Autonomous Local Search Algorithms with Island Representation

Adrien Goëffon and Frédéric Lardeux

LERIA, University of Angers (France)
name@info.univ-angers.fr

Abstract. The aim of this work is to use island models to autonomously select local search operators within a classical population-based local search algorithm. In this representation, each island is associated to a particular LS operator. Here, islands partitionning does not allow to forecast the most promising crossovers between individuals but to detect at each time of the search the most relevant operators. This application constitutes an original approach in defining autonomous algorithms.

1 Introduction

Island Models [9] are simultaneously considering a set of populations clustered in islands which are evolving independently during some search stages while interacting periodically. This model, which constitutes an additional abstraction level in comparison to classical genetic and memetic algorithms, allows to propose several diversification levels and to simplify its parallelization.

Island models are ofter used in a static way, where individuals are migrating from population to population following a determinate scheme [7], or are specifically chosen in order to reinforce the populations diversities [8, 4, 1]. Nevertheless, it is possible to dynamically regulate migrations between islands in considering a transition matrix [5]. Such a model can reinforce or reduce the migration probabilities during the evolutionary process in function of the impact of previous analog migrations. The aim is to auto-adapt migration without any given scheme, to dynamically regulate the gathering or isolation of individuals in function of the search progress, and consequently to adapt the population sizes.

In classical uniform island models, islands are following the same evolutionary rules, so they differ only by their individuals. The dynamic model allows to regulate interactions between individuals or group of individuals. We propose to extend this model in assigning to each island different local search operators. An appropriate and autonomous regulation of migration flows will affect dynamically the resources to the most pertinent operators in function of the search progress. In experimenting this model without crossovers but with a proper local search operator for each island, the objective is not only to regulate the interactions between individuals, but to simulate a reactive controller which assigns individuals to the most promising islands.

2 Island Models Representation

2.1 Topology

In [5] we proposed an island model framework which dynamically supervises the commonly-used specification parameters [1] like the number of individuals undergoing migration, the policy for selecting immigrants or the topology of the communication among subpopulations. An island model topology is represented by a complete labeled digraph $G = (X, X^2)$.

Migration policies are given by a transition (stochastic) matrix T, where $T(i, j)$ represents the probability for an individual to migrate from island i to island j (or to stay at the same island if $i = j$). One can denote T_t the matrix at time (or generation) t.

An application of this dynamic evolution of the model topology is to determine pertinent migration probabilities at each time of the search, considering a classical multi-population based genetic algorithm. The dynamic regulation of migration policies can produce different size islands, which prevents poor-quality subpopulations or islands to require as many computational effort as promising ones. However, if different islands represent different mutation or local search operators, then the aim is to dynamically provide a well-adapted repartition of individuals in function of these operators and considering the search progression, which can be assimilated to an operator selection process.

2.2 Migration Policy

Algorithm 1 is the generic algorithm we used for the autonomous operator selection within an island model context. In order to allow a maximum of adaptability, we chose to update the migration process after each local search iteration (for the whole population). Ideational, less frequent mutations process do not minimize the effective number of mutations (individuals moving to other islands) but only provide a less reactive search. As a dynamic algorithm, transition values are expected to be regulate accordingly.

```
Initialize population;
repeat
    foreach population do
        foreach individual do
            One local search iteration;
    Update the Transition Matrix T;
    Migration Process;
until stop condition;
```

Algorithm 1: Generic Dynamic Island Model (DIM) Algorithm.

The crucial point concerns the transition matrix update, which follows a learning process:

$$T_t = (1-\beta)(\alpha.T_{t-1} + (1-\alpha).R_t) + \beta.N_t$$

T_t is the transition matrix computed after migration process $t-1$ and LS step t. R_t is a *reward matrix* which takes into account the comparative pertinence of last migrations (migration process $t-1$). Using an *intensive* strategy, for each island, migration which have brought the best average accuracy score *acc* of individuals (typically their fitness improvement) receives a most important reward. More formally, if M_{ijt} is the set of individuals which have migrated from island i to island j in migration process $t-1$ ($\cup_i M_{ijt}$ is the set of individuals in island j during iteration t):

$$R_t(i,j) = \begin{cases} 1/|B| & \text{if } j \in B, \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{with } B = \underset{j'}{\operatorname{argmax}} \frac{\sum_{x \in M_{ij't}} acc(x)}{|M_{ij't}|}$$

N_t is a noise stochastic matrix with random values.

Both parameters α and β allow to manage the transition matrix update. α represents the importance of the knowledge accumulated during last migrations and β the amount of noise which is necessary to explore alternative ways and to keep the model reactive.

3 Experimentations

In this section we show that the behavior of our population-based local search algorithm is very close of the theoretical results. Moreover, we remark that it is not very dependent of the parameter tuning.

3.1 One-Max Problem

The One-Max problem is commonly used to assess the performance of Adaptive Operator Selection algorithms [3, 2]. The n-bits One-Max problem considers n-length bit strings; starting from 0^n individuals (*i.e.* strings made up of n *zeros*), the aim is to maximize the number of *ones*, that is to reach the 1^n bit string. The *score* of a bit string x, noted $|x|_1$, corresponds to its number of *ones*.

Recent works cited above use four mutation (or local search) operators: *bit-flip*, which flips every bit with probability $1/n$, and *k-flip* (with $k = \{1, 3, 5\}$), which flips exactly k bits. In the following and depending on the context, bit-flip and k-flip can denote the mutation operator as well as the corresponding neighborhood relation. k-flip can easily be modelized as a neighborhood relation $\mathcal{N}_k : \{0,1\}^n \to 2^{\{0,1\}^n}$ such as $x' \in \mathcal{N}_k(x)$ if and only if $|h(x,x')| = k$ (hamming distance). It is more difficult to exprim the bit-flip operator with a neighborhood relation, since it corresponds to a complete neighborhood with a non-uniform move probability. However, a bit-flip move can be reduced in a k-flip move with a determined probability of chosing k.

Intuitively, the 5-flip operator mutation will be more efficient when applied on weak individuals (with a majority of *zeros*), while 1-flip will improve with a higher probability individuals with a high proportion of *ones*. The domination rates evolution of the four considered operators in function of the score of an individual is shown in Figure 1 (with $M = \{$ 1-flip, 3-flip, 5-flip, bit-flip $\}$).

3.2 Theoretical vs Empirical Results

The expected behavior during the search is to use the 5-flip operator when the population quality is weak (at the beginning), then the 3-flip operator and finally the 1-flip and bit-flip operators when the population quality is sufficiently high. In our experiments, this can be observed by the evolution of the population size in each island with respect to the migrations. More an island attracts individuals, more its assigned operator is applied.

Parameters for this experiment are:

- number of islands: 4 (one for each LS operator)
- population size: 400
- initial probabilities of migrations: 1 to stay in the same island
- (α, β): $(0.8, 0.01)$

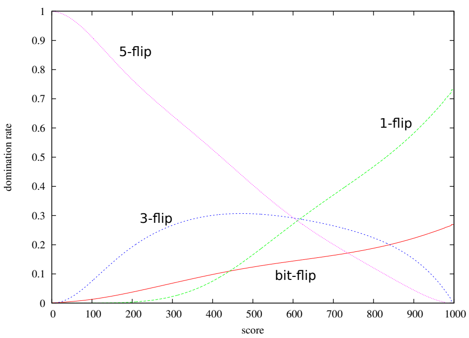


Fig. 1. Domination rates evolution for the 1000-bits One-Max problem.

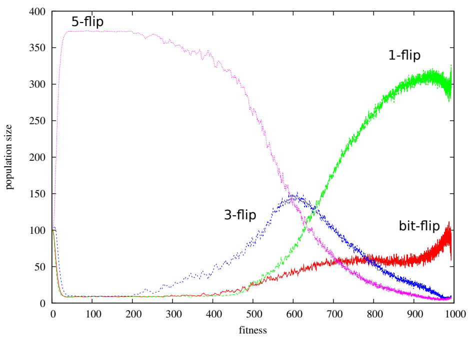


Fig. 2. Evolution of the population size in each island with respect to the average fitness of the population.

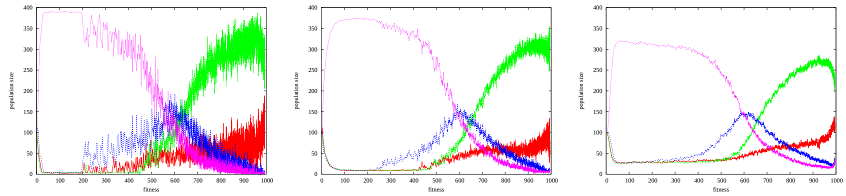
To compare the experimental results with the theoretical values, we represent in Figure 2 the population size in each island with respect to the average fitness of the population. The fact that this evolution of population sizes, *i.e.* the computational effort of each operators, match with the theoretical domination rates, show the accuracy of the proposed model and its pertinence to simulate an operator selection mechanism.

3.3 Dynamic Model Parameters

Default used values for α and β are respectively 0.8 and 0.01. An increasing value of α makes the search slower since informations obtained by recent migrations are less considered for the update. On the contrary, decreasing value of α minimizes the impact of the knowledge (learning process) and overestimates the last migration effects, so the search can be wrong oriented by a migration which provides exceptionally a good result.

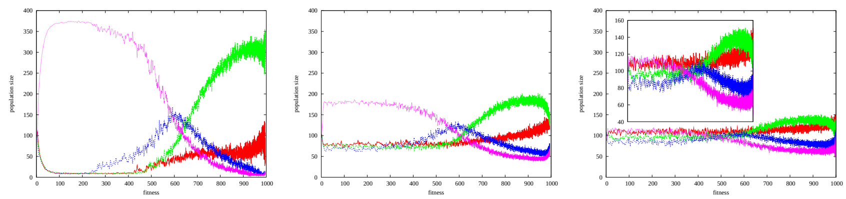
The influence of β is important, but its exact setting is not crucial to the smooth-running of the algorithm, even if a too high value of β make the search slower. On the other hand, it must make sure that $\beta \neq 0$, otherwise some islands can become and stay unreachable (transition probability equal to 0).

Effect of parameters α and β on the model are experimentally shown Figures 7 and 8.



(a) (α,β)=(0.4,0.01) (b) (α,β)=(0.8,0.01) (c) (α,β)=(0.95,0.01)

Fig. 3. Changing the value of α: less or more inertness makes the model more stable but does not modify the global repartition of individuals



(a) (α,β)=(0.8,0.01) (b) (α,β)=(0.8,0.2) (c) (α,β)=(0.8,0.5)

Fig. 4. Changing the value of β: more noise makes the repartition of individuals more uniform

4 Conclusion

This paper presents an original and efficient approach to design an autonomous local search algorithm with an accurate selection of operators. Contrary to other approaches like [6], the proposed mechanism use a dynamic island model, where each island represents an operator. A learning process regulates and adapts migration policies during the search depending to the impact of previous migrations. At each stage of the search, the more efficient operators receive dynamically the great majority of computational resources. In other words, the model is able to auto-adapt the attractive power of each islands.

This application is an extension of the dynamic island model approach. In previous work, we focus on the capacity for the model to dynamically regulate the interaction between individuals in an evolutionary context, with crossovers and the same configuration on each island, with promising results. Here, we dissociate the exploitation / exploration dilemma to focus on the capacity to allocate with relevance the resources to the most suitable operators. For that, we used an experimental protocol which makes possible to assess the real efficiency of the model (One-Max problem and comparison with theoretical values). The next step is to apply this operator selection strategy to difficult problems, and then to assemble this heterogeneous model within a more general evolutionary context.

References

1. 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.
2. Bilel Derbel and Sébastien Verel. DAMS: Distributed Adaptive Metaheuristic Selection. In *GECCO*, pages 1–18, United Kingdom, July 2011.
3. Álvaro Fialho, Luís Da Costa, Marc Schoenauer, and Michèle Sebag. Extreme value based adaptive operator selection. In *PPSN*, pages 175–184, 2008.
4. Steven Gustafson and Edmund K. Burke. The speciating island model: An alternative parallel evolutionary algorithm. *Journal of Parallel and Distributed Computing*, 66(8):1025–1036, 2006.
5. Frédéric Lardeux and Adrien Goëffon. A dynamic island-based genetic algorithms framework. In *SEAL*, pages 156–165, 2010.
6. Jorge Maturana, Frédéric Lardeux, and Frédéric Saubion. Autonomous operator management for evolutionary algorithms. *J. Heuristics*, 16(6):881–909, 2010.
7. Marek Rucinski, Dario Izzo, and Francesco Biscani. On the impact of the migration topology on the island model. *CoRR*, abs/1004.4541, 2010.
8. Zbigniew Skolicki and Kenneth A. De Jong. The influence of migration sizes and intervals on island models. In *GECCO*, pages 1295–1302, 2005.
9. 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.