Comparison between Multi-Population Migration and Random Immigrants Algorithms

Jessica Mégane Taveira da Cunha
jessicac@student.dei.uc.pt

Ricardo José Monteiro Paiva
rjpaiva@student.dei.uc.pt

May, 2020

Abstract

In this paper we show the comparison between two algorithms in a dynamic environment. In the first one, two different populations are evolved, and from time to time, some individuals change populations. In the second algorithm, the worst individuals are replaced by new random individuals. To evaluate the performance both algorithms are tested in a dynamic environment using the Knapsack algorithm, where each 10 generations, the problem conditions are changed.

Keywords: *Multiple populations; Multi-Population Migration; Random immigrants; Evolutionary algorithm; Dynamic environment.*

1 Introduction

Most real-world applications operate in non-dynamic environments. A dynamic environment corresponds to an environment that sometimes changes its characteristics and where populations must adapt to those same changes.

An evolutionary algorithm is inspired by biological/natural evolution, where a population exists and each individual in that population may go through a series of interventions (crossover, mutation, selection...) that contribute to the diversity of the population. These individuals are candidate solutions to the problem to be optimized and they are evaluated by a fitness function that determines the quality of the solutions. This type of algorithms can be used in dynamic problems since the solutions are able to adapt to the changes imposed on the environment, as shown in Brindle (1980).

Several approaches were developed, but in this project we'll focus on Random Immigrants and Multi-populations. As shown in Grefenstette (1992), standard genetic algorithms eventually converge to an optimum point, and lose the diversity to explore the search space when a perturbation in the environment occurs. To avoid premature convergence, Grefenstette proposed a modification of the standard genetic algorithm. This proposal was later called Random Immigrants in Cobb (1993), in which it chooses random individuals to be replaced by new ones based on a percentage of replacement. Improvements of this algorithms have been developed, for example Tinós and Yang (2007)

Many researches (Trojanowski and Michalewicz (1999), Cheng and Yang (2010), Branke, Kaußler, Schmidt, and Schmeck (2000) and others) developed different approaches that use multiple populations, but in this project we'll try a simpler approach, in which some individuals are exchanged between two populations.

2 Algorithm overview

2.1 Multi-Population Migration

In this approach two populations co-evolve, being submitted to the same selection methods (tournament and elitism) and genetic operators (mutation and crossover). Every X generations, several individuals are selected from each population and are exchanged. In this algorithm we're going to test the influence of the frequency of the migration, the number of individuals exchanged and the individuals' selection method.

2.2 Random Immigrants

In this strategy, each X generations, the individuals with the lowest fitness of the population are replaced by new randomly generated ones. In this case, we test the influence of the immigration frequency and the number of random immigrants.

3 Experimental Design

3.1 0/1 Knapsack

The benchmark problem used was the 0/1 Dynamic Knapsack. For this problem, the aim is to choose from a list of *n* items x_i, each with a certain weight w_i and value v_i, the items that maximize the total value without surpassing the capacity C of the knapsack. When an environmental perturbation occurs, the values and weights of the items are changed, also changing the knapsack capacity. Formally:

$$max(\sum_{i=1}^{250} v_i * x_i)$$

subject to:

$$\sum_{i=1}^{250} w_i * x_i \leq C$$

with:

$$x_i \in \{0, 1\}$$

The data was generated using 250 items with weak correlation, using the following formulas:

$$w_i = uniform([1, 10])$$

$$v_i = w_i + uniform([-5, 5])$$

$$C = 0.5 * \sum_{i=1}^{250} w_i$$

The fitness of the individuals was calculated based on their phenotype, being n the number of chosen items by the individual, using a linear function:

$$f(x) = \begin{cases} \sum_{i=1}^{n} v_i & \text{, if } \sum_{i=1}^{n} w_i \leq C \\ \sum_{i=1}^{n} v_i - (max(\sum_{i=1}^{n} v_i/w_i) * (\sum_{i=1}^{n} w_i - C)) & \text{, otherwise} \end{cases}$$

3.2 Experimental Setup

The individual's genotype is generated randomly and is composed by a segment of binary integers. The size of the genotype is the same as the number of items in the problem, so each number in the genotype is assigned to an object. When the genotype has a value of 1, means that the item associated with that position is chosen. Thus, the individual's phenotype is the mapping result of the genotype, that is the list of chosen objects.

The population was submitted to selection methods such as tournament and elitism, and as genetic operators, mutation and crossover. The mutation used was random resetting, which for each bit of the genotype, generates a new number (0 or 1). One-point crossover occurs between two individuals, chosen by tournament, and select randomly the position to perform the recombination, generating two new individuals.

In order to validate the results obtained, two datasets were generated with weak correlation, with the same parameters. During the experimentation, an environmental change (perturbation) occurs every 10 generations. Each experiment was repeated 30 times for each dataset.

Parameter	Multi-populations	Random Immigrants
Runs	30	
Population Size	500	
Generations	100	
Genotype	250	
Elitism	10	
Selection Method	Tournament with size 3	
Crossover Rate	0.90	
Mutation Rate	0.10	
Crossover	One point crossover	
Mutation	Random resetting	
Frequency exchanges	each [1,2,5,10] generations	each [1,2,5,10] generations
Individuals exchanged	[5%, 25%]	[5%, 25%, 75%]
Method	[worst, random, best]	[worst]

Table 1: Parameters used for each algorithm

4 Results

In this section we will analyse the results obtained for both algorithms. First we have performed several statistical tests in order to find differences in performance.
We started by doing the Kolmogorov-Smirnov Test, for each experiment, analysing the results with a confidence interval of 95%, based on the following hypothesis:

H_0: The data follow a normal distribution
H_1: The data doesn't follow a normal distribution

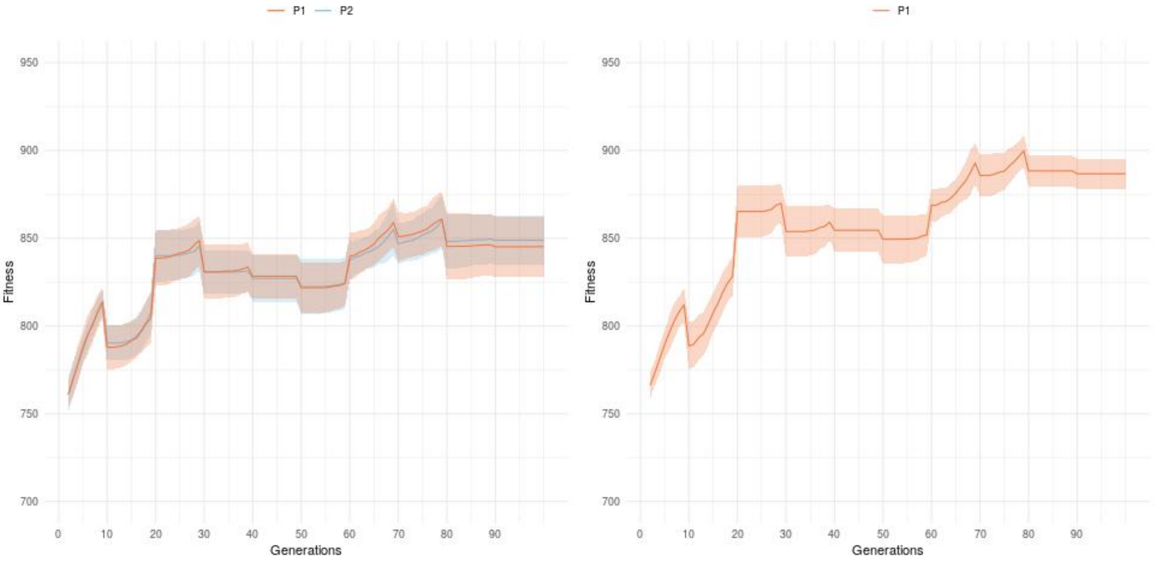
For all the samples, we reject the null hypothesis, so we assume the samples are non-parametric. We applied the Kruskal-Wallis between the samples of each algorithm, since the samples are non-parametric, to see if there are statistical differences in the means.

H_0: The samples are statistically similar
H_1: Fails to reject the null hypothesis

The tests rejected the null hypothesis in both cases, so we did the post-hoc analysis using the Wilcoxon-Mann-Whitney test, with Bonferroni correction of the significance level, to perform pairwise comparisons.
To see the magnitude of the difference between samples, we measured the effect size(r). We used this notation to compare each pair of algorithms: a "$\sim$" means that there is no statistical difference between the algorithms ($r \leq 0.1$ and the Wilcoxon-Mann-Whitney test fails to reject the Null Hypothesis), a "+","-" means that the algorithm on the rows is better/worse than the one in the columns, having a small effect size ($0.1 < r \leq 0.3$), a "+ +","- -" indicates that the algorithm on the rows is statistically better than the algorithm on the columns, with a medium effect size ($0.3 < r \leq 0.5$) and a "+ + +","- - -" means a large effect size ($r > 0.5$).
The results for both datasets are reported bellow.

				Random Immigrants											
				5%				25%				75%			
				1	2	5	10	1	2	5	10	1	2	5	10
Multi-populations	5%	1	worst	-	-	-	∼	- -	- -	- -	- -	- -	- -	- -	- -
			best	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
			random	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
		2	worst	-	-	-	∼	- -	- -	-	-	- -	- -	- -	- -
			best	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
			random	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
		5	worst	-	-	-	∼	- -	- -	- -	- -	- -	- -	- -	- -
			best	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
			random	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
		10	worst	-	-	-	-	- - -	- -	- -	- -	- - -	- -	- -	- - -
			best	-	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
			random	-	-	-	∼	- -	- -	-	-	- -	- -	- -	- -
	25%	1	worst	-	-	-	∼	- -	- -	-	-	- -	- -	- -	- -
			best	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
			random	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
		2	worst	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
			best	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
			random	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -
		5	worst	-	-	-	∼	- -	- -	-	-	- -	- -	- -	- -
			best	-	∼	∼	∼	- -	- -	-	-	- -	- -	- -	- -
			random	∼	∼	∼	∼	- -	-	-	-	- -	- -	-	- -
		10	worst	-	-	-	∼	- -	- -	-	-	- -	- -	- -	- -
			best	∼	∼	∼	∼	- -	- -	-	-	- -	- -	- -	- -
			random	∼	∼	∼	∼	- -	-	-	-	- -	- -	- -	- -

Table 2: Statistical tests - Dataset 1

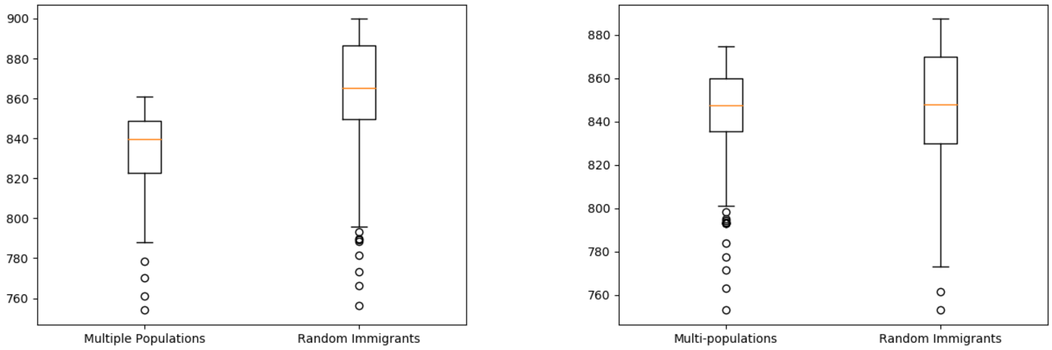


(a) Multi-populations. Individuals exchanged: 5%; Frequency migration: 10 gen; Method: worst individuals

(b) Random Immigrants. Individuals replaced: 75%; Frequency immigration: all generations

Figure 1: Example where algorithm in b) is statistically better than a) - Dataset 1

The Figure 1 shows graphically a case where the Random Immigrants algorithm is statistically better than the Multi-populations. We can see that when a perturbation occurs, in (b) the fitness is better, which means that the population has enough diversity to be more prepared to the perturbation.



(a) Boxplot of the Figure 1 - large effect size

(b) Boxplot of statistical similarity

Figure 2: Comparison between algorithms with different effect sizes - Dataset 1

In the Figure 2 we can see the two box plot comparing the algorithms when they have a large effect size (a) and when are statistically similar (b). The samples used for the first boxplot were the same as described in Figure 1 and for the second boxplot we used for the Multi-population, 5% of random individuals to exchange every generation, and in Random Immigrants, 5% of the worst individuals to be replaced, every generation.

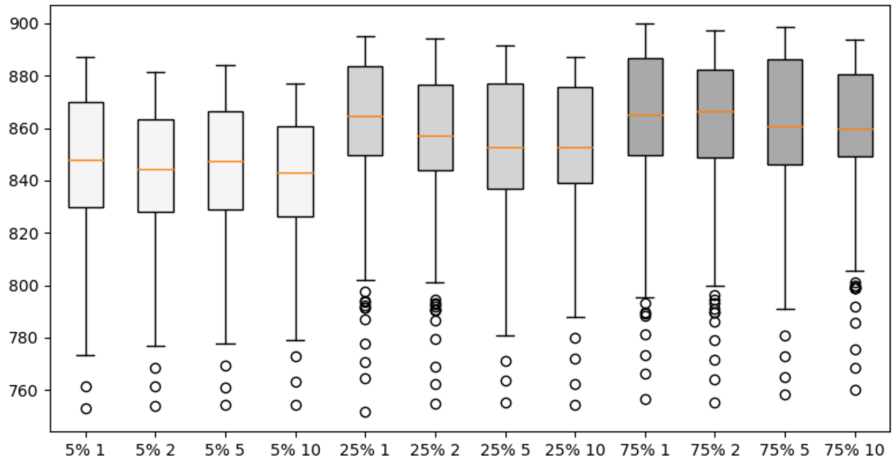


Figure 3: Random Immigrants. Variation of the immigrants percentage and immigration frequency - Dataset 1

The Figure 3 compares the results of different immigrants percentage (grouped by colour) and the immigration frequency (if the label is 5% 5, means that 5% of the population will be replaced by immigrants each 5 generations). The figure shows that, for this dataset, higher percentages and higher frequency have slightly better results.

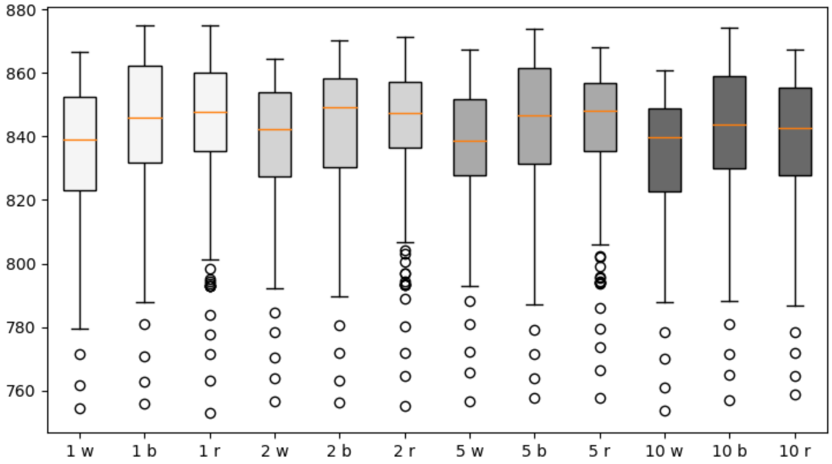


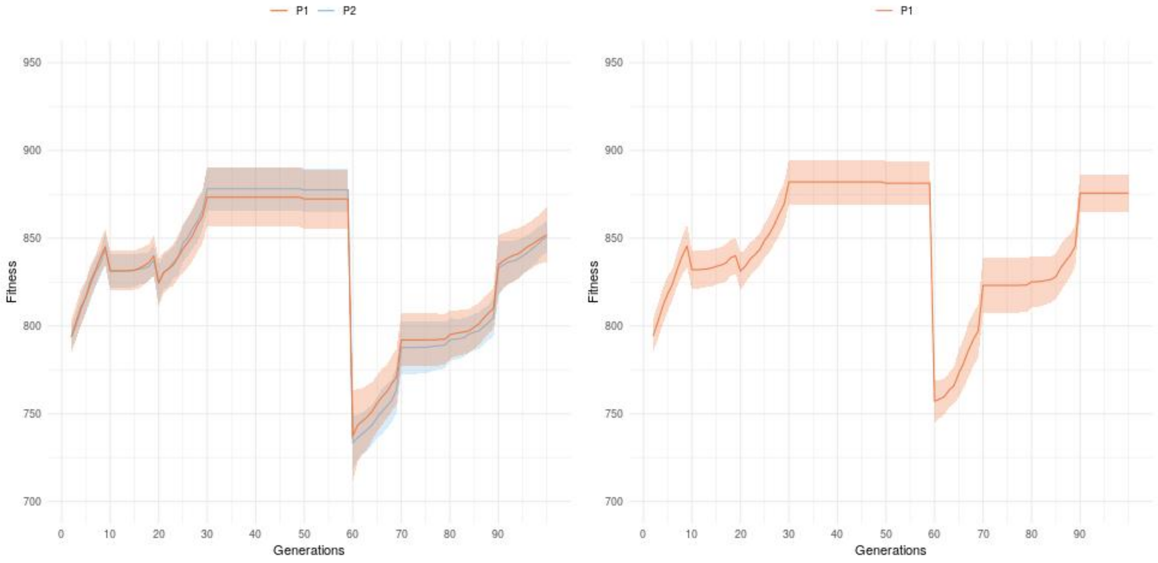
Figure 4: Multiple Populations. Comparison between method Random, Worst and Best, for the different migration frequencies tested and for 5% of individuals to exchange - Dataset 1

The method of selecting the worst fitted individuals to exchange populations showed a worst performance. Selecting the best and random individuals both had a similar performance.

				Random Immigrants											
				5%				25%				75%			
				1	2	5	10	1	2	5	10	1	2	5	10
Multi-populations	5%	1	worst	-	-	∼	∼	-	-	-	-	-	∼	∼	-
			best	∼	∼	∼	∼	-	-	-	∼	∼	∼	∼	-
			random	∼	-	∼	∼	-	-	-	-	∼	∼	∼	-
		2	worst	-	-	∼	∼	- -	-	-	-	-	-	∼	- -
			best	∼	∼	∼	∼	-	-	-	∼	∼	∼	∼	-
			random	∼	∼	∼	∼	-	-	-	∼	∼	∼	∼	-
		5	worst	-	-	-	∼	- -	-	-	-	-	∼	-	- -
			best	∼	∼	∼	∼	-	-	-	∼	∼	∼	∼	-
			random	∼	-	∼	∼	-	-	-	-	∼	∼	∼	-
		10	worst	-	-	∼	∼	- -	-	-	-	-	∼	∼	- -
			best	∼	-	∼	∼	-	-	-	-	∼	∼	∼	-
			random	∼	-	∼	∼	-	-	-	-	∼	∼	∼	-
	25%	1	worst	∼	-	∼	∼	-	-	-	-	-	∼	∼	-
			best	∼	∼	∼	∼	-	-	-	∼	∼	∼	∼	-
			random	∼	∼	∼	∼	-	-	-	-	∼	∼	∼	-
		2	worst	∼	-	∼	∼	-	-	-	-	-	∼	∼	-
			best	∼	∼	∼	∼	-	-	-	∼	∼	∼	∼	-
			random	∼	∼	∼	∼	-	-	-	∼	∼	∼	∼	-
		5	worst	∼	-	∼	∼	-	-	-	-	∼	∼	∼	-
			best	∼	-	∼	∼	-	-	-	∼	∼	∼	∼	-
			random	∼	∼	∼	∼	-	-	-	∼	∼	∼	∼	-
		10	worst	-	-	∼	∼	-	-	-	-	-	∼	∼	-
			best	∼	-	∼	∼	-	-	-	∼	∼	∼	∼	-
			random	∼	∼	∼	∼	-	-	-	∼	∼	∼	∼	-

Table 3: Statistical test - Dataset 2

The statistical results for the second dataset are sligthly different from the dataset 1. For the Random Immigrants' algorithm, with 75% of individuals to be replaced, in this case had similar results with the other algorithm, but in Table 2 the results were significantly better. As in Table 2 the algorithm in the columns had better results with 25%. Also we can see that in case there wasn't any large effect size.

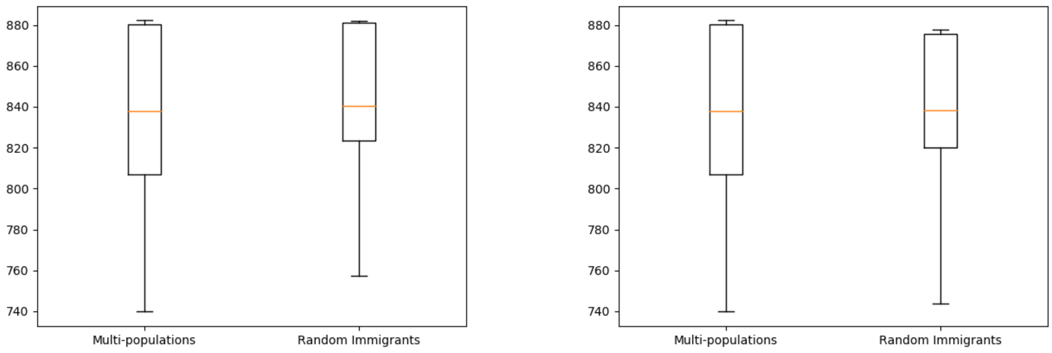


(a) Multi-populations. Individuals exchanged: 5%; Frequency migration: 10 generations; Method: worst individuals;

(b) Random Immigrants. Individuals replaced: 75%; Frequency immigration: all generations

Figure 5: Example where algorithm in b) is statistically better than a) - Dataset2

The Figure 5 shows graphically a case where the Random Immigrants algorithm is statistically better than the Multi-populations, for the second dataset. In this case, we can observe that after that big tumble on the fitness value, the Random Immigrants algorithm has a better recovery in comparison with the Multi population approach.



(a) Boxplot of the Fig 5 - medium effect size

(b) Boxplot of statistical similarity

Figure 6: Comparison between algorithms with different effect sizes - Dataset 2

In the Figure 6 we can see the two boxplots comparing the algorithms when they have a medium effect size (a) and when are statistically similar (b). For this dataset, the difference between both algorithms is not easily perceptible, since both boxplots have similar median, even though they have different effect sizes. Random Immigrants still performed better than the other algorithm.

The samples used for the first boxplot were the same as described in Figure 5 and for the second boxplot we used for the Multi-population, 5% of worst individuals to exchange every 10 generations, and in Random Immigrants, 75% of the worst individuals to be replaced, every 2 generation.

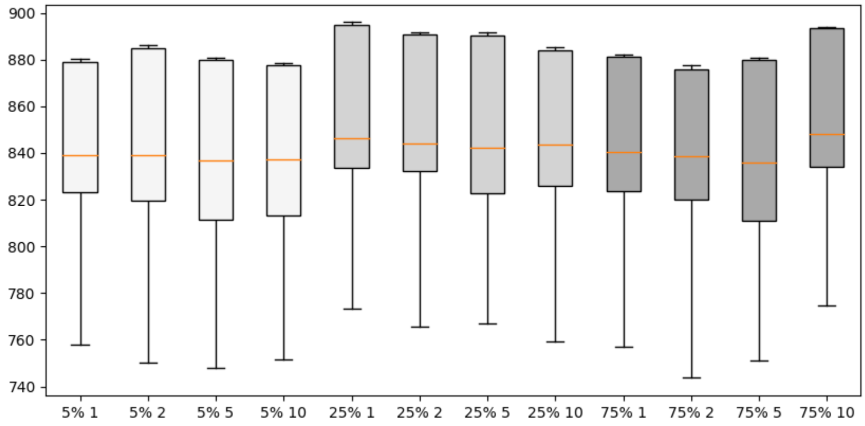


Figure 7: Random Immigrants. Variation of the immigrants percentage and immigration frequency - Dataset 2.

The Figure 7 compares the results of different immigrants percentage (grouped by colour) and the immigration frequency (if the label is 5% 5, means that 5% of the population will be replaced by immigrants each 5 generations). The figure shows that, as opposite of the previous dataset in Figure 3, that there are almost no difference between the parameters, but we can confirm that 25% of percentage of individuals chosen has slightly better performance.

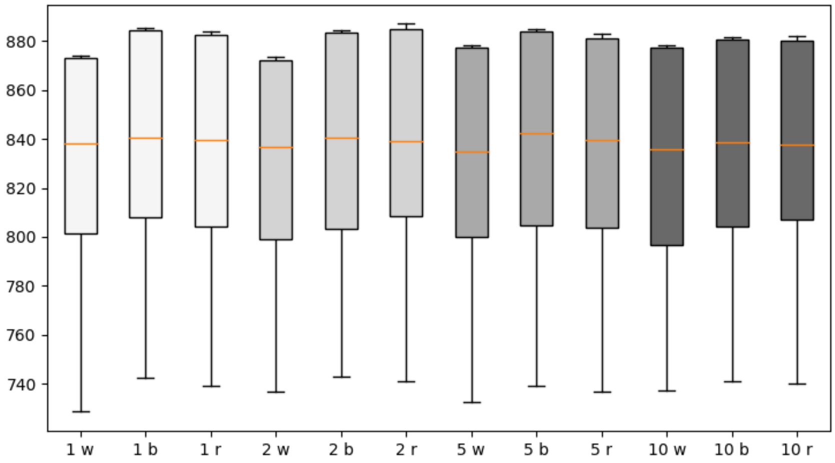


Figure 8: Multiple Populations. Comparison between method Random, Worst and Best, for the different migration frequencies tested and for 25% of individuals to exchange - Dataset 2

The method of selecting the worst fitted individuals to exchange populations showed a worst performance, as seen in Figure 4. Selecting the best and random individuals both had a similar performance.

5 Discussion and Conclusions

As we can see through the analysis of the statistical tests (table 2 and 3), the Random Immigrants approach is better.
In both datasets, the algorithm performed better with 25% of individuals to be replaced. In the other hand, when the Random Individuals approach has a small percentage of immigration (in our tests, 5%), both algorithms are statistically similar, which means that it has a worst performance with a smaller percentage of random individuals. Since the results for the second dataset were different from those for the 75% of the first dataset, we can not say which values are better.
In the case of the multi-population algorithm, changing the percentage of individuals to exchange population doesn't lead to notable results.
In the case of the frequency of migration/immigration, we can say the results are inconclusive in both algorithms.
The Multi-population algorithm obtained better results when selecting the best or random individuals to exchange populations, than choosing the worst individuals.
To improve the results for the Multi-population algorithm, we should've tried starting the populations in different search spaces, or being submitted to different selection methods and genetic operators. A more deeper analysis must be done, with different problem difficulties and test with more percentages, to make conclusions about the parameters.

References

Branke, J., Kaußler, T., Schmidt, C., & Schmeck, H. (2000). A multi-population approach to dynamic optimization problems. In *In adaptive computing in design and manufacturing* (pp. 299–308). Springer.

Brindle, A. (1980). Genetic algorithms for function optimization. *Unpublished doctoral dissertation, University of Alberta, Edmonton, Canada*. Retrieved from `https://doi.org/10.7939/R3FB4WS2W`

Cheng, H., & Yang, S. (2010). *Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks.*

Cobb, H. G. (1993). Genetic algorithms for tracking changing environments. In *Proceedings of the fifth international conference on genetic algorithms* (pp. 523–530). Morgan Kaufmann.

Grefenstette, J. (1992). Genetic algorithms for changing environments. In *Parallel problem solving from nature 2* (pp. 137–144). Elsevier.

Tinós, R., & Yang, S. (2007). Genetic algorithms with self-organizing behaviour in dynamic environments. In S. Yang, Y.-S. Ong, & Y. Jin (Eds.), *Evolutionary computation in dynamic and uncertain environments* (pp. 105–127). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from `https://doi.org/10.1007/978-3-540-49774-5_5` doi: 10.1007/978-3-540-49774-5_5

Trojanowski, K., & Michalewicz, Z. (1999). Evolutionary algorithms for non stationary environments. *Intelligent Inf. Systems, Proc. of the 8th Int. Workshop on Intelligent Inf. Syst.*, 229–240.

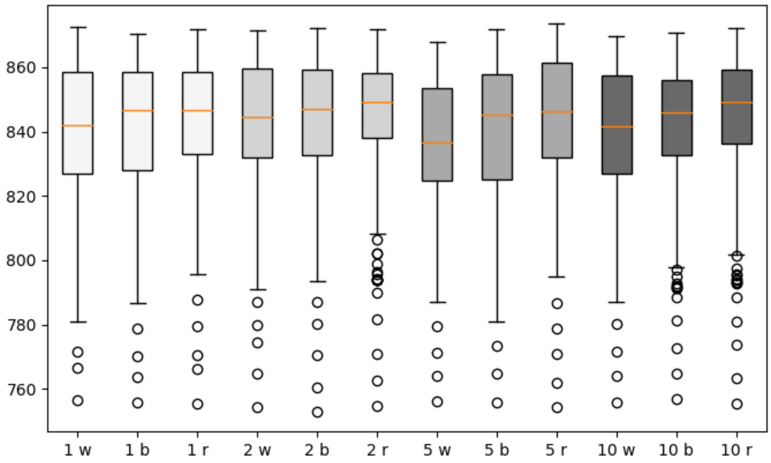


Figure 9: Multiple Populations. Comparison between method Random, Worst and Best, for the different migration frequencies tested and for 25% of individuals to exchange - Dataset 1

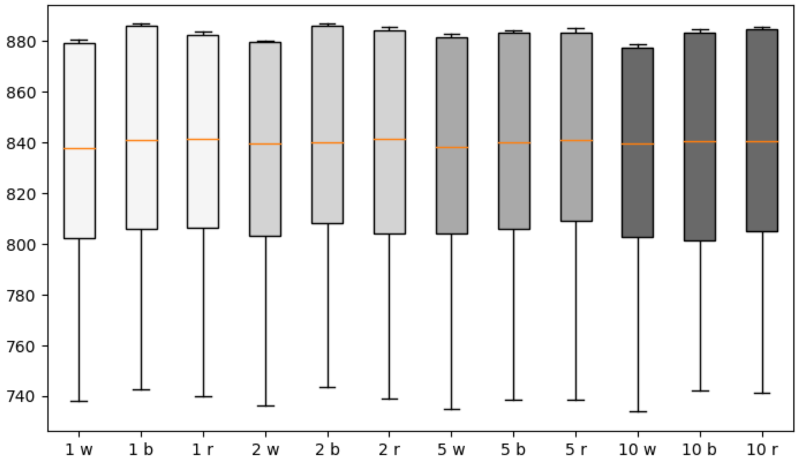


Figure 10: Multiple Populations. Comparison between method Random, Worst and Best, for the different migration frequencies tested and for 25% of individuals to exchange - Dataset 2