Universität Hamburg
Department Informatik

Evaluation of Island-Based Artificial Evolution

Seminar Paper

Bio-Inspired Artificial Intelligence

Irina Barykina
6barykin@informatik.uni-hamburg.de

01.02.2017

Abstract

The migration policy plays a distinctive role in the island-based evolutionary algorithms. Thus, it is essential to know what impact the migration process has on performance of the algorithm to properly configure the island model and achieve the best possible results. In this paper, we experimentally studied a correlation between model behaviour and two fundamental parameters of the migration policy - the interval and the size, applying the island-based evolutionary algorithm to the Travelling Salesman Problem. We also investigated if the properly configured island model can outperform the single population model with the same total number of individuals.

Contents

1 Introduction

With increasing availability and cheapness of computational resources, such as clouds, clusters or computers with multiple CPUs, parallel versions of different algorithms are constantly getting more popular. This trend has certainly touched the evolutionary algorithms - bio-inspired metaheuristic optimisation algorithms, where different approaches to parallelization popped up.

One of the possible ways to parallelize the evolutionary algorithm is to split a population to subpopulations and run them in parallel on different computational units. The eventual result of the algorithm, in this case, is the best solution that occurs across all the subpopulations. A naive way to implement this kind of parallelization is to perform independent runs without any communication between subpopulations. According to D.Sudholt [8], despite its simplicity, the model with independent runs can significantly raise a success probability of the algorithm (so-called probability amplification).

Similarly, the island (or multi-demes) model divides a population into subpopulations (demes) and evolves them in parallel and independently most of the time. However, islands could periodically exchange solutions in the migration process. Besides the algorithm speedup due to parallelization, this approach enforces diversification of a population by spatial separation of islands. In the evolutionary algorithms, the diversity of a population plays a significant role: the poor diversity might lead to premature converging of the algorithm to some local minima. In the island model, demes maintain the better diversity, compared to the single population model, and, at the same time, the migration process prevents them from getting stuck in a local minima by infusion of new genes from other islands.

As soon as the migration process plays a distinctive role in the island-based evolution, it is important to understand how exactly it correlates with the behaviour of the algorithm. In this paper, we are aiming to clarify this correlation, by addressing following questions:

1. How does the migration policy affect performance of the island model?
2. What is the best configuration of the migration parameters?
3. Could the island model outperform the single population model?

We begin with a brief overview of design decisions that could have an impact on performance of the island model in Section 2. In Section 3, we introduce a computation problem used in experiments and describe an experimental setup. In Section 4, we present obtained experimental results and analyze them briefly from the perspective of the questions stated above.

2 Background

The migration process in the island model happens according to some migration topology, which could be represented as a graph with islands as nodes and directed edges, determining the directions of the migrant movements.

Besides the topology, there are several design factors, which affect the behaviour of the island model. They were listed by D.Sudholt in his comprehensive overview of the parallel evolutionary algorithms [8]. The *emigration policy* determines the quantity and quality of migrants that will be moved to another island. It also specifies if we should remove the migrants from an origin or keep copies of them. The *immigration policy* specifies how migrants from another island will be added to a local population: by replacement of the best, worst or random individuals. The *migration interval* is a speed at which information propagates through the island model. The bigger migration interval the more independent islands we have. The *number of migrants* is another parameter that affects how quickly solutions spread to other islands. The *homogeneousness* of the model determines if all demes evolve similarly.

Z. Skolicki and K. De Jong [7] experimentally investigated the influence of the migration interval and the number of migrants to behaviour of the island model. In their work, they showed that for frequent migrations, the migration interval has a stronger effect on performance of the island model than the migration size. They also demonstrated that performance degrades drastically with too small the migration intervals or the migration sizes approaching the size of subpopulations.

The migration process directly affects a speed of information propagation through the island model. Information spreading is usually represented by the growth curves, which show how a number of the best solutions increases over time. G.Luque and E. Alba [1] analysed dependencies between the migration policy and the growth curves behaviour. They revealed that the curves tend to make jumps during migrations and then form plateaus during independent evaluations. Thus, the large migration intervals lead to the vast plateaus. On the other hand, the migration size determines a probability that an island will be taken over by migrants, and on the growth curve, it is expressed as the incline of jump slopes. Finally, G.Luque and E.Alba [1] came to a conclusion that the migration sizes have a much smoother effect to propagation of information than the migration intervals, confirming the results of Z. Skolicki et al. [7].

M. Tomassini [9] compared the empirical efficiency of the island-based and conventional evolutionary algorithms. In his work, he used the conventional model with 2500 individuals and the system of 5 randomly connected islands with 500 individuals each and with the commonly-used migration policy (10% of the population, every 10 generations). His experimental results showed that the distributed model achieved the better quality of results with a higher probability. M. Tomassini came to a conclusion, that if subpopulations are big enough to provide the sufficient population diversity, the distributed model could outperform the single population model with the same total number of individuals.

3 Experimental setup

The Travelling Salesman Problem (TSP) is a problem of combinatorial optimisation that aims to find the shortest path through a set of cities so that every city is visited only once. TSP is a proven NP-hard problem and cannot be solved optimally in polynomial time [4].

The exact algorithms that are designed to find the optimal solution of TSP are very computationally expensive because they must implicitly check all possible solutions in order to find one with the shortest length. This circumstance explains why at the scale of the real world problems it is preferable to use the heuristic algorithms that could find some solution with a length within a few percents from the optimal one but much quicker. The evolutionary algorithm is one of the examples of such algorithms, and due to the simplicity of TSP formulation, it could be easily applied to this problem.

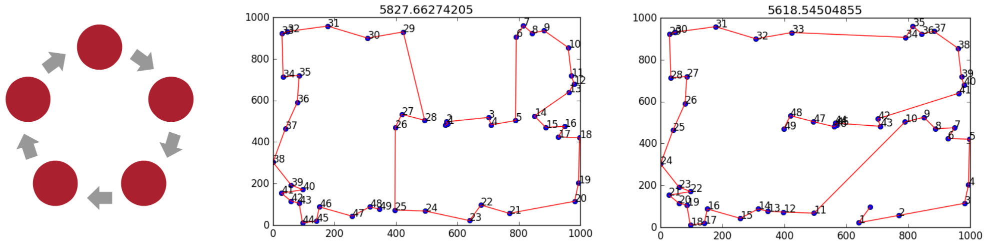
In this work we used a representation of TSP in terms of the artificial evolution, initially suggested by J.Potvin [5]. A chromosome or gene encodes a route, so it is a sequence of cities in the order they should be visited. Whereas TSP explicitly states, that a solution must contain each city only once, all operators used in the algorithm should support this gene invariant. The order crossover (OX) operator, initially introduced by I.M.Oliver et al. [3], was utilised in this work with a slight variation (Table 1). The operator generates two random cut points and copies the subsequence between these points from a first parent to an offspring. Then the remaining space in the offspring is filled by subsequences from a second parent keeping the order if possible and maintaining the invariant of the offspring gene. Also, the simple bio-inspired mutation operator that just swaps randomly selected cities in the chromosome was used.

Parent 1:	ABC	*DEF*	GHI
Parent 2:	GHA	IFB	CDE
Offspring:	GHA	DEF	IBC

Table 1: The example of the OX crossover operator. Symbol * represents cut points.

We used in experiments a homogeneous system of 5 islands with a unidirectional ring topology (Fig.1a). M. Tomassini [9] showed that the topology does not have a significant influence on performance of the algorithm, so the ring topology was chosen due to the simplicity of the implementation. The size of the population at every island was 80, the mutation rate was 0.015, and the tournament size was 5. The algorithm always transferred the best solution in a population to the next generation without applying the mutation and crossover operators (so-called elitism). The elitism guarantees a nondecreasing fitness function of the best solution on an island. This fact makes the graphs, which we present later, consistent and easy to compare. All experiments were performed on a single map with 50 cities, randomly placed within an 8-shaped region. Such a map helps to see clearly the quality of a solution on an image and in the same time gives islands a chance to discover dissimilar crafty ways to solve the problem. On Fig.1b, we presented the city map with two different solutions suggested by the algorithm.

In the context of TSP, the measure of the solution quality is a length of a route. However, in most of the literature, the fitness function is in a direct variation



(a)

(b)

Figure 1: a) The migration topology used in the experiments and b) the world map with two solutions proposed by the algorithm.

with a solution quality, so 1/length of a route measure was used to suit standard terminology.

H. Muhlenbein [2] proposed at least two different measures of performance in the context of the evolutionary algorithms. One of them is a quality of the best solution in a population that could be achieved by the algorithm. The second one is a speed at which the algorithm improves the quality of the best solution, or simply the quality of the best solution after a given number of generations. In this work, we take into account both of them.

We used the phenotypic entropy $\mathrm{H_p(P)}$, proposed by J.Rosca [6], as the diversity measure of a population:

$$H_p(P) = -\sum_{j=1}^{N} f_j log(f_j)$$

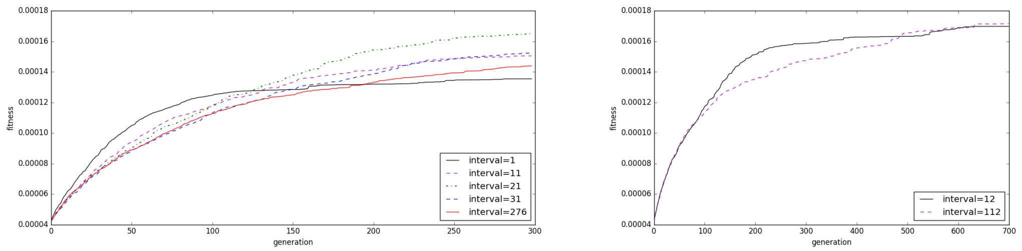
where $\mathrm{f_j}$ is the fraction $\mathrm{n_j/N}$ of individuals in the population P having fitness j, and N is the total number of fitness values in P. The phenotypic measure was not chosen arbitrarily. In the gene representation described above the general order of the cities is meaningless, so the simple comparison of the gene strings cannot express their similarity. The viable genotypic measure, in this case, should match all possible subsequences of the cities in the genes. Thus it would have a high computational cost.

Finally, we performed 10 trials for every combination of parameters to obtain the average values of the measures and avoid misleading results, due to stochasticity of the evolutionary process.

4 Experimental Results

4.1 Influence of migration interval

A priori, too rare migrations should lead to the lack of cooperation between islands and, in the case of a limited number of generations, degeneration of the island model to the model with independent runs. On the other hand, too frequent



(a)

(b)

Figure 2: a) The average fitness in runs with different intervals and the same migration size and b) the comparison of the average fitness in runs with significantly different intervals (12 and 112), the same migration size and the large total number of generations.

migrations do not allow islands to elaborate different solutions and consequently decrease the genetic diversity of the whole population. It should make the island model prone to the premature convergence, which we observe in the single population model. These assumptions were supported by the experimental results of Z.Skolicki et al. [7]. The goal of our first experiment was to investigate correlation between the migration interval and performance of the island model and confirm if expectations described above are true in the context of TSP.

For that, we explored how the island model behaves with the migration intervals in the ranges [1; 51] and [201; 276], the maximum number of generations equal to 300 and the number of migrants always equal to 2.

The experimental results showed that the model with the shortest migration interval (1) converged faster than other configurations, but on the other hand, the maximum overall fitness achieved by it was significantly smaller (Fig.2a). With the migration intervals in the range [10; 20], the model achieved a good overall fitness with an adequate speed. With larger intervals, the model was unable to reach comparably good fitness within the given number of generations, but the curve shape let us assume that the algorithm did not prematurely converge.

To confirm the assumption that the large intervals affect only the speed of the model, but do not lead to the poor quality of achieved results, we did auxiliary runs with the significantly different intervals (12 and 112) and the larger maximum number of generations (1500). As expected, the model with the bigger migration interval (112) was able to reach the average maximum fitness of the model with the smaller interval (12), but much later (Fig.2b).

Overall, the experimental data confirmed that too short the migration intervals make a strong negative influence on performance of the island model. On the other hand, the long migration intervals negatively affect the speed of the model. The best combination of the speed and achieved fitness was observed with the intervals within the range [10; 20].

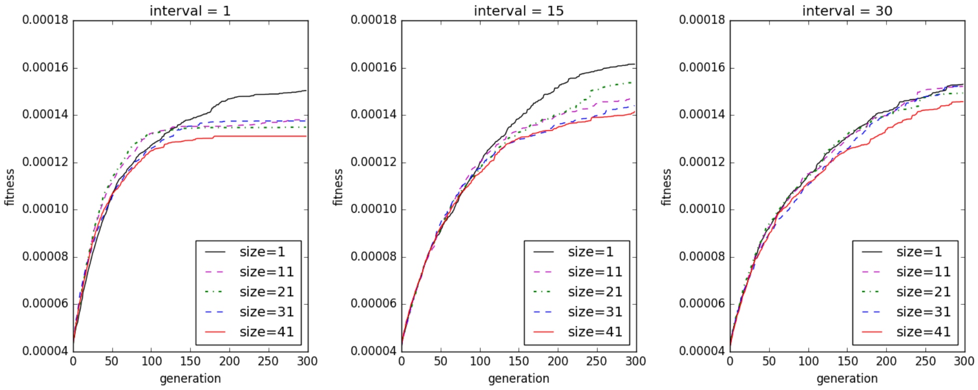


Figure 3: The average fitness in runs with the varying migration sizes and intervals.

4.2 Influence of migration size

As mentioned in Section 2, Z.Skolicki et al. [7] and G.Luque et al. [1] came to the interesting conclusion that the migration size has a minor impact on performance of the algorithm, while the migration interval seems to be a dominating factor.

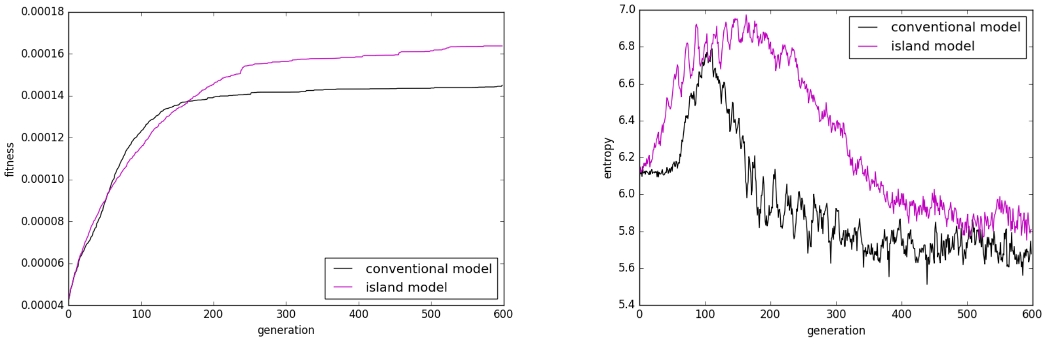
The goal of our second experiment was to confirm the conclusion of these research groups. For that, we explored how the island model behaves with the migration sizes from the set of values {1, 11, 21, 31, 41} in combination with the different migrations intervals {1, 15, 30, 50, 112}.

The experimental results showed thatthe models with the large migration intervals (>50) were almost indifferent to the change of the migration size (Fig.3). Whilst, in combination with the moderate and small migration intervals, the small migration sizes led to the better quality of results. However, the impact of the intervals was more substantial, in agreement with the results of [7] and [1].

4.3 Comparison with single population model

Due to the spatial separation of demes, the island model should be able to maintain a greater diversity of a population and consequently achieve a better quality of solutions with a higher probability. M.Tomassini [9] experimentally showed that this assumption is valid if subpopulations are reasonably large. The goal of our third experiment was to confirm the conclusion of M.Tomassini in the context of TSP and see if the island model with the best migration parameters can achieve better performance compared to the single population model.

In the experiment, we used the conventional model with the total number of individuals equal to 400. The island model had 5 islands with 80 individuals each, exchanging 1 migrant every 15 evaluations. The models had the same crossover and mutation operators. In both cases, we ran 10 trials with the maximum number of generations equals to 600. Besides the fitness, we also measured the phenotypic entropy, that was introduced in Section 3, to show the diversity of populations in these runs.



(a)

(b)

Figure 4: a) The average fitness and b) the phenotypic entropy in runs of the conventional (black) and island model (magenta).

The obtained experimental results showed that the island model achieved the better average fitness than the single population model (Fig.4a). Furthermore, the average phenotypic entropy in the island model remained higher during the whole evolutionary process (Fig.4b). The conventional model had a steep decline of entropy after 100 generations, while the island model was able to maintain the diversity much longer. These observations are in agreement with the experimental results of M.Tomassini.

5 Conclusion

The island model brings some spatial structure to the process of the artificial evolution. Consequently, it is expected to maintain the better diversity of a population and show the better performance and quality of achieved results than the single population model.

In this work, our goal was to confirm empirically that these expectations are valid in the context of the Travelling Salesman Problem. We also aimed to clarify the correlation between the algorithm performance and the migration policy and identify the best configuration of the migration process.

We showed that the island model easily degrades to the single population model when migration process is configured in a way that subpopulations do not have time to evolve enough. Thus, for very frequent migrations, we observed that the behaviour of the island model was very similar to the behaviour of the single population model. They both converged fast and showed poor quality of results. In the same time, we clearly saw that the large intervals do not affect the quality, but only the speed of the algorithm. This indicates that overall dynamics of the models with the average and large intervals was the same. Most likely, with the large intervals between migrations, subpopulations converged to some local minima and remained idle rest of the time until next migration happened.

Based on these results, we could presume that following condition should be satisfied to achieve good performance of the island model. The diversity of populations on islands has to diminish enough before subsequent migration so that the good solutions start to take over islands. In this case, the incorporation of a new gene might be a stimulus to an extensive exploration again. If this "partial convergence" of islands does not happen between migrations, the island model turns to the model with a single population and starts to converge to the single minima. Thus, in the future work, we are planning to confirm this hypothesis. If we figure out the correlation between the diversity decrease on islands and performance of the model as a whole, we will be able to formalise the choice of the migration interval or even make it adaptive during execution of the algorithm.

References

[1] Gabriel Luque and Enrique Alba. *Parallel genetic algorithms: theory and real world applications*, volume 367. Springer, 2011.

[2] Heinz Mühlenbein. Evolution in time and space - the parallel genetic algorithm. In *FOUNDATIONS OF GENETIC ALGORITHMS*, pages 316–337. Morgan Kaufmann, 1991.

[3] I. M. Oliver, D. J. Smith, and J. R. C. Holland. A study of permutation crossover operators on the traveling salesman problem. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 224–230, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[4] Christos H. Papadimitriou. The euclidean travelling salesman problem is np-complete. *Theoretical Computer Science*, 4(3):237 – 244, 1977.

[5] Jean-Yves Potvin. Genetic algorithms for the traveling salesman problem. *Annals of Operations Research*, 63(3):337–370, 1996.

[6] Justinian P Rosca. Entropy-driven adaptive representation. In *Proceedings of the workshop on genetic programming: From theory to real-world applications*, volume 9, pages 23–32. Citeseer, 1995.

[7] Zbigniew Skolicki. The influence of migration sizes and intervals on island models. In *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2005*, pages 1295–1302. ACM Press, 2005.

[8] Dirk Sudholt. *Parallel Evolutionary Algorithms*, pages 929–959. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[9] Marco Tomassini. *Island Models: Empirical Properties*, pages 19–51. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.