

The Importance of Proper Diversity Management in Evolutionary Algorithms for Combinatorial Optimization

Carlos Segura, Arturo Hernández Aguirre, Sergio Ivvan Valdez Peña and Salvador Botello Rionda

Abstract Premature convergence is one of the most important recurrent drawbacks of Evolutionary Algorithms and other metaheuristics. As a result, several methods to alleviate this problem have been devised. One alternative is to explicitly control the diversity of the population. In this chapter, a recently proposed survivor selection strategy is incorporated into a memetic algorithm and analyzed using three different combinatorial optimization problems. This strategy is based on adopting multi-objective concepts for solving single-objective problems by considering the contribution to diversity as an explicit objective. Additionally, it incorporates the principle of adapting the balance between exploration and exploitation to the different stages of the optimization by taking into account the stopping criterion and elapsed time. These new methods provide important benefits when compared to more mature methods that rely on different principles to delay convergence of the population. Additionally, new best-known solutions are generated for several instances of the problems, thus providing proofs of the considerable benefits and robustness yielded by the schemes that incorporate this novel replacement strategy.

Keywords Diversity preservation · Premature convergence · Evolutionary algorithms · Sudoku · Frequency assignment · Two-dimensional packing

C. Segura (✉) · A.H. Aguirre · S.I.V. Peña · S.B. Rionda
Centro de Investigación en Matemáticas, A.C. (CIMAT), Área de Computación,
Jalisco S/N, Col. Valenciana, 36023 Guanajuato, Mexico
e-mail: carlos.segura@cimat.mx

A.H. Aguirre
e-mail: artha@cimat.mx

S.I.V. Peña
e-mail: ivvan@cimat.mx

S.B. Rionda
e-mail: botello@cimat.mx

1 Introduction

Evolutionary Algorithms (EAs) are probably the most widely used population-based metaheuristics, having been successfully applied to many different problems [12] and in many different fields of optimization, such as in multi-objective optimization [8], combinatorial optimization [9] and continuous optimization [53]. In spite of the large number of different areas where they have been applied, designing a proper EA usually involves many difficult design decisions [5]. Many efforts have been made and advances achieved to automate the creation of components and selection of parameters of EAs with the aim of facilitating their design [26]. However, applying these approaches to automatically design state of the art schemes with ad-hoc operators suitable to the problems at hand is not yet a reality. Among the difficulties that the designers of EAs might encounter, the appearance of premature convergence is probably one of the most frequent drawbacks [10]. Premature convergence arises when all the population members are located in a small region of the search space that differs from the region containing the optimal solution, and the genetic operators and components included in the EA do not allow escaping from this region. Thus, it is closely related to the management of diversity in the population. In fact, many studies have shown that maintaining a diverse population offers a way to alleviate the problems related to premature convergence [10]. However, a population that is too diverse might prevent exploitation, resulting in poor-quality solutions. For this reason, Mahfoud coined the concept of *useful diversity* [31] to refer to those amounts of diversity that result in proper solutions.

In the literature, several different ways of alleviating the problem of premature convergence have been devised [40]. In the 90s, most proposals for alleviating premature convergence were focused on modifying the parent selection scheme. The reason is that at the time, most of the schemes were generational methods, so the main source of selection pressure was the parent selection. However, it was soon discovered that trying to alleviate this drawback by only taking into account parent selection was not enough [4]. In subsequent years, most EAs incorporated the use of a replacement phase that abandoned, at least partially, the initial generational replacement methods. Many authors discovered the possibility of incorporating methods to alleviate the premature convergence problem into this phase [10]. Note that, even when the generational replacement methods were more popular, some authors had already taken this choice into account [30]. However, with the popularization of elitism and other kinds of replacement strategies, the number of schemes that adopted these principles grew enormously [27, 35, 54].

One important aspect of premature convergence is that it completely depends on the amount of time and/or generations that is granted to the executions of the EA, i.e. its stopping criterion. For instance, a specific EA might be executed to solve a given problem for one hour and it might yield promising results. However, when executed for 24h, it might converge prematurely after 4h, for instance, and then the results would not be as good as expected for such a long execution. While the above discussion is quite logical and not too surprising, it is perhaps more surprising that

most of the methods that have been proposed to alleviate the premature convergence drawback do not take into account the stopping criterion set by the user to alter their internal operation. This means that, depending on the stopping criterion, different parametrizations might be required. As a result, for each different stopping criterion, the user must study the effect of the different parameters and tune them accordingly. An example of this is the *Restricted Tournament Selection* [14] (RTS) method, which is one of the most popular methods to delay the convergence of EAs. It incorporates a parameter that can be used to alter the balance between exploration and intensification. However, the loss of diversity, and consequently the balance between exploration and exploitation, does not depend solely on this parameter, meaning that each problem requires an analysis of the implications of using different parametrizations, and that different parameter values should be used for each problem and for each stopping criterion.

The basic principle of the the diversity-preservation techniques that affect the replacement phase is that by diversifying the survivors, more exploration can be induced. The reason for this is two-fold. First, if the diversity of a given population is large, it means that several regions of the search space are maintained. Second, most crossover operators tend to be more explorative when distant individuals are involved [13]. However, the most popular proposals that try to maintain proper diversity by altering the replacement phase do not take into account the stopping criterion set by the user, which is an important drawback. In fact, to our knowledge, our recent proposal [52] (MULTI_DYN) is the first replacement strategy designed to alleviate the premature convergence drawback by taking into account the stopping criterion and elapsed time or generations to select the survivors of the population. Note that the principle of relating survivor selection to the stopping criterion can be done in several ways. Thus, one of the aims behind the development of MULTI_DYN is to show that using this principle is helpful and can yield quite important benefits. Specifically, MULTI_DYN combines the idea of transforming a single-objective problem into a multi-objective one by considering diversity as an explicit objective, with the idea of adapting the balance induced between exploration and exploitation to the various optimization stages.

In this chapter, this novel method and some other more mature schemes are incorporated into a memetic algorithm following a common framework, and they are tested on three complex combinatorial optimization problems. The main aim of this paper is to show the generality and robustness of MULTI_DYN. This is done by incorporating this method into this common framework and testing it with three different problems. The first one is the Sudoku, which is an NP-Complete problem [56] that has been recently used extensively as a way of measuring the abilities of different optimization frameworks [7, 25]. Second, the results for a two-dimensional packing problem (2DPP) that was proposed in a contest at the *Genetic and Evolutionary Computation Conference* are also shown.¹ Note that this was the only problem used in [52].

¹The original website (<http://www.sigevo.org/gecco-2008/competitions.html>) is not being maintained. We have created a new website from which the evaluator and instances can be downloaded (<http://2dpp.cimat.mx>).

In this chapter, the experimentation is extended with the aim of further improving the results obtained and to better validate our scheme against other variants. Finally, a variant of the *Frequency Assignment Problem* [22] (FAP), which is known to be an NP-hard optimization problem, is also considered. While the main contribution is to show the robustness and generality of MULTI_DYN, the state of the art in the application of metaheuristics to the problems addressed in this chapter is also improved upon. In fact, new best-known solutions are attained for some well-known instances of these problems. Particularly, since a large number of different methods have been proposed in the case of the FAP, further improving the results obtained for publicly available instances is quite a remarkable achievement.

This chapter is organized as follows. Section 2 presents a summary of the methods designed to alleviate the premature convergence drawback in EAs. Our novel replacement strategy is discussed in Sect. 3. A memetic algorithm that is used to test several ways of delaying the convergence of the population, is presented in Sect. 4. In this last section, the mathematical definition of the three problems tested, as well as the way of integrating them with the memetic algorithm, is also provided. Section 5 is devoted to our experimental validation. Finally, our conclusions and future work are presented in Sect. 6.

2 Control of Premature Convergence in Evolutionary Algorithms

Premature convergence is one of the most typical issues that appears in the application of EAs to complex problems. This has resulted in a large number of techniques being devised to deal with this drawback [40]. In this section, a brief introduction to the most popular schemes is given. In order to classify the different proposals, a taxonomy was proposed in [10]. The methods are classified in base of the component of the EA that they modify. Some of the most popular schemes fall into the following categories: *selection-based*, *population-based*, *crossover/mutation-based*, *fitness-based*, and *replacement-based*. These schemes modify the parent selection, the population model, the variation phase, the way of establishing the individual's fitness, and the replacement strategy respectively. Additionally, schemes are referred to as *uniprocess-driven* approaches when a single component is modified, whereas the term *multiprocess-driven* is used to refer to those schemes that act over more than one component. In this section, some of the most popular general techniques designed for addressing premature convergence are reviewed and the components that are modified are specified. Many of these schemes are used to show the benefits of the novel proposals presented in this chapter. In such cases, some additional details such as the parameters required by them are specified. The reader is referred to [10, 40] and to the original papers of each scheme for further details.

Most of the initial approaches were selection-based schemes based on explicitly controlling the selection pressure of the parent selection phase [12]. However, it was

found both theoretically and experimentally that these kinds of schemes were not able to preserve diversity in the long term, meaning additional measures were required for the most complex problems [4]. Thus, currently when these schemes are used, they are usually combined with other strategies that rely on additional mechanisms to delay convergence.

The population-based schemes alter the panmictic population model of typical EAs. EAs with structured populations, such as the island-based model or the cellular schemes, have gained popularity in recent years [2]. In these schemes, individuals can only be recombined with a subset of the population. Most of these schemes were proposed as a mean of parallelizing metaheuristics [2]. However, practitioners soon discovered that structured populations have important implications on the way of managing the diversity. Specifically, since the amount of interactions between individuals are diminished, the convergence is usually delayed. Methods based on the infusion technique are also quite popular approaches categorized as population-based. In these schemes some new individuals are inserted in the population when some conditions appear. For instance, a partial restart of the population might be done after a fixed period or when some event related to the loss of diversity occur [11]. The saw-tooth genetic algorithm (saw-tooth GA) [20] is a popular scheme that applies this principle. Specifically, this scheme uses a variable population size and periodic partial reinitialization of the population in the shape of a saw-tooth function. In order to adapt the scheme to the particular needs of the problem, two different parameters are used: the period (P) and the amplitude (D) of a saw-tooth function. This function is used to manage the dynamic population size and to trigger the partial reinitializations.

In the group of the crossover/mutation-based schemes, the approaches that impose some mating restrictions are one of the most popular [25]. These schemes follow some principles that are quite similar to those applied in the island-based schemes. Specifically, some restrictions on the interactions between individuals are also imposed. However, in these cases, the features of the individuals that are taken into account to establish the restrictions differ from those used in the structured populations. For instance, in the cross generational elitist selection, Heterogeneous recombination, Cataclysmic mutation scheme (CHC), the recombination of too close individuals is avoided. Note that CHC includes additional modifications, such as the use of a highly disruptive crossover operator and the reinitialization process through the application of special mutation. Finally, in other cases the variation stage is modified with the aim of better controlling the balance between exploration and exploitation in a problem-specific way. For instance, in [42] several complementary genetic operators are considered simultaneously in an effort to adapt the variation phase to the needs of the different optimization stages. However, since this kind of control depends on the problem, on the features of the genetic operators and on the stopping criterion set by the user, adapting this scheme to different problems is a complex task.

In the fitness-based schemes, the way of calculating the fitness of each individual is altered. Probably, the *fitness sharing* scheme is the most popular proposal of this group. Fitness sharing reduces the fitness of individuals placed in densely populated regions. Specifically, the fitness of each individual is normalized depending on the number of individuals in its region, which is defined via the parameter σ . Thus, the

fitness of any individual is affected by any other individual whose distance is not larger than σ . The *clearing* strategy (CLR) [41] is as an extension of fitness sharing and it alters both the fitness assignment procedure and the replacement selection phase. In the clearing procedure the individuals are grouped into niches—defined via the parameter σ —and the resources of a niche are attributed to the best W elements in each niche. Moreover, the winners of each niche are copied to the next population, so the best members of the selected regions are preserved. Note that if too many niches are detected, this might lead to large immobilization of the population. As a result, Petrowsky proposed discarding the winners with a fitness lower than the mean [41], which is the alternative used in this chapter. Note that the *clearing* procedure is categorized as a fitness-based scheme in [10]. However, since it also affects the replacement phase it might be categorized as a multiprocess-driven hybrid method. Finally, methods based on multiobjectivization [19, 51] use simultaneously several fitness functions. In some cases, one of the fitness function is the original one that must be optimized. However, decomposing the original function in several functions, and using each of them as a fitness function to optimize is also a plausible choice.

Finally, since the appearance of elitism and with the popularization of the inclusion of a replacement phase different from the generational replacement, several replacement-based methods have also been devised. The basic principle of these schemes is to induce larger levels of exploration in successive generations by diversifying the survivor of the population [52]. For instance, in *crowding* the basic principle is that offspring should replace similar individuals from the previous generation. Some of the most popular crowding methods are the following:

- In *Mahfoud's deterministic crowding* [30] (DETCR), each pair of parents and their corresponding offspring are paired by minimizing the sum of the distances between them. Then, each offspring survives if it is at least as good as the parent.
- *Probabilistic crowding* [34] (POBCR) operates similarly to DETCR but it uses a non-deterministic rule to establish the winner, with each individual's survival probability being proportional to its fitness value. Scaled variants of probabilistic crowding (SPOBCR) have also been devised [3].
- In *adaptive generalized crowding* [35] (AGCR) the selection pressure is adapted to the needs of the different stages. Two different mechanisms to adapt the selection pressure were devised: a self-adaptive and an adaptive scheme. In the adaptive scheme the entropy of the population is used to alter the selection pressure. In this chapter, this latter version is the one that we have adopted.
- *Restricted Tournament Selection* [14] (RTS) is a popular steady-state scheme in which after each new individual (C) is created, CF individuals from the current population are selected at random. Then, the best individual between C and its most similar individual in the selected set survives. In case of a tie, the offspring is preserved.

Several replacement strategies that do not rely on the principles of crowding have also been devised. In some methods, diversity maximization is considered as an objective that is combined with the original objective to calculate the fitness of

each individual. However, since both objectives measure different things, such a combination is complex and problem-dependent. As a result, other ways of combining them have been proposed. For instance, in COMB [54] the individuals are sorted by their contribution to diversity and by their original cost. Then, the rankings of the individuals, instead of the original objective values, are used to calculate a score using two parameters (N_{Close} and N_{Elite}). In each step of the replacement phase, the individual with the lowest score is erased and the ranks are recalculated. This process is repeated until the desired population size is reached. The *contribution-diversity replace-worst* method [27] (CD/RW) also uses diversity but in a different way. CD/RW is a steady-state EA in which a new individual enters the population by replacing another one that is worse both in diversity contribution and quality. If such an individual does not exist, the *replace-worst* strategy (RW) is applied, i.e., the best individuals are preserved. Finally, using concepts that arise in the field of multi-objective optimization to consider both the original objective and the diversity contribution of each individual is also a plausible choice [6, 39]. In these kinds of schemes, which are usually termed diversity-based multi-objective EAs [48], the diversity contribution is referred to as the auxiliary objective. Several variants of these kinds of schemes have been devised. For instance, several different ways of calculating the auxiliary objective have been proposed [51]. Among them, one of the most popular is probably the *distance to the closest neighbor* (DCN) metric [6], which is used in the extensions applied in this chapter. In DCN, the auxiliary objective of a given individual is calculated as its distance to the closest member in a reference set. Note that this principle can be used to modify several components of the EAs. A uniprocess-driven replacement-based approach (MULTI) based in these principles is proposed in [49].

3 A Novel Diversity-Based Replacement Strategy

The MULTI_DYN replacement strategy analyzed in this chapter is an extension of the strategy presented in [49] (MULTI). The MULTI replacement strategy (Algorithm 1) operates as follows. First, the population of the previous generation and the offspring are joined. Then, the best individual—taking into account the original objective value—is selected to survive. Then, until the new population is filled with N individuals, the following steps are executed. First, each individual's contribution to diversity is calculated. The reference set used in this calculation is the set of already selected individuals, i.e., for each pending individual, the distance to the nearest survivor already selected is taken into account. Then, considering the individuals that have not been selected, the non-dominated front is calculated. Finally, a non-dominated individual is selected at random to survive. The two most expensive steps in each iteration of the survivor selection scheme are the identification of the non-dominated individuals and the calculation of DCN. The non-dominated individuals can be identified in $O(N \log N)$, whereas the calculation of DCN depends on the problem. In each step the reference set consists of an additional individual, meaning that in each step $O(N)$ distances are calculated. The complexity associated

Algorithm 1 MULTI survivor selection scheme

```

1: CurrentMembers = Population  $\cup$  Offspring
2: Best = Individual with best  $f(x)$  in CurrentMembers
3: NewPop = { Best }
4: CurrentMembers = CurrentMembers  $\setminus$  { Best }
5: while ( $|\text{NewPop}| < N$ ) do
6:   Calculate DCN of CurrentMembers, taking NewPop as reference
7:   ND = Non-dominated individuals of CurrentMembers (without repetitions)
8:   Selected = Randomly select an individual from ND
9:   NewPop = NewPop  $\cup$  Selected
10:  CurrentMembers = CurrentMembers  $\setminus$  {Selected}
11: end while
12: Population = NewPop

```

Algorithm 2 MULTI_DYN survivor selection scheme

```

1: CurrentMembers = Population  $\cup$  Offspring
2: Best = Individual with best  $f(x)$  in CurrentMembers
3: NewPop = { Best }
4: CurrentMembers = CurrentMembers  $\setminus$  { Best }
5: while ( $|\text{NewPop}| < N$ ) do
6:   Calculate DCN of CurrentMembers, taking NewPop as reference
7:    $D = D_I - D_I * \frac{T_{Elapsed}}{T_{End}}$ 
8:   Penalize(CurrentMembers, D)
9:   ND = Non-dominated individuals of CurrentMembers (without repetitions)
10:  Selected = Randomly select an individual from ND
11:  NewPop = NewPop  $\cup$  Selected
12:  CurrentMembers = CurrentMembers  $\setminus$  {Selected}
13: end while
14: Population = NewPop

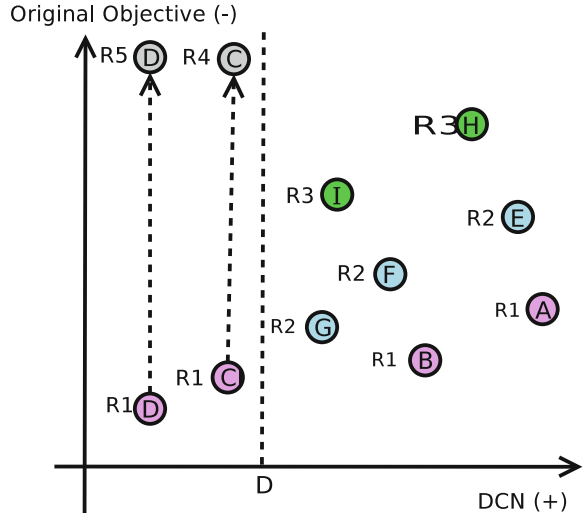
```

with calculating the distance metric depends on how the metric is defined, but in our cases they are $O(V)$, where V is the number of variables. Since N steps are done in each generation, the complexity of the survivor selection scheme is $O(VN^2 + N^2 \log N)$. Finally, we would like to remark that in every problem considered in this chapter, the time associated with the replacement phase is negligible. In every case, the much more expensive local search integrated in our schemes consumes the majority of the time.

The proposal used in this chapter (MULTI_DYN) [52]—see Algorithm 2—extends the MULTI strategy by adapting the balance between exploration and exploitation to the different optimization stages. In order to effect this adaptation, the stopping criterion, as well as the elapsed time, are used as inputs to the replacement strategy. Specifically, for shorter stopping criteria the method induces a faster reduction in diversity than for longer stopping criteria. The adaption is done by defining a dynamic penalty approach that is independent of the scale of the original objective. In this way, the effort required to adapt MULTI_DYN to different problems is highly reduced.

One of the keys behind the development of MULTI_DYN is that individuals that contribute too little to diversity should not survive regardless of their original objective

Fig. 1 Effect of the penalty approach with distance D



value. In our approach, this principle is realized by setting the original objective of individuals that contribute too little to diversity to a very low quality value. For instance, in a minimization function the maximum number that can be represented in the data type associated with the objective function might be used, whereas in a maximization function, the minimum representable number might be considered. This is illustrated in Fig. 1. In this figure, the value D represents the minimum DCN required to avoid the penalty. Any individual whose DCN value is lower than D is penalized. As a result, the non-domination rank (shown at the left of each individual) of the penalized individuals might increase. Obviously, the penalized individuals will not belong to the non-dominated front so they will not survive, unless every pending individual has been penalized. Note that the computational complexity of the survivor selection strategy does not change due to the inclusion of the penalty method.

While the above approach is quite sensible, one of the key choices is how to set the value D . Taking into account the benefits that can be attained by adapting the balance between exploration and exploitation to the needs of the different stages of the optimization, it is clear that the value of D should vary during the optimization process. Specifically, this value should decrease as the stopping criterion is approached with the aim of inducing a higher degree of intensification. In our scheme, an initial D_I value must be selected by the user. Then, a linear reduction of D is carried out in such a way that by the end of the execution, the resulting value is 0. In this chapter, the stopping criterion is set by time. Thus, if T_{End} is the stopping criterion and $T_{Elapsed}$ the elapsed time, D can be calculated with (1). Note that some more complex ways of setting D have also been tested [52]; however, these methods are more complex to use and they do not provide important benefits, so the linear reduction is used in this chapter.

Algorithm 3 Lamarckian Memetic Algorithm

- 1: **Initialization:** Generate an initial population P_0 with N individuals. Assign $t = 0$.
 - 2: **Local Search:** Perform a local search for every individual in the population.
 - 3: **while** (not stopping criterion) **do**
 - 4: **Evaluation:** Evaluate all individuals in the population.
 - 5: **Mating selection:** Perform binary tournament selection on P_t in order to fill the mating pool.
 - 6: **Variation:** Apply genetic operators to the mating pool to create a child population CP .
 - 7: **Local Search:** Perform a local search for every individual in the offspring.
 - 8: **Survivor selection:** Apply the replacement scheme to create P_{t+1} .
 - 9: $t = t + 1$
 - 10: **end while**
-

$$D = D_I - D_I * \frac{T_{Elapsed}}{T_{End}} \quad (1)$$

4 Proposals

We used a common framework to address the three different optimization problems covered in this chapter. In every problem considered, it is clear that incorporating some intensification mechanisms is quite useful. For instance, in the case of the competition of the 2DPP problem, the best three schemes used memetic algorithms, whereas the surveys provided in [29, 47] show that most of the promising schemes for the FAP combine population-based and trajectory-based optimization. Additionally, in the case of Sudoku, some of the most promising schemes are also hybrid approaches [32, 43] that combine a global search with some specific mechanisms to induce additional intensification. As a result, in this chapter a memetic algorithm is used. Algorithm 3 shows the approach applied. Note that it is a fairly standard memetic algorithm, where the local search is incorporated after each individual is created, i.e. in the creation of the initial population and after the variation stage. Specifically, the local search is applied to all the offspring. Note that some schemes to balance the resources granted to the genetic and local searches have been devised [16]. In these cases, a local search is usually applied to a subset of individuals or only in some generations. For instance, one typical approach is to apply a local search to the best individuals with higher probabilities. This goes against some of the principles of our scheme since, due to the way in which the replacement is carried out, any individual kept in the population is promising, so it is important to intensify in every region that is maintained; otherwise, some regions might be abandoned prematurely. In any case, we conducted some experiments by trying to incorporate some of these ideas, but they were unsuccessful.

Note that in the problems at hand, our local search is always a stochastic hill, i.e. the neighbors of a given candidate solution are considered in a random order and only improvement movements are accepted. The local search continues until a local

optimum is reached. Note that some more complex trajectory-based metaheuristics, such as tabu search or simulated annealing, can be incorporated in this step. While these schemes are promising, they are more expensive because they do not finish when a local optimum is found. Thus, more complex mechanisms for specifying the resources granted to global and local searches should be included. As a result, the application of these schemes is left for future work.

In order to test different ways of alleviating the problems associated with premature convergence, several schemes are included in Algorithm 3. In all of the problems tested, a variant that applies the `MULTI_DYN` scheme is considered, i.e. `MULTI_DYN` is selected as the survivor selection scheme. Depending on each problem, other schemes are also incorporated. The set of schemes applied and their corresponding parametrizations are given in Sect. 5 for each problem.

In order to properly adapt the previous framework to a given problem, the following components are required. First, the representation of the individuals and an associated distance metric must be given. Second, the objective function used to evaluate each individual must be established. Third, the variation stage, i.e. the crossover and mutation operators, must be selected. Finally, the neighborhood must be defined. In the following sections, we describe the components selected for each of the problems addressed in this chapter.

4.1 *Sudoku*

4.1.1 Problem Definition

The Sudoku puzzle is a popular logic puzzle consisting of an $N^2 \times N^2$ grid that is subdivided into N^2 blocks of size $N \times N$. An initial board with numbers in some cells is given and the objective is to fill the remaining cells with digits from 1 to N^2 such that each block, row and column can contain only one instance of each digit. The most popular variant of Sudoku involves a 9×9 grid. The reason to use so small grids is probably that since this problem is NP-Complete [56], large grids might be too complicated to be solved by hand.

In recent years, Sudoku has been widely used to test the abilities and performance of different optimization frameworks. For instance, the most popular metaheuristics have been applied to Sudoku solving [7, 17, 37]. In the particular case of EAs, their basic variants can only solve simple Sudoku puzzles [33], as they face significant issues when tackling more difficult puzzles. In fact, even with the incorporation of more complex components some of these issues also appear [25, 38, 55]. In order to alleviate these drawbacks, hybrid variants and ad-hoc genetic operators have been proposed [32]. Thus, most efforts have focused on incorporating problem-specific knowledge into the design of EAs. However, as the experimental validation in this chapter shows, successful simpler EAs that do not incorporate ad-hoc knowledge can be designed by properly managing diversity.

4.1.2 Selected Components

In the case of the Sudoku problem, we have selected very straightforward components. One of the most popular representations is used [33]; specifically, an individual consists of a set of sub-chromosomes, each of which is a permutation of the numbers absent in a block. Thus, the constraints associated with the numbers given in the initial board and with the blocks are fulfilled by any individual. The fitness function is calculated as the sum of two terms: the first one is the number of repetitions that appear in the rows and columns; the second one is the number of conflicts with the numbers given in the initial board multiplied by 100. In the case of the distance metric, the Hamming distance is applied, i.e. the distance between two individuals is the number of cells that differ.

The variation scheme applies simple mutation and crossover operators. The crossover operator uniformly exchanges complete blocks between solutions [11]. The mutation operator iterates over each gene and performs a swap with another element selected at random from its corresponding sub-chromosome with probability p_m [33]. Note that these operators have only been successful with easy Sudoku puzzles, so they are suitable choices to show that general and simple operators can be successfully applied by properly managing diversity. Finally, a neighbor of a candidate solution is generated by swapping two elements of a block.

4.2 Two-Dimensional Packing Problem

4.2.1 Problem Definition

The two-dimensional packing problem (2DPP) variant used in this chapter was defined in the GECCO 2008 competition session. Problem instances are described by the following data:

- The sizes of a rectangular grid: X, Y .
- The maximum number that can be assigned to each grid position: M . The value assigned to each grid location is an integer in the range $[0, M]$.
- The score associated with the appearance of each pair (a, b) where $a, b \in [0, M]$: $v(a, b)$.

A candidate solution is generated by assigning a number to each position on the grid. Thus, the search space consists of $(M + 1)^{X \cdot Y}$ candidate solutions. The objective of the problem is to pack a grid so that the sum of the point scores for every pair of adjacent numbers is maximized. Two positions are considered to be adjacent if they are neighbors in the same diagonal, row, or column of the grid. A pair cannot be collected more than once. Mathematically, the objective is to find the grid G which maximizes the objective function f :

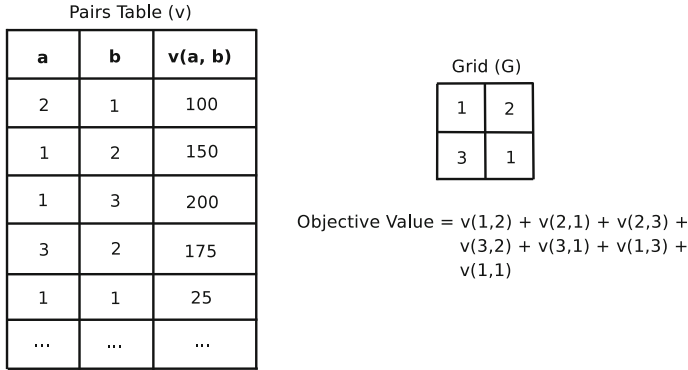


Fig. 2 Assignment of the original objective function for the 2DPP

$$f = \sum_{a=0}^M \sum_{b=0}^M v_2(a, b) \tag{2}$$

where

$$v_2(a, b) = \begin{cases} 0 & \text{if } (a, b) \text{ are not adjacent in } G \\ v(a, b) & \text{if } (a, b) \text{ are adjacent in } G \end{cases} \tag{3}$$

Figure 2 illustrates the assignment of the objective function of a candidate solution for a 2 × 2 grid. Note that although the pairs (1, 2) and (2, 1) are repeated in the grid, they are only considered once.

One of the reasons for selecting the 2DPP is that this problem was tackled by several research groups during the competition. In the competition, the best results were reported by using memetic algorithms [18, 23], results that could be further improved by taking into account the diversity of the population [44, 46]. The current best result for this problem is offered by the MULTI_DYN scheme [52]. The initial results obtained in [52] are extended in this chapter and new best-known solutions are found for the largest instances.

4.2.2 Selected Components

In our proposal, candidate solutions are encoded using a two-dimensional chromosome. Specifically, there are as many genes as there are cells in the grid, and each gene can take a value in the range [0, M]. A brief description of the variation operators and the definition of the neighborhood are attached. For a more comprehensive description, readers are referred to [23]. The variation stage is based on the application of crossover and mutation. Crossover is applied with probability p_c and two different parameters (min_p_m and max_p_m) are used to control the mutation. The crossover operator is the 2D Sub-String Crossover (SSX) [15], which is an extension of

one-point crossover for two-dimensional chromosomes. The mutation operator applied is the *Uniform Mutation with Domain Information* (UMD). First, a random number (ap_m) between min_p_m and max_p_m is generated. Then, each gene is mutated with a probability ap_m . In order to make new assignments, a random value is selected from among those that increase the objective value. If such a value does not exist, a random value between 0 and M is used.

In our proposal, Hamming distances are used to calculate the DCN value. As a result, the most distant individuals are at a distance equal to $X \times Y$. Taking this into account, relating the value of D_I to the number of genes (G) seems appropriate. Particularly, in this chapter the value of D_I is set to a percentage ($R_I \times 100$) of G .

The definition of neighborhood applied considers a new neighbor for each pair of adjacent grid positions (i, j) and (k, l) . Each neighbor is generated by assigning the best possible values to the positions (i, j) and (k, l) while keeping intact the values in all other grid locations. A pruning procedure is applied so as to efficiently assign the best values to both locations. Finally, the fitness function is the score associated to each candidate solution, which is evaluated with Eq. (2)

4.3 Frequency Assignment Problem

4.3.1 Problem Definition

The Frequency Assignment Problem (FAP) is a very popular NP-hard combinatorial optimization problem with practical applications that emerges in the deployment of networks in different environments. The FAP involves different objectives, features, and constraints. This has resulted in quite a large number of different mathematical formulations being defined [1]. In recent years, the basic FAP formulation has been widely expanded in order to address real-world issues [21]. For this chapter, we selected a commonly applied definition and used some popular instances inspired on realistic global system for mobile communications (GSM) networks [36]. In this definition, an instance is given by a set of transmitters, a set of constraints and the number of admissible channels. The constraints define the minimum distance allowed between the channels assigned to pairs of transmitters. When a given constraint is violated, a given penalty term associated with each constraint is added to the cost function. The objective of the problem is to find a complete assignment that minimizes this cost function.

Several different meta-heuristics have been applied to the FAP [28]. Initially, in the case of the instances tackled in this chapter, some methods based on a tabu search yielded promising results [36]. Currently, the best reported results have been obtained using path relinking [22]. This successful variant of path relinking incorporates a special mechanism to preserve the population diversity.

4.3.2 Selected Components

In this problem we use a straightforward representation of candidate solutions. Specifically, the number of genes in the chromosomes is equal to the number of transmitters, and each one represents the channel assigned to a transmitter. As in previous cases, the Hamming distance was used to calculate the distance between individuals. Finally, the fitness function is just the cost associated with each candidate solution, i.e. the sum of the penalties associated with the constraints that are violated.

The genetic operators are more complex because they are ad-hoc operators for the FAP. Specifically, they are extensions of the ones that were proposed in [45] for a different variant of the FAP. The crossover is the *Multi-Interference-based Crossover*, which operates as follows. First, a transmitter t is selected at random. Then, every gene associated with a transmitter that interferes with t , including the gene that represents t , is marked. This process is repeated a number of times that is selected randomly between one and a number set by the user (R), which in this chapter is eight. However, in the subsequent iterations, instead of selecting t at random from the complete set of transmitters, it is selected, without repetition, from those that were previously marked. Finally, the parents swap the channels assigned to the transmitters that were marked to generate the offspring. The mutation operator is the *Neighborhood-based Mutation*. First, a transmitter t is selected randomly and it is mutated. Then, the transmitters that interfere with t are mutated with a probability p_m . The above step is repeated R times, but in the subsequent iterations the transceiver is randomly selected from among those that interfere with transmitters mutated in previous iterations. Thus, similarly to the crossover operator, the mutation scheme focuses on altering subsets of transmitters affected by common constraints.

Finally, a simple definition of neighborhood is used. A neighbor of a candidate solution is generated by reassigning the channel of two transmitters that might potentially have interference between them, while leaving intact the assignments of the remaining ones.

5 Experimental Validation

In this section we describe the experiments conducted with our memetic algorithm for the three different optimization problems considered, and discuss the results obtained by using a diverse set of methods to control convergence. In every problem, the memetic algorithm with the MULTI_DYN scheme is applied. Additionally, some of the more mature schemes described above are also tested. A subsection is included for each of the problems and in each case, the parametrizations and methods used in the comparisons are detailed.

The optimization schemes were implemented in METCO (*Metaheuristic-based Extensible Tool for Cooperative Optimization*) [24]. The experiments were run on dual processor machines with 32Gb RAM. Since the algorithms considered in this

study are stochastic, each execution was repeated 30 times. In order to compare the results (fitness and speed), a series of statistical tests that relied on a guideline similar to that applied in [52] was conducted. Specifically, the following tests were applied, assuming a significance level of 5%. First, a *Shapiro–Wilk test* was applied to check whether or not the results followed a Gaussian distribution. If they did, the *Levene test* was used to check for the homogeneity of the variances. If the samples had equal variance, an *ANOVA test* was done; if not, a *Welch test* was performed. For non-Gaussian distributions, the non-parametric *Kruskal–Wallis* test was used to test whether samples were drawn from the same distribution.

5.1 Sudoku

In the case of the Sudoku problem, the analyses were performed with 26 different puzzles.² Twenty puzzles were generated using two of the most popular web pages devoted to Sudoku [57, 58]. In the case of the Sudoku puzzles generated in [58] (tagged as SS), the hard level was used, whereas for the ones generated in [57] (tagged as SW), the evil level was selected. The three puzzles tagged as SD in [43] were also included in our set. Note that one of them is the popular AI-Escargot puzzle. Finally, the last three are now regarded as the most difficult puzzles [59]. Two of them were designed by David Filmer, whereas the last one was proposed by Arto Inkala. Note that in the Sudoku puzzle it makes no sense to compare the fitness values when the problem is not solved, i.e. it is no more valuable to obtain a candidate solution with two conflicts than one with four conflicts. Therefore, our comparisons are done based on the success rate achieved. Since the time factor is also important, the time required to solve the puzzles is also taken into account.

In every variant of our memetic algorithm, typical values for the population size (N), mutation probability (p_m) and crossover probability (p_c) were selected. Specifically, they were set to 100, 0.01 and 0.8, respectively. Additionally, the stopping criterion was set to 5 min. The following variants were tested for this problem. First, the memetic scheme was tested with two replacement strategies, the generational scheme with elitism (GEN_ELIT) and the replace-worst strategy (RW), which did not include any special mechanism to delay convergence. In GEN_ELIT, the best individual of the population is copied to the next generation while the rest of the population is created with the variation scheme. In RW the offspring and the previous population are joined in a set and the best N individuals survive. Additionally, seven schemes from among those previously discussed, and that incorporate mechanisms to delay convergence, were tested. They are the following: MULTI_DYN, CLR, RTS, DETCR, SAW-TOOTH GA, COMB and CD/RW. The parametrization of these methods is shown in Table 1. In order to select the parametrization of each scheme, some initial

²The Sudoku puzzles are available at <http://www.cimat.mx/~carlos.segura/Sudoku/SudokuPuzzles.tar.gz>.

Table 1 Parametrization of the methods applied in the Sudoku Problem

Method	Parametrization
MULTI_DYN	$D_I = 20$
GEN_ELIT	No parametrization required
RW	No parametrization required
CLR	$\sigma = 10, W = 1$
RTS	$CF = 75$
DETCR	No parametrization required
SAW- TOOTH GA	$D = 99, P = 50$
COMB	$N_{Close} = 3, N_{Elit} = 8$
CD/RW	No parametrization required

experiments were done with the last six puzzles selected. The parameter values that maximized the success rate were selected for the final experimentation.

Table 2 shows the results obtained with each scheme in every puzzle analyzed. Specifically, it shows the success rate (SR) and the time in seconds (T) required to obtain a success rate of at least 50%. A dash is shown in those cases where the success rate was lower than 50%. The methods are sorted by the number of successful runs.

It is important to note that the worst schemes were the two methods that did not incorporate any mechanisms to delay convergence, which quite clearly shows the important benefits provided by any of the mechanisms that delay convergence. However, there are important differences in their behaviors. The scheme that provided the highest success rate was the MULTI_DYN method. The results provided by RTS, COMB, SAW- TOOTH GA and CLR were also quite good. In fact, they only encountered difficulties with SS4 and with the puzzles generated by Arto Inkala and David Filmer. Furthermore, they converged faster than MULTI_DYN in most cases, so they might be more useful than MULTI_DYN for the easiest Sudoku puzzles. Some analyses were carried out to compare the times required by MULTI_DYN and these other schemes. Among them, the best one was CLR. The statistical tests confirmed that CLR was faster than MULTI_DYN in 20 Sudoku puzzles, whereas the opposite never happened.

In light of these results and those in Table 2, it is clear that in the less challenging instances, the CLR strategy provides important benefits; whereas in the most difficult puzzles, the MULTI_DYN method is preferred. Note that this behavior also appeared when MULTI_DYN was applied to the Traveling Salesman Problem [50], where the advantages were clearer when dealing with large-scale and difficult instances. Additionally, we would like to note that all the schemes were also executed by setting the stopping criterion to 30 min with the Filmer1 and Filmer2 puzzles, which are the cases where the EAs encountered more difficulties. The only method that was able to obtain a 100% success rate was MULTI_DYN, whereas the second best method (RTS) obtained a success rate of 90% with Filmer1 and 70% with Filmer2. This last experiment also confirms the robustness of the MULTI_DYN scheme.

Table 2 Summary of the results obtained by the different tested schemes (5 min)

Inst.	MULTI_DYN		RTS		COMB		SAW-TOOTH		CLR		DETCR		CD/RW		RW		GEN_ELIT	
	SR	T	SR	T	SR	T	SR	T	SR	T	SR	T	SR	T	SR	T	SR	T
SW1	100	3.6	100	2.4	100	3.7	100	2.1	100	0.6	100	1	73.3	0.8	63.3	0.5	53.3	3.2
SW2	100	2.4	100	2	100	1.9	100	0.6	100	0.3	100	0.5	96.7	0.6	100	0.3	100	0.5
SW3	100	5.4	100	3	100	2.6	100	0.8	100	0.5	100	0.9	83.3	0.9	63.3	3.1	73.3	1.4
SW4	100	1.2	100	1.2	100	1.2	100	0.6	100	0.3	100	0.4	96.7	0.5	80	0.5	76.7	0.7
SW5	100	4.8	100	3.6	100	2.9	100	1	100	0.8	96.7	1	70	2.9	53.3	6.9	66.7	8
SW6	100	3.7	100	5.1	100	5.2	100	8.8	100	2.1	96.7	2.1	33.3	-	13.3	-	20	-
SW7	100	2.3	100	1.8	100	1.3	100	0.5	100	0.3	100	0.4	90	0.5	80	0.5	83.3	0.3
SW8	100	2.1	100	1.9	100	1.8	100	0.8	100	0.4	100	0.7	86.7	0.8	86.7	0.6	80	0.6
SW9	100	5.4	100	2.4	100	3.6	100	1	100	0.7	100	1	70	2.1	43.3	-	50	1.6
SW10	100	4.2	100	3	100	2.3	100	0.7	100	0.5	100	0.8	96.7	0.7	90	0.8	90	0.9
SS1	100	1.8	100	1	100	1.1	100	0.4	100	0.2	100	0.4	96.7	0.7	93.3	0.5	93.3	0.7
SS2	100	12.5	100	4.8	100	8.6	100	11.3	100	1.9	96.7	1.5	16.7	-	36.7	-	10	-
SS3	100	4.8	100	4.1	100	2.2	100	2.2	100	0.9	93.3	1.1	73.3	5.2	36.7	-	46.7	-
SS4	100	63.4	96.7	37.7	90	87.4	66.7	201.5	16.7	-	13.3	-	10	-	6.7	-	0	-
SS5	100	5.9	100	3.2	100	4.1	100	1.2	100	0.7	100	1.8	60	1.4	43.3	-	56.7	13.6
SS6	100	7.1	100	3.8	100	3.2	100	1.1	100	0.6	100	1.3	83.3	1.1	70	6	80	1.2
SS7	100	1.4	100	1.6	100	1.2	100	0.5	100	0.4	100	0.5	90	0.9	73.3	0.6	70	0.7
SS8	100	2.2	100	1.6	100	2.2	100	0.6	100	0.5	100	0.6	93.3	0.8	66.7	0.9	70	0.9
SS9	100	1.5	100	2.3	100	2	100	0.6	100	0.4	100	0.6	86.7	1.1	73.3	0.9	66.7	12
SS10	100	5.4	100	4.4	100	6.5	100	2.3	100	0.6	96.7	1.2	86.7	4.2	40	-	26.7	-
SD1	100	14.5	100	12.5	100	23	100	10.2	100	6.1	90	5	23.3	-	13.3	-	10	-
SD2	100	7.6	100	22.6	100	10.2	100	6.9	100	4.4	96.7	2.9	30	-	33.3	-	16.7	-
SD3	100	10.8	100	8.6	100	6.6	100	5.3	100	3	100	3.9	60	55.2	30	-	23.3	-
Inkala	100	67.3	86.7	71.6	63.3	199.2	76.7	147.1	80	154.6	30	-	3.3	-	0	-	0	-
Film1	83.3	155	33.3	-	16.7	-	20	-	36.7	-	0	-	3.3	-	0	-	3.3	-
Film2	63.3	208	20	-	23.3	-	16.7	-	0	-	3.3	-	6.7	-	0	-	0	-

Table 3 Parametrization of the methods applied in the 2DPP Problem

Method	Parametrization
MULTI_DYN	Several parametrizations
MULTI	No parametrization required
GEN_ELIT	No parametrization required
CD/RW	No parametrization required
CLR	$\sigma = 0.20 \times X \times Y$, $W = \{1, 2, 5\}$
DETCR	No parametrization required
SPOBCR	No parametrization required
AGCR	$\phi = \{0.25, 0.75\}$
RTS	$CF = \{2, 5, 10, 25, 50\}$
COMB	$N_{Close} = 3$, $N_{Elit} = 8$
RW	No parametrization required
CHC	$p_m = 0.35$
SAW-TOOTH GA	$D = 99$, $P = 50$

5.2 Two-Dimensional Packing Problem

In the case of the 2DPP, the experimentation presented in this chapter is an extension of the one developed in [52]. The most important achievement is that new best-known solutions could be obtained for the most complex instances of this problem. Some new tests that better show the benefits of MULTI_DYN are also included. Since one of the objectives of this chapter is to demonstrate the robustness of the MULTI_DYN method and its ability to perform properly in different problems with minor effort, some of the most illustrative results obtained previously for this problem are also shown. The ten instances that were created in [52] are used in our performance evaluation. They were created with a random generator that allows selecting different parameters to alter the difficulty and size of the instances created.

As in the previous problem, in order to prove the proper performance of our memetic algorithm with MULTI_DYN, an extensive comparison with other more mature methods was carried out. In particular, the following schemes were used: CD/RW, CLR, DETCR, SPOBCR, AGCR, RTS, COMB, RW, CHC and Saw-Tooth GA. In addition, a generational scheme with elitism (GEN_ELIT) was used. As in previous studies [44], the parameters p_c , min_p_m and max_p_m , which are associated with the genetic operators, were set to 1, 0.1 and 0.15, respectively, while the population size was set to 50. The parametrization of the different methods is included in Table 3. In the methods where several values are used for a specific parameter, these are expressed as comma-separated values, and in order to denote the method with its specific parametrization, the name of the method is followed by the value of the parameter. For instance, CLR_5 means that the parameter W is set to 5.

Table 4 Statistical comparison of every considered scheme in 24h

	SMALL			MEDIUM			LARGE			GECCO			TOTAL		Score
	↑	↓	↔	↑	↓	↔	↑	↓	↔	↑	↓	↔	↑	↓	
MULTI_DYN	57	0	0	57	0	0	57	0	0	19	0	0	190	0	190
MULTI	47	4	6	48	5	4	51	6	0	17	2	0	163	17	146
CHC	39	12	6	47	7	3	54	3	0	18	1	0	158	23	135
COMB	46	4	7	39	12	6	39	9	9	12	4	3	136	29	107
CLR_1	33	14	10	41	12	4	40	9	8	15	3	1	129	38	91
CLR_2	27	18	12	35	17	5	39	9	9	12	4	3	113	48	65
GEN_ELIT	15	32	10	35	17	5	37	10	10	12	3	4	99	62	37
SPOBCR	40	9	8	52	3	2	8	48	1	3	16	0	103	76	27
CLR_5	21	22	14	27	24	6	30	20	7	11	7	1	89	73	16
CD/RW	3	43	11	14	25	18	29	20	8	11	4	4	57	92	-35
Saw-Tooth GA	16	27	14	15	25	17	15	37	5	5	9	5	51	98	-47
RTS_10	10	35	12	11	27	19	21	27	9	5	9	5	47	98	-51
RTS_25	16	29	12	14	26	17	15	38	4	5	9	5	50	102	-52
AGCR_0.25	50	3	4	3	37	17	3	54	0	1	18	0	57	112	-55
RTS_50	27	19	11	8	33	16	12	45	0	4	15	0	51	112	-61
RTS_5	3	43	11	8	33	16	22	23	12	5	9	5	38	108	-70
RTS_2	3	44	10	8	33	16	20	24	13	5	9	5	36	110	-74
RW	3	45	9	8	31	18	19	28	10	5	9	5	35	113	-78
DETCR	23	19	15	5	51	1	6	50	1	2	17	0	36	137	-101
AGCR_0.75	0	57	0	0	57	0	0	57	0	0	19	0	0	190	-190

In our first experiment, the memetic algorithm was executed with the above schemes, as well as with the MULTI_DYN replacement strategy, by setting the stopping criterion to 24h. The parameter R_I in MULTI_DYN was set to 0.5. Additionally, the MULTI scheme was also executed, which is similar to MULTI_DYN with R_I set to 0. In order to obtain a ranking of the different approaches, pairwise statistical comparisons between the 20 configurations were carried out, meaning that 190 statistical tests were done for each scheme. Table 4 shows, grouped by category, the results of these statistical tests. The groups “small”, “medium” and “large” contain three instances, while the last column (“GECCO”) refers to the instance that was used during the competition to perform the final comparisons. For each category, columns with the symbol \uparrow refer to the number of cases where the model listed in each row is statistically better. The number of cases where it is worse is shown in the column with the symbol \downarrow . Finally, the column with the symbol \leftrightarrow denotes the number of cases where the differences are not statistically significant. In addition, a score is assigned to each model that is equal to the number of cases where the model was superior minus the number of cases where the model was inferior. The superiority of the MULTI_DYN model is quite clear. In fact, it was superior to all the remaining models in every instance.

In order to better understand the reasons for the superiority of MULTI_DYN, it is important to analyze the diversity induced by the different schemes throughout the optimization process. Figure 3 shows the evolution of the mean entropy for all of the

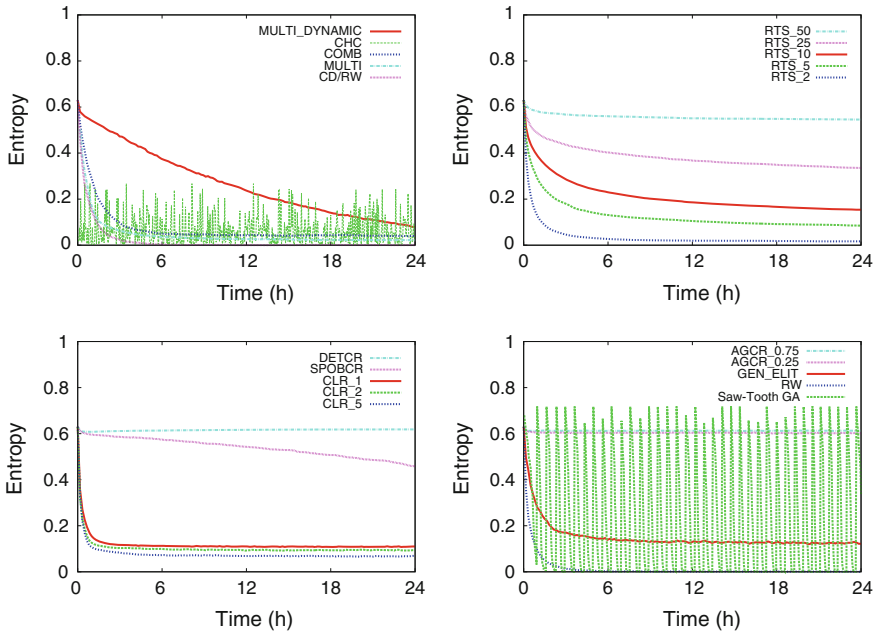


Fig. 3 Evolution of the mean entropy for all of the schemes considered (GECCO instance)

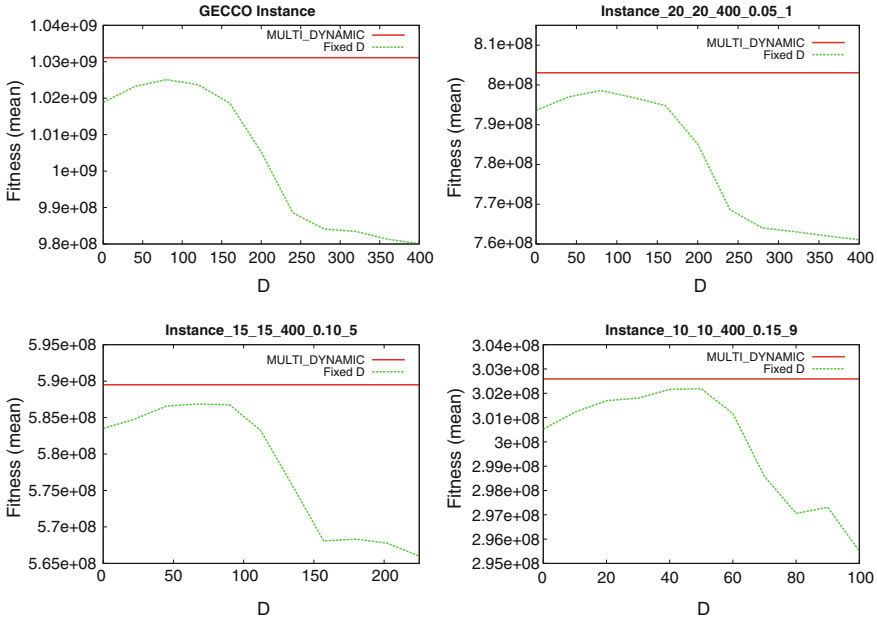


Fig. 4 Fitness (mean) achieved with fixed D values and with MULTI_DYN

schemes considered in the GECCO instance. In the case of CHC and Saw-Tooth GA, only one execution was used to acquire the data because the exact times where the restarts are triggered differ for each run, meaning that the mean of several executions is less representative of the behavior of these schemes. MULTI_DYN is the only model that induces a slow but continuous decrease in the entropy, meaning that it is the only scheme where the balance between exploration and exploitation is altered gradually, as expected.

In order to better show the advantages provided by the linear decrease in D , additional experiments were conducted that consider fixed values of D . Specifically, for each instance, 11 fixed values equidistributed among 0 and the number of variables of the corresponding instance were considered. Figure 4 shows, for four different instances, the mean of the fitness obtained with MULTI_DYN and with the schemes that consider a fixed D . In MULTI_DYN, R_I was set to 0.6 in every instance, which proved to be quite a robust value in [52]. The advantages of using a dynamic D are quite clear. In fact, in every instance the mean value attained by MULTI_DYN is higher than the means obtained with any fixed value of D . Similar analyses were done with the remaining instances and in every case the statistical tests show that the results obtained by MULTI_DYN are superior. We also note that in the small instances, the differences are not so large. In fact, the p-values obtained when comparing the results of MULTI_DYN to those output by the best fixed D values are only slightly below 0.05, meaning that the benefits provided by MULTI_DYN are more important in the most complex instances.

Table 5 Results obtained by MULTI_DYN in the long-term (3 Days)

Instance	Best	Mean	Median	Previous best
GECCO	1, 041, 931, 347	1, 037, 842, 560	1, 037, 529, 370	1, 039, 822, 096
20_20_400_0.05_1	814, 869, 977	809, 457, 130	809, 218, 641	811, 360, 334
20_20_400_0.10_2	1, 057, 813, 297	1, 052, 064, 375	1, 052, 412, 440	1, 053, 754, 765
20_20_400_0.15_3	1, 233, 613, 667	1, 227, 245, 581	1, 227, 541, 149	1, 229, 316, 361

In order to see the specific mean and best values obtained by 24h executions of the MULTI_DYN scheme, the readers are referred to [52]. Note that while the MULTI_DYN method provided the currently best-known solution for the ten instances above, it is important to analyze whether larger executions can provide additional benefits. In order to carry out this experiment, the four most complex instances were selected, i.e. the GECCO instance and the ones in the large group. In this case, only the MULTI_DYN scheme was used and the stopping criterion was set to 3 days. Table 5 shows, for each instance, the mean, median and best of the solutions attained. In addition, it also shows the previously attained best-known solution as reported in [52]. In every instance, the best-known results could be improved substantially. For instance, in the GECCO instance, the best result could be improved from 1, 039, 822, 096 to 1, 041, 931, 347, which represents a significant advance. This probably means that due to the huge search space of these instances, there is already room for improvement. Considering how the execution times of the runs carried out in this chapter are already quite long, it seems particularly important to develop parallel models capable of managing diversity in a similar way to the sequential case but in parallel environments.

5.3 Frequency Assignment Problem

In the case of the FAP, the main objective was to improve all of the results obtained in literature through the application of MULTI_DYN. Additionally, in order to show that the benefits come from the explicit control of diversity, the GEN_ELIT replacement phase was also used, which is probably the most popular replacement phase. In both schemes the same parametrization was considered. First, the population size, crossover probability and mutation probability were set to 50, 1 and 0.05 respectively. The R parameter, which is used in the mutation operator, was set to eight. Finally, the stopping criterion was set to 48 h. Additionally, in the case of MULTI_DYN, D_i was set to 0.75 multiplied by the number of transmitters of the considered instance. In this chapter, a subset of the problem instances provided in [36] was used. Specifically, these are the ones whose tags start with GSM2, which are a set of instances that were created with realistic data from GSM networks. The best results obtained so far for these instances were recently compiled in [22].

Table 6 shows the results obtained with our memetic algorithm using MULTI_DYN and GEN_ELIT in the nine GSM2 instances. Specifically, the mean, median and best

Table 6 Results obtained by our memetic algorithm with MULTI_DYN and GEN_ELIT in the GSM2 instances (48 h)

Instance	MULTI_DYN			GEN_ELIT			Best known
	Best	Mean	Median	Best	Mean	Median	
GSM2- 184- 39	5250	5252.1	5250	5250	5282.2	5258	5250
GSM2- 184- 49	874	874	874	874	874	874	874
GSM2- 184- 52	162	162	162	162	162	162	162
GSM2- 227- 29	55367	56427	56472.5	56608	58876.4	59023	56955
GSM2- 227- 39	8391	8639.9	8642.5	8680	9116.9	9100	8656
GSM2- 227- 49	1998	1998	1998	1998	2011.2	1998	1998
GSM2- 272- 34	51046	52095	52130.5	53275	54877.5	54453.5	53080
GSM2- 272- 39	25535	26318.2	26295.5	27258	28675.3	28484	26237
GSM2- 272- 49	6957	7122.6	7122.5	7185	7361.7	7334.5	6997

costs obtained are shown. The table also shows the previously attained best-known solution as reported in [22]. We can see that our memetic algorithm provides quite a significant improvement by applying the MULTI_DYN replacement strategy. In fact, new best-known frequency plans were obtained in five instances, whereas in the remaining ones, frequency plans with the same cost as the currently best-known solutions were obtained. The cases where the best known solution could be improved upon are shown in **boldface**. Moreover, the benefits of the MULTI_DYN scheme are also apparent when compared to the memetic algorithm with the GEN_ELIT replacement strategy. In fact, the statistical tests confirm that the scheme that applies the MULTI_DYN replacement scheme obtained a superior performance in 7 out of the 9 instances. In the two remaining instances, both schemes attained the best known solution in every run. Thus, as with the results of the previous problems, in the case of the FAP, the state of the art in the application of optimizers could be improved substantially, demonstrating once again the robustness and remarkable performance of the methods that incorporate MULTI_DYN as a way to control premature convergence.

6 Conclusions and Future Work

Premature convergence is one of the most studied drawbacks of EAs and other population-based metaheuristics. The appearance of premature convergence and its consequences obviously depend on the specified stopping criterion. Thus, it is quite surprising that most of the methods designed to alleviate this drawback do not take into account the stopping criterion set by the users to bias the decisions made by them. MULTI_DYN is the first survivor selection strategy proposed that takes into account the stopping criterion and elapsed time to bias its decision with the aim of properly adapting the balance between exploration and exploitation to the needs of

the different optimization stages. Specifically, it uses the contribution to diversity as an additional optimization objective and it adopts ideas from multi-objective optimization. Additionally, it uses an adaptive penalty approach to induce a higher degree of exploration in the initial phases and a higher degree of intensification in the final ones.

A fairly standard memetic algorithm is used to test the abilities of MULTI_DYN by applying it to three different complex optimization problems; namely, the Sudoku puzzles, the 2DPP and a variant of the FAP. Note that in order to apply the MULTI_DYN replacement, the only additional task that the user must carry out is to define a distance metric to calculate the dissimilarity between pairs of individuals. Thus the additional effort required with respect to standard EAs is minimal.

The experimental validation carried out with these three problems shows that schemes that apply the MULTI_DYN survivor selection strategy provide important benefits when compared to more mature methods that have been designed with the aim of delaying the convergence of the population. Furthermore, benefits with respect to some other traditional schemes, such as the GEN_ELIT or RW methods, are even more noticeable. An important additional contribution of this chapter is that the state of the art in the application of metaheuristics to the three problems considered could be substantially improved upon. In the case of the Sudoku, three puzzles regarded as the most difficult ever designed were successfully solved for the first time with an EA. As for the 2DPP and FAP, new best-known solutions were obtained for popular instances of these problems.

Several lines of future work might be explored. First, we would like to apply MULTI_DYN to a larger set of combinatorial problems and adopt similar principles to different fields of optimization, such as continuous optimization and multi-objective optimization. Second, we would like to make some changes to accelerate convergence in easy problems; specifically, adopting some of the ideas from SAW-TOOTH GA to design new replacement phases seems promising. Third, replacing the simple stochastic hill-climbing with more complex trajectory-based metaheuristics seems promising. Finally, in light of the long times required to solve these problems, developing parallel schemes seems quite important. We would like to include the principles of MULTI_DYN in some coarse-grained models by developing distributed and centralized ways to explicitly manage diversity.

References

1. Aardal, K.I., Hoeseel, S.P.M.V., Koster, A.M.C.A., Mannino, C., Sassano, A.: Models and solution techniques for frequency assignment problems. *Ann. Oper. Res.* **153**(1), 79–129 (2007)
2. Alba, E.: *Parallel Metaheuristics: A New Class of Algorithms*. Wiley Series on Parallel and Distributed Computing. Wiley-Interscience, Hoboken (2005)
3. Ballester, P.J., Carter, J.N.: An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimisation. In: Deb, K. (ed.) *Genetic and Evolutionary Computation (GECCO 2004)*. Lecture Notes in Computer Science, vol. 3102, pp. 901–913. Springer, Berlin Heidelberg (2004)

4. Blickle, T., Thiele, L.: A comparison of selection schemes used in evolutionary algorithms. *Evol. Comput.* **4**(4), 361–394 (1996)
5. Borenstein, Y., Moraglio, A.: *Theory and Principled Methods for the Design of Metaheuristics*. Springer, Berlin (2014)
6. Bui, L.T., Abbass, H.A., Branke, J.: Multiobjective optimization for dynamic environments. In: 2005 IEEE Congress on Evolutionary Computation CEC'05, vol. 3, pp. 2349–2356 (2005)
7. Clementis, L.: Advantage of parallel simulated annealing optimization by solving sudoku puzzle. In: Sink, P., Hartono, P., Virkov, M., Vak, J., Jaka, R. (eds.) *Emergent Trends in Robotics and Intelligent Systems. Advances in Intelligent Systems and Computing*, vol. 316, pp. 207–213. Springer, Heidelberg (2015)
8. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: *Evolutionary Algorithms for Solving Multi-objective Problems*. Springer, New York (2006)
9. Cotta, C., van Hemert, J.I. (eds.): *Recent advances in evolutionary computation for combinatorial optimization. Studies in Computational Intelligence*, vol. 153. Springer, Heidelberg (2008)
10. Črepinšek, M., Liu, S.H., Mernik, M.: Exploration and exploitation in evolutionary algorithms: a survey. *ACM Comput. Surv.* **45**(3), 35:1–35:33 (2013)
11. Das, K.N., Bhatia, S., Puri, S., Deep, K.: A retrievable GA for solving sudoku puzzles. Technical report, Department of Electrical Engineering, Indian Institute of Technology Roorkee (2007)
12. Eiben, A., Smith, J.: *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, Heidelberg (2003)
13. Eiben, A.E., Schippers, C.A.: On evolutionary exploration and exploitation. *Fundamenta Informaticae* **35**(1–4), 35–50 (1998)
14. Harik, G.R.: Finding multimodal solutions using restricted tournament selection. In: *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 24–31. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)
15. Hong, T.P., Tsai, M.W., Liu, T.K.: Two-dimensional encoding schema and genetic operators. In: *Proceedings of the 2006 Joint Conference on Information Sciences (JCIS 2006)*. Atlantis Press (2006)
16. Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Trans. Evol. Comput.* **7**(2), 204–223 (2003)
17. Jilg, J., Carter, J.: Sudoku evolution. In: 2009 IEEE International Games Innovations Conference, pp. 173–185 (2009)
18. Kim, J., Moon, B.R.: A hybrid genetic algorithm for a variant of two-dimensional packing problem. In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO '09*, pp. 287–292. ACM, New York, NY, USA (2009)
19. Knowles, J., Watson, R.A., Corne, D.: Reducing local optima in single-objective problems by multi-objectivization. In: *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, EMO '01*, pp. 269–283. Springer, London, UK (2001)
20. Koumousis, V., Katsaras, C.: A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. *IEEE Trans. Evol. Comput.* **10**(1), 19–28 (2006)
21. Kuurne, A.: On GSM mobile measurement based interference matrix generation. In: *IEEE 55th Vehicular Technology Conference (VTC Spring 2002)*, vol. 4, pp. 1965–1969 (2002)
22. Lai, X., Hao, J.K.: Path relinking for the fixed spectrum frequency assignment problem. *Expert Syst Appl* **42**(10), 4755–4767 (2015)
23. León, C., Miranda, G., Segura, C.: 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*, pp. 1371–1378. ACM, New York, NY, USA (2009)
24. León, C., Miranda, G., Segura, C.: METCO: A parallel plugin-based framework for multi-objective optimization. *Int. J. Artif. Intell. Tools* **18**(4), 569–588 (2009)

25. Lim, T.Y., Al-Betar, M., Khader, A.: Monogamous pair bonding in genetic algorithm. In: 2015 IEEE Congress on Evolutionary Computation (CEC'15), pp. 15–22 (2015)
26. Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.): Parameter Setting in Evolutionary Algorithms. Studies in Computational Intelligence, vol. 54. Springer, Heidelberg (2007)
27. Lozano, M., Herrera, F., Cano, J.R.: Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Inf. Sci.* **178**(23), 4421–4433 (2008)
28. Luna, F., Estébanez, C., León, C., Chaves-González, J.M., Alba, E., Aler, R., Segura, C., Vega-Rodríguez, M.A., Nebro, A.J., Valls, J.M., Miranda, G., Gómez-Pulido, J.A.: Metaheuristics for solving a real-world frequency assignment problem in GSM networks. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO '08, pp. 1579–1586. ACM, New York, NY, USA (2008)
29. Luna, F., Estébanez, C., León, C., Chaves-González, J.M., Nebro, A.J., Aler, R., Segura, C., Vega-Rodríguez, M.A., Alba, E., Valls, J.M., Miranda, G., Gómez-Pulido, J.A.: Optimization algorithms for large-scale real-world instances of the frequency assignment problem. *Soft Comput.* **15**(5), 975–990 (2010)
30. Mahfoud, S.W.: Crowding and preselection revisited. In: Männer, R., Manderick, B. (eds.) *Parallel Problem Solving from Nature 2*, pp. 27–36. North-Holland, Amsterdam (1992)
31. Mahfoud, S.W.: Niching methods for genetic algorithms. Technical report, University of Illinois at Urbana Champaign (1995). IlliGAL Report No. 95001
32. Mantere, T.: Improved ant colony genetic algorithm hybrid for sudoku solving. In: Third World Congress on Information and Communication Technologies (WICT), pp. 274–279 (2013)
33. Mantere, T., Koljonen, J.: Solving and rating sudoku puzzles with genetic algorithms. In: Proceedings of the 12th Finnish Artificial Intelligence Conference (STeP 2006), pp. 86–92. Finnish Artificial Intelligence Society, Espoo, Finland (2006)
34. Mengshoel, O.J., Goldberg, D.E.: Probabilistic crowding: Deterministic crowding with probabilistic replacement. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99), pp. 409–416, Orlando, FL (1999)
35. Mengshoel, O.J., Galán, S.F., de Dios, A.: Adaptive generalized crowding for genetic algorithms. *Inf. Sci.* **258**, 140–159 (2014)
36. Montemanni, R., Moon, J., Smith, D.: An improved tabu search algorithm for the fixed-spectrum frequency-assignment problem. *IEEE Trans. Veh. Technol.* **52**(4), 891–901 (2003)
37. Moraglio, A., Togelius, J.: Geometric particle swarm optimization for the sudoku puzzle. Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation. GECCO '07, pp. 118–125. ACM, New York, NY, USA (2007)
38. Moraglio, A., Togelius, J., Lucas, S.: Product geometric crossover for the sudoku puzzle. In: IEEE Congress on Evolutionary Computation (CEC'06), pp. 470–476 (2006)
39. Mouret, J.B.: Novelty-based multiobjectivization. In: Doncieux, S., Bredèche, N., Mouret, J.B. (eds.) *New Horizons in Evolutionary Robotics. Studies in Computational Intelligence*, vol. 341, pp. 139–154. Springer, Berlin (2011)
40. Pandey, H.M., Chaudhary, A., Mehrotra, D.: A comparative review of approaches to prevent premature convergence in GA. *Appl. Soft Comput.* **24**, 1047–1077 (2014)
41. Petrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: Proceedings of IEEE International Conference on Evolutionary Computation (CEC'96), pp. 798–803 (1996)
42. Qin, A.K., Huang, V.L., Suganthan, P.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **13**(2), 398–417 (2009)
43. Sato, Y., Inoue, H.: Solving sudoku with genetic operations that preserve building blocks. In: 2010 IEEE Symposium on Computational Intelligence and Games (CIG), pp. 23–29 (2010)
44. Segredo, E., Segura, C., León, C.: Memetic algorithms and hyperheuristics applied to a multi-objectivised two-dimensional packing problem. *J. Glob. Optim.* **58**(4), 769–794 (2014)
45. Segura, C., Miranda, G., León, C.: Parallel hyperheuristics for the frequency assignment problem. *Memet. Comput.* **3**(1), 33–49 (2010)
46. Segura, C., Segredo, E., León, C.: Parallel island-based multiobjectivised memetic algorithms for a 2D packing problem. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO '11, pp. 1611–1618. ACM, New York, NY, USA (2011)

47. Segura, C., Segredo, E., León, C.: Scalability and robustness of parallel hyperheuristics applied to a multiobjectivised frequency assignment problem. *Soft Comput.* **17**(6), 1077–1093 (2012)
48. Segura, C., Coello Coello, C.A., Miranda, G., León, C.: Using multi-objective evolutionary algorithms for single-objective optimization. *4OR* **11**(3), 201–228 (2013)
49. Segura, C., Coello, C., Segredo, E., Miranda, G., Leon, C.: Improving the diversity preservation of multi-objective approaches used for single-objective optimization. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 3198–3205 (2013)
50. Segura, C., Botello Rionda, S., Hernández Aguirre, A., Valdez Peña, S.I.: A novel diversity-based evolutionary algorithm for the traveling salesman problem. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO'15, pp. 489–496. ACM, New York, NY, USA (2015)
51. Segura, C., Coello, C.A.C., Miranda, G., León, C.: Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization. *Ann. Oper. Res.* **240**(1), 217–250 (2016)
52. Segura, C., Coello Coello, C., Segredo, E., Aguirre, A.: A novel diversity-based replacement strategy for evolutionary algorithms. *IEEE Trans. Cybern.* 1–14 (2016 in Press)
53. Storn, R., Price, K.: Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341–359 (1997)
54. Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Comput. Oper. Res.* **40**(1), 475–489 (2013)
55. Wang, Z., Yasuda, T., Ohkura, K.: An evolutionary approach to sudoku puzzles with filtered mutations. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1732–1737 (2015)
56. Yato, T., Seta, T.: Complexity and completeness of finding another solution and its application to puzzles. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E86-A**(5), 1052–1060 (2003)
57. <http://www.websudoku.com/>. Accessed 21 Jan 2016
58. <http://www.sudoku-solutions.com/>. Accessed 21 Jan 2016
59. http://www.sudokuwiki.org/Arto_Inkala_Sudoku. Accessed 21 Jan 2016