Migration study on a Pareto-based island model for multi-objective ant colony optimization algorithms

A.M. Mora, P. García-Sánchez, J.J. Merelo, P.A. Castillo

Abstract— **Pareto-based island model is a multi-colony distribution scheme recently presented for the resolution, by means of ant colony optimization algorithms, of bi-criteria problems. It yielded very promising results, but the model was implemented considering a unique Pareto-front-shaped unidirectional neighborhood migration topology, and a constant migration rate. In the present work two additional neighborhood topology schemes, and four different migration rates have been tested, considering the algorithm which obtained the best results in average in the model presentation article: MOACS (Multi-Objective Ant Colony System). Several experiments have been conducted, including statistical tests for better support the study. High values for the migration rate and the use of a bidirectional neighborhood migration topology yields the best results.**

I. Introduction

Island model is one of the most extended distribution paradigms, since it arose in the Evolutionary Algorithms (EA) scope [1]. It proposed the division of the whole population into different subpopulations called islands. They evolve independently a number of generations and then, some individuals (normally the best) migrate between them. Thus, the optimization process is improved due to the parallel exploration of different areas of the search space that every island is focused on, and in addition, thanks to the inclusion of different promising individuals in a population, which guides the search and probably avoids local optima stagnation. Moreover, the distribution means an inherent running time improvement. There are several factors to consider in an island model, such as the migration rate (number of times the individuals are moved between islands), the neighborhood topology (island connections) or the replacement policy (which individuals are removed of the receiver population to include the migrants).

This model was initially proposed for the resolution of problems with just one objective, but it was later adapted to solve multi-objective problems [2]. Multi-objective optimization problems (MOPs) are those where more than one independent function must be optimized at once, every function is an objective and the improvement of one does not mean an improvement in the rest. There is a set of the so-called non-dominated solutions (those which better satisfy each of the functions) rather than just one, which constitutes the Pareto set (PS). The graphical representation of this Pareto-optimal set of solutions is named the Pareto front (PF). Thus, the main issue of a good multi-objective

Department of Architecture and Computer Technology, ETSIIT, CITIC, University of Granada, Spain, {amorag,pgarcia,jmerelo,pedro}@geneura.ugr.es

(MO) algorithm [3] is to obtain the maximum number of non-dominated solutions, having a high diversity and spread, and being as closer to the optimal set as possible.

In addition to EAs, Ant colony optimization (ACO) is another metaheuristic well-suited for distribution. ACO [4] algorithms are based in the behaviour of natural ants when searching for food, and the way they build shortest paths between their nest and any interesting point. This behaviour has been explained through the concept of *stigmergy*, that is, communication using the environment, so every ant, while walking, deposits on the ground a chemical substance called *pheromone* which the others can smell (and which is evaporated after some time). One ant tends to follow strong concentrations of pheromone caused by repeated passes of ants; a pheromone trail is then formed from nest to food source, so in intersections between several trails an ant moves with high probability following the highest pheromone level.

Ants are modelled as a set of artificial agents which search in a weighted graph for the optimal paths. These agents cooperate to explore the search space and to build the solution for a problem. The ACOs adapted to the resolution of problems with several objectives are known as multi-objective ant colony optimization algorithms or MOACOs.

As stated, ACO algorithms can be directly parallelized due to the inherent independent work of the ants. Thus, several parallel ACO approaches have been proposed since the metaheuristic was presented [5], most of them devoted to address specific problems.

There are two main distribution levels: coarse-grained (parallelization at colony level) [6], and fine-grained (parallelization at ant level) [7]. The first approach is the most extended and is known as multi-colony model, since the ants are divided in sub-colonies which can operate independently from the rest [8] or can share any information (pheromone trails for instance) [9].

In this line, there was proposed in a recent paper a multi-colony distribution model named *Pareto-based island model* [10], which spread the ants, grouped in colonies, into several computational nodes (processors in clusters) or islands. Everyone focuses on a different area of the search space, promoting the exploration and finally yielding a better set of results than a mono-colony standard approach. Thus, a bigger amount of non-dominated solutions, with higher quality (closer to the optimal values), and with a better spread or distribution over the ideal set of solutions were obtained. To this end, some migrations were performed between the colonies (moving ants/solutions), increasing the explorative component which turns on getting a better performance. This

model defined a neighborhood topology based on the Pareto front form and was compared with two other methods based in independent *multi-colony* structures. The proposed island model, based in the Pareto front, obtained excellent results in the comparison with the other models (multi-start and independent multi-colonies). All of them were implemented as substrates for three different MOACOs from the literature: Iredi's et al. *BIANT* [11], Barán's et al. *MOACS* [12]; and an algorithm named *CHAC* [13]. MOACS and CHAC yielded the best results, very similar between them. But since it was an initial proof of concept, the island model just implemented a neighborhood topology (unidirectional) and was tested with just a migration rate value (every 50 iterations).

There are just a few papers studying island models in ACOs scope [14], [15], but to our best knowledge, just the works [16], [10] involve the distribution of MOACOs following an island-model shape.

Thus in the present work a deeper analysis of the Pareto-based island model is proposed. Three different neighborhood topologies will be tested: *unidirectional*, *bidirectional* and *broadcast*; along with four different migration rates: every 25, 50, 75 and 100 iterations. Due to space limitations, just MOACS algorithm will be considered, according to the good performance it presented in previous papers. The results, including different metrics, obtained with every configuration solving the *Symmetric Bi-criteria Travelling Salesman Problem* (bTSP), will be analyzed by means of statistical tools, such as the Analysis of variance (ANOVA) [17] and post-hoc Tukey's Honestly Significant Difference tests (Tukey HSD) [18].

II. MULTI-OBJECTIVE OPTIMIZATION

Multi-Criteria or Multi-Objective Optimization Problems (MOP) [2] are those where several objectives have to be simultaneously optimized. These problems do not have a single best solution, one that is better than any other with respect to every objective. In fact, frequently improving the solution for one objective implies a worsening for another. Thus in a MOP there is a set of solutions that are better than the rest considering all the objectives; this set is known as the *Pareto set* (PS). They try to approach the ideal set of solutions which is usually represented graphically as the *Pareto front* (PF). These are related to an important concept, the *dominance*, defined as follows (*a* dominates *b*):

$$a \prec b \ \ if :$$
$$\forall i \in 1, 2, ..., k \mid C_i(a) \leq C_i(b) \ \land \ \exists j \in 1, 2, ...k \mid C_j(a) < C_j(b) \tag{1}$$

where $a \in X$ and $b \in X$ are two different decision vectors of n values, and every C is a cost function (one per objective). If it intends to minimize the cost and Equation 1 is true, then *b is dominated by a*.

Hence, the solutions in the PS are known as *non-dominated solutions*, while the remainder are known as dominated solutions. Since none of the solutions in the PS is absolutely better than the other non-dominated solutions, all of them are equally acceptable as regards the satisfaction of all the objectives. These concepts can be extended in [3].

III. MOACS ALGORITHM

The algorithm to study is the well-known *MOACS* (Multi-Objective Ant Colony System), which was proposed by Barán et al. [12], to solve the Vehicle Routing Problem with Time Windows (VRPTW), a bi-objective problem. The good results it yielded in the article [10], where the Pareto-based island model was presented and tested, is the reason why it has been chosen as the algorithm to study in the present work.

As main features it considers a single pheromone matrix for both objectives, and follows a State Transition Rule (STR), for building the solutions, described as:

If $(q \leq q_0)$
$$j = \arg\max_{j \in N_i} \left\{ \tau(i,j) \cdot \eta_1(i,j)^{\beta \cdot \lambda} \cdot \eta_2(i,j)^{\beta \cdot (1-\lambda)} \right\} \tag{2}$$

Else
$$P(i,j) = \begin{cases} \dfrac{\tau(i,j) \cdot \eta_1(i,j)^{\beta \cdot \lambda} \cdot \eta_2(i,j)^{\beta \cdot (1-\lambda)}}{\displaystyle\sum_{u \in N_i} \tau(i,u) \cdot \eta_1(i,u)^{\beta \cdot \lambda} \cdot \eta_2(i,u)^{\beta \cdot (1-\lambda)}} & if \ j \in N_i \\ \\ 0 & otherwise \end{cases} \tag{3}$$

where τ is the pheromone value of the arc (between the nodes/cities), η_1 and η_2 are the heuristic functions related to each objective, β is the weighting parameter to set the relative importance of the heuristic information, and N_i is the current feasible neighborhood for the node i. This rule applies parameter $\lambda \in [0,1]$ for balancing the priority of the objectives in the search.

In the initial proposal of MOACS, λ parameter has a value for every ant, varying from 0 to 1. In this work, as will be commented in the next section, λ has been used to determine the area of the search space that each sub-colony has to explore, so it will be constant for all the ants in a colony (and different to the rest of the colonies).

This STR works as follows: when an ant is building a solution path and is placed at one node i, a random number q in the range [0,1] is generated, if $q \leq q_0$ the best neighbor j is selected as the next node in the path (Equation 2). Otherwise, the algorithm decides which node is the next by using a roulette wheel considering $P(i,j)$ as probability for every feasible neighbor j (Equation 3).

Since *MOACS* is an Ant Colony System (ACS), there are two levels of pheromone updating, local and global. The equation for *local pheromone updating* is:
$$\tau(i,j)^t = (1 - \rho) \cdot \tau(i,j)^{t-1} + \rho \cdot \tau_0 \tag{4}$$

considering:
$$\tau_0 = \frac{1}{L_1 \cdot L_2} \tag{5}$$

ρ is the evaporation factor, t marks the new pheromone value and $t-1$ the preceding one. L_1 and L_2 are the cost obtained in each objective using a Greedy algorithm for solving the problem.

MOACS applies a *pheromone reinitialization mechanism*, so τ_0 value is not constant during the algorithm run, as usual

in ACS, but it undergoes adaptation. Every time an ant h builds a complete solution, it is compared to the Pareto set P generated until the moment, to check if the former is a non-dominated solution. At the end of each iteration, τ_0' is calculated following the formula:

$$\tau_0' = \frac{1}{\bar{F}_1 \cdot \bar{F}_2} \tag{6}$$

where $\bar{F}_1$ and $\bar{F}_2$ are respectively the average costs in each objective for the solution paths currently included in the Pareto set.

Then, if $\tau_0' > \tau_0$ (the current initial pheromone value), a better PS has been found, and the pheromone trails are reinitialized considering the new value for $\tau_0 \leftarrow \tau_0'$.

Otherwise the *global pheromone updating* is performed for every solution in the PS:

$$\tau(i,j)^t = (1 - \rho) \cdot \tau(i,j)^{t-1} + \frac{\rho}{F_1 \cdot F_2} \tag{7}$$

where, as in previous equations, F_1 and F_2 are the evaluation functions for each objective, ρ is the common evaporation factor; and t and $t - 1$ stands for actual and previous pheromone values.

MOACS has been adapted to solve the symmetric bi-criteria TSP (bTSP), defined as: given an undirected weighted graph $G(N, A)$, with a set of nodes (cities) N, and arcs (links) A joining them, having associated two different costs per link; find the permutation of cities (solution path), composing a Hamiltonian circuit (a set of closed tours visiting each of the nodes exactly once), which minimizes both the global sum of each cost.

Thus the *heuristic functions* (one per objective) are described as:

$$\eta_1(i,j) = \frac{1}{d_1(i,j)} \tag{8}$$

$$\eta_2(i,j) = \frac{1}{d_2(i,j)} \tag{9}$$

where $d_1(i,j)$ and $d_2(i,j)$ are the correspondent Euclidean distances between nodes (cities) i and j in each objective.

Finally, this approach has been improved by means of a *local search* (LS) method: the 2-OPT algorithm [19], since it has been widely proved in the literature that this hybridization mechanism yields a better performance both in sequential and parallel ACOs [20].

IV. PARETO-BASED ISLAND MODEL

Following the conclusions reached in previous work [10], the sub-colony and multi-colony models presented the problem of having 'gaps' of solutions between the colonies. In order to address it, the authors proposed an *island model* which includes communication between colonies aiming to yield some extra solutions in the less explored areas.

The sub-colonies in this model are called *islands*. In addition we refer to migrant ants as an analogy to migrant individuals in distributed GAs, but what is actually migrated is a solution itself, i.e., in the case of the bTSP, a list of cities which compose a valid Hamiltonian cycle.

A. Model factors

There are some components to be set in the island model.

- *Migration policy* (which ants are migrated?): following the conclusions reached by several authors [21], [20] the circulation of the *locally best solutions* to the neighbor island yields better results than other possibilities, so this is the policy we have considered. But since we are dealing with a multi-objective problem, there is not an absolutely best solution, but a set of non-dominated ones. Thus the migrant will be *the best solution with respect to the objective with higher priority in the island*. This policy along with the neighboring topology tries to guide the search in the neighbors to the same area than the island which sends the migrants (explore new areas in the receiver).
- *Influence of Migrants* (what effect have the migrants in the receiver island?): The ant/solution firstly performs a pheromone contribution (in the matrix of the receiver colony) to those edges of the solution according the correspondent equation depending on the algorithm, i.e. Equation 7 in MOACS. This contribution tries to guide (in part) the search in the following iterations. Moreover, the changes in the PS (if the immigrant solution is included) will also globally update the pheromone matrix in the future steps of the process.
- *Replacement policy* (which ants/solutions are replaced in the receiver island?): Considering the multi-objective scope, the immigrant ant/solution is compared with those in the current PS, removing from it those dominated by the new one.
- *Migration rate* (how often are the migrants sent?): According to previous researches [21], [20], it is important to set a balance between the communication costs that the migrations imply and the profit these migrations give. In this work four different values have been set, from 25 to 100 (from 2.5% to 10% of the total number of iterations). Thus 40 to 10 migration events will be performed during the execution, so the colonies have enough time to increase the exploration factor being influenced by the new solutions, and the costs in communications are not critically increased for maintaining a good computational performance.

B. Model topologies

The islands are distributed along the PF, i.e. everyone searches in a different area of the front. They have been placed considering a value for λ, which weights the objectives in the state transition rule (STR) of the algorithm which implements the model. Thus, when $\lambda > 0.5$, the objective function 1 (F1) has a bigger weight in the search, and the other way round when $\lambda < 0.5$. The equation which defines the value for this parameter, considered in an island i (for N_i islands) is:

$$\lambda_i = 1 - \frac{i - 1}{N_i - 1} \tag{10}$$

Then *neighborhood topology* is defined in order to cover the inter-colony 'gaps'. There are three different models, all of them based in the Pareto front shape:

- *Unidirectional topology*: Every island migrates ants (solutions) just to the closer neighbor and just in one direction (aimed to the optimal). This direction depends on the objective which has higher priority in the island optimization process, so the islands devoted to mainly optimize objective function 1 ($\lambda > 0.5$), migrate the ants to the immediate left neighbors. On the other hand, the islands optimizing objective function 2 ($\lambda < 0.5$) share the solutions with the immediate right neighbors. The two central islands (on the left and right sides of $\lambda = 0.5$ value) interchange ants between them. If there is an island with $\lambda = 0.5$, it receives ants from two sides neighbors and shares its own solutions with them. In addition, the bounding islands just send solutions and do not receive at all, since the gaps to cover appear between the colonies.

- *Bidirectional topology*: Every island migrates ants to its two closest neighbors, one on the right and one on the left, and receives migrants from them. This model tries to improve the exploration in the gaps between islands, i.e. the ants in one island lightly move to search in areas closer to the neighbors rather than their own colony zone. The objective is getting a better spread in the inter-island areas. As a disadvantage this will increase the communication costs (in time) in the algorithm run.

- *Broadcast topology*: Every island sends migrants to the rest. This would increase the exploration along the whole PF and thus, the global spread of solutions. On the other hand, this will mean a decrease in the exploitation factor of the colonies. The communication cost is higher than in the other models, since all the colonies send and receive ants in every migration event.

The ideal scheme of these proposals, when minimizing the objectives, can be seen in Figure 1.

V. EXPERIMENTS AND RESULTS

In order to test the proposed model, with the different neighborhood migration topologies and migration rates, several experiments have been conducted using three different instances of the symmetric TSP[1], in sizes 100, 150 and 200 cities. $kroA$ and $kroB$ instances have been taken in every case, considering each of them as one independent objective of a bi-criteria TSP (bTSP). These problems, named respectively *kroAB100*, *kroAB150* and *kroAB200*, have been widely used in the literature. In this case, due to the huge amount of data and figures generated, and considering the space limitations, we will analyze just the results for the hardest one, kroAB200, being the others equivalent, so the conclusions reached for this instance can be extended to the others.

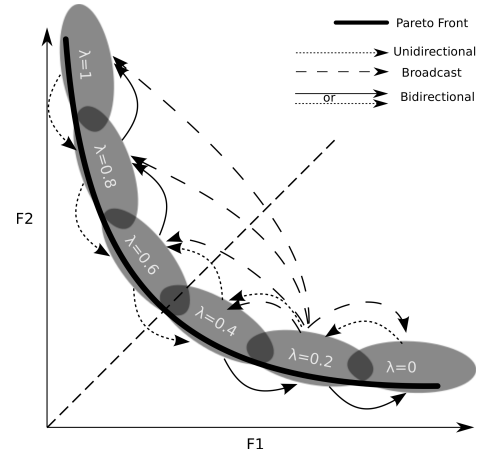[1]Available from `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`

Fig. 1. Pareto-based island model schemes examples for six islands. Each neighborhood migration topology aims to explore the whole Pareto front. The arrows show the migration flows between islands, in order to cover the gaps between colonies/islands. For the sake of clarity broadcast scheme just presents the migrations from one of the islands (to the others), but the model migrates individuals from every island to the rest. The dashed line represents the value $\lambda = 0.5$.

The twelve approaches (MOACS, topology + migration rate) have been run in a cluster platform using 16 processors, being each of them a 2.6 GHz AMD Opteron (AMD64e) with 2GB of memory running Linux. The cluster has a shared-memory architecture which allows running large threaded jobs (e.g. OpenMP) as well as message-passing jobs. The parallelization has been implemented using the Message Passing Interface (MPI), not using the shared memory but the own memory of every processor.

Table I shows the parameter setting for MOACS algorithm and the variations in neighborhood topology and migration rate. They have been set starting from standard values and performing a systematic experimentation process for fine-tuning them.

TABLE I
PARAMETER VALUES USED IN MOACS ALGORITHM (IN EVERY ISLAND).

Num. iterations	1000
Num. ants	4 per island
β (heuristic weight)	2
q_0 (pseudo-proportional factor)	0.8
ρ (global evaporation rate)	0.2
φ (local evaporation rate)	0.1
Num. iterations in Local Search (2-OPT)	15
Migration topology	unidirectional bidirectional broadcast
Migration rate	25, 50, 75, 100

The number of iterations and ants per island are the same considered in the previous work. It should be noticed that the value for q_0 is smaller than usual in ACSs (normally very close to 1), in order to choose in the STR paths which are not the most promising (in principle) more than

usual. Moreover, the global evaporation rate (ρ) takes a value doubled considering most works, in order to get a faster evaporation of the ant's trails. The reason in both cases is the aim of having a higher exploration component.

The first study makes a comparison between the different PSs obtained by each configuration. To this end the *median attainment surfaces*[22] have been obtained for every approach, considering the solutions of 20 runs (PSs) for every approach. Each attainment surface graphically shows the distribution of the solutions, summarizing in a graph the outcomes of several runs, by dividing the space of solutions in two parts: the attained solutions (in a percentage) and those not attained. A median attainment surface plots the graph which limits a region in the space attained by the 50% of the runs. We have used the software implemented by Knowles[2] considering resolution 100.

The graphs[3] for each neighborhood migration topology, using every migration rate are plotted in Figure 2. In it, it can be seen that all the attainment functions are similar (very close). Thus we cannot extract relevant conclusions from these graphs so, in principle, almost all the migration rates could be a good option. Maybe it could be considered as the best those yielded for the migration rate 100 in every migration topology model (as we will probe in the metrics results). Having this in mind, a comparison between the best attainment function graphs is shown in Figure 3.
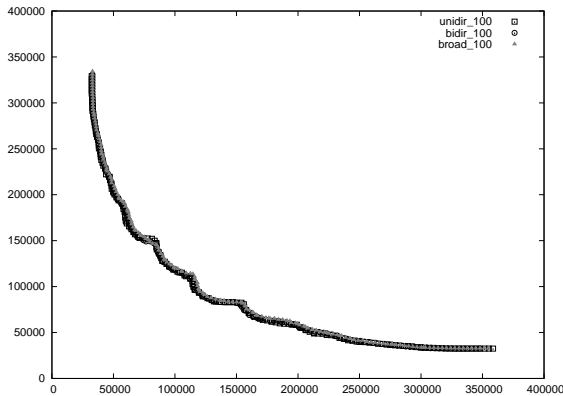

Fig. 3. Attainment Surfaces of the best results on each topology approach for MOACS in kroAB200 problem.

Looking at this figure, we can conclude that *broadcast* model is the worse in this sense, due to the higher exploration factor which means a lower exploitation component than the others, and thus, worse quality in the set of solutions. In the comparison between *unidirectional* and *bidirectional*, the latter performs better in some specific areas (mainly in the gaps between islands), but the first obtains a wider set of solutions in the bounds and better quality ones in the center of the island's search areas (it has a higher exploitation component).

In order to do a fair comparison, some multi-objective metrics and indicators have been computed for every approach, considering diversity, distribution, convergence and general quality. They are:

- *Hypervolume* ($HV \in [0, 1]$): it calculates the volume, in the objective space, covered by a set of non-dominated solutions (PS). A higher value means a better result, being 1 the maximum.
- *Spread* ($Spr \in [0, 1]$): it measures the extent of spread of a set of non-dominated solutions. It considers the Euclidean distance between consecutive solutions on average and extreme distances. A value 0 means an ideal spread.
- *Epsilon indicator* (ϵ): it is a measure of the smallest distance it would be necessary to translate every solution in a PS so that it dominates the optimal PF of the problem. Smaller values are better.
- *Cardinality of the Pareto set* ($|PS|$): number of non-dominated solutions in the obtained PS.

We have used $jMetal$ software [23] [4] to compute them. As a clarification point, a pseudo-optimal Pareto set has been created per each problem, since some of the applied metrics and indicators require an optimal Pareto set to be computed. This, named PS_G, has been built by combining all the solutions yielded by all the approaches in the 20 runs; then the dominated solutions have been removed.

The results are presented in the boxplots in Figure 4. The HV results in Figure 4.(a) show that *unidirectional* solutions have better quality than the solutions obtained by the other topologies, due to its higher exploitation component which means more optimal solutions are yielded. *Bidirectional* results are similar but worse, because it has a higher exploration factor, and *broadcast* has obtained the worst, since it is the most explorative method. With respect to the *migration rates*, 100 seems to yield the best results. It is not surprising considering that a higher migration rate means less migration events are performed and thus, less ants migrate and influence in the search of the other islands, i.e., the exploration factor is decreased. So the higher the migration rate is, the higher the exploitation component is in every island, and the better the solutions are achieved.

Figure 4.(b) shows this time the contrary situation, being *broadcast* the best option with regard to the spread/distribution (Spr) of solutions, i.e. to the wide of the Pareto set. The reason is the same as before but on the contrary sense. Thus the higher the exploration component is, the wider the PS is. As expected, the lowest value for the *migration rate* yields the best results for this indicator (closer to 0).

ϵ metric results (in Figure 4.(c)) show that the best convergence factor is reached by the *bidirectional* approach, and for high values of the *migration rates* (75 and 100). The reason lies in the higher exploitation performed in this model in every island, since the migrants are solutions
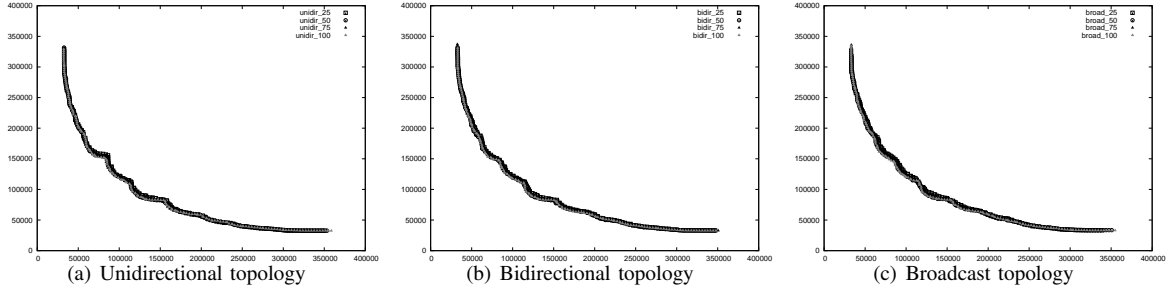
Fig. 2. Attainment Surfaces of each topology approach for MOACS in kroAB200 problem. Each graph shows the results for every migration rate.

from close search areas. High values of the *migration rate* lead to less migration events, so the algorithm can perform more iterations to improve the solutions in an area, i.e. the exploitation grows and the obtained solutions are closer to the optimal ones.

With respect to $|PS|$ (cardinality), plotted in Figure 4.(d), high values for *migration rate* correspond to higher number of solutions in the PS. *unidirectional* model seems to show the best performance in this sense. Thus, the exploitation is more desirable in this problem, because more non-dominated solutions are obtained in the specific search area of every island, than close to the neighbors (when more migrants arrive to an island).

The final boxplot (Figure 5) considers the distributed nature of the model, and shows the running time results. The worst time among all the processors in one execution has been chosen as the running time of that execution.

Looking at this figure, it can be noticed that *bidirectional* model consumes the highest computational time. These results may seem not expected, since, in principle, there should be more communication costs in the broadcast model. The main reason lies in the algorithm implementation, since the PS updating is quite expensive. Thus in *bidirectional* model there are several updates due to found solutions in the inter-colony gaps, because of the exploitation factor increasing that the migrations in this model provide (close solutions in the same area are received). This along with a higher communication cost than *unidirectional* model leads to these results. In the *broadcast* case, the communication costs are higher, but there are many fewer updates in the PS, since it finds less non-dominated solutions than the other models. This conclusion is supported by the similar results obtained for the different *migration rates*, which should increase the running time consumption when these rates decrease, however it does not happen. Thus communication cost is not the main factor in these results. Anyway the differences are around 3 seconds, so they are not significative from the user point of view.

Finally, ANOVA statistical tests [17] have been applied to complete this analysis. These tests allow us to determine whether a change in the results of every metric is due to a change in a factor or to a random effect. In this case, the factors to consider are the possible approaches (topology + migration rate). After applying ANOVA, if the test has

shown significative differences, post-hoc Tukey's Honestly Significant Difference (HSD) test [18] has been applied to find which factors are causing these differences.

The ANOVA results are shown in Table II. It can be noticed that the approaches (topologies + migration rates) have significative influence on the values of every metric or indicator, since the *P-value (Sig. Level)* is always much smaller than 0.05 (is the standard minimum value 2.2e-16). Moreover, they have significative influence at a confidence level of 99%. However in the case of $Time$ no influence has been found for *migration rate* factor.

TABLE II
ANOVA test results for the metrics and indicators values.
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05.

	Param.	Df	F	P (Sig. Level)
	topology	2	327.219	< **2.20e-16 *****
HV	*mig. rate*	1	272.867	< **2.20e-16 *****
	Residuals	234	0.0016073	0.00000687
	topology	2	51.1573	< **2.20e-16 *****
ϵ	*mig. rate*	1	32.4955	**3.58e-08 *****
	Residuals	234	453518905	1938115
	topology	2	104.750	< **2.20e-16 *****
Spr	*mig. rate*	1	57.4511	**8.06e-13 *****
	Residuals	234	0.36810	0.001573
	topology	2	77.3939	< **2.20e-16 *****
$|PS|$	*mig. rate*	1	28.1965	**2.54e-07 *****
	Residuals	234	35762	152.8
	topology	2	873.522	< **2.20e-16 *****
$Time$	*mig. rate*	1	0.1675	**0.6828**
	Residuals	234	43473869	185786

Given this conclusion, post-hoc Tukey's HSD test can be used. It is a versatile, easily calculated technique that allows determining exactly where the significant differences are, once ANOVA has found a significant effect. This test compares pairs of the factor values, i.e. topology on one side, and migration rate on the other, showing a segment (confidence level) for each comparison. Additionally, there is a vertical dotted line (distance equals zero) that intersects some segments. Significant differences can be found in those cases where the vertical line does not intersect a segment (those values compared are significantly different). Tukey's HSD results can be seen in Figure 6.
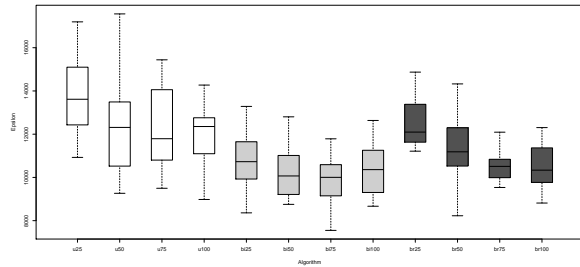
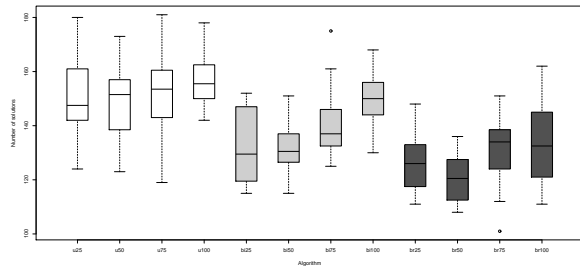The first conclusion that can be extracted from these graphs is that, in most of cases, there are no significative

(a) HV

(b) Spr

(c) ϵ
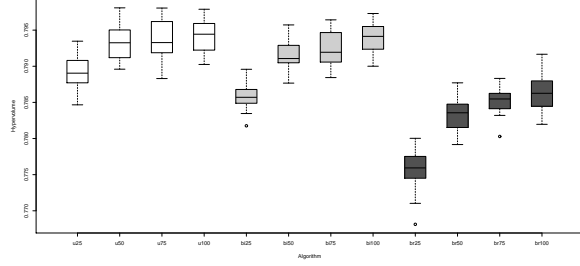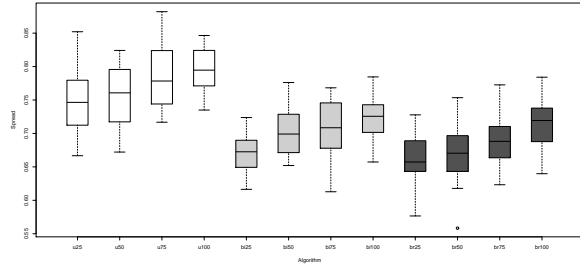
(d) $|PS|$

Fig. 4. Boxplots showing the metrics and indicators values of the 20 PSs (one per run) for the kroAB200 instance of the problem. The topology models are plotted in different shades of gray. Each one of the approaches (topology + migration rate) is labelled with its two initial letters and the migration rate value, from unidirectional-25 to broadcast-100.
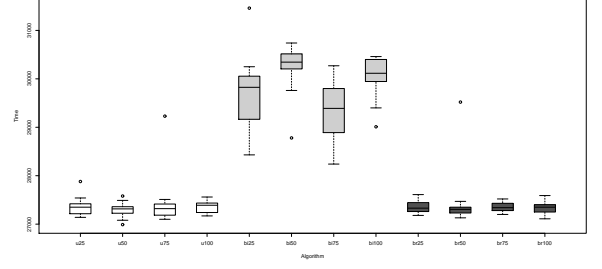

Fig. 5. Boxplots showing the running time (in milliseconds) of the 20 runs for the kroAB200 instance of the problem. The colors and labels are the same as in previous figure.

differences between the results when using close *migration rates*, and just when the difference in rate is more than 50, the differences arise. With respect the *migration topology*, the differences are always significatives, but in the comparison between *unidirectional* and *bidirectional* in HV metric, between *broadcast* and *bidirectional* in Spr metric, and between *unidirectional* and *broadcast* in $Time$ measure. These facts could be also inferred looking respectively at Figures 4.(a), 4.(b) and 5.

A set of conclusions can be extracted from the experiments presented in this section:

- High values for the *migration rate* are recommended for any *migration topology* considered. The balance between the quality of the solutions and their spread obtained for 100 leads us to choose this value as the best.
- *bidirectional* migration topology can be considered as the best with regard most of the indicators. It gets a good balance between quality, spread and convergence, with the only handicap of requiring higher computational time than the other approaches, and yielding less non-dominated solutions in average, but getting a number close to *unidirectional*.
- However, if having the smaller running time as possible is considered as a requirement, *unidirectional* turns in the best option when searching for quality in the solutions, and *broadcast* when the aim if get a well distributed PS (good spread).

VI. CONCLUSIONS

This work presents an analysis of the Pareto-based island model introduced in the recent article [10]. It is a distribution scheme inspired in the island model widely used in evolutionary algorithms. But this is applied in the scope of multi-objective problems and is aimed to distribute a set of sub-colonies in an ant colony optimization algorithm. Thus it is a shape of multi-colony (or coarse-grained) model.

A neighborhood migration topology named *unidirectional* was initially proposed. It is focused on the Pareto front (representation of the ideal set of solutions) shape, so it establish the communication links in order to 'cover' the whole Pareto front, i.e. trying that the sub-colonies/islands search

(a) HV

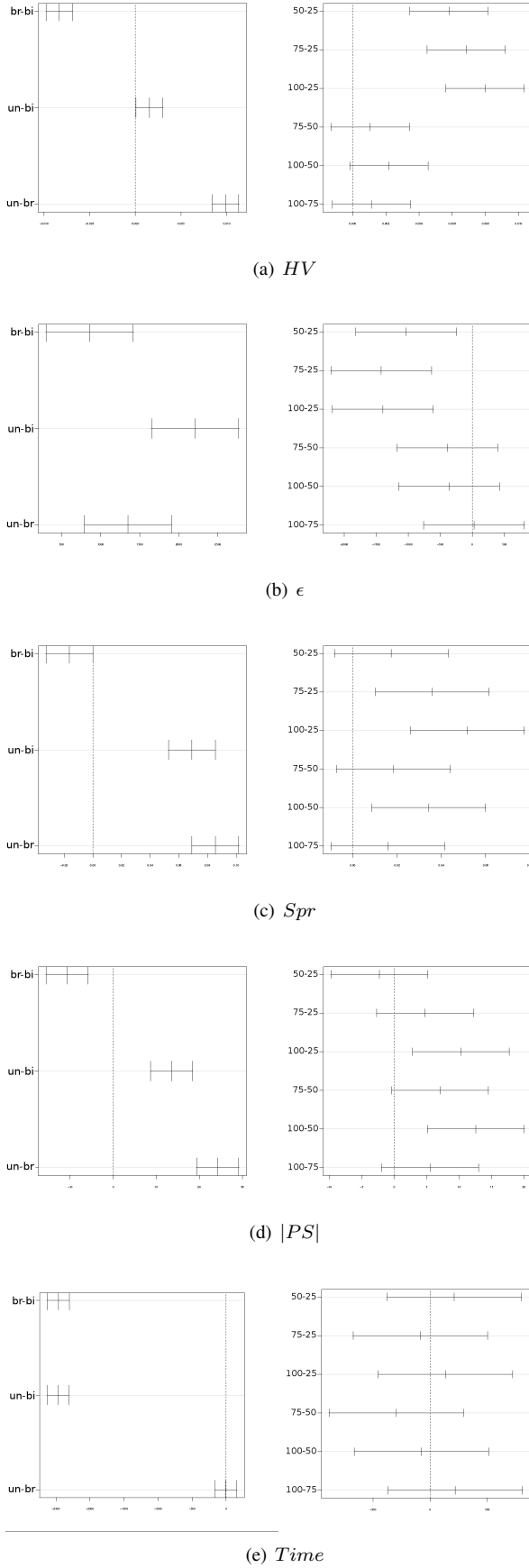(b) ϵ

(c) Spr

(d) $|PS|$

(e) $Time$

Fig. 6. Tukey's HSD graphs for topology (left) and migration rate (right)

in all the possible areas of the space of solutions. In this paper two additional topologies are presented: *bidirectional*, which completes the communications between islands in both directions, and *broadcast*, which migrates individuals from every island to the rest.

In addition, four different values for the *migration rate* parameter have been tested. The aim is to decide the best combination in these two factors of the proposed island model. In order to do this, MOACS algorithm [12] has been implemented using the proposed topologies. It has been adapted to solve the symmetric bi-criteria TSP in three different instances (100, 150 and 200 cities).

Several experiments have been conducted, computing classical metrics and indicators (hypervolume, spread, epsilon), along with graphs (attainment functions). Then a deep analysis of the results has been done in order to determine the approach with the best performance. In addition, statistical tests ANOVA and post-hoc Tukey's HSD have been applied to support this analysis.

The conclusions reached say that high values for the *migration rate* (around 100) yield better results. In the comparison between the *migration topologies*, the *bidirectional* approach has been chosen as the best option due to its good balance between the quality of the obtained solutions and the diversity and convergence level which it gets.

Following this line of work it could be interesting to perform another analysis with regard other factors of the island model, such as the migration or replacement policies. The comparison with others MOACO multi-colony models is another interesting line of research. There are several models proposed in the literature such as the one by Iredi at al. [11]. It would be also interesting to conduct a comparison with state-of-the-art multi-objective evolutionary algorithms implementing the island model.

REFERENCES

[1] E. Cantú-Paz, "Topologies, migration rates, and multi-population parallel genetic algorithms," in *Genetic and Evolutionary Computation Conference, GECCO-99*. ACM, 1999, pp. 13–17.

[2] A. Osyczka, "Multicriteria optimization for engineering design," *In John S. Gero, editor, Design Optimization, pp.193-227. Academic Press*, 1985.

[3] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.

[4] M. Dorigo and T. Stützle, "The ant colony optimization metaheuristic: Algorithms, applications, and advances," in *Handbook of Metaheuristics*, G. K. F. Glover, Ed. Kluwer, 2002, pp. 251–285.

[5] M. Pedemonte, S. Nesmachnow, and H. Cancela, "A survey on parallel ant colony optimization," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 5181–5197, 2011.

[6] M. Middendorf, F. Reischle, and H. Schmeck, "Multi colony ant algorithms," *Journal of Heuristics*, vol. 8, pp. 305–320, 2002.

[7] M. Bolondi and M. Bondanza, "Parallelizzazione di un algoritmo per la risoluzione del problema del commesso viaggiatore," Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, 2003.

[8] T. Stützle, "Parallelization strategies for ant colony optimization," in *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, 1998, pp. 722–741, lNCS 1498.

[9] B. Bullnheimer, R. Hart, and C. Strauss, *Parallelization Strategies for the Ant System*, ser. Applied Optimization, 1998, vol. 24, ch. High Performance Algorithms and Software in Nonlinear Optimization.

[10] T. authors, "MOACO Island Model Presentation," *One Journal*, vol. V, no. N, p. In Press., 2013.

[11] S. Iredi, D. Merkle, and M. Middendorf, "Bi-criterion optimization with multi colony ant algorithms," in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, ser. LNCS, E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne, Eds., vol. 1993.   Berlin: Springer, 2001, pp. 359–372.

[12] B. Barán and M. Schaerer, "A multiobjective ant colony system for vehicle routing problem with time windows," in *IASTED International Multi-Conference on Applied Informatics*, ser. IASTED IMCAI, no. 21, 2003, pp. 97–102.

[13] A. M. Mora, J. J. Merelo, P. A. Castillo, and M. G. Arenas, "hchac: A family of moaco algorithms for the resolution of the bi-criteria military unit pathfinding problem," *Computers & Operations Research*, vol. In Press., 2011, dOI: http://dx.doi.org/10.1016/j.cor.2011.11.015.

[14] R. Michel and M. Middendorf, "An island model based ant system with lookahead for the shortest supersequence problem," in *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, 1998, lNCS 1498.

[15] R. Jovanovic, M. Tuba, and D. Simian, "Comparison of different topologies for island-based multi-colony ant algorithms for the minimum weight vertex cover problem," *Transactions on Computers*, vol. 9, no. 1, 2010.

[16] A. Mora, J. Merelo, P. Castillo, M. Arenas, P. García-Sánchez, J. Laredo, and G. Romero, "A study of parallel approaches in moacos for solving the bicriteria tsp." in *Proceedings of the International Work Conference on Artificial Neural Networks (IWANN 2011)*, 2011, pp. 316–324, lNCS 6692.

[17] R. Fisher, "Theory of statistical estimation," in *Proceedings of the Cambridge Philosophical Society*, 1925, pp. 700–725.

[18] P. Dickinson and B. Chow, "Some properties of the tukey test to duckworth's specification," Office of Institutional Research, University of Southwestern Louisiana, Lafayette, USA, Tech. Rep., 1971.

[19] G. Croes, "A method for solving traveling salesman problems," *Operations Res.*, vol. 6, pp. 791–812, 1958.

[20] C. Twomey, T. Stützle, M. Dorigo, M. Manfrin, and M. Birattari, "An analysis of communication policies for homogeneous multi-colony aco algorithms," *Inf. Sci.*, vol. 2010, no. 12, pp. 2390–2404, 180.

[21] M. Middendorf, F. Reischle, and H. Schmeck, "Information exchange in multi colony ant algorithms," in *Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing*, 2000.

[22] C. Fonseca and P. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," in *Fourth International Conference on Parallel Problem Solving from Nature (PPSN-IV)*, 1996, pp. 584–593, lNCS 1141.

[23] J. Durillo, A. Nebro, and E. Alba, "The jmetal framework for multi-objective optimization: Design and architecture," in *IEEE Conference on Evolutionary Computation, CEC-2010*, 2010, pp. 4138–4325.