until the next reunification phase is reached. Such a reunification phase is initiated, if all subpopulations have converged prematurely.

On the other hand, authors such as Alba and Troya [4] found that in the island model, the migration of a random string prevents the "conquest" effect in the target island for small or medium sized subpopulations; this could be explained by the immigrant individual behaving as an *intermediate* disturbance, as opposed to the strong disturbance initially provoked by the high-fitness individual. Finally, Noda *et al.* [21] proposed choosing which individuals to migrate and/or replace adaptively depending on some knowledge-oriented rules. To do this, each agent receives information about the fitness function from its peers. Besides, it considers, among other policies, one in which the individuals sent are chosen to be quite different from others previously sent. The tested adaptive policies have been proved useful, providing best solutions than the sequential execution. A later paper by Yang *et al.* [30] proposes selecting immigrants from an *elite* set, instead of random ones, and using them and their bit-wise complements as a pool for creating a set of immigrants; this yields good results in dynamic environments, which are known to need a high population diversity.

However, even as these mentioned policies having the effect of increased diversity and thus performance, diversity, by itself, is not acknowledged as one of the factors that should be dealt with explicitly. And one of the first papers to do so is the one mentioned above by Denzinger and Kidney in [11]. Their approach derives from sequential evolutionary algorithms that take into account diversity (in the form of difference to the rest of the member of the population) explicitly when selecting an individual like the algorithm proposed by de Jong and coauthors in [16], which converts a single-objective evolutionary algorithm in a multiobjective one that considers diversity as the second objective to optimize; and also the *diversity-guided evolutionary algorithm* [27], which alternates phases of exploration and exploitation depending on the *concentration* of individuals around a single point in search space. Denzinger *et al.* aggregate the difference of every individual to its fitness for a *quality* value that is then used to select the individuals that are going to be sent to other nodes. They test different weights for fitness and quality, concluding that a 70:30 proportion (fitness:difference) or 50:50 is a better choice that just fitness by itself, of using the difference in a larger proportion.

It should be noticed that so far, all policies of immigrant selection concern just the emitting node, not the receiving node. However, as remarked by Eldos in [12], the receiving node can also play a role in the selection of immigrants. Eldos uses a fitness threshold in the receiving node as the criteria for accepting or not an immigrant, without taking into account diversity, but other, earlier paper by Power *et al.* [22] considered it not only to select for migration a *representative* of the population (along with others, if the migration rate is high), but also to select which individual in the receiving population is going to be eliminated: the first criteria for replacement is the closeness to other members of the population, so as not to reduce diversity, eliminating members of the population which contain unique information.

The authors previously proposed the multikulti algorithm [6], which applied diversity considerations to the receiving population, such as the algorithms described above. It was tested on some discrete optimization problems, finding that it yields better results than random or best-individual migration policies in some cases. The multikulti algorithm selects individuals to send to other populations based on its difference with a representative chromosome of the receiving population. By sending strings that are *different enough*, it ultimately tries to increase performance via the diversity-boosting effect that an incoming *multicultural* immigrant might have.

However, in the previous instance of the algorithm (let's call it *multikulti*$_1$) the receiving population did not have a choice; in some cases, even if the received individual was the most different to the representative of the population (be it the best individual or the consensus sequence), it might actually be very similar to some individuals in the population, having thus no influence on diversity. That is why the multikulti policy described in this work (let's call it *multikulti*$_2$) also aims at exploiting genotypic differences in the various subpopulation by putting more emphasis on the receiving population than the emitting one. Our thesis is that migrating individuals different enough to the destination subpopulation instead of the best (or random) individuals can result in a better performance through the enhancement of diversity it produces. Instead of sending a single individual selected by the emitting population, we propose to send several individuals $b_1,\cdots,b_m$, in order to allow the receiving population to choose the most different one to promote its own diversity. In general, it can lead to explore new areas of the search space where the global optimum may be placed. In order to combine diversity and quality, the random sample selected to be sent is constrained to be above a threshold fitness value, which is previously given by the receiver node.

In this paper we will perform systematic experimentation of the selection of immigrants on two functions, and parallel environments with different number of nodes; the number of nodes has an impact on performance, but also on diversity, with more nodes usually meaning better performance (in average number of evaluations), but also a higher chance of premature convergence.

The rest of the paper is organized as follows: section 2 describes the model details, the evolutionary algorithm implemented and the problems to solve using it; section 3 presents and discusses the experimental results, and section 4 draws the main conclusions of this work.

2. MATERIALS AND METHODS

Before detailing the algorithms and problems considered, let us firstly describe more in depth how the multikulti principle is put at work and the problems used to test it.

2.1 Multikulti Policy

Without losing generality, we have considered a ring topology (Figure 1), in which each node can only send or receive information to/from the next and previous nodes in the ring. The node P_i receives from node P_{i+1} a message with a fitness value below which the immigrant (or any other individual, for that matter) would not be selected for reproduction; for instance, if 20% of the population is going to be eliminated in the rank-based selection scheme, the fitness of the first individual over that threshold is sent.

Once node P_i has got this information, it sends to node P_{i+1} a set of random individuals whose fitness is above that limit. The number of individuals is an adjustable parameter; in this work it is equivalent to the number of generations between migrations; that is, if every node waits for k generations before communicating with the others, k individuals are sent. The only rationale for doing this is to reduce the number of free parameters of the system, but obviously a more thorough investigation of these values should be done. Node P_{i+1} then uses some distance measure (in this work, the Hamming distance) to compute the average distance from the population to each immigrant, and picks the immigrant with the higher average distance. Please note that this model, while keeping the multicultural spirit of the former multikulti algorithm [6], is different in the sense that previously the most different individual was picked by the emitting node, while now it is done by the receiving node from a selected sample of immigrants, thus enabling more flexible strategies for immigrant management.

2.2 Experimental Setup

We use a traditional binary representation in this evolutionary algorithm; the selection mechanism used is linear ranking, population replacement follows a steady state policy (which substitutes only a part of the population in every generation; the number of individuals substituted is computed multiplying the selection rate by the population size); two-point crossover operator (except in some cases) and single-bit-flip mutation are used.

Two problems have been considered, namely P-Peaks and the massively multimodal deceptive problem (MMDP), which are discrete combinatorial optimization problems presented by Giacobini *et al.* in [13]. These problems, while being both multimodal, represent different degrees of difficulty for distributed evolutionary optimization. They will be described in next subsection.

The problems have been implemented and integrated in the `Algorithm::Evolutionary` library, which is freely available under the GPL license at `http://fon.gs/ae-perl/`. In order to simulate a parallel algorithm, the *cooperative multitasking* Perl module POE has been used; each node is represented by a POE *session*. The rest of the evolutionary algorithm has been implemented using the same `Algorithm::Evolutionary` Perl module [15]. This module can be used to simulate a parallel execution via a sequential program; since this paper was not concerned with running times, but algorithmic efficiency, the way it is implemented at a low level (parallel or sequential) is not important. The program, along with the parameter sets used, is also available under an open source license in the same site.

Tests were initially made for 2, 4 and 8 nodes, to see how the different migration methods fared under different initial diversity conditions. The total population was kept constant independently of the number of nodes, dividing it among the nodes. In this simulated distributed scenario, each node runs a deterministic rank-based substitution (the worst are always eliminated and substituted by descendants of the rest), steady state algorithm. At the end of a preset number of generations, each one sends a single individual to the other node according to the policy being tested. These problems, and the specific issues of their implementation, will be explained below.

2.3 Problems Considered

P-Peaks is a multimodal problem generator proposed by

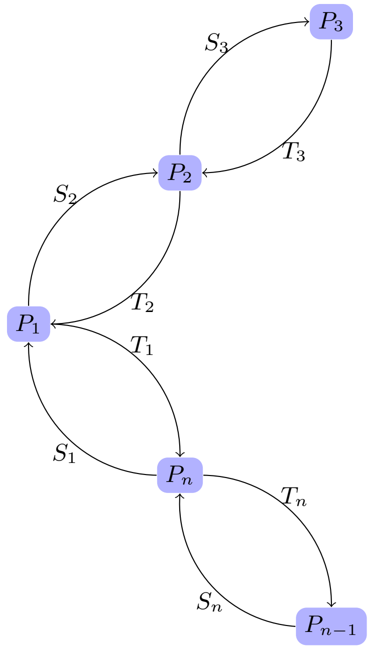

Figure 1: Scheme of the multikulti algorithm. A node, such as P_2, sends to the previous node P_1 the threshold fitness value T_2 above which the received immigrant's fitness must be. Then P_1 sends to the node P_2 a set S_2 of individuals randomly chosen among those with a fitness above the threshold.

Table 1: Evolutionary algorithm parameters used in the P-Peaks and MMDP experiments. Operator rates have been selected heuristically, by testing several values, with the objective of maximizing success rate.

	P-Peaks		MMDP	
Parameter			xOver	GBX
Chromosome length	100	128	90	120
Total population	512	256	2048	
Selection rate	20%		60%	20%
Migration period	10		20	
Mutation rate	1		40%	20%
2-point crossover rate	9		60%	80%
Max number of evaluations	100000		200000	

De Jong in [17], and is created by generating P random $N - bit$ strings $\vec{p}_i$, $1 \leqslant i \leqslant P$, where the fitness value of a string $\vec{x}$ is the number of bits that $\vec{x}$ has in common with the nearest peak divided by N, i.e.,

$$f_{P-Peaks}(\vec{x}) = \frac{\max_{1 \leqslant i \leqslant p}\{N - H(\vec{x}, \vec{p}_i)\}}{N} \qquad (1)$$

where $H(\vec{x}, \vec{y})$ is the Hamming distance between binary strings $\vec{x}$ and $\vec{y}$.

We consider an instance of $P = 100$ and 100 and 128 bits where the optimum fitness is 1.0. The parameters used in the EA are shown in Table 1. As mentioned above, the source for the experiments as well as the parameter files are available from our group's CVS server `http://fon.gs/ae-perl`.

On the other hand, MMDP [14] is a deceptive problem composed of k subproblems of 6 bits each one (s_i, $1 \leqslant i \leqslant k$). The fitness of a $6k-$bit string is computed as:

$$f_{MMDP}(\vec{x}) = \sum_{i=0}^{k-1} \phi\left(\sum_{j=1}^{6} x_{ik+j}\right) \qquad (2)$$

where function $\phi(u)$ is defined on the unitation of any $6-$bit segment as

$$\phi(u) = \begin{cases} 1 & u \in \{0,6\} \\ 0 & u \in \{1,5\} \\ 0.360384 & u \in \{2,4\} \\ 0.640576 & u = 3 \end{cases} \qquad (3)$$

The number of local optima is quite large (22^k), while there are only 2^k globally optimal solutions. In this paper, we have considered an instance with $k = 20$ subproblems, whose maximum is then $f_{MMDP}(\vec{s}) = 20$; this will be represented in a chromosome of length 120. Due to the nature of this problem, and the fact that in the initial tests the algorithm was not able to find the solution in many cases, we hypothesized that it might be due to the fact that the crossover was acting as a macromutation operator, thus disrupting the subproblems already *solved* by the EA. That is why, besides the traditional one point crossover, we used another crossover operator that *respects gene boundaries* (which we have called GBX), that is, only interchanges whole genes (whole subproblems, in this case) between parents, leaving the mutation function to the bit-flip mutation operator. Besides, these initial tests showed that results obtained by the 4- and 8-node system were much worse than for two nodes, possibly due to the lack of initial diversity in the smaller populations. The parameters used in the EA are shown in Table 1.

3. EXPERIMENTAL RESULTS

Each combination was run 30 times, with termination condition being success or a maximum of evaluations. Figures 2 and 3 show the results for the P-Peaks problem with 100 and 128 bits respectively.

There are several conclusions, which are tangential to the target of this paper. Firstly, the highest influence on the number of evaluations (algorithmic performance) is the number of nodes, with more nodes meaning less evaluations. As has been indicated in the state of the art, EAs profit from division in islands, achieving a double benefit: more evaluations happening at the same time, and also less evaluations needed to reach target. From two to eight nodes, P-Peaks needs half as many evaluations for all lengths (100 and 128).

Next conclusion is that while differences are not too big for the easiest problem ($l = 100$), on average, the multikulti method is better than non-adaptive *random* or *best* strategies for $l = 100, 128$. Multikulti algorithm is generally better on average (except for $n = 8$ nodes, when it is worse than the random strategy), and much better in median than the other methods for $l = 100$. A Mann-Whitney U test indicates that these differences are statistically significant at 0.15 level for four nodes. This difference does not follow a clear trend with the increasing number of nodes (and thus decreasing population in each node). Please note also that the other two methods, best and random do not have a clear advantage over each other, and the decreasing number of evaluations needed to reach optimum when the number of nodes increases. We think that the multikulti policy could be more advantageous for smaller populations where diversity is usually depleted

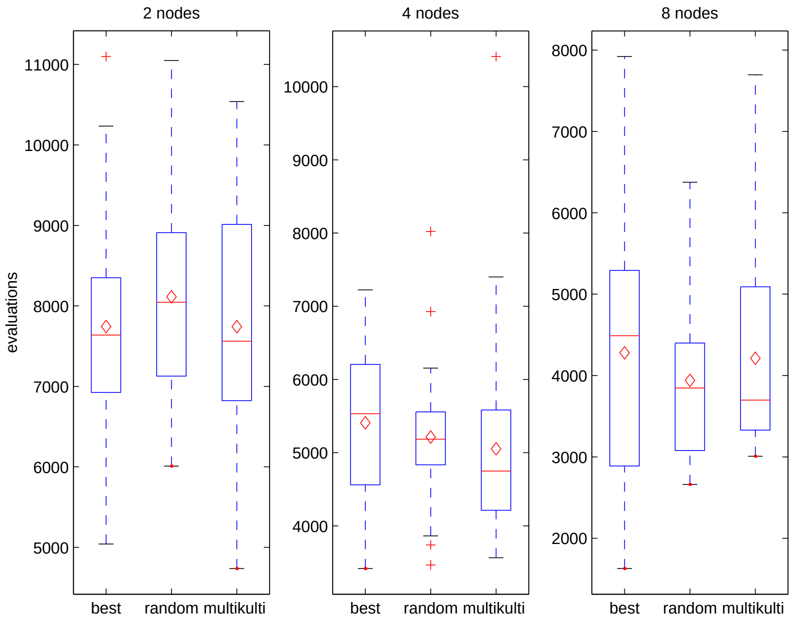

Figure 2: Boxplot of the number of evaluations needed to find the solution for P-Peaks with 100 bits for 2, 4 and 8 nodes. Averages are represented as red rhombs.

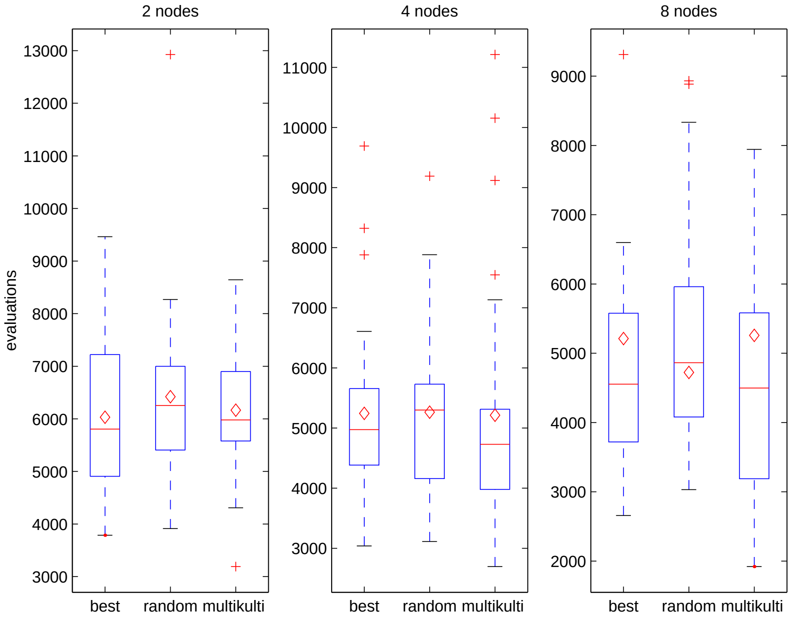

Figure 3: Boxplot of the number of evaluations needed to find the solution for P-Peaks with 128 bits for 2, 4 and 8 nodes. Averages are represented as red rhombs.

after few generations. This behavior can be observed in the experiments, in which multikulti obtains better results for 4 nodes than for 2. However in the case of 8 nodes, the population of each node, 64 for P-peaks with 100 peaks and 32 for P-peaks with 128 peaks, may be too small to provide samples that are different enough from the target population.

The same pattern appears with $l = 128$ (Figure 3), except that the initial population is smaller. Medians are better, once again except for $n = 2$, which is the second best. Statistical significance is achieved just for eight nodes, against the random strategy. In this case, the behavior of the multikulti algorithm seems to be better for small initial populations.

The conclusion from these experiments is that while trading immigrants that are different enough does have a definite effect on the performance, it is relatively easy to overshoot

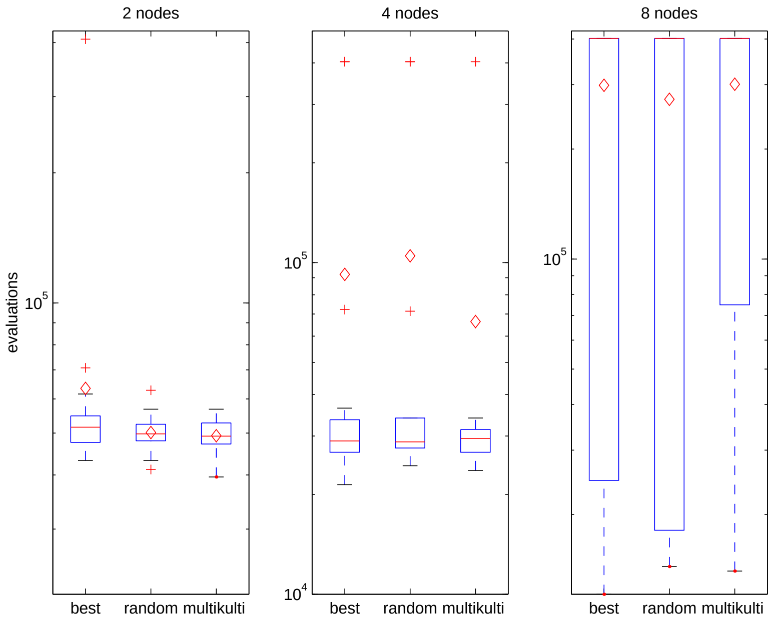

Figure 4: Boxplot (with logarithmic y axis) of the number of evaluations needed to find the solution for the 90-bit MMDP using two to eight nodes (left to right). Red rhombs represent the average number of evaluations.

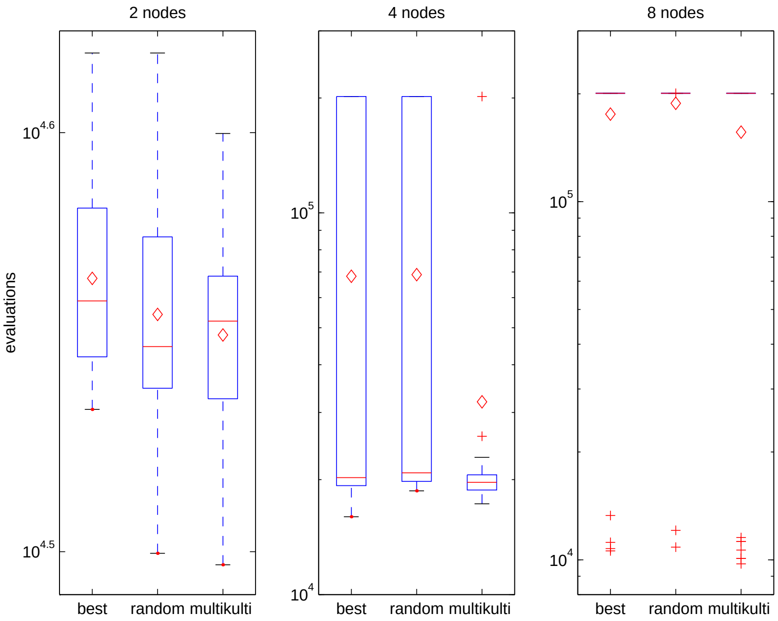

Figure 5: Boxplot (with logarithmic y axis) of the number of evaluations needed to find the solution for the 120-bit MMDP using (left to right) two to eight nodes; red rhombs represent the average number of evaluations. These results use the GBX crossover

the difference, or else being too similar to have any effect. It is easy to imagine, for instance, that at the end of the run all solutions are close to each other, and the difference between emitting and receiving population are not so big. In any case, sending the best or a random one might achieve good or bad results, depending on many factors (number of nodes, the problem length); using a *multiculturalism policy* is always close to the best or, in some cases, better than the rest.

In any case, we have used a more difficult problem, MMDP, to test the hypothesis of validity of multikulti policies, and how they become increasingly relevant with the difficulty of the problem. Experiments made on MMDP were, as above, performed with 30 runs for every parameter setting, and with two, four and eight nodes.

Figure 4 shows the results obtained for the MMDP problem. Multikulti is better on average for two and four nodes, and in median for two nodes but only slightly worse for four nodes than the others. Note however that statistical significance is achieved precisely on two nodes, against the best strategy. Once again, there are good results and they are apparently better when there is enough initial diversity available; the MMDP algorithm obtains worse results when initial diversity is less (the population is smaller), so the fact that multikulti is better in this situation probably indicates that it helps in keeping diversity (in the shape of entropy) high, but fails to increase diversity via exploration when whatever is available is small to start with.

The same problem tested with the GBX crossover operator yields the results shown in Figure 5. Please note that, in this case, crossover is set to a 80% rate, as opposed to the last graph; besides, this crossover respects gene boundaries, avoiding the introduction of additional mutations. As the graph shows clearly, average number of evaluations is much better for the multikulti policy, although the median is slightly worse for two nodes than the one for the random policy. At any rate, there is no statistical significance in this latter case. This significance is achieved on two nodes (against best), on four nodes (against both best and random), and on eight nodes (against random). This shows that, since the higher priority of crossover increases the degree of mixture with the incoming population, the impact of incoming immigrants is much bigger, and since they have been selected to have a difference as big as possible, they have a noticeable impact in the performance, decreasing the average number of evaluations by a few percentage points for two nodes, and almost halving it for four nodes. This improvement arises from the fact that the number of unsuccessful runs is much lower (just two for multikulti, closer to ten for the other methods); and this, in turn, means that the multikulti is able to successfully keep diversity high and prevent the populations inhabiting the nodes to fall into inbreeding.

4. CONCLUSIONS

This paper has explored a new alternative to promote diversity in an island model based on the concept of *multiculturalism*, and which we have called *multikulti*. This is achieved by selecting as immigrant individuals with a genotype different enough to the destination population.

Results in several problems show that, in most cases, the multikulti policies are able to outperform non-adaptive random and best-individual migration policies. While these policies do not show a clear trend of being the one better than the other, the version of multikulti presented in this paper is consistently better (or equal at the very least) than them in most circumstances, independently of the number of nodes (that is, diversity present in the initial population) and the problem tested, showing a clear advantage for the P-Peaks problem with 100 bits and 20-subproblem MMDP with the GBX (gene-boundary respecting crossover) operator, and slight advantages or disadvantages in the case of the 128-bit P-Peaks. However, the parameters used in every problem have been different, which is bound to have an impact in the results.

There is clearly also a trade-off between the different sources of diversity. Interchanging individuals is more an exploitation than exploration method, and thus is very limited for the creation of diversity. However, it can increase it locally and during some time in the receiving node, but mechanisms must be in place to make sure that it enters the reproductive tool, or at least that it has got a high probability of doing it; this provision applies to every migration method. That is why the combination of a high crossover (high degree of exploration) with big populations (higher initial diversity) often makes the multikulti method obtain better results.

This is probably one of the avenues of research we should use in the future: try to choose the method that best represents the whole population (consensus chromosome, such as the one used in [6], or others), and the method that better ensures that there is going to be a difference. The number of candidates to be sent is also an issue. It has been kept constant in this paper, but there is a trade-off between the amount of information that can be sent between nodes and what performance increases that can be obtained from it, so algorithmic performance will have to be analyzed along with (simulated or real) network traffic to obtain the best results. Past experiments by the authors [18] also show that starting populations at different time will have a positive impact on performance by decreasing the number of evaluations; we will study how late start and migration policies interact.

We also intend to develop a parallel implementation of the system, which will allow us to measure execution times too. In addition, we are working on alternative mechanisms to characterize the destination population, and thus select the more appropriate immigrants. We will also test results obtained by changing other algorithm parameter such as number of immigrants.

Acknowledgements

This paper has been funded in part by the Spanish MICINN projects NoHNES (Spanish Ministerio de Educación y Ciencia - TIN2007-68083) and TIN2008-05941 and the Junta de Andalucía P06-TIC-02025.

5. REFERENCES

[1] M. Affenzeller and S. Wagner. Sasegasa: A new generic parallel evolutionary algorithm for achieving highest quality results. *Journal of Heuristics*, 10(3):243–267, 2004.

[2] E. Alba and B. Dorronsoro. *Cellular Genetic Algorithms*. Springer, New York, 2008. ISBN 978-0-387-77609-5.

[3] E. Alba, A. J. Nebro, and J. M. Troya. Heterogeneous computing and parallel genetic algorithms. *J. Parallel Distrib. Comput.*, 62(9):1362–1385, 2002.

[4] E. Alba and J. M. Troya. Influence of the migration policy in parallel distributed gas with structured and panmictic populations. *Appl. Intell.*, 12(3):163–181, 2000.

[5] E. Alba and J. M. Troya. Analyzing synchronous and asynchronous parallel distributed genetic algorithms. *Future Generation Comp. Syst.*, 17(4):451–465, 2001.

[6] L. Araujo, J.J. Merelo, C. Cotta, and F. de Vega. MultiKulti Algorithm: Migrating the Most Different Genotypes in an Island Model. *Arxiv preprint arXiv:0806.2843*, 2008.

[7] E. Cantú-Paz. Migration policies and takeover times in genetic algorithms. In *GECCO*, page 775, 1999.

[8] E. Cantú-Paz. Migration policies, selection pressure, and parallel evolutionary algorithms. *Journal of Heuristics*, 7(4):311–334, 2001.

[9] R. J. Collins and D. R. Jefferson. Selection in massively parallel genetic algorithms. In R. K. Belew and L. B. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 249–256, San Diego, CA, 1991. Morgan Kaufmann.

[10] K. De Jong. Evolutionary computation: a unified approach. In *Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*, pages 2245–2258. ACM New York, NY, 2008.

[11] J. Denzinger and J. Kidney. Improving migration by diversity. In *2003 Congress on Evolutionary Computation*, volume 1, pages 700–707. IEEE Press, 2003.

[12] T. Eldos. A New Migration Model For Distributed Genetic Algorithms. In *Proceedings of the International Conference on Scientific Computing (CSC' 06), Las Vegas, NV*, pages 128–134, 2006.

[13] M. Giacobini, M. Preuss, and M. Tomassini. Effects of scale-free and small-world topologies on binary coded self-adaptive CEA. In J. Gottlieb and G. R. Raidl, editors, *Evolutionary Computation in Combinatorial Optimization – EvoCOP 2006*, volume 3906 of *LNCS*, pages 85–96, Budapest, 10-12 April 2006. Springer Verlag.

[14] D. E. Goldberg, K. Deb, and J. Horn. Massive multimodality, deception, and genetic algorithms. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, Amsterdam, 1992. Elsevier Science Publishers, B. V.

[15] J. J. Merelo. Evolutionary computation in Perl. In M. Perl Mongers, editor, *YAPC::Europe::2002*, pages 2–22, 2002.

[16] E. D. D. Jong, R. A. Watson, and J. B. Pollack. Reducing Bloat and Promoting Diversity using Multi-Objective Methods. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2001*, pages 11–18. Morgan Kaufmann Publishers, 2001.

[17] K. A. D. Jong, M. A. Potter, and W. M. Spears. Using problem generators to explore the effects of epistasis. In T. Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, San Francisco, CA, 1997. Morgan Kaufmann.

[18] J. J. Merelo, A. M. Mora, P. A. Castillo, J. L. J. Laredo, L. Araujo, K. C. Sharman, A. I. Esparcia-Alc'azar, E. Alfaro-Cid, and C. Cotta. Testing the intermediate disturbance hypothesis: Effect of asynchronous population incorporation on multi-deme evolutionary algorithms. In *Parallel Problem Solving from Nature - PPSN X*, pages 266–275, 2008.

[19] P. Moscato and M. G. Norman. A Memetic Approach for the Traveling Salesman Problem Implementation of a Computational Ecology for Combinatorial Optimization on Message-Passing Systems. In M. Valero, E. Onate, M. Jane, J. L. Larriba, and B. Suarez, editors, *Parallel Computing and Transputer Applications*, pages 177–186, Amsterdam, 1992. IOS Press.

[20] H. Muhlenbein. Evolution in time and space - the parallel genetic algorithm. In G. Rawlings, editor, *Foundations of Genetic Algorithms*, pages 316–337. Morgan Kaufmann, 1991.

[21] E. Noda, A. Coelho, I. Ricarte, A. Yamakami, and A. Freitas. Devising adaptive migration policies for cooperative distributed genetic algorithms. *IEEE International Conference on Systems, Man and Cybernetics*, page 6, 2002.

[22] D. Power, C. Ryan, and R. Azad. Promoting diversity using migration strategies in distributed genetic algorithms. *IEEE Conqress on Evolutionary Computation*, 2:1831–1838, 2005.

[23] P. Spiessens and B. Manderick. A massively parallel genetic algorithm: Implementation and first analysis. In R. K. Belew and L. B. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 279–287, San Diego, CA, 1991. Morgan Kaufmann.

[24] J. Stender. *Parallel Genetic Algorithms: Theory and Applications*, chapter Implementation in Occam of Parallel Genetic Algorithms on Transputer Networks. IOS Press, 1993.

[25] R. Tanese. Distributed genetic algorithms. In J. D. Schaeffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 434–439, San Mateo, California, 1989. Morgan Kaufmann Publishers.

[26] M. Tomassini. *Spatially Distributed Evolutionary Algorithms*. Natural Computing Series. Springer-Verlag, Berlin Heidelberg, 2005.

[27] R. K. Ursem. Diversity-guided evolutionary algorithms. In J. J. Merelo, P. Adamidis, H.-G. Beyer, J. L. F.-V. Martín, and H.-P. Schwefel, editors, *PPSN*, volume 2439 of *Lecture Notes in Computer Science*, pages 462–474. Springer, 2002.

[28] J. Ward and J. Stanford. Intermediate-Disturbance Hypothesis: An Explanation for Biotic Diversity Patterns in Lotic Ecosystems. *Dynamics of Lotic Systems, Ann Arbor Science, Ann Arbor MI. 1983. 347-356 p, 2 fig, 35 ref.*, 1983.

[29] D. Whitley, S. Rana, and R. Heckendorn. The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7:33–47, 1999.

[30] S. Yang and R. Tinós. A hybrid immigrants scheme for genetic algorithms in dynamic environments. *International Journal of Automation and Computing*, 4(3):243–254, 2007.