Parallel Island-Based Multiobjectivised Memetic Algorithms for a 2D Packing Problem

Carlos Segura
Dpto. Estadistica, I. O. y Computación
Universidad de La Laguna
38271, Tenerife, Spain
csegura@ull.es

Eduardo Segredo
Dpto. Estadistica, I. O. y Computación
Universidad de La Laguna
38271, Tenerife, Spain
esegredo@ull.es

Coromoto León
Dpto. Estadistica, I. O. y Computación
Universidad de La Laguna
38271, Tenerife, Spain
cleon@ull.es

ABSTRACT

Bin Packing problems are NP-hard problems with many practical applications. A variant of a Bin Packing Problem was proposed in the GECCO 2008 competition session. The best results were achieved by a mono-objective Memetic Algorithm (MA). In order to reduce the execution time, it was parallelised using an island-based model. High quality results were obtained for the proposed instance. However, subsequent studies concluded that stagnation may occur for other instances. The term *multiobjectivisation* refers to the transformation of originally mono-objective problems as multi-objective ones. Its main aim is to avoid local optima. In this work, a multiobjectivised MA has been applied to the GECCO 2008 Bin Packing Problem. Several multiobjectivisation schemes, which use problem-dependent and problem-independent information have been tested. Also, a parallelisation of the multiobjectivised MA has been developed. Results have been compared with the best up to date mono-objective approaches. Computational results have demonstrated the validity of the proposals. They have provided benefits in terms of solution quality, and in terms of time saving.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Inteligence—*Problem Solving, Control Methods and Search* Heuristic Methods; D.1.3 [**Software Engineering**]: Programming Techniques—*Concurrent Programming* Parallel Programming

General Terms

Algorithms

Keywords

Multiobjectivisation, Island-Based Models, Memetic Algorithms, Cutting and Packing Problems

1. INTRODUCTION

Bin Packing problems are combinatorial NP-hard problems in which items with different shapes, volumes and/or areas must be packed into a finite number of bins. They are closely related to cutting problems, whose main goal is to cut large stock sheets into a set of smaller pieces. In many cases, both problems are analysed together, referring them as cutting and packing problems. Cutting and packing problems have many applications and are widely used inside more complex systems, e.g., filling up containers and trucks, pallet loading, optimisation of the layout of electrical circuits, multiprocessor scheduling, etc. Cutting and packing problems can be classified according to several characteristics: the number of dimensions, the number of available patterns, the shape of the patterns (regular or irregular), the orientation, the objective that must be optimised, etc. Some popular variants are: *2D strip packing*, *constrained 2D cutting stock*, *knapsack problems*, *packing with cost*, and *online packing*. In the GECCO 2008 competition session[1] a variant of a 2D bin packing problem (2DPP) was proposed.

Some exact approaches have been analysed [16] for the resolution of cutting and packing problems. In order to reduce the execution time, some parallel exact approaches have been designed. However, since exact approaches are practically unaffordable for many real-world instances, a wide variety of approximated algorithms have also been developed. Among them, meta-heuristics are a family of techniques which have become very popular. *Memetic Algorithms* (MAs) [19] are a synergy of *Evolutionary Algorithms* (EAs) or any population-based approach with separate individual learning. They are of great value because they perform some orders of magnitude faster than traditional genetic algorithms for some problem domains.

In order to reduce the computational time, several studies have considered the parallelisation of EAs [1]. Parallel evolutionary algorithms (pEAs) can be classified [4] in three major computational paradigms: *master-slave*, *island-based*, and *diffusion*. When compared to the other parallel proposals, the island-based approach brings two benefits: it maps easily onto the parallel architectures (thanks to its distributed and coarse-grained structure), and it extends the search area (due to multiplicity of islands) preventing from sticking in local optima. Island-based models, also known as multideme models, have shown good performance and scalability in many areas [1]. Island-based models conceptually divide the overall pEA population into a number of independent

[1]http://www.sigevo.org/gecco-2008/competitions.html

and separate populations, i.e., there are separate and simultaneously executing EAs (one per processor or island). Each island evolves in isolation for the majority of the pEA execution, but occasionally, some individuals can be migrated between neighbour islands. Such paradigms can be extended to MAs parallelisation (pMAs) substituting EAs by MAs.

The best results for the instance proposed in the competition were achieved by a mono-objective MA. In [14] an island-based model in which each island executes a configuration based in this MA was proposed. Although high quality results were obtained for the proposed competition instance, subsequent studies concluded that stagnation may occur for other ones. In order to deal with stagnation, several methods have been designed [7]. Some of the simplest techniques rely on performing a restart of the approach when stagnation is detected. In other cases, a component which inserts randomness or noise in the search is used. Maintaining some memory, in order to avoid exploring the same zones several times, is also a typical approach. Finally, population-based strategies try to maintain the solutions set diversity. By recombining such a set of solutions, more areas of the decision space can be explored.

The term *multiobjectivisation* was introduced in [12] to refer to the reformulation of originally mono-objective problems as multi-objective ones. Multiobjectivisation changes the fitness landscape, so it can be useful to avoid local optima [8], and consequently, to make easier the resolution of the problem. However, it can also produce a harder problem [2]. There are two different ways of multiobjectivising a problem. The first one is based on a decomposition of the original objective, while the second one is based on aggregating new objective functions. The aggregation of alternative functions can be performed by considering problem-dependent or problem-independent information.

The first objective of this work has been the analysis of the 2DPP multiobjectivisation. Several multiobjectivisations which take into account problem-dependent and problem-independent information have been used. The applied optimisation methods have been compared with the best sequential approach published in [14]. Depending on the tested instance, multiobjectivisation methods could behave better or worse than the mono-objective strategies. Given that, the possibility of applying higher-level resolution methods such as hyper-heuristics to automate the selection of the proper approach seems very promising. In many cases, island-based models have been hybridised with hyper-heuristics [20]. Consequently, the second objective of this work has been the validation of island-based models together with the proposed multiobjectivisation methods. For this second study, two island-based models have been taken into account. These models make use of the mono-objective approach and the best behaved multiobjectivisation method detected in the first aforementioned study. The validity of the proposals have been demonstrated in the experimental evaluation.

The rest of the paper is structured as follows: the mathematical formulation for the 2DPP is given in Section 2. Section 3 is devoted to describe the applied optimisation scheme. MAs with its learning process and the employed genetic operators are presented. Also, the multiobjectivisation methods are detailed. The main features of island-based models are depicted in Section 4. Then, the experimental evaluation is described in Section 5. Finally, the conclusions and some lines of future work are given in Section 6.

2. FORMAL DEFINITION OF 2DPP

The proposed problem in the competition session is a two-dimensional variant of a bin packing problem. Problem instances are described by the following data:

- The sizes of a rectangular grid: X, Y.
- The maximum number which can be assigned to a grid position: N. The value assigned to each grid location is an integer in the range $[0, N]$.
- The score or value associated to the appearance of each pair (a, b) where $a, b \in [0, N]$: $v(a, b)$. Note that $v(a, b)$ is not necessarily equal to $v(b, a)$.

A candidate solution is obtained by assigning a number to each grid position. Thus, the search space is constituted by $(N+1)^{X \cdot Y}$ candidate solutions. The objective of the proposed problem is to best pack a grid so that the sum of the point scores for every pair of adjacent numbers is maximised. It is considered that two positions are adjacent if they are neighbours in the same row, column, or diagonal of the grid. Once that a particular pair is collected, it cannot be collected a second time in the same grid.

Mathematically, the problem objective is to find the grid G which maximises the fitness function f:

$$f = \sum_{a=0}^{N} \sum_{b=0}^{N} v_2(a, b)$$

where

$$v_2(a, b) = \begin{cases} 0 & if\ (a, b)\ are\ not\ adjacent\ in\ G \\ v(a, b) & if\ (a, b)\ are\ adjacent\ in\ G \end{cases}$$

3. OPTIMISATION SCHEME

3.1 Memetic Algorithms

Memetic algorithms [13, 19] are a synergy of a population-based approach with separate individual learning or local improvement procedures. These algorithms are also referred to in the literature as *Cultural Algorithms*, *Baldwinian Evolutionary Algorithms*, *Lamarckian Evolutionary Algorithms*, or *Genetic Local Search*. They are of great value because they perform some orders of magnitude faster than traditional genetic algorithms for some problem domains [6]. These algorithms have been applied in the mono-objective field [18] and also in the multi-objective one [22].

Algorithm 1 shows a general pseudocode of a memetic strategy. The main difference with respect to the corresponding original algorithm is the addition of a learning process step (line 5). There are two different ways to incorporate the individual learning [24]. On the one hand, the *Lamarckian learning* forces the genotype to reflect the result of improvement in the learning by placing the locally improved individual back into the population to compete for reproduction. On the other hand, the *Baldwinian learning* modifies the fitness of the individuals to reflect the improvement after the learning process. However, the improved genotype is not encoded back into the population. Both kinds of approaches have been successfully applied [15]. Learning processes usually make use of problem domain information. Nevertheless, some general methods have also been used.

Algorithm 1 MA Pseudocode

```
1: Generate an initial population
2: while (not stopping criterion) do
3:     Evaluate all individuals in the population
4:     Variation phase
5:     Perform individual learning process in the population with
       a probability p_l
6:     Select the individuals which survive to the next generation
7: end while
```

The learning process step is usually applied with a probability p_l. In [10] the effect of the probability p_l is analysed for a set of multi-objective benchmark problems. Authors concluded that the performance of MAs can be improved by dynamically changing the probability p_l. However, in the considered approach the individuals obtained after the variation phase may not have high enough quality components. Moreover, they can be easily improved by the individual learning process. Thus, the best results were obtained by applying the learning process in each generation. An effort to highly reduce the computational requirements of the learning process has been performed.

In the present work, two different MAs have been compared. They belong to the first generation of MAs [17]. The first one (*VarPopEA*) is the mono-objective approach presented in [14]. It is a MA which combines a modified evolutionary algorithm with a $(1+1)$ selection operator and a learning process specifically designed to face the 2DPP. The algorithm has the ability to perform as a trajectory-based algorithm when no stagnation is detected. However, it increases the population size in order to avoid strong local optima when necessary, behaving then as a population-based algorithm. Nevertheless, the method may not escape from local optima for some instances. The second one is a modified version of the well-known multi-objective evolutionary algorithm NSGA-II. This version incorporates the learning process after the variation stage of the original algorithm. For both MAs, individuals are encoded as a bidimensional array of integer values, G, where $G(x,y)$ is the number assigned to the grid position (x,y).

3.2 Learning Process for the 2DPP

Usually, multi-objective MAs make use of a multi-objective learning process [11]. However, since in the current paper the 2DPP has been multiobjectivised, the learning process has only taken into account the original objective. The applied process can be classified as a Lamarckian learning, i.e., the genotype reflects the learning process improvements. It is based on a mono-objective stochastic hill-climbing local search. The application of a local search allows admissible solutions to be achieved in relatively short times. The applied local search strategy [14] has the following features. For each pair of adjacent grid positions (i,j) and (k,l), a neighbour is considered. Each neighbour is constituted by assigning the best possible values to the positions (i,j) and (k,l), leaving intact the assignments in any other grid location. In order to assign the best values to both locations, the trivial solution consists in enumerating all possible pairs, for a later selection of the best one. As such approach is computationally too expensive, a mechanism to prune the explored values has been used. First, all the possible assignments $n \in [0,N]$ to the grid position (i,j) are considered, and the fitness contribution of each assignment $v_{ij}(n)$, assuming position (k,l) unassigned, is calculated. The same process is performed for the position (k,l), assuming the position (i,j) unassigned, and thus calculating $v_{kl}(n)$. The fitness contribution obtained by assigning a value a to the position (i,j), and a value b to the position (k,l), is given by:

$$v_{ij}(a) + v_{kl}(b) + v'(a,b) - v_{rep}$$

where $v'(a,b) = v(a,b) + v(b,a)$ if the pair (a,b) was not already in the grid, or 0 if it was, and v_{rep} is the value associated to pairs that are constituted by both, the assignment of the value a to (i,j) and the assignment of the value b to (k,l), which must be considered only once. An upper bound of such fitness contribution is given by:

$$v_{ij}(a) + v_{kl}(b) + min(bestV(a), bestV(b))$$

where $bestV(n)$ is the maximum value associated to any pair (n,m), $m \in [0,N]$, i.e., $\max\{(v(n,m)+v(m,n)\}$. Being $bestFit$ the best fitness currently achieved for an assignment of the positions (i,j) and (k,l), the only values a', b' that must be considered are the ones in which $v_{ij}(a') + v_{kl}(b') + min(bestV(a'), bestV(b')) > bestFit$ holds. By omitting the values in which the previous inequality does not hold, the neighbourhood to be considered is highly reduced.

The order in which neighbours are analysed is determined in a random way. The local search moves to the first new generated neighbour that improves the current solution. The local search stops when none of the neighbours improves the current solution.

3.3 Genetic Operators

A mutation and a crossover operator are applied on each generation (line 4 of the algorithm). Several variation operators were tested in [14]. The best behaved ones have been selected. The crossover operator consists in a two-dimensional sub-string crossover (SSX) [9]. First, a grid position is selected as the division point. Then, it randomly decides to do a vertical or horizontal crossover. SSX is illustrated in Figure 1. $H1$ and $H2$ are generated by means of an horizontal crossover, while $V1$ and $V2$ are generated by the application of the vertical one. In both cases, the position $(3,2)$ is selected as the division point.

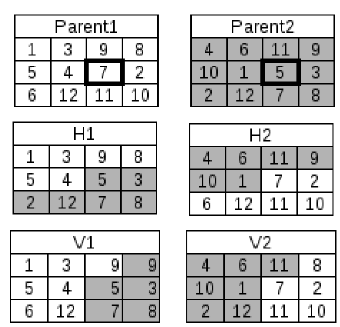

Figure 1: Sub-string Crossover (SSX)

The applied mutation operator was the *Uniform Mutation with Domain Information (UMD)*. Each gene is mutated with a probability between min_p_m and max_p_m. In order to perform the new assignment to the gene, a random value is selected among the ones that produce a non-zero increase in the fitness value.

3.4 Multiobjectivisations

Several multiobjectivisation schemes have been explored for the 2DPP. Multiobjectivisation usually decreases the selection pressure of the original approach. Therefore, some low quality individuals could survive in the population with a higher probability. However, in the long term these individuals could help to avoid stagnation in local optima, so higher quality solutions might be obtained.

Most of the applied multiobjectivisation strategies have considered problem-independent information, in which an artificial objective function has been added to multiobjectivise the 2DPP. The first objective has been selected as the fitness function of the 2DPP, while for the second one, an artificial function which tries to maximise the diversity has been used. One of the main challenges has been the selection of this artificial function. In fact, it has been demonstrated that the proper artificial function depends on the considered problem and even instance [21]. A comparison of a set of well-known schemes has been carried out. Moreover, a novel artificial objective has also been tested.

Several options have been proposed to define the artificial objective [3]. Some schemes based on the usage of the Euclidean distance on the decision space have been analysed:

- DCN: Distance to the closest population neighbour.
- ADI: Average distance to all population individuals.
- DBI: Distance to the best population individual, i.e., the one with the highest 2DPP fitness.

Also, the following ones have been taken into account:

- *Random*: A random value is assigned as the second objective to be minimised. Smaller random values may be assigned to some low quality individuals which would get a chance to survive.
- *Reverse*: In this case, the optimisation direction of the original objective function is inverted and it is used as the artificial objective. This approach highly decreases the selection pressure, so a large number of Pareto-optimal solutions could be included at each generation.

Finally, a novel variant of the DBI scheme has also been considered. It is based on the addition of a threshold which penalises those solutions that may have a very poor quality. In DBI_THR a threshold is established over the 2DPP objective function. Thus, individuals that are not capable to achieve the fixed threshold are penalised by assigning a zero value to the second objective function.

A novel multiobjectivisation by aggregation which considers problem-dependent information (*Dependent*) has also been tested. It makes use of two objectives. The first one is the original fitness function. For the second objective, the original 2DPP fitness function is decomposed in two independent fitness functions f_0 and f_1. The decomposition is performed in the following way. First, a table containing all possible pairs whose score is not equal to zero is constituted. Then, this table is sorted based on the score of the appearance of each pair p. The resultant position of each p, after the sort, is denoted as i_p. The fitness associated to each p is taken into account to calculate the function f_n where $n = i_p \bmod 2$. Finally, f_0 is used as the second objective.

4. ISLAND-BASED MODELS

In island-based models, the population is divided into a number of independent subpopulations or *demes*. Each subpopulation is associated to an island and a MA configuration is executed over each subpopulation. Usually, each available processor constitutes an island which evolves in isolation for the majority of the parallel run. However, collaborative schemes could lead to a better behaviour. Therefore, a migration stage which enables the transfer of individuals among islands is generally incorporated.

Four basic island-based models are seen to exist [4]: all islands execute identical configurations (homogeneous), all islands execute different configurations (heterogeneous), each island evaluates different objective function subsets, and each island represents a different region of the genotype or phenotype domains. In the first two variants, the population of each island represents solutions to the same problem. In the third variant, each island searches a reduced problem domain space. The last variant isolates each processor to solve specific, non-overlapping regions of genotype/phenotype domain space. The parallel strategy presented in this paper is based on the homogeneous island-based model.

Migrations are an essential operation on these parallel schemes, so that they allow the collaboration among islands. A well designed migration stage could provide a successful collaboration. Thus, the solution search space could be better explored and higher quality solutions could be obtained. However, if an unsuitable migration stage is introduced in the model, the effect could be similar, or even worse, than having separate MAs simultaneously executing on several processors with no communication among them. Therefore, the migration stage must be carefully defined. In order to configure the migration stage, it is necessary to establish the migration topology (where to migrate the individuals) and the migration rate (how many individuals are migrated and how often). Also, individuals which are going to be migrated and those which are going to be replaced must be selected. Such a selection is performed by the use of the migration scheme and the replacement scheme, respectively.

Applying island-based models, landscapes may be completely different from those produced by its corresponding sequential MA. As such the pMA may find better or equivalent solution in a lower amount of time. Depending on the selected migration stage the landscape is affected on different ways [23]. In [14], an island-based model is successfully applied to the 2DPP. In such a case, every island configuration is based on VarPopEA. The migration stage has the following characteristics. An all to all connected migration topology is used. An elitist migration scheme is performed. Specifically, a subpopulation individual is migrated when it is better than any member of its previous generation. Replacements are also performed following an elitist scheme. They only take place when the migrated individual is better than any of the individuals in the destination island.

For the island-based multiobjectivised approach here presented, a similar migration stage has been used. The only one difference with the aforementioned migration phase resides on the replacement scheme. In this case, the elitist ranking scheme [23] has been applied. It ranks all Pareto fronts and replaces an individual from the worst ranked front with the immigrant. This scheme also provides high selection pressure, but it has been specifically designed for the multi-objective field.

5. EXPERIMENTAL EVALUATION

This section is devoted to describe the experiments performed with the multiobjectivised memetic algorithm exposed in Section 3 and with the island-based multiobjectivised memetic algorithm detailed in Section 4. The obtained results are compared with the corresponding mono-objective versions of both approaches. These mono-objective versions are based on the VarPopEA algorithm, which is the algorithm that up to date has reported the highest quality results for 2DPP. Tests have been run on a Debian GNU/Linux computer with four AMD ® Opteron ™ 6164 HE at 1.7 GHz and 64 Gb RAM. The compiler which has been used is *gcc 4.4.5*. Comparisons have been performed considering two instances. The first one is characterised by the following parameters: $X = 10$, $Y = 10$, $N = 99$, and 9032 possible pair scores. The second one is the one which was proposed in the competition session. Its parameters are the following: $X = 20$, $Y = 20$, $N = 399$, and 15962 possible pair scores. Since we are dealing with stochastic algorithms, each execution has been repeated 30 times. In order to provide the results with confidence, comparisons have been performed applying the following statistical analysis [5]. First, a *Shapiro-Wilk test* is performed in order to check whether the values of the results follow a normal distribution or not. If so, the *Levene test* checks for the homogeneity of the variances. If samples have equal variance, an ANOVA *test* is done. Otherwise, a *Welch test* is performed. For non-gaussian distributions, the non-parametric *Kruskal-Wallis* test is used to compare the medians of the algorithms. A confidence level of 95% has been considered.

The first experiment is devoted to analyse the behaviour of multiobjectivisation when applied to 2DPP. An analysis of the solutions quality of seven different configurations of the sequential multiobjectivised memetic algorithm has been performed. Each configuration has applied one of the seven multiobjectivisation schemes proposed in Section 3.4. A threshold value equal to 0.99 has been fixed for DBI_THR. Results have been compared with the ones obtained by VarPopEA. In every case, the stopping criterion has been fixed to 24 hours. In the case of the multiobjectivised memetic algorithm, a population size equal to 10 individuals has been fixed. The same parameters used in [14] have been applied in the case of VarPopEA. For both memetic algorithms, the UMD operator and the SSX operator have been applied. The UMD operator has used the next parameterisation: $min\_p_m = 0.1$ and $max\_p_m = 0.15$. The SSX operator was applied for each offspring, i.e., $p_c = 1$.

Figure 2 shows, for the first instance, the evolution of the fitness function average value for the different schemes. Four multiobjectivised models obtain a higher value than VarPopEA. The *Dependent* multiobjectivisation was not able to obtain high quality results. Table 1 shows, for 24 hours of execution, whether the row configuration is statistically better (↑), not different (↔), or worse(↓), than the corresponding column configuration. The differences among the three best multiobjectivised approaches and VarPopEA are significant, showing the benefits of multiobjectivisation for the analysed instance. Similar information is given for the second instance in Figure 3 and Table 2. In this instance VarPopEA average value is higher than the ones obtained by the multiobjectivised approaches. Moreover, differences among VarPopEA and the multiobjectivised approaches are significant. Thus, with the considered parameterisation, the

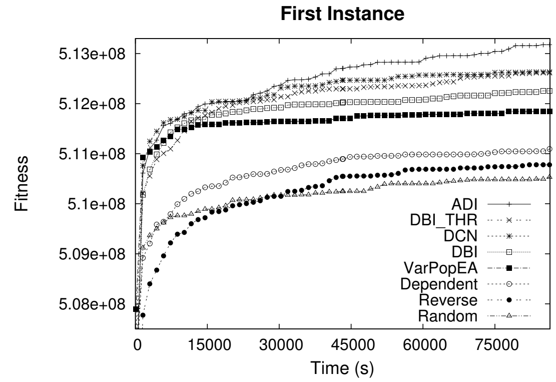

Figure 2: Fitness Evolution for the First Instance

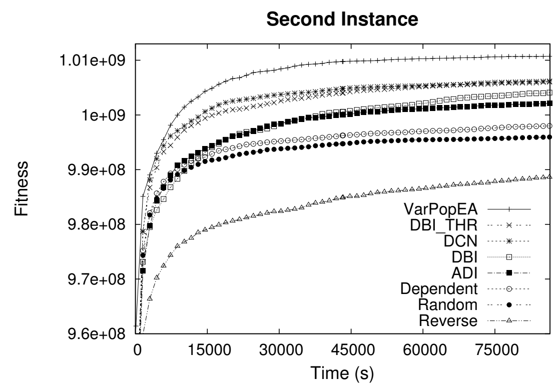

Figure 3: Fitness Evolution for the Second Instance

most adequate algorithm depends on the instance. Multiobjectivisation has been useful only for small instances because it requires the evolution of a large amount of generations. In the 2DPP first instance note that about 17 times more generations than in the second instance have been evolved.

The main aim of the second experiment is to study the behaviour of island-based models. Two homogeneous island-based models configured as explained in Section 4, have been executed. The first one (*Mono-Island*) makes use of VarPopEA. The second one (*Multi-Island*) uses, for each instance, the best behaved multiobjectivised approach of the first experiment. Both models have been executed using four islands and a stopping criterion of 12 hours. Figure 4 shows, for the first instance, the boxplots obtained by the sequential and parallel models in a fixed time of 12 hours. Both parallel models have clearly improved the results obtained by their corresponding sequential versions. In this case, both multiobjectivised approaches have improved the mono-objective ones. In fact, the best results have been achieved by Multi-Island. For the sequential schemes, differences have been more noticeable than for the parallel models. For the second instance, the same information is shown in Figure 5. In this case, both parallel models have also improved their corresponding sequential schemes. Solutions obtained by VarPopEA have clearly improved the ones obtained by DBI_THR. Moreover, Multi-Island has not been able to achieve the high

Table 1: Statistical Comparison of Configurations for the First Instance

	ADI	DBI_THR	DCN	DBI	VarPopEA	Dependent	Reverse	Random
ADI	↔	↔	↔	↑	↑	↑	↑	↑
DBI_THR	↔	↔	↔	↔	↑	↑	↑	↑
DCN	↔	↔	↔	↔	↑	↑	↑	↑
DBI	↓	↔	↔	↔	↔	↑	↑	↑
VarPopEA	↓	↓	↓	↔	↔	↑	↑	↑
Dependent	↓	↓	↓	↓	↓	↔	↔	↔
Reverse	↓	↓	↓	↓	↓	↔	↔	↔
Random	↓	↓	↓	↓	↓	↔	↔	↔

Table 2: Statistical Comparison of Configurations for the Second Instance

	VarPopEA	DBI_THR	DCN	DBI	ADI	Dependent	Random	Reverse
VarPopEA	↔	↑	↑	↑	↑	↑	↑	↑
DBI_THR	↓	↔	↔	↔	↑	↑	↑	↑
DCN	↓	↔	↔	↔	↑	↑	↑	↑
DBI	↓	↔	↔	↔	↔	↑	↑	↑
ADI	↓	↓	↓	↔	↔	↑	↑	↑
Dependent	↓	↓	↓	↓	↓	↔	↑	↑
Random	↓	↓	↓	↓	↓	↓	↔	↑
Reverse	↓	↓	↓	↓	↓	↓	↓	↔

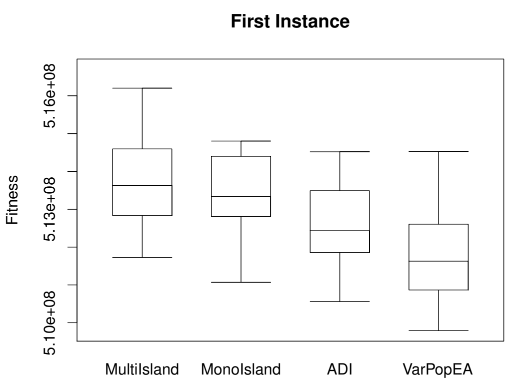

Figure 4: First Instance Boxplots in 12 Hours

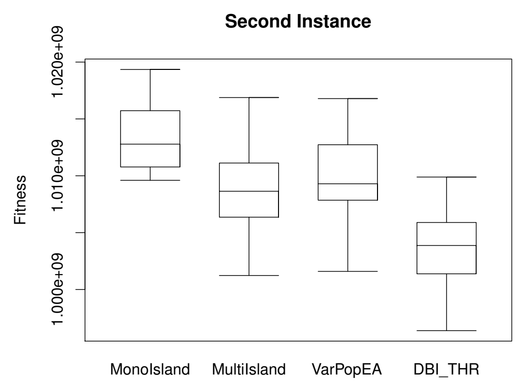

Figure 5: Second Instance Boxplots in 12 Hours

quality solutions achieved by VarPopEA. Therefore, for this instance, multiobjectivisation has not reported benefits.

The previous analysis has shown that parallel approaches have obtained higher quality solutions than their corresponding sequential models when the same stopping criterion is considered. In order to check which model has made a better usage of the computational resources, Figure 6 shows, for the first instance, the boxplots of each model when a similar amount of computational resources is used. The same information is shown in Figure 8 for the second instance. Since four islands have been considered for the parallel models, results of sequential schemes are shown taking into consideration a stopping criterion of 24 hours, while for the parallel schemes the stopping criterion has been fixed to 6 hours. The memetic island-based models have made a similar usage of computational resources than their corresponding sequential approaches for both instances. Even for the first instance, Mono-Island has made a better usage of computational resources than VarPopEA. Since computational resources are used in a parallel way with the memetic island-based models, the validity of the here presented parallel approaches has been demonstrated.

Run-length distributions are a useful tool to quantify the improvement of the parallel models. They show the relation between success ratios and time. Success ratio is defined as the probability of achieving a certain quality level. Run-length distributions have been calculated for the best mono-objective and multiobjectivised approaches, as well as their parallelisations. In order to establish a high enough quality level for each instance, it has been fixed as the average fitness obtained in 6 hours of execution by the worst of the aforementioned models.

Figure 7 shows the run-length distribution for the first instance. It shows the superiority of the parallel schemes. Considering the times required to achieve a 50% of success ratio, superlinear speedups have been obtained, comparing each sequential algorithm with its corresponding parallel scheme. The main reason has been that the population increment in the parallel approaches has allowed avoiding stagnation. Since VarPopEA behaves essentially as a trajectory-based algorithm, its parallelisation had a higher impact over the performance. In fact, for the same computational effort, parallel schemes have obtained higher quality levels (Figure 6).

The run-length distributions for the second instance are shown in Figure 9. In this case, the parallel schemes benefits can also be appreciated. Considering times to achieve a 50% of success ratio, the speedup value for Mono-Island

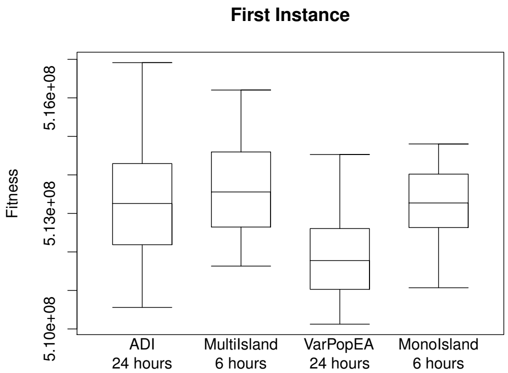

Figure 6: Boxplots - Fixed Computational Effort

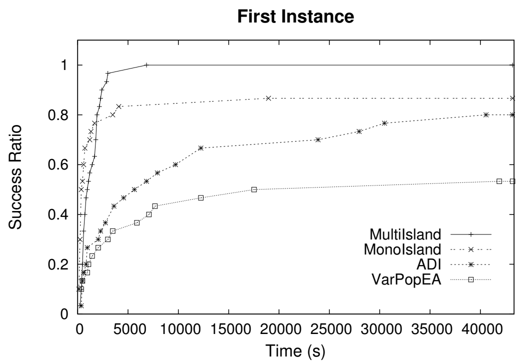

Figure 7: Run-length Distributions

has been 1.95, when it has been compared with VarPopEA. The speedup value for Multi-Island has been 2.42, when it has been compared with DBI_THR. When success ratios from 25% to 75% have been taken into account, the speedup factors have changed. In the case of the mono-objective approaches, the speedup values have ranged from 1.95 to 2.13. The speedup values have ranged from 1.89 to 3.58 for the multiobjectivised schemes.

6. CONCLUSIONS AND FUTURE WORK

Bin Packing problems are NP-hard problems with many practical applications. A variant of a Bin Packing Problem (2DPP) was proposed in the GECCO 2008 competition session. VarPopEA is a memetic approach which has provided the best known results for the 2DPP. However, it suffers of stagnation with some instances. This paper analyses the usage of multiobjectivisation as a technique that facilitates the escaping of local optima. Multiobjectivised configurations of a memetic algorithm which have considered problem-dependent and problem-independent information have been applied. The memetic approach is based on NSGA-II. Results obtained for two different instances have been compared. The *Dependent* multiobjectivisation has not been able to obtain high quality results. However, three multiobjectivised models have obtained better results than VarPopEA, for the first instance. By contrast, in the case of the second instance, VarPopEA has been the best behaved approach. Therefore, the most adequate algorithm

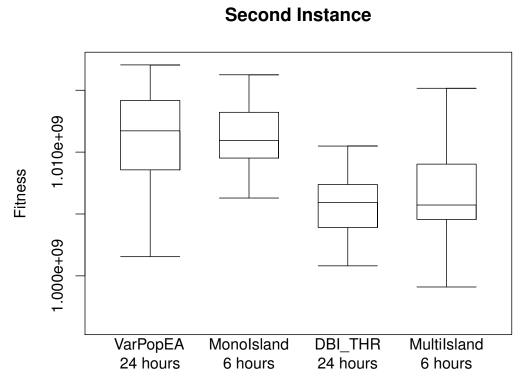

Figure 8: Boxplots - Fixed Computational Effort

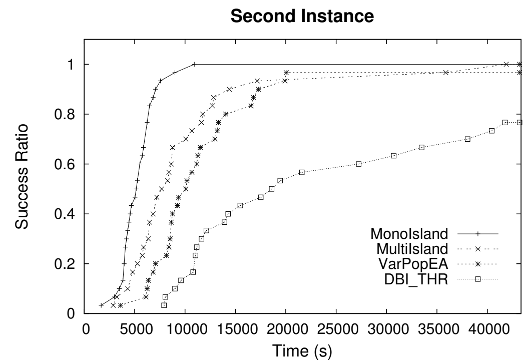

Figure 9: Run-length Distributions

depends on the instance. Given that, the possibility of applying hyper-heuristics to automate the selection of the proper approach seems very promising. Models which hybridise hyper-heuristics and island-based models have been previously applied in many fields with success. Thus, island-based models have also been analysed in this work. Mono-objective and multiobjectivised homogeneous island-based models have been tested. For both analysed instances the proper behaviour of island-based models has been demonstrated. They have provided benefits in terms of solution quality, and in terms of time saving. In the first instance, superlinear speedups have been obtained. The main reason has been that the population increment in the parallel approaches has allowed avoiding stagnation. In the second instance, satisfactory speedup ratios have also been obtained.

Future work will focus on the application of parallel multiobjectivised hyper-heuristics to solve the 2DPP. A scalability study should be performed, in order to analyse the limitations of the proposed island-based models. Moreover, since the appropriate optimisation method depends on the instance that is being solved, the application of hyper-heuristics seems a promising approach. Thus, the selection of which optimisation method must be used, could be performed in an automatic way. Also, it would be interesting to analyse the usage of multiobjectivisation with other 2DPP instances. That could help to identify the kind of instances in which multiobjectivisation is a valid approach.

7. ACKNOWLEDGEMENTS

This work has been partially supported by the EC (FEDER) and the Spanish Ministry of Science and Innovation as part of the 'Plan Nacional de I+D+i', with the contract number TIN2008-06491-C04-02 and by the Canary Government project number PI2007/015. The work of Eduardo Segredo has been funded by grant FPU-AP2009-0457. The work of Carlos Segura has been funded by grant FPU-AP2008-03213. The work has also been funded by the HPC-EUROPA2 project (project number: 228398) with the support of the European Commission - Capacities Area - Research Infrastructures

8. REFERENCES

[1] E. Alba. *Parallel Metaheuristics: A New Class of Algorithms*. Wiley-Interscience, 2005.

[2] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. Do additional objectives make a problem harder? In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, pages 765–772, New York, NY, USA, 2007. ACM.

[3] L. Bui, H. Abbass, and J. Branke. Multiobjective optimization for dynamic environments. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pages 2349 – 2356 Vol. 3, 2005.

[4] C. A. Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, 2007.

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

[6] P. Garg. A comparison between memetic algorithm and genetic algorithm for the cryptanalysis of simplified data encryption standard algorithm. *International Journal of Network Security & Its Applications*, 1(1):34 – 42, April 2009.

[7] F. W. Glover and G. A. Kochenberger. *Handbook of Metaheuristics (International Series in Operations Research & Management Science)*. Springer, January 2003.

[8] J. Handl, S. C. Lovell, and J. Knowles. Multiobjectivization by decomposition of scalar cost functions. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X*, pages 31–40, Berlin, Heidelberg, 2008. Springer-Verlag.

[9] T.-P. Hong, M.-W. Tsai, and T.-K. Liu. Two-dimentional encoding schema and genetic operators. In *JCIS*. Atlantis Press, 2006.

[10] H. Ishibuchi, Y. Hitotsuyanagi, and Y. Nojima. An empirical study on the specification of the local search application probability in multiobjective memetic algorithms. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 2788 –2795, 2007.

[11] A. Jaszkiewicz. Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, 137(1):50 – 71, 2002.

[12] J. D. Knowles, R. A. Watson, and D. Corne. Reducing local optima in single-objective problems by multi-objectivization. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, EMO '01, pages 269–283, London, UK, 2001. Springer-Verlag.

[13] M. N. Le, Y.-S. Ong, Y. Jin, and B. Sendhoff. Lamarckian memetic algorithms: local optimum and connectivity structure analysis. *Memetic Computing*, 1(3):175–190, 2009.

[14] C. Leon, G. Miranda, and C. Segura. A memetic algorithm and a parallel hyperheuristic island-based model for a 2d packing problem. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, GECCO '09, pages 1371–1378, New York, NY, USA, 2009. ACM.

[15] Z.-f. Z. Li-xiao Ma, Kun-qi Liu and N. Li. Exploring the effects of lamarckian evolution and baldwin effect in differential evolution. In *Communications in Computer and Information Science*, volume 107 of *Computational Intelligence and Intelligent Systems*, pages 127–136. Springer, 2010.

[16] S. Martello, M. Monaci, and D. Vigo. An exact approach to the strip-packing problem. *INFORMS Journal on Computing*, 15(3):310–319, 2003.

[17] Q. H. Nguyen, Y. S. Ong, and M. H. Lim. Non-genetic transmission of memes by diffusion. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, GECCO '08, pages 1017–1024, New York, NY, USA, 2008. ACM.

[18] Q. H. Nguyen, Y. S. Ong, and M. H. Lim. A Probabilistic Memetic Framework. *Evolutionary Computation, IEEE Transactions on*, 13(3):604–623, 2009.

[19] Y.-S. Ong, M.-H. Lim, N. Zhu, and K. W. Wong. Classification of adaptive memetic algorithms: a comparative study. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 36(1):141–152, 2006.

[20] C. Segura, G. Miranda, and C. León. Parallel hyperheuristics for the frequency assignment problem. *Memetic Computing*, pages 1–17, 2010.

[21] C. Segura, E. Segredo, Y. González, and C. León. Multiobjectivisation of the antenna positioning problem. In A. Abraham, J. Corchado, S. González, and J. De Paz Santana, editors, *International Symposium on Distributed Computing and Artificial Intelligence*, volume 91 of *Advances in Intelligent and Soft Computing*, pages 319–327. Springer Berlin / Heidelberg, 2011.

[22] K. Sindhya, A. Sinha, K. Deb, and K. Miettinen. Local search based evolutionary multi-objective optimization algorithm for constrained and unconstrained problems. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 2919–2926, Piscataway, NJ, USA, 2009. IEEE Press.

[23] D. A. V. Veldhuizen, J. B. Zydallis, and G. B. Lamont. Considerations in engineering parallel multiobjective evolutionary algorithms. *IEEE Trans. Evolutionary Computation*, 7(2):144–173, 2003.

[24] L. D. Whitley, V. S. Gordon, and K. E. Mathias. Lamarckian evolution, the baldwin effect and function optimization. In *Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature*, PPSN III, pages 6–15, London, UK, 1994. Springer-Verlag.