

# Improving Evolutionary Algorithms with Multi-representation Island Models

Zbigniew Skolicki and Kenneth De Jong

George Mason University, 4400 University Drive, Fairfax, VA 22031, USA,  
zskolick@gmu.edu, kdejong@gmu.edu

**Abstract.** We present an island model that uses different representations in each island. The model transforms individuals from one representation to another during migrations. We show that such a model helps the evolutionary algorithm to escape from local optima and to solve problems that are difficult for single representation EAs. We illustrate this approach with a two population island model in which one island uses a standard binary encoding and the other island uses a standard reflective Gray code. We compare the performance of this multi-representation island model with single population EAs using only binary or Gray codes. We show that, on a variety of difficult multi-modal test functions, the multi-representation island model does no worse than a standard EA on all of the functions, and produces significant improvements on a subset of them.

## 1 Introduction

The continuing successes of EA applications have resulted in exploring their use on new and more challenging problems. As the complexity of these new applications increases, there is a corresponding need for more sophisticated evolutionary algorithms. One direction that is being explored is the use of island models to increase the problem-solving capabilities of EAs. The research described in this paper is in this area and presents some preliminary results on the advantages of using island models in which different islands use different representations. Specifically we show that by using a two-island model, each with a different representation, and allowing for migrations between them, we are able to effectively solve problems that are difficult to solve with single representation EAs.

In section 2 we present a short background on island models and on using several different representations. In section 3 we describe our multi-representation island model with transforming migrations and verify our suppositions with some initial experiments. In section 4 we present the results of a more systematic set of experiments, and we finish with conclusions in section 5.

## 2 Island models and multiple representations

Separating individuals spatially from each other results in a qualitative change in the behavior of evolutionary algorithms. The two most well known approaches

to separating individuals is to either group them into populations (*island models*) or put them on a grid and restrict their interaction to some neighborhood (*neighborhood models*).

The first most obvious outcome of separating individuals is the slowing down of the information flow between individuals [1]. The results of this may be two-fold. On one hand it may be undesirable - it can prevent successful mixing, which would lead to constructing a novel solution. On the other hand, the results may be positive - it may stop individual solutions from dominating the population and allow different building blocks to be discovered in separated regions.

In island models the exchange of information is carried out by migrating individuals between populations. Migrations are characterized by several parameters such as size, frequency and migration policy [2]. Although initially people used island models simply for the purpose of distributing computations on many machines, often island models produce better qualitative results [3].

In island models different algorithms and/or representations can be used in the same time inside populations. Seemingly useless redundancy of different representations may lead to different evolvability characteristics, and ultimately to a more effective search - as suggested by the neutrality theory [4, 5]. Switching representation changes the exploration distribution around a given solution, and opens a possibility for genetic operators to discover new regions of the search space.

One of the problems of search algorithms, including evolutionary algorithms, is getting locked in a local optimum. A local optimum is defined as a solution, whose all direct neighbors are worse in terms of fitness. However, the neighborhood relation depends on representation. Therefore a solution which is a local optimum in one representation may not necessarily be so in another. Also, a solution located in the basin of attraction of one local optimum, may belong to the basin of some other optimum in another representation.

A well known implementation of island model with different representations is the injection island model [6]. In this model there are several islands with different precisions of representation and different fitnesses. Some islands have simpler representations and faster fitness evaluation. Their task is to evolve the first, coarse approximations of the final solution. The best solutions migrate to the islands with more detailed representations and more expensive evaluations. There might be several levels of islands and the final solution is taken from the most complex one.

Another interesting approach to dynamically switching representations was proposed in [7, 8]. This algorithm uses single population and changes the representation for all the individuals, when it gets trapped in a minimum. The representations used are Gray encodings and the method switches from one Gray code to another by "rotating" the encoding, which is called *shifting*. By doing so the authors are often able to make two local minima become neighbors and thus "collapse" the worse of them. The usefulness of the method was experimentally confirmed. More information on shifting and properties of both binary and Gray encodings is included in [9] and [10].

### 3 Transforming migrations

The model that we propose in this paper is an island model in which we transform individuals from one representation to another during migrations. Such a model allows for changing representations in either direction, as opposed to injection islands, where the flow of individuals is one-directional, from coarser to more precise representations. In other approaches (like shifting) the representation changes for all individuals in the same time. Again, our model is more flexible, because it allows for having different representations simultaneously, and the control of representation is more at the individual level than at the whole population level. The general model we just described doesn't make any assumptions about the sort of representations used, or number of populations.

For every problem there exist better and worse representations. Some of them may create deceptive regions, whereas others will result in an easy landscape. In many cases it is impossible to decide *a priori* what representation to choose for a given problem. It may even happen that all the representations we are trying produce a deceptive landscape. However, even then, a local optimum in one representation need not be a local optimum in the other representation. Therefore by switching representations in some cases we should be able to escape from such situations. Optimally, if a function is easy for some representation in places where it is difficult for others, the island model using multiple representations should behave better on this function than a standard single population EA using any of those representations.

In this paper we try to support these intuitions by analyzing a particular example of the described model. We will use two populations. Migrations will be occurring at the same time from the first population to the second and vice versa and their parameters will be the same throughout any single experiment. We will use two relatively simple encodings: a standard binary encoding and a standard reflected Gray encoding. For those representations we will show a function  $F$  which has the characteristics of the problem described above, and we will show that the multi-representation island model indeed behaves better than standard EAs.

#### 3.1 Exemplary function

We will construct a function  $F$  representing a minimization problem. Let us denote the solution search space by  $\Omega$  and by  $G_n$  denote a graph, in which nodes are binary strings of length  $n$  and the edges represent the neighborhood relation (for a more formal definition and discussion see [9]). An encoding  $f$  is a function  $f : \Omega \rightarrow G_n$ . We will denote the binary and Gray encodings by  $binary : \Omega \rightarrow G_n$  and  $gray : \Omega \rightarrow G_n$ .

To construct our function we first define two functions from  $G_n$  to  $\mathbb{N}$ .

Let us define a function  $easy : G_n \rightarrow \mathbb{N}$  by

$$easy(a) = hamming(a, 0^n)/4$$

where the function *hamming* returns the number of bits by which its two arguments differ and  $0^n$  denotes a string of length  $n$  consisting of 0's. It is easy to see that it is a simple unimodal function with one global optimum equal to  $0^n$ .

Let us also define a function *deceptive* :  $G_n \rightarrow \mathbb{N}$  by

$$deceptive(a) = \begin{cases} 3 * hamming(a, 0^n) & \text{if } hamming(a, 0^n) < \frac{1}{4}n \\ n - hamming(a, 0^n) & \text{otherwise} \end{cases}$$

The function has two optima. One is global ( $0^n$ ), and the other one is local ( $1^n$ ) and has a bigger basin of attraction. Therefore the function is deceptive - it is easy for EA to get stuck in the local minimum.

We further define a function  $F_1 : \Omega \rightarrow \mathbb{N}$  by

$$F_1(x) = deceptive(gray(x)) + easy(binary(x))$$

The second component (*easy*) of the function is usually smaller and thus has a smaller impact when using the Gray encoding. However, it influences the trajectory of EA when using binary encoding.  $F_1$  is difficult for Gray encoding and relatively easy for binary encoding.

Similarly, we define a function  $F_2 : \Omega \rightarrow \mathbb{N}$  by

$$F_2(x) = deceptive(binary(x)) + easy(gray(x))$$

Analogously,  $F_2$  is difficult for binary and relatively easy for Gray encoding.

Finally we can construct the function  $F$  to be

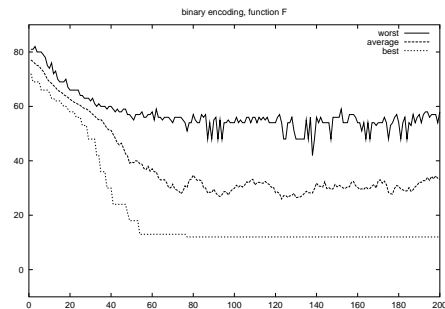
$$F(x) = \min(F_1(x), F_2(x))$$

The function inherits both local minima from  $F_1$  and  $F_2$  and has one global optimum at  $0^n$  with value 0. It is a difficult function when using only binary or only Gray encoding, but turns out to be much easier when switching between representation is allowed.

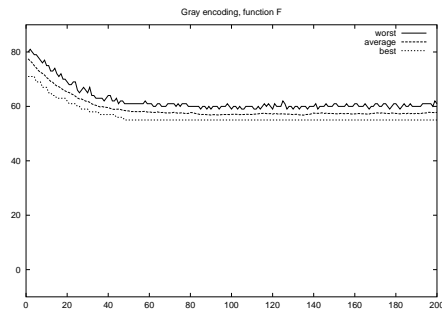
### 3.2 Initial observations

To verify our assumptions about the behavior of the model and the properties of the function  $F$ , we start with analyzing single runs of EA. Not surprisingly, single populations using just one encoding usually fail to find the global optimum. A typical run for the binary encoding is shown in Figure 1. and a typical run for Gray encoding is shown in Figure 2. We plot the best, average and worst fitness of the population. In both cases the EA was trapped in local optima, which are neighborhoods of respectively  $binary^{-1}(1^n)$  and  $gray^{-1}(1^n)$ .

The situation changes when we allow for the island model with both representations in the same time. On Figure 3. we show an analog run with two representations. A careful inspection of the figure shows that the representations "help each other", not allowing each other to get stuck in local minima. One may notice that it is the binary representation that is leading in the beginning but it



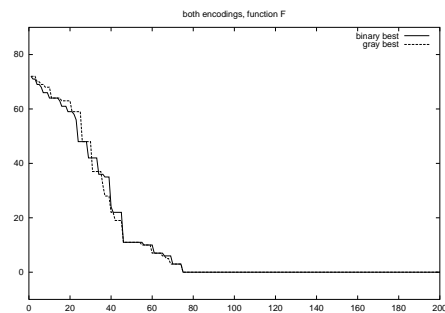
**Fig. 1.** A typical single run with binary encoding on function  $F$ . EA got trapped in a local minimum



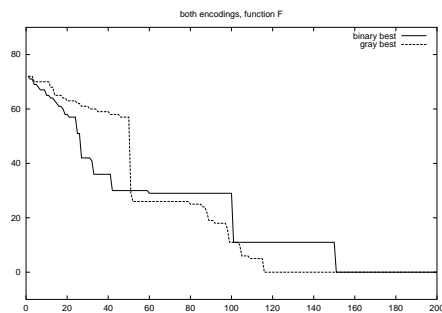
**Fig. 2.** A typical single run with Gray encoding on function  $F$ . EA got trapped in a local minimum

is the Gray representation that plays an important role towards the end. Vertical changes in best fitness correspond to migrations from the other population. The influence of such migrations is very distinct in generations 20–35 for the Gray encoding and in generations 55–65 for the binary encoding.

To make the influence of migrations even more visible, we have run a set of experiments in which migrations occur very rarely (every 50 generations). Again, a typical run is shown in Figure 4. One can clearly see the initial leading role of the population with binary representation. By generation 50 the Gray encoding population is already stuck at its local minimum. A migration helps it escape from this minimum. Later the roles of the populations get reversed and it is the Gray encoding population that helps the binary encoding population escape from its local minimum, by means of migrations at generations 100 and 150.



**Fig. 3.** A typical single run with both encodings on function  $F$ . EA found the global optimum



**Fig. 4.** A typical single run with both encodings and rare migrations, on function  $F$ . Mutual influence of populations by means of migrations is very visible in generations 50, 100 and 150

## 4 Experiments

We have conducted several sets of experiments, the statistical results of which we report in this section. Firstly, we changed EA parameters to study the robustness of the model. Then we explored the influence of migration level. Finally, we ran experiments on different functions known in the literature to confirm the applicability of the model to a wider set of problems.

Unless explicitly stated otherwise, we used the following parameters. For each experiment we made 50 runs for a single population of size 100, using the binary encoding, a single population of size 100, using the Gray encoding and an island model of two populations each of size 50, one using the binary and the other using the Gray encoding. The migrations occurred every 5 generations by copying 10% of best individuals from each population, transforming them to the other representation and immigrating by replacing random individuals. The parameters used are summarized in Table 1.

**Table 1.** Parameters used

| Parameter                      | Value                      |
|--------------------------------|----------------------------|
| mutation rate                  | $1/n$                      |
| recombination type             | two-point                  |
| recombination rate             | 1.0                        |
| selection strategy             | ranked                     |
| population size                | 100 or 50+50               |
| migration policy               | copy best - replace random |
| migration size                 | 10%                        |
| generations between migrations | 5                          |
| total number of generations    | 500                        |
| number of runs                 | 50                         |

### 4.1 Changing EA parameters

We have tested the model using two crossover types (two-point and uniform) and two selection strategies (ranked and binary tournament). In every test the island model performed better than single population ones. It was the only one to find the global optimum, and for experiment 2 it found the global optimum every time. In other experiments both island model EA and single population EAs were finding different minima in different runs. For example, in experiment 3 the binary encoding EA was finding optima with values 55 and 11, whereas island model EA was finding optima of 55 and 0. Therefore the averages are not significantly different. The results of experiments are presented in Table 2. For

each model – binary (B), Gray (G) and island model (IM) – we give the average best fitness at the end of run (Av), confidence interval (Err) for  $\alpha = 0.05$  and number of times the model found the global optimum (#Opt)

**Table 2.** Changing EA parameters.

| No | Description         | B Av  | B Err      | #Opt | G Av  | G Err      | #Opt | IM Av | IM Err     | #Opt |
|----|---------------------|-------|------------|------|-------|------------|------|-------|------------|------|
| 1  | two-point, ranked   | 21.08 | $\pm 4.99$ | 0    | 55.00 | $\pm 0.00$ | 0    | 22.00 | $\pm 7.54$ | 30   |
| 2  | uniform, ranked     | 12.00 | $\pm 0.00$ | 0    | 55.00 | $\pm 0.00$ | 0    | 0.00  | $\pm 0.00$ | 50   |
| 3  | two-point, b.tourn. | 25.38 | $\pm 5.76$ | 0    | 55.00 | $\pm 0.00$ | 0    | 27.50 | $\pm 7.70$ | 25   |
| 4  | uniform, b.tourn.   | 12.00 | $\pm 0.00$ | 0    | 55.00 | $\pm 0.00$ | 0    | 3.48  | $\pm 2.16$ | 19   |

## 4.2 Changing migrations parameters

Similar analysis was performed for three different levels of migration. Again, because the EAs converge to several very different optima, the averages and confidence intervals don't tell us much. However the comparison of numbers of times the global optimum was found by each model is much more informative. The results are given in Table 3. in a similar format to the previous one. It seems that the migration of 1%, even occurring every generation, is too weak to have any significant impact on the behavior of the algorithm (although it did find the global optimum once). On the other hand migrations of level 10% seem to affect the behavior significantly, even if they occur rarely (every 20 generations).

**Table 3.** Changing migration parameters

| No | Description   | Average | Confidence | #Opt |
|----|---------------|---------|------------|------|
| 1  | 10 %, 5 gen.  | 22.0    | $\pm 7.54$ | 30   |
| 5  | 1 %, 1 gen.   | 26.8    | $\pm 5.49$ | 1    |
| 6  | 10 %, 20 gen. | 18.7    | $\pm 7.29$ | 30   |

## 4.3 Changing fitness functions

The last experiments were aimed at verifying the applicability of the model for functions other than the one presented so far. Therefore a set of 4 functions which are well-known in the literature, was taken. The functions are multi-modal test

functions, known as Rosenbrock, Schwefel, Rastrigin and Griewangk (see Table 4.). A 10-dimensional versions of each of them were used and each parameter was represented with 12 bits. To maintain the mutation rate of  $\frac{1}{n}$ , it was set to 0.0083.

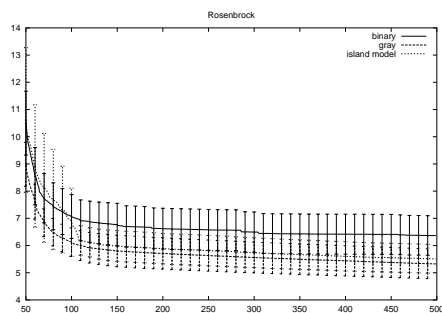
**Table 4.** Test functions

| Name       | Formula   |
|------------|---|
| Rosenbrock | $f(x_1, \dots, x_n) = \sum_{i=2}^n (100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2)$                        |
| Schwefel   | $f(x_1, \dots, x_n) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$   |
| Rastrigin  | $f(x_1, \dots, x_n) = 10n + \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i))$                                  |
| Griewangk  | $f(x_1, \dots, x_n) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})$ |

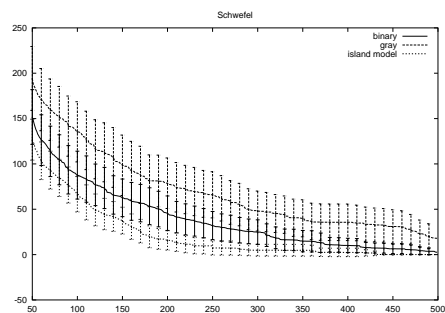
The figures and the table below shows the results. We must note that we plot the figures with error bars based on confidence interval calculated for  $\alpha = 0.05$  and assumption of Gaussian distribution. The latter is not necessarily true because it strongly depends on what minima exist in the landscape. However, the error bars still give us some insight into how reliable the data is.

For the Rosenbrock function the proposed model doesn't differ significantly from the single-population single-representation models. Its performance seems to be slightly better than that of binary representation and comparable with Gray representation. However the differences are within a statistical error.

For Schwefel function the performance of the multi-representation model looks better. It is significantly better than both that of binary as well as of Gray representation for short term runs (around generation 200) and at the end, when it converges to very low minimum of value around 0.015.



**Fig. 5.** Average results from 50 runs for Rosenbrock function, shown from generation 50

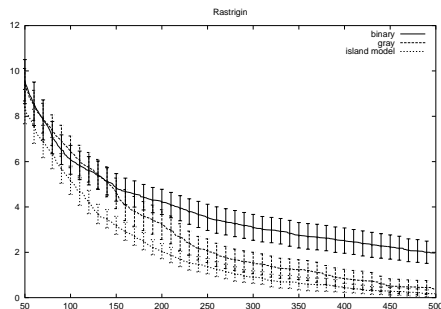


**Fig. 6.** Average results from 50 runs for Schwefel function, shown from generation 50

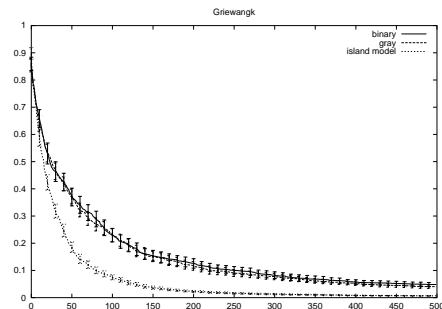


The model behaves well on Rastrigin function, again with most visible advantage in short runs. After 500 generations the Gray encoding model becomes statistically indistinguishable from the island model while the binary encoding model occurs significantly worse.

Finally, for Griewangk function, we observe a huge advantage of using the multi-representation island model. It not only converges faster but also finds a better optimum.



**Fig. 7.** Average results from 50 runs for Rastrigin function, shown from generation 50



**Fig. 8.** Average results from 50 runs for Griewangk function, all generations shown

We are currently analyzing these landscapes in more detail in order to understand more precisely what causes the island model to behave better on certain functions than on others, and hope to present those results in a later paper.

## 5 Conclusions

The results presented in this paper suggest that island models with different representations and transforming migrations may behave better than standard EAs for some problems without making things worse on others. Therefore they are a promising approach for difficult, multi-modal problems. What is interesting is that they often behaved better than EAs using either of the two representations. This suggests that representations with dynamic neighborhood (i.e. those in which the neighborhood structure can change, which happens when we switch from one representation to another) can be in general better than any static-neighborhood representation. Of course, the No Free Lunch theorem states that they cannot be better for any function - but they may be better for commonly used test and real-life functions.

The observations from single runs convinces us that the multi-representation island models produce better results because the representations "help each other" by means of migrations. A further investigation into this process and possibly making it dependent on some population-based feedback would definitely be an interesting area for future work.

## References

1. Sarma, J.: An Analysis of Decentralized and Spatially Distributed Genetic Algorithms. PhD thesis, George Mason University, Fairfax, VA (1998)
2. Cantú-Paz, E.: Migration policies, selection pressure, and parallel evolutionary algorithms. *Journal of Heuristics* **7** (2001) 311 – 334
3. Whitley, D., Rana, S., Heckendorn, R.B.: The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology* **7** (1999) 33–47
4. Toussaint, M.: Self-adaptive exploration in evolutionary search. Technical Report IRINI-2001-05, Institute for Neuroinformatics, Ruhr-University Bochum (2001)
5. Toussaint, M., Igel, C.: Neutrality: A necessity for self-adaptation. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002). (2002) 1354–1359
6. Eby, D., Averill, R., Goodman, E., Punch, W.: The optimization of flywheels using an injection island genetic algorithm. In Bentley, P., ed.: *Evolutionary Design by Computers*. Morgan Kaufmann, San Francisco (1999) 167–190
7. Rana, S., Whitley, L.: Bit representation with a twist. In: Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA-97), Morgan Kaufmann (1997)
8. Barbulescu, L., Watson, J.P., Whitley, D.: Dynamic representations and escaping local optima: Improving genetic algorithms and local search. In: AAAI/IAAI. (2000) 879–884
9. Rowe, J., Whitley, D., Barbulescu, L., Watson, J.P.: Properties of gray and binary representations. *Evolutionary Computation* **12** (2004) 47–76
10. Whitley, D.L., Rana, S., Heckendorn, R.B.: Representation issues in neighborhood search and evolutionary algorithms. In Quagliarelli, D., Periaux, J., Poloni, C., Winter, G., eds.: *Genetic Algorithms in Engineering and Computer Science*. John Wiley & Sons Ltd., Chichester (1997) 39–57