Parallel Island model Genetic Algorithms applied in $\mathcal{NP}$-Hard problems

Lucas A. da Silveira[†], José L. Soncco-Álvarez[*], Thaynara A. de Lima[‡] and Mauricio Ayala-Rincón[†]
[†]Department of Computer Science, Universidade de Brasília, 70900–010 Brasília D.F., Brazil
[*]Departamento Académico de Informática, Universidad Nacional de San Antonio Abad del Cusco, Cusco, Peru
[‡]Institute of Mathematics and Statistics, Universidade Federal de Goiás — Campus II, 74690–900 Goiânia, Brazil

***Abstract*—Designing efficient parallel island Genetic Algorithms (GA) is a difficult task: several decisions are needed related to the adequate structure of the islands, how they are connected, how many individuals should migrate, and how often they should migrate. The impact of these choices has not yet been fully understood since they might vary for different problems. In previous work, a variety of island model GAs to solve Reversal Distance Problem (RDP) over uni-crhomosomal genomes were proposed from which adequate choices were pointed out that provided results with an excellent balance among accuracy and performance. In this work, another evolutionary problem is considered in order to analyze how general were the decisions taken for island model GAs over RDP. The problem is translocation distance over multi-chromosomal genomes, which involves the interchange of gene between different chromosomes. Despite the fact that this problem falls also in the category of evolutionary distance problems, it is different from the RDP. Regarding accuracy, island models using a dynamic communication topology for exchange of individuals between islands provided the best solutions; while regarding performance, models using a static topology reached the highest speedup. Comparing with previous work on RDP, it was observed that islands models that did not provided good accuracy in RDP provided good quality solutions for translocation distance problem, while the best island models for RDP did not repeat the same success for translocation distance problem. The only invariant is that all the island model GAs in addition to competitive speedups provided better results than the corresponding sequential GA.**

I. Introduction

Genetic Algorithms (GAs) are stochastic search methods that have been successfully applied to many search, optimization, and machine learning problems. The evolutionary process consists of maintaining a population of coded solutions which are manipulated by variation operators to find a satisfactory optimized solution. A characteristic of GAs is the high computational time required for their executions when it involves treatment of $\mathcal{NP}$-hard and $\mathcal{NP}$-complete problems. One solution to speeding up the computation of solutions is to apply parallelism, since GAs can be implemented relatively easy to run in parallel architectures.

There is a variety of parallel model architectures in the literature [1], which can be used in GAs. A good option is the *coarse granularity model*, also known as *island model*, which favors achievement of both higher accuracy and performance. This model consists of a set of islands holding instances of GAs evolving their own populations locally and periodically; candidate solutions are exchanged between islands in a process called *migration*. This movement of solutions is conditioned by a migration policy and involves parameters that need to be carefully calibrated to obtain good results.

There are several papers showing the effectiveness of working with island models looking for performance [2], [3], [4] and accuracy [5], [6], [7], [8]. Motivated by these results, previous authors' work analyzed the impact caused by the parameters *communication topology* and *migration policy* varying the amount of individuals allocated in each island. The proposed island models were applied in evolutionary distance problems [9], [10].

Results of high quality were obtained for evolutionary distance problems using as measure the operation of *reversion* over unsigned uni-chromosomal genomes [10], which is a well-known $\mathcal{NP}$-hard problem [11]. In that work parallel homogeneous island models were proposed, where in all islands a GA with the same configuration of parameters is applied; and, the migration strategy exchanges the same number of individuals through the communication topology of the islands. Experiments, with models with different amount of islands, populations, and topologies, but maintaining the same total population, were performed and compared. Experiments were performed for a unique case-of-study (reversal distance problem), which implies that general conclusions over the adequability of the behaviour of the different parallel model were restricted. The current work deals with the interesting case of *translocation* distance problem with the aim of analyzing if the same behaviors are observed as for island models using as case-study the reversal distance problem. From this perspective, calibration of parameters such as communication topology between islands, number of individuals involved in the migratory process, migration interval and number of islands also are replicated when using the translocation distance problem were considered.

The first parallel island models treatment for the translocation distance problem was the memetic approach given in [12], where two communication topologies and a regular migration policy repeated at each generation were proposed reusing parameters calibrated for a sequential version. However, the results provided by the island models were not satisfactory, presenting less accuracy than the sequential version. In contrast to the last approach, the current work develops and performs such GA island models for the translocation distance problem with a careful setting phase to find the best values

for the parameters involved in migration policies and breeding cycle.

Reversions apply over uni-chromosomal genomes containing a simple genetic structure inverting just a contiguous subsequence of the whole genome, while translocations are applicable only to multi-chromosomal genomes. A translocation operation splits two different chromosomes of a multi-chromosomal genome and then joins the resulting blocks to construct two new chromosomes. Thus, genome distance can be modelled as an optimization problem which consists in finding the minimum number of such operations necessary to transform one genome into another one. According to the used operation, reversal and translocation distance are discriminated. These problems have two versions: one using *signed* genomes with polynomial solution and the other using *unsigned* genomes that is $\mathcal{NP}$-hard. This paper deals with the unsigned translocation distance problem (UTD). Of course, as $\mathcal{NP}$-hard problems, these problems are expressive enough to "encode" any other $\mathcal{NP}$ problem.

It is important to stress that reversal and translocation distance are different problems not only because they apply to uni- and multi- chromosomal genomes, but also because an instance of the reversal distance problem will always have a solution, which is not true for translocation distance problem. Two genomes with the same genes must satisfy specific properties to evolve one into the other through translocations as presented in Section II-A.

The paper is organized as follows: Section II presents definitions and terminology; Section III presents the parallel GA approaches for the UTD problem; Section IV presents experiments and results; and, before concluding and discussing future work in Section VI, Section V discusses the results. The source code and data used in the experiments is available at `genoma.cic.unb.br`.

II. Background

A. Definitions and Terminology

A *gene* is represented by an integer number, where oriented and non oriented genes are modeled by signed and unsigned integers, respectively. A *chromosome* is a finite sequence of genes and a *genome* is a set of chromosomes. Formally, a genome G with N chromosomes and n genes is a set $\{(x_{11},\ldots,x_{1r_1}),\cdots,(x_{N1},\ldots,x_{Nr_N})\}$, where $n=\sum_{k=1}^{N} r_k$ and $|x_{ij}|\neq|x_{kl}|$ whenever $i\neq k$ or $j\neq l$. For $1\leq i\leq N$ and $1\leq j_i\leq r_i$, if $x_{ij_i}\in[\pm n]=\{\pm1,\ldots,\pm n\}$ then G is called a signed genome, whereas if $x_{ij_i}\in[n]=\{1,\ldots,n\}$ then G is an unsigned genome. Notice that, if $N=1$, an unsigned genome is a permutation that belongs to the well-known symmetric group S_n, whereas if G is a signed genome then it is a permutation of S_{2n} [10], [13].

For an interval $I=[x_i,\ldots,x_j]$ within a signed (respec. unsigned) chromosome $X=(x_1,\ldots,x_k)$, one denotes $-\overleftarrow{I}=[-x_j,\ldots,-x_i]$ (respec. $\overleftarrow{I}=[x_j,\ldots,x_i]$) the reversed interval builded from I. A chromosome does not have orientation. It means that the chromosome X is equal to $-\overleftarrow{X}$, if X is over $[\pm n]$, whereas X and $\overleftarrow{X}$ are the same if X is over $[n]$.

A translocation is an operation that acts over two chromosomes of a genome. There are two types of translocations: prefix-prefix and prefix-suffix. Let $G=\{X_1,\ldots,X,\ldots,Y,\ldots,X_N\}$ be a genome and $X=(x_1,\ldots,x_l)$ and $Y=(y_1,\ldots,y_m)$ two chromosomes of G. A *prefix-prefix translocation* $\rho(X,Y,x_i,y_j)$, $1\leq i<l$, $1\leq j<m$, applied over G maintains the prefixes of X and Y and switches their suffixes, that is,

$$G\rho(X,Y,x_i,y_j)=\{X_1,\ldots,X',\ldots,Y',\ldots,X_N\},\text{ where}$$
$$X'=(x_1,\ldots,x_i,y_{j+1},\ldots,y_m)\text{ and}$$
$$Y'=(y_1,\ldots,y_j,x_{i+1},\ldots,x_l).$$

It is not necessary to specify whether G is a signed or an unsigned genome when one defines a prefix-prefix translocation once this concept is similar for both cases. A *prefix-suffix translocation* $\theta(X,Y,x_i,y_j)$, $1\leq i<l$, $1\leq j<m$, applied over G maintains the prefix of X and the suffix of Y and interchanges the suffix of X with the prefix of Y, that is,

$$G\theta(X,Y,x_i,y_j)=\{X_1,\ldots,X',\ldots,Y',\ldots,X_N\},\text{ where}$$
$$X'=(x_1,\ldots,x_i,-y_j,\ldots,-y_1)\text{ and}$$
$$Y'=(-x_l,\ldots,-x_{i+1},y_{j+1},\ldots,y_m),$$

when G is a signed genome and

$$X'=(x_1,\ldots,x_i,y_j,\ldots,y_1)\text{ and}$$
$$Y'=(x_l,\ldots,x_{i+1},y_{j+1},\ldots,y_m),$$

when G is an unsigned genome (See Figure 1).

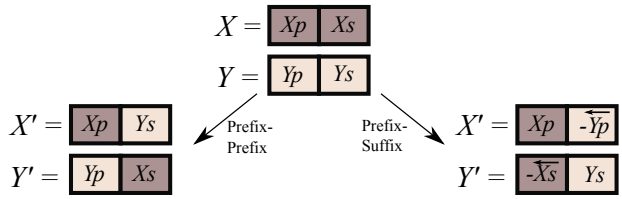

Fig. 1. Prefix-prefix and prefix-suffix translocations over a signed genome.

Example: Consider the signed genomes

$$A=\{(+1,+3),(+4,-8,+5,+2,+6),(+7,+9)\},\text{ and}$$
$$B=\{(+1,+2,+3),(+4,+5,+6),(+7,+8,+9)\}$$

Observe that,

$$\begin{aligned}
A &= \{(+1,\underline{+3}),(+4,-8,+5,\underline{+2,+6}),(+7,+9)\}\\
A\rho_1 &= \{(+1,+2,\underline{+6}),(+4,-8,+5,\underline{+3}),(+7,+9)\}\\
A\rho_1\rho_2 &= \{(+1,+2,+3),(+4,-8,\underline{+5,+6}),(\underline{+7},+9)\}\\
A\rho_1\rho_2\theta_1 &= \{(+1,+2,+3),(+4,\underline{-8,-7}),(\underline{-6,-5},+9)\}\\
A\rho_1\rho_2\theta_1\theta_2 &= \{(+1,+2,+3),(+4,+5,+6),(+7,+8,+9)\}
\end{aligned}$$

Thus, the prefix-prefix translocations ρ_1 and ρ_2 together with the prefix-suffix translocations θ_1 and θ_2 transform A into B.

Given a chromosome $X=(x_1,\ldots,x_l)$ the elements of the set $\{x_1,-x_l\}$ are called *tails* of X, if X is a signed chromosome, whereas the elements of the set $\{x_1,x_l\}$ are the tails of X in the unsigned case. Two genomes are called *co-tails* if they have the same set of tails. Notice that if G is a

genome and ρ is a translocation, then $G\rho$ and G are co-tails; thus, to ensure that a genome A can be transformed into a genome B by translocations, one must require that A and B are co-tails. The two versions of the genome rearrangement problem by translocations are defined as below.

- **Unsigned Translocation Distance Problem *(UTD)*:** given two unsigned co-tails genomes A and B with the same number and over the same set of genes, determine the minimum number of translocations needed to transform A into B. One can assume that the genes of B are in increasing order. One calls B an identity genome.
- **Signed Translocation Distance Problem *STD*:** it is defined analogously to *UTD*, but in this case A and B are signed co-tails genomes and the genes of the signed identity genome B are positive and in increasing order.

If $\rho_1, \ldots, \rho_k$ is a shortest sequence of translocations such that $A\rho_1 \ldots \rho_k = B$ then k is called the *translocation distance* between A and B. In the previous example, the translocation distance between A and B is 4. Since the identity is sorted, the UTD and STD problems from A to an identity (that is co-tail with A) are also referred as the problem of *sorting* genome A by translocations. Observe that to sort a genome A over $[\pm n]$, it is necessary that its tails include the extremes $+1$ and $-n$, and, for $1 < i < n$, if $-i$ is tail then $+(i+1)$ must be tail of A. See the example above that sorts A into the identity $A\rho_1\rho_2\theta_1\theta_2$ and notice that the genome below cannot be sorted.

$$\{(+1, +5), (+3, -8, +4, +2, +6), (+7, +9)\}$$

This happens since the genes in all identities are ordered increasingly and there exist no possible identity with the same tails than this genome: $\{+1, +3, -5, -6, +7, -9\}$.

Zhu and Wang showed that computing the translocation distance between unsigned genomes is $\mathcal{NP}$-hard [14], thus algorithms that provide approximated solutions for UTD are of great interest. On the other hand, STD is in class $\mathcal{P}$. Linear time algorithms for computing the translocation distance between signed genomes are presented in [15], [16]. In this work, a parallel GAs that approximate solutions for UTD are proposed, whose heuristic use exact solutions for associated STD computed by the linear time algorithm in [16].

B. Genetic Algorithm for solving UTD

In [17] we present a sequential GA, $\mathcal{GA}_{\text{S}}$ shown in Algorithm 1, of time complexity $O(n^3 \log n)$ to solve UTD for genomes with n genes with the aim of improving the accuracy of the results provided by the 1.5+ε-approximation algorithm in [18]. Experiments showed that the proposed $\mathcal{GA}_{\text{S}}$ outperforms the quality of solutions computed by the approximation algorithm. The proposed $\mathcal{GA}_{\text{S}}$ takes as input a genome A (to be transformed into a co-tail identity) and its output is the number of translocations to sort A. Initially, a population of $n \log n$ individuals is built. These individuals are signed genomes generated by a random attribution of signs to the unsigned genes in A. Each of these individuals corresponds to a STD problem with exact solutions that are also feasible solutions for the unsigned genome A; thus, to compute the fitness of these signed individuals, the linear algorithm proposed in [16] is applied. The genetic operations in $\mathcal{GA}_{\text{S}}$ work as follows: *selection* considers 80% of the best individuals in the population; *crossover* uses single-point cut mechanism with probability of 90%, and as a policy of insertion of the offspring, the candidates to leave the current population are those that have the 70% worst fitness values. To improve the population quality, in each generation the two best individuals of the current population are selected for which *crossover* and *mutation* operations are applied (with 2% probability) generating two new descendants. If the new offspring individuals are better than those in the population (regarding their fitness values), they are incorporated in the current population. Algorithm $\mathcal{GA}_{\text{S}}$ finishes after completing n generations.

Algorithm 1: GA for Calculating UTD - $\mathcal{GA}_{\text{S}}$

Input: Unsigned genomes A and B (identity genome)
Output: Number of translocations to transforming A into B

```
Generate the initial population of signed genomes;
Compute fitness of the initial population;
for i = 1 to Length(A) do
    Perform selection and save the best solution found;
    Apply the crossover operator;
    Apply the mutation operator;
    Compute fitness of the offspring;
    Perform replacement of the worst individuals;
```

C. Parallel Island Models

Islands models are used to implement GAs in parallel with an island usually representing a processor. In an implementation of the model, each island runs an instance of the GA maintaining its own population, such that each island evolves independently and, periodically, a portion of its population is exchanged in a process called migration. Migration involves a set of parameters that impact both the performance and accuracy of the multi-island algorithm [1]. In this work, a *target* island refers to the island that will receive individuals from a local *source* island. A brief description of the set of parameters involved in the migration is given below.

- *NumMigIndividuals:* represents the amount of individuals that will be sent and received from one island to another.
- *TypeEmIndividual:* represents the type of individuals selected for emigration that will be sent to the target island. The types of individuals are:
  1) Better individuals.
  2) Worse individuals.
  3) Random individuals.
- *EmPolicy:* represents the emigration policy in the local island that can be:
  1) Clone individuals.

2) Remove individuals.

- *TypeImIndividual:* represents the immigration policy, selects the type of individuals to be replaced by individuals who are arriving to the target island, and can be classified as:
  1) Worse individuals.
  2) Random individuals.
  3) Similar individuals (with the same fitness).
- *MigrationInterval:* represents the migration interval defining generations in which exchange of individuals between islands is allowed and performed.
- *Topology*: is the organizational model that defines how islands are connected.

As in [9] *MigrationInterval* is defined as being a breeding cycle percentage. One way to increase accuracy and performance in island models is to explore the topology parameter, that can be done both statically and dynamically. In the static approach, given a set of islands $[x_1, x_2, \cdots, x_n]$, the relationships between two islands x_i and x_j are fixed at the beginning of the algorithm and persisted until the end. In dynamic models, relationships between island may be altered at each migration cycle to increase the diversity of the genetic material.

The research to solve URD problem with islands models proposing a master-slave model started in [19], where isolated islands were proposed such that each slave process maintains its own population and executes an instance of the GA. The master process receives the best fitness values computed by the slave processes in each stage of the GA. Even applying the simplest isolated island model, the results of this work were encouraging providing better solutions regarding the sequential GA approach.

Two island models to solve UTD problem using a $24\times$ larger search space than the sequential GA under comparison were proposed in [12]. The first with islands progressing in isolation, and the second using a complete graph topology for communication between islands, and preserving the parameters used in the breeding cycle fixed in [20]. As migration policy each island sends its the best individual to replace the worst individual on the target island in each generation. Surprisingly, the best results were obtained by the isolated island model; the second proposed model provided worse results, even than the sequential GA, due to the premature convergence established by the elitist migration policy, which, according to [21] and [22], negatively impacts the quality of the solutions.

The URD problem was further addressed in [23], using a ring and a complete graph topology. Experiments were performed in order to set the parameters of $\mathcal{GA}_{\mathrm{S}}$ and the migration policy obtaining a configuration which provided the best results. In addition, two population generation variants were used: **1)** the population of $\mathcal{GA}_{\mathrm{S}}$ was partitioned into populations of equal size and distributed among the islands; **2)** different populations were separately generated for each island. From the analysis of results and statistical tests, the models using population generation **1** provided better solutions for both ring and complete graph topologies, highlighting ring topology. Later in [10], static (Fig. 2) and a dynamic topologies were used, where the dynamism applied consist of classifying the islands as good, medium and bad, and then organize the communication between islands according to their ranking. Indeed, three types of communication organization

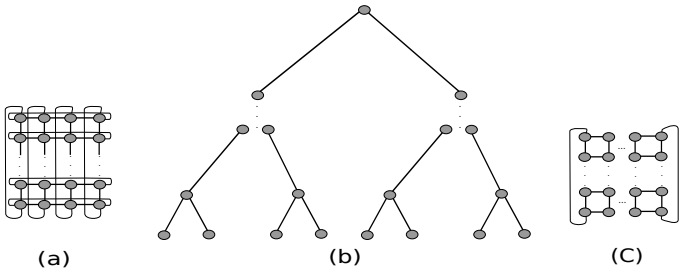

Fig. 2. (a) toru (b) tree (c) x×y net, where x and y represent rows and columns in the net.

were proposed: **1)** communication between islands with the same classification; **2)** communication of good islands with bad islands and, medium with medium islands; and, **3)** random communication among pairs of islands. The best results considering accuracy and performance were obtained from models using a 4×3-net static topology and a dynamic topology using random communication. In both cases, the best option was partitioning the population generated for the sequential version. In particular, the model using static topology reached almost linear speedup (i.e. speedup almost equal to the number of processors) and retained an accuracy similar to dynamic model.

III. Parallel Island Model GAs

Algorithm 2: Generation of the population for each island

Input: Unsigned genome A
Output: Each island with its own initial population

```
1 p = numberIslands;
2 Island 0 generates n log n signed genomes from A for
    itself;
3 for i = 1 to p − 1 do
4 |   Island 0 generates n log n signed genomes from A
  |     and send them to Island i (MPI_Send);
5 for i = 1 to p − 1 do
6 |   Island i receives n log n signed genomes from Island
  |     0 (MPI_Recv);
```

The models with static and dynamic topologies (proposed in [9], [10]) are adapted for UTD. In these models the breeding cycle and the migration interval follows as described in Section II-B. Since island models in which the population in each island comes from partitioning the population generated for the sequential GA provided (statistically validated) better results than island models in which each island generates its own

initial population, the former option to populate islands was selected. Sections III-A and III-B provide the nomenclature used in the adopted islands models.

A. Parallel island model GA with Static Topology

For each static topology two Parallel Island Model GAs (for short, *PIMGAs*) were proposed, organized as below, where the subscript 12 and 24 stand for the number of islands in the model.

PIMGAs where each island has the population generated according to Algorithm 2:

- $\mathcal{GA}_{\text{F12}}$ and $\mathcal{GA}_{\text{F24}}$, for the complete graph topology.
- $\mathcal{GA}_{\text{R12}}$ and $\mathcal{GA}_{\text{R24}}$, for the ring topology.
- $\mathcal{GA}_{\text{TR12}}$ and $\mathcal{GA}_{\text{TR24}}$, for the tree topology.
- $\mathcal{GA}_{\text{TO12}}$ and $\mathcal{GA}_{\text{TO24}}$, for the torus topology.
- $\mathcal{GA}_{\text{N12}}$ with 4×3-net topology and $\mathcal{GA}_{\text{N24}}$ using 6×4-net topology.

B. Parallel island model GA with Dynamic Topology

The dynamism consists of initially qualify the islands as being: good, medium and bad as explained in [10], and from this classification the connections between islands are generated.

In the same way as for static topologies, the models are divided into models with 12 and 24 islands. The proposed algorithms are listed below, where the subscripts 12 and 24 refer to the number of islands in the model.

PIMGAs with population generated according to Algorithm 2:

- $\mathcal{GA}_{\text{same12}}$ and $\mathcal{GA}_{\text{same24}}$: communication between islands with same classification: good with good, medium with medium and bad with bad.
- $\mathcal{GA}_{\text{gbmm12}}$ and $\mathcal{GA}_{\text{gbmm24}}$: communication between bad and good, and medium and medium islands.
- $\mathcal{GA}_{\text{Rand12}}$ and $\mathcal{GA}_{\text{Rand24}}$: random communication between islands. Before each migration, connections between islands are generated randomly.

IV. Experiments and Results

The island models were implemented using the `MPI` library of the `C` language. The experiments were executed on a server with 256GB of RAM with two processors Xeon E5-2620. Each processor has six cores with hyper-threading enabled.

A. Experiments Setup

The population on each island is composed of signed genomes created from unsigned genome A provided as input.

The sequential GA has populations of size $24 \times (n \log n)$, where n is the length of the input. On the other hand, parallel island models have two configurations. For models with 24 islands, each island has $n\ logn$ individuals, while for those with 12 islands, $2 \times (n\ logn)$ individuals are allocated for each island. Thus, all models are performed with total populations of the same size. Each individual in the population is generated from the input A, assigning randomly either a positive or negative signal to each internal gene and a positive signal to the end genes of each chromosome of A. In this way, it is guaranteed that the generated individuals are co-tails with the identity (and thus they can be sorted).

TABLE I
Estimated Values for the Parameters

Parameter	estimated values
Crossover probability	$10\%, 12\%, \cdots, 98\%, 100\%$
Mutation probability	$1\%, 2\%, \cdots, 5\%$
Percentage for selection	$10\%, 12\%, \cdots, 98\%, 100\%$
Percentage for replacement	$10\%, 12\%, \cdots, 98\%, 100\%$
NumMigIndividuals	1,2,3,4,5,6,7,8,9,10,11,12,13
TypeEmIndividual	1=Better, 2=Worse, 3=Random
EmPolicy	1=Clone, 2=Remove
TypeImIndividual	1=Worse, 2=Random, 3=Similar
MigrationInterval	$1\%, 2\%, \cdots, 99\%, 100\%$

Searching for island models that can provide solutions with good accuracy, a careful parameter definition phase was performed, where given a set of possible estimated values (Table I) for each parameter, the models were submitted to evaluation by testing each referenced value. In the end, the parameters that provided the best solutions (Table II) were selected. In the setting phase the used input packages consisted of one hundred synthetic genomes for genomes of sizes $n \in \{50, 100, 150\}$. For each package of one hundred genomes, the island models were executed ten times and the average results of the executions (number of translocations) for each model was computed.

B. Experiments: Performance and Accuracy

Two experiments were performed:

1) Speed-up of the PIMGAs regarding the sequential $\mathcal{GA}_{\text{S}}$: one hundred genomes with 150 genes and 5 chromosomes were randomly generated. Subsequently, each PIMGA was executed ten times for each genome in the package and the average run-time was taken as mean time. The speedups are presented in Table III.
2) Accuracy of results provided by PIMGAs:
the input data for this experiment consists of packages of one hundred genomes, generated as mentioned in previous section, with n genes varying by 10 genes from $n = 100$ until $n = 150$, and m chromosomes, for $m \in \{3, 4, 5\}$. The island models were executed ten times for each genome into each package, and the average (number of translocations) was taken as result for each genome. Tables IV, V, VI, VII, VIII, IX, X compare the island models with 12 and 24 islands presented in Sections III-A and III-B showing the results, where the value in each cell represents the average translocation distance computed for each package of one hundred genomes and the best results are highlighted. In addition, Tables XII, XIII, XIV compare only the island models that provided the best solutions in previous tables (IV, V, VI, VII, VIII, IX, X) for genomes with the same number of chromosomes.

TABLE II

PARAMETER SETTINGS FOR THE *PIMGAs*. 1=$\mathcal{GA}_{\text{R12}}$, 2=$\mathcal{GA}_{\text{R24}}$, 3= $\mathcal{GA}_{\text{F24}}$, 4=$\mathcal{GA}_{\text{F12}}$, 5=$\mathcal{GA}_{\text{TR12}}$, 6=$\mathcal{GA}_{\text{TR24}}$, 7=$\mathcal{GA}_{\text{TO12}}$, 8=$\mathcal{GA}_{\text{TO24}}$, 9=$\mathcal{GA}_{\text{N12}}$, 10=$\mathcal{GA}_{\text{N24}}$, 11=$\mathcal{GA}_{\text{SAME12}}$ 12=$\mathcal{GA}_{\text{SAME24}}$, 13=$\mathcal{GA}_{\text{GBMM12}}$, 14=$\mathcal{GA}_{\text{GBMM24}}$, 15=$\mathcal{GA}_{\text{RAND12}}$, 16= $\mathcal{GA}_{\text{RAND24}}$

Parameter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Crossover probability	96%	97%	94%	96%	92%	84%	96%	84%	88%	97%	94%	97%	96%	96%	94%	95%
Mutation probability	1%	1%	1%	1%	1%	1%	1%	1.4%	1%	1%	1%	1%	1%	1.4%	2%	1.1%
Pct. for selection	94%	93%	96%	93%	80%	96%	82%	96%	94%	68%	97%	96%	47%	94%	40%	54%
Pct. for replacement	40%	20%	70%	40%	55%	70%	32%	70%	40%	30%	30%	72%	34%	50%	30%	56%
NumMigIndividuals	9	2	5	4	8	6	8	7	3	5	9	7	6	5	3	5
TypeEmIndividual	3	2	1	1	1	1	1	1	1	1	3	1	3	1	1	1
EmPolicy	1	2	2	1	2	1	2	1	2	1	2	1	1	1	2	2
TypeImIndividual	1	2	1	2	1	2	1	2	2	1	3	3	2	3	1	1
MigrationInterval	10%	60%	30%	10%	90%	55%	42%	55%	32%	30%	10%	34%	40%	10%	10%	11%

TABLE III

SPEED-UP FOR THE EXPERIMENT WITH GENOMES OF LENGTH 150. 1=$\mathcal{GA}_{\text{R12}}$, 2=$\mathcal{GA}_{\text{R24}}$, 3= $\mathcal{GA}_{\text{F24}}$, 4=$\mathcal{GA}_{\text{F12}}$, 5=$\mathcal{GA}_{\text{TR12}}$, 6=$\mathcal{GA}_{\text{TR24}}$, 7=$\mathcal{GA}_{\text{TO12}}$, 8=$\mathcal{GA}_{\text{TO24}}$, 9=$\mathcal{GA}_{\text{N12}}$, 10=$\mathcal{GA}_{\text{N24}}$, 11=$\mathcal{GA}_{\text{SAME12}}$ 12=$\mathcal{GA}_{\text{SAME24}}$, 13=$\mathcal{GA}_{\text{GBMM12}}$, 14=$\mathcal{GA}_{\text{GBMM24}}$, 15=$\mathcal{GA}_{\text{RAND12}}$, 16= $\mathcal{GA}_{\text{RAND24}}$

Length	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
150	7.9	14.35	8.01	14.33	16.88	8.75	16.25	8.71	15.16	10.45	22.75	14.10	14.01	8.12	**22.81**	15.96

TABLE IV

EXPERIMENTS WITH HUNDRED GENOMES WITH 3,4 AND 5 CHROMOSOMES. 1=$\mathcal{GA}_{\text{SAME24}}$, $\mathcal{GA}_{\text{SAME12}}$

	3 chrom.		**4 chrom.**		**5 chrom.**	
n	1	2	1	2	1	2
100	63.43	**63.4**	77.86	**61.15**	75.03	**57.68**
110	89.26	**70.94**	86.73	**67.76**	84.36	**64.69**
120	98.4	**78.08**	96.03	**74.9**	93.46	**71.89**
130	107.85	**85.87**	104.73	**81.18**	102.27	**77.96**
140	117.15	**93.09**	113.85	**88.28**	111.4	**84.81**
150	126.57	**100.69**	122.91	**95.23**	119.97	**91.05**

TABLE V

EXPERIMENTS WITH HUNDRED GENOMES WITH 3,4 AND 5 CHROMOSOMES. 1=$\mathcal{GA}_{\text{GBMM24}}$, 2=$\mathcal{GA}_{\text{GBMM12}}$

	3 chrom.		**4 chrom.**		**5 chrom.**	
n	1	2	1	2	1	2
100	**63.26**	63.39	**61.07**	61.15	**57.62**	57.67
110	**70.76**	70.93	**67.58**	67.71	**64.57**	64.65
120	**77.84**	78.03	**74.71**	74.86	**71.75**	71.88
130	**85.57**	85.8	**80.95**	81.15	**77.81**	77.95
140	**92.8**	93.06	**88.01**	88.22	**84.61**	84.8
150	**100.4**	100.67	**94.97**	95.19	**90.8**	91.01

TABLE VI

EXPERIMENT WITH HUNDRED GENOMES WITH 3,4 AND 5 CHROMOSOMES. 1=$\mathcal{GA}_{\text{RAND24}}$, 2=$\mathcal{GA}_{\text{RAND12}}$

	3 chrom.		**4 chrom.**		**5 chrom.**	
n	1	2	1	2	1	2
100	**63.41**	63.57	**61.16**	61.24	**57.66**	57.76
110	**71.0**	71.18	**67.74**	67.9	**64.67**	64.78
120	**78.13**	78.39	**74.94**	75.14	**71.88**	72.03
130	**85.91**	86.19	**81.2**	81.45	**77.95**	78.23
140	**93.2**	93.47	**88.36**	88.61	**84.83**	85.04
150	**100.85**	101.2	**95.28**	95.57	**91.1**	91.25

TABLE VII

EXPERIMENT WITH HUNDRED GENOMES WITH 3,4 AND 5 CHROMOSOMES. 1=$\mathcal{GA}_{\text{N24}}$, 2=$\mathcal{GA}_{\text{N12}}$

	3 chrom.		**4 chrom.**		**5 chrom.**	
n	1	2	1	2	1	2
100	**63.32**	63.41	**61.1**	61.16	**57.65**	57.69
110	**70.84**	70.95	**67.65**	67.71	**64.61**	64.64
120	**77.97**	78.06	**74.82**	74.86	**71.83**	71.88
130	**85.71**	85.84	**81.05**	81.14	**77.91**	77.94
140	**92.95**	93.07	**88.13**	88.24	**84.7**	84.78
150	**100.55**	100.73	**95.07**	95.21	**90.88**	90.99

TABLE VIII

EXPERIMENT WITH HUNDRED GENOMES WITH 3,4 AND 5 CHROMOSOMES. 1=$\mathcal{GA}_{\text{R24}}$ AND $\mathcal{GA}_{\text{R12}}$

	3 chrom.		**4 chrom.**		**5 chrom.**	
n	1	2	1	2	1	2
100	**63.27**	63.41	**61.09**	61.14	**57.63**	57.67
110	**70.82**	70.94	**67.59**	67.73	**64.56**	64.65
120	**77.88**	78.08	**74.76**	74.88	**71.78**	71.92
130	**85.66**	85.85	**81.02**	81.14	**77.83**	77.94
140	**92.87**	93.1	**88.08**	88.25	**84.67**	84.77
150	**100.46**	100.67	**95.03**	95.2	**90.85**	90.99

TABLE IX

EXPERIMENT WITH HUNDRED GENOMES WITH 3,4 AND 5 CHROMOSOMES. 1=$\mathcal{GA}_{\text{TR24}}$, 2=$\mathcal{GA}_{\text{TR12}}$

	3 chrom.		**4 chrom.**		**5 chrom.**	
n	1	2	1	2	1	2
100	**63.31**	63.44	**61.1**	61.19	**57.63**	57.72
110	**70.85**	71.01	**67.63**	67.76	**64.61**	64.69
120	**77.95**	78.14	**74.74**	74.97	**71.8**	71.93
130	**85.72**	85.92	**81.03**	81.22	**77.85**	78.03
140	**92.91**	93.21	**88.1**	88.35	**84.69**	84.89
150	**100.52**	100.84	**95.05**	95.29	**90.9**	91.13

V. DISCUSSION

According to the experiments to calculate the speedup with respect to the sequential GA, $\mathcal{GA}_{\text{S}}$, given in Table III, the better performances were obtained by the models using 12 islands: $\mathcal{GA}_{\text{Rand24}}$, $\mathcal{GA}_{\text{same24}}$, $\mathcal{GA}_{\text{TR24}}$, and $\mathcal{GA}_{\text{TO24}}$, with $\mathcal{GA}_{\text{Rand24}}$ almost reaching linear speedup (22.81). In contrast, these models provided the worst solutions regarding accuracy, in general. For the models that provided the best quality solutions, which are those with 24 islands, the speedups are much smaller, but competitive. This is explained since for models we have extra overhead cost for creating and destroying 12 additional processes.

TABLE X
EXPERIMENT WITH HUNDRED GENOMES WITH 3,4 AND 5 CHROMOSOMES. 1=$\mathcal{GA}_{\text{F24}}$, 2=$\mathcal{GA}_{\text{F12}}$

	3 chrom.		4 chrom.		5 chrom.	
n	1	2	1	2	1	2
100	**63.31**	63.41	**61.1**	61.17	**57.62**	57.66
110	**70.83**	70.96	**67.62**	67.71	**64.59**	64.66
120	**77.94**	78.09	**74.71**	74.9	**71.8**	71.84
130	**85.71**	85.86	**81.0**	81.14	**77.86**	78.01
140	**92.91**	93.11	**88.06**	88.29	**84.68**	84.74
150	**100.53**	100.75	**95.01**	95.2	**90.84**	90.99

TABLE XI
EXPERIMENT WITH HUNDRED GENOMES WITH 3,4 AND 5 CHROMOSOMES. 1=$\mathcal{GA}_{\text{TO24}}$, 2=$\mathcal{GA}_{\text{TO12}}$

	3 chrom.		4 chrom.		5 chrom.	
n	1	2	1	2	1	2
100	**63.28**	63.4	**61.1**	61.17	**57.63**	57.69
110	**70.8**	70.96	**67.61**	67.73	**64.61**	64.66
120	**77.91**	78.06	**74.75**	74.91	**71.8**	71.92
130	**85.66**	85.82	**81.02**	81.18	**77.85**	77.98
140	**92.85**	93.06	**88.11**	88.28	**84.69**	84.83
150	**100.48**	100.7	**95.07**	95.26	**90.88**	91.03

Analyzing Tables V to XI, it can be seen that models using 24 island provide better solutions on average than those using 12 island. The exception occurs in Table IV where the dynamic model $\mathcal{GA}_{\text{same12}}$ computes better results. Experiments in Tables XII, XIII and XIV, comparing the results for the better models were performed. From these experiments it can be observed that all models provided better solutions than the sequential version $\mathcal{GA}_{\text{S}}$. The dynamic model $\mathcal{GA}_{\text{gbmm24}}$ provided on average the best solutions for all inputs and the worst solutions were obtained with the dynamic model $\mathcal{GA}_{\text{same12}}$.

To validate the results in Tables XII, XIII and XIV, a statistical analysis (according to the methodology in [24]) was performed over the multiplicative inverse of the results computed for 100 genomes with lengths varying from 100 until 150 for 3,4 and 5 chromosomes. The statistical analysis consists in first applying the Friedman test; if the null hypothesis ($h0$: all algorithms have the same performance) is rejected then a post-hoc test is performed, which in this case is the Holm Test. The Holm test allows one to test the null hypothesis that a control algorithm has the same performance with respect to another algorithm. For both tests, a significance level of $\alpha = 0,05$ was used. Tables XV, XVI, XVII show the results from the statistical analysis, where we can see that $\mathcal{GA}_{\text{gbmm24}}$ is chosen as the control algorithm since it provided better accuracy when compared to the other models used.

VI. Conclusions and future work

Parallel homogeneous island GAs proposed in [9], [10] were applied to solve the unsigned translocation distance problem concluding how these models perform regarding accuracy and speedup. As for reversal distance problem, the algorithms outperformed the sequential GA.

The choice of UTD and URD problems that involve evolutionary scenarios, initially left the hypothesis that the proposed island models could provide similar behavior. However, from the parameter calibration phase, it was noticed that the settings were not the same for parameters as migration and breeding cycle. From experiments, it became even clearer that accuracy and performance had different behavior. If one considers $\mathcal{GA}_{\text{N12}}$ and $\mathcal{GA}_{\text{Rand12}}$ which provided the best accuracy for URD, when applied to UTD they obtained low and medium quality solutions, respectively. Considering performance, the speedup of $\mathcal{GA}_{\text{Rand12}}$ for URD was reasonable, while in this work for UTD it reached almost linear speedup , already for $\mathcal{GA}_{\text{N12}}$ the speedup was similar to the performance achieved with URD problem.

Regarding number of islands in the model, the best results for URD were provided using 12 islands, while for UTD using 24 islands, providing the model $\mathcal{GA}_{\text{gbmm24}}$ solutions in average with the best quality. Thus, the hypothesis that the models might provide similar results was refuted concluding that the behavior of island models vary a lot from one problem to another.

As future work, it will be of interest considering problems out of the evolutionary scenario.

Acknowledgment

Research partially funded by FAPDF. Fourth author partially funded by a CNPq grant.

References

[1] D. Sudholt, *Springer Handbook of Computational Intelligence*. Springer, 2015, ch. Parallel Evolutionary Algorithms, pp. 929–959.

[2] R. Tanese, "Parallel genetic algorithms for a hypercube," in *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1987, pp. 177–183. [Online]. Available: http://dl.acm.org/citation.cfm?id=42512.42536

[3] E. Cantu-Paz, "Migration policies, selection pressure, and parallel evolutionary algorithms," *J. of Heuristics*, vol. 7, no. 4, pp. 311–334, Jul 2001. [Online]. Available: https://doi.org/10.1023/A:1011375326814

[4] A. Mambrini and D. Sudholt, "Design and analysis of schemes for adapting migration intervals in parallel evolutionary algorithms," *Evolutionary Computation*, vol. 23, no. 4, pp. 559–582, 2015.

[5] R. Tanese, "Distributed genetic algorithms," in *Proceedings of the Third International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 434–439. [Online]. Available: http://dl.acm.org/citation.cfm?id=93126.94044

[6] T. Starkweather, D. Whitley, and K. Mathias, *Optimization using distributed genetic algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 176–185. [Online]. Available: https://doi.org/10.1007/BFb0029750

[7] R. Bianchini and C. Brown, "Parallel genetic algorithms on distributed-memory architectures," in *Proceedings of the Sixth Conference of the North American Transputer Users Group on Transputer Research and Applications 6*, ser. NATUG-6. Amsterdam, The Netherlands, The Netherlands: IOS Press, 1993, pp. 67–82. [Online]. Available: http://dl.acm.org/citation.cfm?id=162785.162806

[8] G. Duarte, A. Lemonge, and L. Goliatt, "A new strategy to evaluate the attractiveness in a dynamic island model," in *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2018, pp. 1–8.

[9] L. A. da Silveira, J. L. Soncco-Álvarez, and M. Ayala-Rincón, "Parallel genetic algorithms with sharing of individuals for sorting unsigned genomes by reversals," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 741–748.

[10] L. A. da Silveira, J. L. Soncco-Álvarez, T. A. de Lima, and M. Ayala-Rincón, "Parallel multi-island genetic algorithm for sorting unsigned genomes by reversals," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, July 2018, pp. 1–8.

TABLE XII

AVERAGE RESULTS FOR THE EXPERIMENT WITH HUNDRED GENOMES WITH 3 CHROMOSOMES. 1= $\mathcal{GA}_{\mathrm{S}}$, 2=$\mathcal{GA}_{\mathrm{SAME12}}$, 3=$\mathcal{GA}_{\mathrm{GBMM24}}$, 4= $\mathcal{GA}_{\mathrm{RAND24}}$, 5=$\mathcal{GA}_{\mathrm{N24}}$, 6=$\mathcal{GA}_{\mathrm{R24}}$, 7=$\mathcal{GA}_{\mathrm{F24}}$, 8=$\mathcal{GA}_{\mathrm{TO24}}$, 9=$\mathcal{GA}_{\mathrm{TR24}}$.

n	1	2	3	4	5	6	7	8	9
100	63.54	63.4	**63.26**	63.41	63.32	63.27	63.28	63.31	63.31
110	71.33	70.94	**70.76**	71.0	70.84	70.82	70.8	70.83	70.85
120	78.77	78.08	**77.84**	78.13	77.97	77.88	77.91	77.94	77.95
130	87.05	85.87	**85.57**	85.91	85.71	85.66	85.66	85.71	85.72
140	94.84	93.09	**92.8**	93.2	92.95	92.87	92.85	92.91	92.91
150	102.92	100.69	**100.4**	100.85	100.55	100.46	100.48	100.53	100.52

TABLE XIII

AVERAGE RESULTS FOR THE EXPERIMENT WITH HUNDRED GENOMES WITH 4 CHROMOSOMES. 1= $\mathcal{GA}_{\mathrm{S}}$, 2=$\mathcal{GA}_{\mathrm{SAME12}}$, 3=$\mathcal{GA}_{\mathrm{GBMM24}}$, 4= $\mathcal{GA}_{\mathrm{RAND24}}$, 5=$\mathcal{GA}_{\mathrm{N24}}$, 6=$\mathcal{GA}_{\mathrm{R24}}$, 7=$\mathcal{GA}_{\mathrm{F24}}$, 8=$\mathcal{GA}_{\mathrm{TO24}}$, 9=$\mathcal{GA}_{\mathrm{TR24}}$.

n	1	2	3	4	5	6	7	8	9
100	61.23	61.15	**61.07**	61.16	61.1	61.09	61.1	61.1	61.1
110	67.93	67.76	**67.58**	67.74	67.65	67.59	67.62	67.61	67.63
120	75.24	74.9	**74.71**	74.94	74.82	74.76	**74.71**	74.75	74.74
130	81.85	81.18	**80.95**	81.2	81.05	81.02	81.0	81.02	81.03
140	89.33	88.28	**88.01**	88.36	88.13	88.08	88.06	88.11	88.1
150	96.82	95.23	**94.97**	95.28	95.07	95.03	95.01	95.07	95.05

TABLE XIV

AVERAGE RESULTS FOR THE EXPERIMENT WITH HUNDRED GENOMES WITH 5 CHROMOSOMES. 1= $\mathcal{GA}_{\mathrm{S}}$, 2=$\mathcal{GA}_{\mathrm{SAME12}}$, 3=$\mathcal{GA}_{\mathrm{GBMM24}}$, 4= $\mathcal{GA}_{\mathrm{RAND24}}$, 5=$\mathcal{GA}_{\mathrm{N24}}$, 6=$\mathcal{GA}_{\mathrm{R24}}$, 7=$\mathcal{GA}_{\mathrm{F24}}$, 8=$\mathcal{GA}_{\mathrm{TO24}}$, 9=$\mathcal{GA}_{\mathrm{TR24}}$.

n	1	2	3	4	5	6	7	8	9
100	57.73	57.68	**57.62**	57.66	57.65	57.63	57.62	57.63	57.63
110	64.74	64.69	64.57	64.67	64.61	**64.56**	64.59	64.61	64.61
120	72.08	71.89	**71.75**	71.88	71.83	71.78	71.8	71.8	71.8
130	78.32	77.96	**77.81**	77.95	77.91	77.83	77.86	77.85	77.85
140	85.42	84.81	**84.61**	84.83	84.7	84.67	84.68	84.69	84.69
150	92.32	91.05	**90.8**	91.1	90.88	90.85	90.84	90.88	90.9

[11] A. Caprara, "Sorting by reversals is difficult," in *Proc. of the first annual Int. Conf. on Computational molecular Biology*. ACM, 1997, pp. 75–83.

[12] L. A. da Silveira, J. L. Soncco-Álvarez, and M. Ayala-Rincón, "Parallel memetic genetic algorithms for sorting unsigned genomes by translocations," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 185–192.

[13] T. A. de Lima and M. Ayala-Rincón, "On the average number of reversals needed to sort signed permutations," *Discrete Applied Mathematics*, vol. 235, no. Supplement C, pp. 59 – 80, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166218X17304377

[14] D. Zhu and L. Wang, "On the complexity of unsigned translocation distance," *Theor. Comput. Sci.*, vol. 352, no. 1-3, pp. 322–328, 2006. [Online]. Available: https://doi.org/10.1016/j.tcs.2005.09.078

[15] G. Li, X. Qi, X. Wang, and B. Zhu, "A linear-time algorithm for computing translocation distance between signed genomes," in *Combinatorial Pattern Matching, 15th Annual Symposium, CPM 2004, Istanbul,Turkey, July 5-7, 2004, Proceedings*, ser. Lecture Notes in Computer Science, S. C. Sahinalp, S. Muthukrishnan, and U. Dogrusöz, Eds., vol. 3109. Springer, 2004, pp. 323–332. [Online]. Available: https://doi.org/10.1007/978-3-540-27801-6_24

[16] A. Bergeron, J. Mixtacki, and J. Stoye, "On sorting by translocations," *jcb*, vol. 13, no. 2, pp. 567–578, 2006.

[17] L. A. da Silveira, J. L. Soncco-Álvarez, T. A. de Lima, and M. Ayala-Rincón, "Computing translocation distance by a genetic algorithm," in *2015 Latin American Computing Conference (CLEI)*, Oct 2015, pp. 1–12.

[18] Y. Cui, L. Wang, D. Zhu, and X. Liu, "A (1.5+ε)-approximation algorithm for unsigned translocation distance," *ieeeacmtcbb*, vol. 5, no. 1, pp. 56–66, 2008.

[19] J. L. Soncco-Álvarez, G. M. Almeida, J. Becker, and M. Ayala-Rincón, "Parallelization and virtualization of genetic algorithms for sorting permutations by reversals," in *Nature and Biologically Inspired Computing (NaBIC), 2013 World Congress on*. IEEE, 2013, pp. 29–35.

[20] L. A. da Silveira, J. L. Soncco-Álvarez, T. A. de Lima, and M. Ayala-Rincón, "Memetic and opposition-based learning genetic algorithms for sorting unsigned genomes by translocations," in *Advances in Nature and Biologically Inspired Computing*. Springer, 2016, pp. 73–85.

[21] W. Martin, J. Lienig, and J. P. Cohoon, "C6. 3 island (migration) models: evolutionary algorithms based on punctuated equilibria," *B ack et al. BFM97], Seiten C*, vol. 6, pp. 101–124, 1997.

[22] A. Leitão, F. B. Pereira, and P. Machado, "Island models for cluster geometry optimization: how design options impact effectiveness and diversity," *Journal of Global Optimization*, vol. 63, no. 4, pp. 677–707, 2015.

[23] L. A. da Silveira, J. L. Soncco-Álvarez, and M. Ayala-Rincón, "Parallel genetic algorithms with sharing of individuals for sorting unsigned genomes by reversals," in *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 2017, pp. 741–748.

[24] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

TABLE XV
RESULTS OF THE HOLM TEST FOR ONE HUNDRED GENOMES WITH 3 CHROMOSOMES

L	Control Algorithm	i	Algorithm	Rank	P-value	α/i
100	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 3.80)	8	$\mathcal{GA}_{\text{same12}}$	6.47	5.94337E-12	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	6.27	1.79987E-10	0.00714
		6	$\mathcal{GA}_{\text{S}}$	5.64	2.15891E-6	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.92	0.00399	0.01
		4	$\mathcal{GA}_{\text{TO24}}$	4.77	0.01226	0.0125
		3	$\mathcal{GA}_{\text{TR24}}$	4.76	**0.01367**	0.01667
		2	$\mathcal{GA}_{\text{R24}}$	4.18	0.33292	0.025
		1	$\mathcal{GA}_{\text{F24}}$	4.17	0.33941	0.05
110	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 3.29)	8	$\mathcal{GA}_{\text{Rand24}}$	6.79	1.43137E-19	0.00625
		7	$\mathcal{GA}_{\text{S}}$	6.66	3.28096E-18	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	6.17	9.40233E-14	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.75	1.63441E-4	0.01
		4	$\mathcal{GA}_{\text{TR24}}$	4.72	2.11262E-4	0.0125
		3	$\mathcal{GA}_{\text{TO24}}$	4.45	0.00263	0.01667
		2	$\mathcal{GA}_{\text{R24}}$	4.24	**0.01367**	0.025
		1	$\mathcal{GA}_{\text{F24}}$	3.90	0.11230	0.05
120	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 3.08)	8	$\mathcal{GA}_{\text{S}}$	7.35	2.16998E-28	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	6.72	5.53782E-21	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	6.30	1.03157E-16	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.91	2.15891E-6	0.01
		4	$\mathcal{GA}_{\text{TR24}}$	4.56	1.25946E-4	0.0125
		3	$\mathcal{GA}_{\text{TO24}}$	4.47	3.31994E-4	0.01667
		2	$\mathcal{GA}_{\text{F24}}$	3.95	**0.02387**	0.025
		1	$\mathcal{GA}_{\text{R24}}$	3.64	0.14820	0.05
130	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 2.64)	8	$\mathcal{GA}_{\text{S}}$	7.96	6.16427E-43	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	6.91	3.33817E-28	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	6.39	4.06166E-22	0.00833
		5	$\mathcal{GA}_{\text{TR24}}$	4.57	5.84866E-7	0.01
		4	$\mathcal{GA}_{\text{N24}}$	4.55	7.63361E-7	0.0125
		3	$\mathcal{GA}_{\text{TO24}}$	4.42	4.04902E-6	0.01667
		2	$\mathcal{GA}_{\text{F24}}$	3.80	0.00263	0.025
		1	$\mathcal{GA}_{\text{R24}}$	3.75	**0.00416**	0.05
140	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 2.79)	8	$\mathcal{GA}_{\text{S}}$	8.69	1.73314E-52	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	7.15	2.46317E-29	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	6.17	2.61234E-18	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.68	9.93766E-7	0.01
		4	$\mathcal{GA}_{\text{TR24}}$	4.25	1.55190E-4	0.0125
		3	$\mathcal{GA}_{\text{TO24}}$	4.12	**5.66943E-4**	0.01667
		2	$\mathcal{GA}_{\text{R24}}$	3.64	0.02727	0.025
		1	$\mathcal{GA}_{\text{F24}}$	3.48	0.07070	0.05
150	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 2.78)	8	$\mathcal{GA}_{\text{S}}$	8.8	3.93283E-54	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	7.29	2.09678E-31	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	5.99	1.03157E-16	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.52	6.22850E-6	0.01
		4	$\mathcal{GA}_{\text{TO24}}$	4.39	3.04717E-5	0.0125
		3	$\mathcal{GA}_{\text{TR24}}$	4.11	5.40446E-4	0.01667
		2	$\mathcal{GA}_{\text{F24}}$	3.64	**0.02468**	0.025
		1	$\mathcal{GA}_{\text{R24}}$	3.49	0.06302	0.05

TABLE XVI
RESULTS OF THE HOLM TEST FOR ONE HUNDRED GENOMES WITH 4 CHROMOSOMES

L	Control Algorithm	i	Algorithm	Rank	P-value	α/i
100	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 4.47)	8	$\mathcal{GA}_{\text{Rand24}}$	5.69	**0.00163**	0.00625
		7	$\mathcal{GA}_{\text{S}}$	5.46	0.01019	0.00714
		6	sameA	5.40	0.01634	0.00833
		5	fullB	4.87& 0.29570	0.01	
		4	$\mathcal{GA}_{\text{TO24}}$	4.85	0.32652	0.0125
		3	$\mathcal{GA}_{\text{TR24}}$	4.83	0.34597	0.01667
		2	$\mathcal{GA}_{\text{N24}}$	4.80	0.39418	0.025
		1	$\mathcal{GA}_{\text{R24}}$	4.61	0.70812	0.05
110	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 4.10)	8	$\mathcal{GA}_{\text{same12}}$	6.23	3.53679E-8	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	6.14	1.38477E-7	0.00714
		6	$\mathcal{GA}_{\text{S}}$	5.89	**3.57507E-6**	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.89	0.04013	0.01
		4	$\mathcal{GA}_{\text{TR24}}$	4.65	0.15186	0.0125
		3	$\mathcal{GA}_{\text{TO24}}$	4.48	0.32019	0.01667
		2	$\mathcal{GA}_{\text{F24}}$	4.37	0.47768	0.025
		1	$\mathcal{GA}_{\text{R24}}$	4.23	0.75668	0.05
120	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 3.83)	8	$\mathcal{GA}_{\text{S}}$	6.64	4.00510E-13	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	6.56	2.17139E-12	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	6.04	1.24618E-8	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	5.04	**0.00195**	0.01
		4	$\mathcal{GA}_{\text{R24}}$	4.47	0.09844	0.0125
		3	$\mathcal{GA}_{\text{TO24}}$	4.26	0.26689	0.01667
		2	$\mathcal{GA}_{\text{TR24}}$	4.20	0.33941	0.025
		1	$\mathcal{GA}_{\text{F24}}$	3.94	0.77640	0.05
130	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 3.56)	8	$\mathcal{GA}_{\text{S}}$	7.21	4.33151E-21	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	6.33	8.54545E-13	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	5.94	7.98963E-10	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.75	**0.00212**	0.01
		4	$\mathcal{GA}_{\text{R24}}$	4.40	0.03009	0.0125
		3	$\mathcal{GA}_{\text{TR24}}$	4.395	0.03109	0.01667
		2	$\mathcal{GA}_{\text{TO24}}$	4.33	0.04539	0.025
		1	$\mathcal{GA}_{\text{F24}}$	4.08	0.17939	0.05
140	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 3.24)	8	$\mathcal{GA}_{\text{S}}$	8.08	7.76914E-36	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	6.61	3.67603E-18	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	6.02	8.54545E-13	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.61	4.04202E-4	0.01
		4	$\mathcal{GA}_{\text{TO24}}$	4.31	0.00596	0.0125
		3	$\mathcal{GA}_{\text{TR24}}$	4.23	**0.01098**	0.01667
		2	$\mathcal{GA}_{\text{R24}}$	4.04	0.04010	0.025
		1	$\mathcal{GA}_{\text{F24}}$	3.85	0.11826	0.05
150	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 3.38)	8	$\mathcal{GA}_{\text{S}}$	8.37	5.52994E-38	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	6.18	4.84530E-13	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	5.91	7.05318E-11	0.00833
		5	$\mathcal{GA}_{\text{TO24}}$	4.48	0.00433	0.01
		4	$\mathcal{GA}_{\text{N24}}$	4.40	0.00845	0.0125
		3	$\mathcal{GA}_{\text{TR24}}$	4.31	**0.01634**	0.01667
		2	$\mathcal{GA}_{\text{R24}}$	4.06	0.07695	0.025
		1	$\mathcal{GA}_{\text{F24}}$	3.90	0.17524	0.05

TABLE XVII
RESULTS OF THE HOLM TEST FOR ONE HUNDRED GENOMES WITH 5 CHROMOSOMES

L	Control Algorithm	i	Algorithm	Rank	P-value	α/i
110	$\mathcal{GA}_{\text{R24}}$ (Rank: 4.34)	8	$\mathcal{GA}_{\text{same12}}$	5.79	1.81197E-4	0.00625
		7	$\mathcal{GA}_{\text{S}}$	5.76	2.45966E-4	0.00714
		6	$\mathcal{GA}_{\text{Rand24}}$	5.39	**0.00671**	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.88	0.16323	0.01
		4	$\mathcal{GA}_{\text{TR24}}$	4.86	0.18361	0.0125
		3	$\mathcal{GA}_{\text{TO24}}$	4.82	0.21521	0.01667
		2	$\mathcal{GA}_{\text{F24}}$	4.67	0.40139	0.025
		1	$\mathcal{GA}_{\text{gbmm24}}$	4.46	0.76652	0.05
120	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 4.05)	8	$\mathcal{GA}_{\text{S}}$	6.80	1.24334E-12	0.00625
		7	$\mathcal{GA}_{\text{same12}}$	5.52	1.39850E-4	0.00714
		6	$\mathcal{GA}_{\text{Rand24}}$	5.50	**1.72104E-4**	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.98	0.01577	0.01
		4	$\mathcal{GA}_{\text{F24}}$	4.63	0.13093	0.0125
		3	$\mathcal{GA}_{\text{TR24}}$	4.55	0.19671	0.01667
		2	$\mathcal{GA}_{\text{TO24}}$	4.54	0.20581	0.025
		1	$\mathcal{GA}_{\text{R24}}$	4.41	0.35262	0.05
130	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 4.09)	8	$\mathcal{GA}_{\text{S}}$	6.31	9.92443E-9	0.00625
		7	$\mathcal{GA}_{\text{same12}}$	5.63	7.00067E-5	0.00714
		6	$\mathcal{GA}_{\text{Rand24}}$	5.48	3.15931E-4	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	5.20	**0.00416**	0.01
		4	$\mathcal{GA}_{\text{F24}}$	4.73	0.09844	0.0125
		3	$\mathcal{GA}_{\text{TR24}}$	4.59	0.19227	0.01667
		2	$\mathcal{GA}_{\text{TO24}}$	4.57	0.21521	0.025
		1	$\mathcal{GA}_{\text{R24}}$	4.39	0.43858	0.05
140	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 3.75)	8	$\mathcal{GA}_{\text{S}}$	7.07	1.01524E-17	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	5.86	4.73709E-8	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	5.73	**3.18196E-7**	0.00833
		5	$\mathcal{GA}_{\text{N24}}$	4.64	0.02084	0.01
		4	$\mathcal{GA}_{\text{TO24}}$	4.50	0.03887	0.0125
		3	$\mathcal{GA}_{\text{TR24}}$	4.50	0.05281	0.01667
		2	$\mathcal{GA}_{\text{F24}}$	4.48	0.05773	0.025
		1	$\mathcal{GA}_{\text{R24}}$	4.40	0.09080	0.05
150	$\mathcal{GA}_{\text{gbmm24}}$ (Rank: 3.51)	8	$\mathcal{GA}_{\text{S}}$	7.97	9.46922E-31	0.00625
		7	$\mathcal{GA}_{\text{Rand24}}$	6.33	3.30842E-13	0.00714
		6	$\mathcal{GA}_{\text{same12}}$	5.81	2.87511E-9	0.00833
		5	$\mathcal{GA}_{\text{TR24}}$	4.54	**0.00783**	0.01
		4	$\mathcal{GA}_{\text{TO24}}$	4.38	0.02468	0.0125
		3	$\mathcal{GA}_{\text{N24}}$	4.32	0.03649	0.01667
		2	$\mathcal{GA}_{\text{R24}}$	4.11	0.11826	0.025
		1	$\mathcal{GA}_{\text{F24}}$	4.02	0.18790	0.05