A Dynamic Island-Based Genetic Algorithms Framework

Frédéric Lardeux, Adrien Goëffon

A Dynamic Island-Based Genetic Algorithms Framework

Frédéric Lardeux and Adrien Goëffon

LERIA, Université d'Angers
UFR Sciences, 2 bd Lavoisier
49045 Angers, FRANCE
frederic.lardeux@univ-angers.fr
adrien.goeffon@univ-angers.fr

Abstract. This work presents a dynamic island model framework for helping the resolution of combinatorial optimization problems with evolutionary algorithms. In this framework, the possible migrations among islands are represented by a complete graph. The migrations probabilities associated to each edge are dynamically updated with respect to the last migrations impact. This new framework is tested on the well-known 0/1 Knapsack problem and MAX-SAT problem. Good results are obtained and several properties of this framework are studied.

1 Introduction

Genetic algorithms (GAs) [8] are widely used to tackle NP-hard problems. They are easy to implement and can provide good results on classic discrete and continuous optimization problems in term of solutions quality and robustness. Nevertheless, the efficiency of GAs mainly depends on the representation of configurations [15], the fitness function [26], the mutation and crossover operators used [18] as well as global parametrization (population size, mutation frequency, diversity control, selections, elitism, ...) [10, 9, 22]. Even with a good effort to adapt an efficient GA to a given problem, one quickly observes on critical problem instances some limitations in terms of general performance or scalability.

In order to make GAs more powerful, classic techniques include hybridizations with local search (memetic algorithms [16]) and/or multi-island parallelization schemes [25], which we are investigating in this paper.

Since twenty years and the first distributed evolutionary algorithms [23], island-based genetic algorithms (or island models [25]) are more and more studied in the community. The main problem is to define both the model topology and the migration policies in order to slow down the general convergence of the population while preserving the global mixing of promising individuals. Araujo *et al.* give in [2] a nice review of state-of-the-art island models, in particular concerning the question of migration policies. One can observe that an important number of topologies (Gustafson and Burke in [11] or Rucinski *et al.* in [19] cite numerous topologies models like chains, rings, hypercube and many more) and policies [3, 6, 24, 5, 7, 1], greatly based on local or global diversity measures, have been defined. In all cases, migration sizes and intervals remain

difficult to fix [21]. In his recent work, Skolicki [20] emphasizes the fondamental interactions between the two levels of evolution in island models: intra-islands and inter-islands. Ideally, a master intelligent evolution strategy should take advantage of these interactions and maximize the benefits of migrations. But, depending on the current intra- and inter-islands situations (traditionnally only with diversity and fitness measures), it is difficult to predict when individuals have to move, which ones and where, and for which impact.

These considerations motivate us to develop a dynamic island model framework which aims to auto-adapt topology and migration policies during the search in function of some chosen indicators (typically subpopulations properties and previous migrations effects). A particularity of our dynamic island model is to use a complete graph modeling. Nodes represent islands while edges symbolize possible migrations.

Section 2 contains both general and concrete descriptions of our dynamic island model framework. In section 3, we apply our model to two benchmark problems: 0/1 KP and MAX-SAT. Section 4 is a short discussion with additional experiments, with a view to measure the influence of migrations. The conclusion includes future investigations.

2 Dynamic Island Model Framework

2.1 General description

As recalled by [2], several parameters specify an island model, like:

- the number of individuals undergoing migration,
- the frequency of migrations,
- the policy for selecting immigrants,
- the immigrant replacement policy,
- the topology of the communication among subpopulations, and
- the synchronous or asynchronous nature of the connection among subpopulations.

Now, let us propose an island model framework which generalizes all these parameters, while giving us the possibility to make the model dynamic.

The island model is materialised by a graph, where vertices symbolize islands (subpopulations), while edges represent the possibilities of migrations. Each edge is oriented, and valued with the probability for an individual to migrate from an island to a destination one. The auto-adaptation of this modeling is made with a reward/penalty mechanism. Migration probabilities (values of the edges) are updated after each migration cycle in function of the last migration effects. If the island which receives an individual observes any improvement (resp. deterioration) of its population, then the corresponding migration probability increases (resp. decreases). Here, the population quality is impacted by the average fitness of individuals as well as their diversity if the modeling imposes it.

The dynamic control of parameters like migration rate, can produce different size islands (unless we specifically forbid it). This mechanism prevents poor-quality subpopulations to require as many computational effort as promising ones, and manages the merging of populations. If different islands represent different mutation operators, local search effort or local parametrization, then the algorithm will dynamically provides a well-adapted repartition of individuals considering the search progression.

2.2 Practical use of the framework

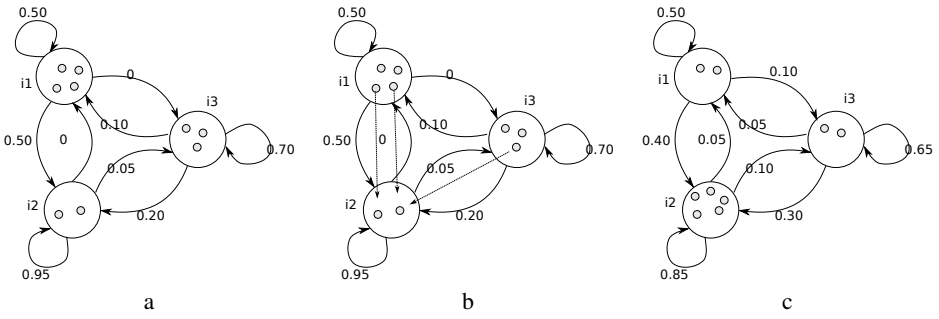

Fig. 1. Communication among subpopulations with a complete graph representation

Figure 1 is an example of our Island Model framework with three islands (i^1, i^2 and i^3). Figure 1.a represents islands with their individuals as well as the migration values (probabilities) from an island to another. In figure 1.b, the destination for each individual are chosen with respect to the values. Most of them remain in the same island (due to the probability values close to 1) but several individuals migrate to other islands (two individuals go from i^1 to i^2 and one from i^3 to i^2). After those migrations, on each island, operators like crossover, migration, or local search, are applied on the individuals. If an offspring individual (*i.e.* an individual obtained by crossover) improves (resp. deteriorates) the population, then its parents are used to update the migration values. For each parent, the values of edges between the last visited island and the current one are increased (resp. decreased) to take into account the impact of the migration. For instance, on Figure 1.c, if we only observe i^2 and a reward/penaly fixed to 5 points (± 0.05), several values are updated:

- $(i^1 \rightarrow i^2)$ decreases from 0.50 to 0.40, because the two indivduals becoming from i^1 have produced individual offspring deterioring the population of i^2;
- $(i^2 \rightarrow i^2)$ decreases from 0.95 to 0.85, because individuals of island i^2 do not improve the population of i^2;
- $(i^3 \rightarrow i^2)$ increases from 0.20 to 0.30, because the individual becoming from i^3 has produced individual offsprings improving the population of i^2;
- due to normalization, other values in relation with i^2 have to be adjusted.

3 Results

In this section we propose to measure the overall efficiency of the Dynamic Island Model scheme (DIM), applied to two well-known NP-hard problems: 0/1 Knapsack problem and MAX-SAT problem. For this study, the main goal is not to propose a ready-to-use algorithm which outperforms best available softwares, but to measure the global relevance of such a model. For this purpose, we compare for both problems the performance of four basic configurations of the GA:

– A classic 1-island GA ($\mathrm{GA}_{\mathrm{classic}}$),
– A standard DIM algorithm ($\mathrm{DIM}_{\mathrm{stand}}$),
– A specially-parametrized DIM algorithm with uni-directional ring topology $\mathrm{DIM}_{\mathrm{ring}}$, which simulates a classic island-based GA with rotative migrations (at each migration process, best individuals migrate to the following populations),
– A parallel GA ($\mathrm{GA}/\!/$), with several islands but no migration.

Let us notice that a classic GA corresponds to an island-based GA where migration intervals are minimal, while the parallel partitioned GA ($\mathrm{GA}/\!/$) is an island-based GA with no migrations (*i.e.* infinite migration intervals).

3.1 Genetic algorithms characteristics

The four configurations of the GA have two types of characteristics. All the numerical values are empirically obtained and confirmed by the REVAC method [17].

1. Intra-islands characteristics:
   – type of population management: steady state
   – elitism: yes
   – selection: tournament
   – mutation: random on offspring with probability 0.5
   – crossover: uniform crossover
2. Inter-islands characterisics:
   – islands number: 20
   – total number of individuals: 600
   – starting repartition: well-balanced (30 individuals per island)
   – total number of crossovers: 216000 (360×600 individuals)
   – initial migration probabilities: see below
   – reward: 5 points
   – penalty: 5 points

The number of crossovers in an island between two migrations is proportional to its number of individuals. This choice ensures the same crossover rate per island, whatever its size.

Initial migration probabilities To give the same attractive power to each island, the initial migration probabilities must be symmetric. At the beginning, we fix a highest probability to stay on the same island than to move to another one, in order to exploit initial populations. For instance, the initial matrix corresponding to an island model with three islands can be the next one:

		Destination		
		i^1	i^2	i^3
Source	i^1	0.75	0.125	0.125
	i^2	0.125	0.75	0.125
	i^3	0.125	0.125	0.75

3.2 Experimental Settings

Algorithms used in our experiments are applied 10 times for each instance. To be sure that the difference of behaviours is not due to the initial populations and other stochastic factors, 10 distinct random seeds are used by each algorithm. Results presented in the tables are averages; standard deviations are not mentioned since they are very low.

3.3 0/1 Knapsack Problem

The Knapsack Problem (KP) is a well-known combinatorial problem. Given n items whose weights w_i and values v_i are known ($x_i \in \{1, \ldots, n\}$), the goal is to find a subset of items of maximal value such that the total weight is less than a given capacity W. In the most common 0/1 KP, each item can be selected only once ($x_i \in \{0, 1\}$, where x_i is the number of selected copies of object i).

More precisely, 0/1 KP is shortly formulated as an optimization problem by:

$$\text{maximize} \sum_{i=1}^{n} v_i x_i, \text{s.t.} \sum_{i=1}^{n} w_i x_i \leq W, x_i \in \{0, 1\}$$

For more information on 0/1 KP, we invite the reader to refer to [12].

Island-based algorithms $\mathrm{DIM}_{\mathrm{stand}}$ and $\mathrm{DIM}_{\mathrm{ring}}$, as well as edgeless topologies $\mathrm{GA}_{\mathrm{classic}}$ and $\mathrm{GA}/\!\!/$, have been tested on five 0/1 KP instances. Instances have been generated according to the definition given by [4] and the generator proposed in [13], with the following parameters:

- number of items $\in \{100, 250, 500, 1000, 2000\}$
- range of coefficient: 10000
- type: avis subset-sum
- number of tests in series: 1000

Experiments have shown that only the three last instances (those with resp. 500, 1000 and 2000 items) are representative for comparison, the two first ones appearing too much easy to solve, with similar results for all algorithms. Consequently, we only focus on three random instances: n500, n1000 and n2000.

Table 1 shows the efficiency of each method on these instances. It is not surprising that the Dynamic Island Model outperforms traditional GAs. However, performance differences are quite important, taking into account that last improvements are particularly hard to find for knapsack problems. An interesting point is that, in this experiment, the classic rotative scheme ($\mathrm{DIM}_{\mathrm{ring}}$) is not competitive; comparatively, the classic GA works even better for the two hardest instances. The main reason is probably the relatively-small size of islands (20 individuals), which is adaptative in $\mathrm{DIM}_{\mathrm{stand}}$ while it remains unchanged during the entire process in $\mathrm{DIM}_{\mathrm{ring}}$.

3.4 MAX-SAT Problem

In order to test the DIM framework with an other problem, we try to handle the MAX-SAT problem. Given a Boolean formula in CNF (conjunction of clauses which are disjunctions of literals), the aim is to provide an assignment to the Boolean variables such

Instance	GA	$\mathrm{DIM}_{\mathrm{stand}}$	$\mathrm{DIM}_{\mathrm{ring}}$	GA$/\!/$
n500	755 626.54	**760 620.50**	755 818.38	748 230.25
n1000	1 485 393.86	**1 502 549.22**	1 483 248.36	1 465 757.47
n2000	2 866 752.92	**2 910 891.46**	2 853 072.50	2 832 290.37

Table 1. Comparison between DIM and classic GAs

that the number of true clauses is maximum. The formula is satisfiable iff it exists an assignment which makes true all the clauses.

Three instances are used for our experiments:

- f600: random instance with 600 variables and 2550 clauses;
- f1000: random instance with 1000 variables and 4250 clauses;
- qg1-7.suffled: latin square instance with 686 variables and 6816 clauses.

All these instances are satisfiable thus there is an assignment of the Boolean variables satisfying all the clauses.

Instance	Nb Clauses	GA	$\mathrm{DIM}_{\mathrm{stand}}$	$\mathrm{DIM}_{\mathrm{ring}}$	GA$/\!/$
f600	2550	2513.80	**2533.40**	2518.70	2357.20
f1000	4250	4174.20	**4208.90**	4174.10	3890.50
qg1-7.shuffled	6816	6756.30	**6787.00**	6776.50	6211.70

Table 2. Comparison between DIM and classic GAs

In table 2, we observe that $\mathrm{DIM}_{\mathrm{stand}}$ provides the best results on the three instances. GA$/\!/$ is the worst and GA obtains a little less interesting results than $\mathrm{DIM}_{\mathrm{ring}}$. The results for $\mathrm{DIM}_{\mathrm{stand}}$ and $\mathrm{DIM}_{\mathrm{ring}}$ are computed with the best migration frequency empirically found. In the next section, a more detailled study of this parameter is given.

4 Discussion

As seen in section 3.3 and 3.4, $\mathrm{DIM}_{\mathrm{stand}}$ provides very promising results with respect to the other GAs. The difference between $\mathrm{DIM}_{\mathrm{stand}}$ and $\mathrm{DIM}_{\mathrm{ring}}$ is only concerning the type of migration, whereas the difference among GA$/\!/$, GA and $\mathrm{DIM}_{\mathrm{stand}}$ is the migrations periodicity. This periodicity is given by a mean number of crossovers per individuals. Then, the number of crossovers between two migrations differs from an island to another and depends on their size (number of individuals).

GA can be considered as an island model with a very weak migrations periodicity and GA$/\!/$ with a very strong migrations periodicity (recall that a weak periodicity corresponds to a high frequency). Between these two algorithms, a dynamic island model can use different frequencies which provide different algorithm behaviours. In figure 2, one can observe the impact of the migrations periodicity on all the studied 0/1 KP and MAX-SAT instances in this paper.

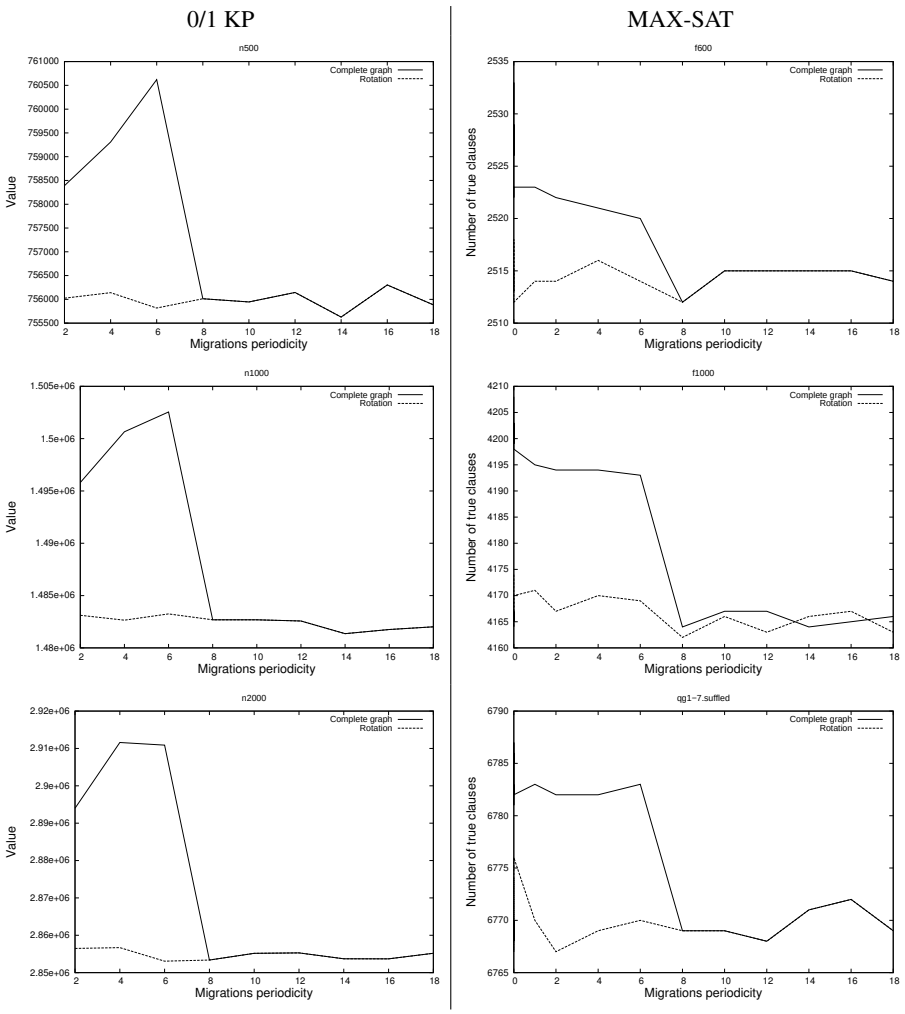

Fig. 2. Powerfull of the DIM with several values of migrations periodicity

It is interesting to see that, when the migrations periodicity is higher than 8 crossovers per individual, $\mathrm{DIM}_{\mathrm{stand}}$ and $\mathrm{DIM}_{\mathrm{ring}}$ provide equivalent results. A possible explanation of this behavior is that the population is converging before the first migration. Indeed, in our experiments, 240 crossovers are applied on the population of each island before the first migration. If each population only contains clones after these crossovers, then rotative migration provides the same effect than the complete graph migration scheme. Let us precise that in this study, we have deliberately not regulated the diversity of the population, in order to observe more precisely the behaviour of the models. That is the reason for which scores are lower than those shown in Table 2.

An other observation is the weak difference among the results obtained by $\mathrm{DIM}_{\mathrm{ring}}$ for all values of migrations periodicity. Except for instance `qg1-7.suffled`, where results are better with a small periodicity, this parameter seems not determinant for the rotative migration scheme.

With a migrations periodicity smaller than 8, the complete graph migration scheme is clearly better than the rotative one. For the 0/1 KP instances, value 6 seems to be the best periodicity. For MAX-SAT instances, a small periodicity provides very good results. The reason is probably that a small periodicity avoid the convergence of all the populations.

It is clear that the migration periodicity is an important parameter for $\mathrm{DIM}_{\mathrm{stand}}$. A next work will be to control it with autonomous mechanism like in [14].

5 Conclusion

In this work, we have introduced a dynamic island model framework, which aims both to generalize migration toplogies and to auto-regulate migration policies. First, the complete graph modeling allows every definition of topologies, from edgeless graphs (standard sequential or parallel genetic algorithms) to well-studied island model topologies like uni-directional or bi-directional rings, lattices, hypercubes, full topologies or anything else. If the dynamic regulation is activated, then the topology is evolving during the search following the rewards and penalties due to previous migration effects. Contrary to traditional island model mechanisms, where migrations are evaluated *a priori* in measuring divergence between individuals (which is much more a guided repartition of individuals to provide diversity, but a nonsense in a nature-inspired algorithm), we encourage (resp. dissuade) moves of which previous executions and produced mixing yield good (resp. weak) offsprings, in term of fitness and/or diversity. During the search, this auto-regulation of migration probabilities makes the model more or less dynamic in terms of number of migrations, which favouring either diversification, or intensification.

Experiments realized on two major combinatorial problems like 0/1 KP and MAX-SAT show that this dynamic scheme, even with basic parametrization, provides good results, notably if we compare its performance with a classic uni-directional ring migration topology.

The most promising prospect of our ongoing and future works is to parametrize differently each island. One can imagine that different islands can work with their own rules in terms of mutation or crossover operators, selection or replacement criterions for instance. In particular, considering local search operators, if different islands working with different operators, or different parametrizations of the search (more intensive or more stochastic), are coexisting within a dynamic island model, it would be interesting to observe the evolution of each island activity during the search. We think that such a model can provide an adaptative operator selection as well as a diversity regulation due to the island topology.

References

1. Lourdes Araujo, Juan Julián Merelo Guervós, Carlos Cotta, and Francisco Fernández de Vega. Multikulti algorithm: Migrating the most different genotypes in an island model.

CoRR, abs/0806.2843, 2008.

2. Lourdes Araujo, Juan Julián Merelo Guervós, Antonio Mora, and Carlos Cotta. Genotypic differences and migration policies in an island model. In *GECCO*, pages 1331–1338, 2009.
3. Erick Cantú-Paz. Migration policies, selection pressure, and parallel evolutionary algorithms. *Journal of Heuristics*, 7(4):311–334, 2001.
4. Vasek Chvátal. Hard knapsack problems. *Operations Research*, 28:1402–1411, 1980.
5. Jörg Denzinger and Jordan Kidney. Improving migration by diversity. In *IEEE Congress on Evolutionary Computation (1)*, pages 700–707, 2003.
6. I. L. M. Ricarte A. Yamakami E. Noda, A. L. V. Coelho and A. A. Freitas. Devising adaptive migration policies for cooperative distributed genetic algorithms. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 438–443, 2002.
7. Taisir Eldos. A new migration model for the distributed genetic algorithms. In *Proceedings of the International Conference on Scientific Computing (CSC06), Las Vegas, NV*, pages 26–29.
8. David E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, Mass., 1989.
9. David E. Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, 1991.
10. David E. Goldberg, Kalyanmoy Deb, and James H. Clark. Genetic algorithms, noise, and the sizing of populations. *COMPLEX SYSTEMS*, 6:333–362, 1991.
11. Steven Gustafson and Edmund K. Burke. The speciating island model: An alternative parallel evolutionary algorithm. *J. Parallel Distrib. Comput.*, 66(8):1025–1036, 2006.
12. Michail G. Lagoudakis. The 0-1 knapsack problem – an introductory survey. Technical report, University of Southwestern Louisiana, 1996.
13. Silvano Martello, David Pisinger, and Paolo Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Manage. Sci.*, 45(3):414–424, 1999.
14. Jorge Maturana, Frédéric Lardeux, and Frédéric Saubion. Autonomous operator management for evolutionary algorithms. *Journal of Heuristics*, in press, 2010.
15. Zbigniew Michalewicz. *Genetic algorithms + data structures = evolution programs (3rd ed.)*. Springer-Verlag, London, UK, 1996.
16. Pablo Moscato. Memetic algorithms: a short introduction. pages 219–234, 1999.
17. Volker Nannen, Selmar K. Smit, and Agoston E. Eiben. Costs and benefits of tuning parameters of evolutionary algorithms. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature*, pages 528–538, Berlin, Heidelberg, 2008. Springer-Verlag.
18. J. N. Richter. *On Mutation and Crossover in the Theory of Evoutionary Algorithms*. PhD thesis, Montana State University, 2010.
19. Marek Rucinski, Dario Izzo, and Francesco Biscani. On the impact of the migration topology on the island model. *CoRR*, abs/1004.4541, 2010.
20. Zbigniew Skolicki. *Linkage in Island Models*, chapter Studies in Computational Intelligence, pages 41–60. Springer Berlin / Heidelberg, 2008.
21. Zbigniew Skolicki and Kenneth A. De Jong. The influence of migration sizes and intervals on island models. In *GECCO*, pages 1295–1302, 2005.
22. William M. Spears. *Evolutionary Algorithms: The Role of Mutation and Recombination*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2000.
23. Reiko Tanese. Distributed genetic algorithms. In *Proceedings of the third international conference on Genetic algorithms*, pages 434–439, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

24. Rasmus K. Ursem. Diversity-guided evolutionary algorithms. In Juan J. Merelo Guervos, Panagiotis Adamidis, Hans-Georg Beyer, Jose Luis Fernandez-Villacanas Martin, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VII, 7th International Conference, Granada, Spain, September 7-11, 2002, Proceedings*, volume 2439 of *Lecture Notes in Computer Science*, pages 462–474. Springer, 2002.
25. Darrell Whitley, Soraya Rana, and Robert B. Heckendorn. The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7:33–47, 1998.
26. D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997.