# 2

# The State of the Art in Cellular Evolutionary Algorithms

> *A person who never made a mistake never tried anything new.*
>
> *Albert Einstein (1879 - 1955) – Physicist*

Before starting any new scientific research, it is necessary to know perfectly well the existing contributions to the considered field in the literature. This documentation step is basic for the right development of science, as it provides important knowledge of the working area, allowing us to take advantage of the contributions of others authors, and thus avoiding the development of low interest works as, for example, studies tackled before by other researchers. Hence, in this chapter we present a wide exploration of the state of art in cellular evolutionary algorithms, including and classifying some of the main existing publications related to this field.

The chapter structure is detailed next. We start with Sect. 2.1 explaining the first appeared models of cEAs in the literature. In Sect. 2.2 the main theoretical studies developed in cEAs are summed up, whereas Sect. 2.3 compiles some of the most relevant works where empiric studies of the functioning of cEAs have been carried out, and also comparisons to other models. A summary of the most important works contributing with algorithmic improvements to the cEAs field is shown in Sect. 2.4. Section 2.5 presents some works with high repercussion in the field of parallel cEAs. Finally, at the end of the chapter we summarize all this and mention some open research lines.

## 2.1 Cellular EAs: a New Algorithmic Model

The cellular evolutionary algorithms were initially designed for working in massive parallel machines, composed of many processors executing simultaneously the same instructions on different data (SIMD machines –Single Instruction Multiple Data) [88]. In the simplest case, the executed cEAs in this sort of machines used a single large population and assigned an only single individual to each processor. In order to avoid a high overload in communications, the mating of the individuals was restricted to the closer individuals (that is, the ones belonging to their neighborhood).

E. Alba, B. Dorronsoro, *Cellular Genetic Algorithms*,
doi: 10.1007/978-0-387-77610-1_2,

In 1976, Bethke [38] made the theoretical study of a GA on a SIMD parallel machine, analyzing the efficiency of the processing capacity use. He concluded that the maximum efficiency is obtained when much more expensive fitness functions than evolution operations are evaluated, a typical case for many applications.

The first cGA model known is the one proposed by Robertson in 1987 [216], implemented on a CM1 computer. It was a model where all the steps of the algorithm (parent selection, replacement, recombination, and mutation) were executed in parallel. This model obtained good results, with an execution time independent from the population size.

A year later, Mühlenbein, Gorges Schleuter, and Kramer published a work [186] where a cGA on massive parallel machines for the TSP problem was proposed. An important characteristic of this cGA was the incorporation of a local search step for improving the generated solutions by the recombination and mutation operators. Therefore, it is considered the first published hybrid cGA.

After these two first works some cGAs appeared in a few years. They were named in terms of the *pollination plants* [105], *parallel individual* [130], *diffusion* [33], *fine grained* [176], *massively parallel* [129, 232] or *local selection* [116] models. The term *cellular GA* was not used until 1993, when Whitley proposed it for the first time in a work where a cellular automaton model was applied on a genetic algorithm [259].

All these cGAs were initially designed for working on massively parallel machines, although, due to the fast loss of popularity suffered by this kind of machines, the model was adopted later for also functioning on mono-processor machines, without any relation to parallelism at all. In fact, since the emergence of cEAs, there were implementations in secuencial environments [59], in *transputers* nets [115], or in parallel distributed environments [177]. This issue should be clear, as many researchers still think about the equivalence between massively parallel EAs and cellular EAs, what represents a wrong connection: cEAs are simply a different class of EAs, as memetic algorithms [184], estimation of distribution algorithms [158, 187], or particle swarm optimization algorithms [40] are.

Since the cGAs appeared, there have been many contributions published in this field. In Table 2.1 we summarize some of the most important ones, which are commented in Sects. 2.2 to 2.4.

## 2.2 The Research in the Theory of the Cellular Models

The number of existing works in the area of the cGAs theory is too low. This is probably due to the difficulty of deducing generic tests in an area where so many possibilities of implementation exist. Maybe, another reason for this lack is the generalized belief that cGAs model in a more accurate way the populations of nature with respect to the islands model or sequential GAs. Apart from the reason of this lack of theory, more research is necessary in this area.

**Table 2.1.** Brief summary of the main contributions to the cEAs field

| **Reference** | **Contributions** |
|---|---|
| (Hillis, 1990) [129] | cGA with two co-evolutionary populations |
| (Collins & Jefferson, 1991) [50] | Study of the influence of different selection methods |
| (Gorges-Schleuter, 1992) [116] | cGA with a migration mechanism |
| (Gordon et al., 1992) [112] | cGA with a migration mechanism |
| (White & Pettey, 1993) [258] | Study of different ways of applying the selection method |
| (Rudolph & Sprave, 1995) [219] | Use of a self-adaptive acceptation threshold |
| (Sarma & De Jong, 1996) [220] | First theoretical study on the selection pressure in cGAs |
| (Sarma & De Jong, 1997) [221] | Study on the influence of different selection methods |
| (Sipper, 1997) [230] | Co-evolutionary cGA |
| (Folino et al., 1998) [91] | A cGA with a local search step for SAT |
| (Laumanns et al., 1998) [159] | Prey/predator algorithm for the multi-objective domain |
| (Gordon et al., 1999) [111] | Heterogeneous cGA: different parameterization in each cell |
| (Gorges-Schleuter, 1999) [119] | Comparison of panmictic versus cellular ESs |
| (Kirley et al., 1999) [149] | A cGA allowing empty cells (without individuals) |
| (Ku, Mak & Siu, 1999) [157] | A cGA with local search for training recurrent neural networks |
| (Sprave, 1999) [234] | Hypergraphs based model for characterizing cEAs |
| (Alba & Troya, 2000) [26] | Influence of the ratio on the exploration/exploitation |
| (Kirley & Green, 2000) [148] | A cGA applied to the continuous optimization domain |
| (Lee, Park & Kim, 2000) [160] | A cGA with migrations |
| (Rudolph, 2000) [218] | Takeover in cGAs with ring and toroidal population |
| (Krink et al., 2000) [215] | A cGA with disasters |
| (Thomsen & Kirley, 2000) [244] | RBGA: cGA based in religions |
| (Krink et al., 2001) [155] | [215] with a sand bag model for disasters frequency |
| (Llor & Garrell, 2001) [167] | GALE: cGA for data mining. Empty cells are allowed |
| (Kirley, 2002) [147] | CGAD: cGA with disasters |
| (Alba & Dorronsoro, 2003) [10, 12] | Proposal of cGAs with adaptive population |
| (Li & Sutherland, 2002) [165] | Prey/predator algorithm for continuous optimization |
| (Giacobini et al., 2003) [100] | Takeover in asynchronous cGAs with ring population |
| (Giacobini et al., 2003) [99] | Selection pressure study in cGAs with ring population |
| (Li, 2003) [164] | Prey/predator algorithm for the multi-objective domain |
| (Alba et al., 2004) [9] | Comparison between cGAs and other EAs |
| (Alba & Dorronsoro, 2004) [11] | Some hybrid cGAs for VRP |
| (Dorronsoro et al., 2004) [17, 74] | Comparison between synchronous and asynchronous cGAs |
| (Giacobini et al., 2004) [98] | Selection pressure in asynchronous cGAs with toroidal pop. |
| (Alba et al., 2005) [7] | Asynchronous cGAs with adaptive populations |
| (Alba et al., 2005) [8, 18] | cMOGA: first orthodox multi-objective cGA |
| (Alba et al., 2005) [15] | First memetic cGA (cMA); applied on SAT |
| (Alba & Saucedo, 2005) [24] | Comparison between cGA and panmictic GAs |
| (Dick, 2005) [68] | A cGA with ring population as a method for preserving niches |
| (Dick, 2005) [69] | Modelling the genetic evolution in cGAs (ring population) |
| (Giacobini et al., 2005) [102] | Modelling cGAs with square and rectangular populations |
| (Giacobini et al., 2005) [101] | Modelling cGAs with small world topology populations |
| (Alba & Dorronsoro, 2006) [13] | Hybrid cGA which improves the state of art in VRP |
| (Alba et al., 2006) [23] | First EDA with population structured in a cellular way |
| (Dorronsoro & Alba, 2006) [73] | A cGA for the numerical optimization domain |
| (Grimme & Schmitt, 2006) [125] | Prey/predator algorithm for multi-objective domain |
| (Ishibuchi et al., 2006) [138] | A cGA with distinct neighborhoods for selection and crossover |
| (Luo & Liu, 2006) [171] | A cGA designed for GPUs |
| (Luna et al., 2006) [169, 170] | Comparison of cMOGA versus other MO algoritms |
| (Nebro et al., 2006) [193, 194] | MOCell: a new orthodox MO cGA |
| (Payne & Eppstein, 2006) [203] | Study of the emergence matching topology in cGAs |
| (Janson et al., 2006) [139] | HcGAs: cGAs with hierarchical population |
| (Simoncini et al., 2006) [229] | A new anisotropic selection operator for cGAs |
| (Xhafa, 2006) [268] | A cMA for task scheduling in *grid computing* |
| (Alba & Dorronsoro, 2007) [14] | Wide study of a memetic cGA on VRP |
| (Xhafa et al., 2007) [269] | A cMA versus other GAs for batch task scheduling in grids |
| (Nebro et al., 2007) [195] | Improvement of MOCell [193, 194] |
| (jMetal, 2007) [80] | Multi-objective algorithms library (including MOCell) |
| (Dorronsoro et al., 2007) [75] | Parallel hybrid cGA for large instances of VRP |

In the following subsections, we summarize the main existing contributions to the theory of cEAs. Thus, Sect. 2.2.1 presents some studies for theoretically modelling the behavior of cEAs, while Sect. 2.2.2 shows a summary of the main works attempting to characterize the behavior of the cEAs according to the neighborhood to population ratio.

### 2.2.1 Characterizing the Behavior of cEAs

An easy way for characterizing the search performed by a cEA lies in using the selection pressure, which is a measure of the diffusion speed of the good solutions through the population. In order to deepen into the knowledge of the functioning of cEAs, some theoretical works compare the algorithms according to the selection pressure showed, and in some cases even intend to mathematically model its behavior.

Sarma and De Jong made in [220, 221] some theoretical studies about the selection pressure induced by cGAs with different selection operators and neighborhood sizes and shapes. For studying the effect on the size of neighborhood in the selection pressure, they proposed in [220] a definition of the radius of the neighborhood as a measure of its size. Moreover, they observed the same effect when changing the size of the population, so they proposed a new measure called *ratio*, defined as the relationship between the radii of the neighborhood and the population. Sarma and De Jong discovered that this ratio is a key issue for controlling the selection pressure of the algorithm. Therefore, two algorithms with different population sizes and neighborhoods but with the same ratio value have a similar selection pressure. Finally, they proposed the use of a logistic function (parameterized with one variable) for approaching the curve of the selection pressure presented by cGAs. This function is based on the family of logistic curves which was demonstrated in a previous work [106] that works in the panmictic case. The model proposed seemed to be a good approach for cGAs with square populations, but later it was demonstrated that this model have some deficiencies when using rectangular populations (see Chap. 4 for more information).

Three years later, Sprave proposed in [234] a unified description of any kind of EA with both structured and non-structured populations based on the concept of *hypergraph*. A hypergraph is an extension of a canonical graph, where the concept of the edges is generalized: instead of the union of a pair of vertex they become unions of subsets of vertexes. Using the concept of hypergraph, Sprave developed in [234] a method for estimating the growth curve of the selection pressure of a GA. This method is based on calculating the diameter of the structure of the population and the probability of the distribution induced by the selection operator.

Gorges-Schleuter studied in [119] the growth curves for a diffusion (cellular) model of Evolutionary Strategy (ES) with populations structured in toroidal or ring shapes. In her studies, she observed that the diffusion model of ES (both the toroidal and the ring model) has a lower selection pressure than the equivalent ES with panmictic population. Moreover, comparing the two diffusion models, she concluded that, using the same neighborhood size, structuring the population in a ring shape allows a lower selection pressure than when using a toroidal population.

In this work, the differences in the behavior of the algorithm with distinct selection schemes were also analyzed. Particulary, two cases were studied: the one in which it is forced that the actual individual is one of the parents, or the case in which the two parents are selected with the same selection method. It was also studied the effect of allowing one single individual to be chosen as both parents (self-matching) or not. In fact, if we force the actual individual to be one of the parents, the error induced by stochastic sampling is reduced, and also the changes in the individuals are more gradual due to one survived offspring will replace that parent.

In [100], Giacobini et al. proposed quantitative models for estimating the *takeover* time (the time for colonizing the population by copies of the best individual only under the selection effects) for synchronous and asynchronous cGAs with structured population in ring shape (one dimension), and using a neighborhood composed by the two nearest individuals to the considered one. This work was later extended in [98, 99] in order to find accurate mathematical models for fixing the selection pressure curves of synchronous and asynchronous cGAs. In these works, the population is structured in a bidimensional grid, but it is forced to be square. In [102], the same authors proposed some probabilistic recurrences for modelling the behavior of the selection pressure of some synchronous and asynchronous cGAs with square, linear (ring), and toroidal population for two different selection schemes. They also studied the case of a rectangular population for synchronous and asynchronous cGAs, but in this case they only validated the experiments on one selection scheme. In Chap. 4 it is demonstrated that the model is not completely satisfactory when other selection schemes different from the one studied in [102] are tested.

Giacobini, Tomassini, and Tettamanzi proposed in [101] some mathematical models for approaching the growth curve of cGAs working on populations where the topology is defined as a random graph or as *small world graphs*, where the distance between any two individuals is, in general, much lower than in the case of the most commonly regular grids used (they are neither regular nor completely irregular [197] graphs).

Recently, in [229], Simoncini et al. proposed a new selection operator for cGAs called *anisotropic selection* for tuning the selection pressure of the algorithm. This new method lies in allowing the selection of the individuals of the neighborhood with different probabilities according to their location. In this way, the authors promote the emergence of niches in the population. In this work the selection pressure of the algorithm on different shapes of population was studied, but the comparison between the new algorithm and the canonical cGA is missed.

Finally, a new contribution to the field of theory of cGAs was presented in [72] through the development of a more accurate mathematical equation than the existing ones for modelling the selection pressure curves of cGAs with rectangular and square populations. The proposed model was demonstrated to be valid for different selection methods.

### 2.2.2 The Influence of the Ratio

In the literature, there exist results (as [189] for the case of large instances of the TSP problem, or [35, 84, 114] for function optimization) that suggest, but do not analyze, that the shape of the grid of the population really influences in the quality of the search performed by the algorithm. However, Sarma and De Jong defined in [220, 221] the concept of ratio. As it was introduced in Sect. 2.2.1, this feature is highly interesting because algorithms with similar ratio values show a similar behavior in the search.

It was not until year 2000 when Alba and Troya [26] published the first quantitative study of the improvement obtained in the efficiency of a cGA when using non square grids. In this work, the behavior of some cGAs with different shapes of grids on some problems was analyzed, concluding that the use of non square grids promotes a very efficient behavior on the algorithms. Moreover, Alba and Troya redefined in [26] the concept of *radius* as the dispersion of a set of patterns, since the definition by Sarma and De Jong [220] can assign the same numeric value to different neighborhoods, which is undesirable. Finally, we can find another really important contribution in [26], consisting in changing the shape of the population in an specific time step of the execution for modifying the tradeoff between exploration and exploitation applied by the algorithm on the search space. So, the authors take advantage of the different behavior showed by the cGAs with distinct population shapes, and in a very easy way (free of computational load) they change the behavior of the algorithm, in the middle of the execution, promoting the exploration of the search space from a local exploiting step or vice versa.

As a contribution to this area, Dorronsoro and Alba developed in [12] a new adaptive model in which the shape of the population is automatically changed (and, therefore, the ratio value) for regulating the balance between exploration and exploitation performed by the algorithm. Different versions of this new adaptive algorithm were compared to static ratio algorithms, and it improved all of them in all the cases (see Chap. 6).

## 2.3 Empirical Studies on the Behavior of cEAs

In this section we present some important works for analyzing the behavior of cEAs, like the evolution process of the individuals in the population, or the algorithm complexity according to the operators used.

Collins and Jefferson characterize in their work [50] the difference between the panmictic GAs and the cGAs according to some factors, as the diversity of the genotype and phenotype, the speed of convergence, or the stockiness of the algorithm, concluding that the local matching performed in the cGAs "*... is more appropriate for the artificial evolution ...*" than the GAs with panmictic population. The authors demonstrate in this work that, for a particular problem with two optima, a panmictic GA rarely finds the two solutions, meanwhile the cGA generally finds both solutions. The reason is that, thanks to the slow diffusion of the best solutions produced by the cGA, the diversity is kept for longer in the population, forming some small *niches* in it, or groups of similar individuals, representing different searching areas of the algorithm. This work inspired some other modern works where cGAs are used as methods for finding multiple optimal solutions to problems [68].

Davidor developed in [58] a study about a cGA with a bidimensional grid and a neighborhood with eight individuals. In this study, the proportional selection was used (according to the fitness value) for both parents, creating two offsprings in the recombination step, and placing both offsprings in the neighborhood with given probability according to their fitness value. Using this model, he discovered that the cGA showed a fast convergence, although in a located way, that is why niches of individuals with fitness values close to the optima were formed in the population. This fast (and located) convergence is not surprising if we consider that the selection is really effective in very small populations. Therefore, we can conclude from this work that a characteristic behavior of cGAs is the forming of diverse niches in the population where the reproductive cycle tends to promote the specialization of the composing individuals (the exploitation inside these areas is promoted). From this statement we conclude that the cGA maintain diverse search paths towards different solutions, as each of these niches can be seen as an exploitation path of the search space.

In the same conference where Davidor presented the commented work, Spiessens and Manderick [232] published a comparative study of the temporal complexity between their cGA and a secuencial GA. Due to the problem dependance of the evaluation step, they ignored it in their studies, and they were able to demonstrate that the complexity of cGAs increases linearly according to the genotype length. On the contrary, the complexity of a secuencial GA increases polynomially according to the size of the population multiplied by the genotype length. As an increment in the length of the individual should theoretically be joined to an increase in the size of the population, an increment in the length of an individual will affect to the execution time of a sequential GA, but not in the case of the cGA. Moreover, in this article the authors deduce the expecting number of individuals when using the common selection methods in cGAs, showing that the proportional selection is the one with the lower selection pressure.

Now, we briefly discuss the results of the experiment presented some years later by Sarma and De Jong in [62]. In this work, they obtained a really important result for any researcher interested in developing cGAs. In their experiments, they compared some cGAs using diverse selection schemes, and they realized that two of the studied selections behaved in a different way even having equivalent selection pressures. In accordance with the authors, "*these results remark the importance of an analysis on the variation of selection schemes. Without this analysis, it is possible to fall into the trap of assuming that the selection algorithms which are expected to have an equivalent selection pressure produce a similar search behavior.*".

In 1994, Gordon et al. [110] studied seven cGAs with different neighborhoods on continuous and discrete optimization problems. In that paper, they concluded that larger neighborhoods work better with simple problems but on the contrary, with more complex problems it is better the use of smaller neighborhoods.

Capcarrère et al. defined in [44] a set of very useful statistical measures for understanding the dynamical behavior of cEAs. In that study two kinds of statistics were used: based on genotype and phenotype. The metrics based on genotype measure issues related to the chromosomes of the individuals of the population, whereas the ones based on phenotypes take account of the adequacy properties of the individual, basically in the fitness.

More recently, Alba et al. performed in [19] a comparative study of the behavior of cGAs with synchronous and asynchronous update policies of the population. The results obtained show that the asynchronous cGAs perform a higher selection pressure than the synchronous ones, so they converge faster, and generally, find the solution sooner than the synchronous in the less complex studied problems. On the contrary, in the case of the hardest problems, the synchronous cGAs seem to be the ones offering a better efficiency, as the asynchronous get stuck in local optima more frequently.

In [84], Eklund performed an empirical study for determining the most appropriate selection method and the shape and size of the neighborhood for a cellular GP. The conclusions were that both the ideal size and shape of the neighborhood depend on the size of the population. Regarding the selection method, any of the studied ones behave well with elitist populations. Moreover, it was discovered that the higher the number of dimensions in a population is, the higher the dispersion speed of the good solutions is and therefore the size of the population should be larger for obtaining a good behavior.

Finally, this book contributes to the theory in cEAs with new and severe theoretical and practical studies of the selection pressure in synchronous and asynchronous cGAs with different population shapes (Chap. 4) [17, 74], and comparative studies between cEAs and panmictic EAs (see Chap. 3 for a deep comparison in the field of GAs, and Chaps. 10 and 13 for the cases of EDAs [23] and memetic GAs [14], respectively).

## 2.4 Algorithmic Improvements to the Canonical Model

In this section we remark some other relevant publications in the field of cGAs which do not directly belong to any of the previous sections, but they deserve a mention because they suppose important advances in the field.

Rudolph and Sprave presented in [219] a synchronous cGA with structured population in a ring topology and with a self-adaptive acceptance threshold, which is used as a way to add elitism to the algorithm. The algorithm was compared to a panmictic GA, which resulted to have a considerably worse efficiency than the cGA.

In [111, 113], the authors presented an heterogeneous algorithm called Terrain-Based GA (TBGA). The idea of TBGA is that the programmer does not need to tune any of the parameters. This is achieved by defining a rank of values for each of the parameters, which disperse along the axis of the population of the cGA. So, in each position of the population there exists a different combination of the parameters, being similar the parameters of neighbor positions. The TBGA algorithm has been also used for finding a good parametrization for a cGA. The authors present two methods for searching a good configuration, and they lie in the storage of the number of times that the best individual of the population in every generation was in each position of the grid. The general idea is that the location with a higher number of best individuals along the different generations should have a good parametrization. In [113] Gordon and Thein conclude that the algorithm with the parameter configuration of that location with a higher registered number of best individuals has a really better efficiency than TBGA, and also than a manually tuned cGA.

In the literature, some cGAs hybridized with local search methods have been published. Some examples are the cGAs for training recurrence artificial neural networks for solving the long-term dependency problem [157] or the XOR function [156], and the most recently hybrid cGAs proposed for the SAT problem by Folino et al. [91, 92], and by Luo and Liu [171], where the mutation operator is replaced by a local search step. This last algorithm has the particularity of being developed for running in the Graphic Processing Unit (GPU) of the computer, instead of using the Central Processing Unit (CPU).

Some models have been also proposed where extinction of the individuals in particular areas of the population is introduced. For example, Kirley proposes CGAD [147, 149], a cGA with perturbations characterized by the possibility of disaster occurrences, which removes all the individuals located in a particular area of the population. CGAD was successfully tested in numerical, dynamical, and multi-objective [146] optimization problems, becoming the only multi-objective approach of a cellular model. Another similar proposal is the one presented by Krink et al. [155, 215], in which some disasters are frequently generated in areas of the population where the individuals are replaced by new individuals. The frequency of these disasters is controlled by a sand bag model (see [155] for more details).

Kirley also proposed, with Thomsen and Rickers, an EA based in religions (RBEA) in [244], where a set of religions is established in the population. In this model, individuals can only belong to one single religion, and they are allowed to mate only with individuals belonging to their same religion. This way, the creation of niches is promoted. Occasionally, individuals can become to another different religion. In this work, the authors demonstrated that the new RBEA improved the results of a panmictic EA and a cEA for a set of numerical functions.

Although cGAs have an implicit migration given by the overlap of the neighborhoods, some authors try to emphasize this issue by adding any other additional kind of explicit migration. Some examples of this class of algorithms are the ones presented in [112, 116], where the migration is introduced (by copying an individual anywhere in the population in determined periods of time) for allowing separate niches to interact. This kind of migration is also used in [160], where Lee et al. additionally propose another new migration policy consisting in applying the mutation operator to the migrating individuals. Another sort of migration is the one in the previously commented CGAD [147], where the extinguished areas are filled by replicas of the best individual. Finally, there is also an implicit migration in the model proposed by Alba and Troya in [26], explained in Sect. 2.2.2, as the change in the shape of the population implies a redistribution of part of the individuals composing it.

In [167] a new cGA was proposed, called *GALE*, for the data mining classification problem. The singularity of *GALE* with respect to a canonical cGA is that it allows the existence of empty cells in the grid. Therefore, the offsprings will be placed in the empty cells of their neighborhoods, and if there are not empty cells, they replace the worst individual in the neighborhood. Another singularity of this model is the existence of a *survival step*, where it is decided whether the individuals are kept for the next generation or not, in terms of the fitness values of each individual and its neighbors.

In 2002, Li and Sutherland presented in [165] a variation of cGA called prey/predator algorithm, where the preys (corresponding to the individuals representing potential solutions to the problem) move freely around the positions of the grid, mating with neighbor preys in each generation. Moreover, there exists a number of predators which are continuously displacing around the population, and they kill the weakest prey of their neighbor in each generation. The algorithm showed good results when comparing it to a panmictic GA and to a distributed heterogeneous GA (both from [127]) for a set of 4 numeric problems. This algorithm was later extended in [164] to the multi-objective domain with good results. In fact, there exist two more prey/predator algorithms proposed for multi-objective problems in the literature, published in [125] and [159].

Another sort of non standard cGA is the one given by the *co-evolutionary* approaches. The most well-known example is the Hillis method [129] for sorting minimal sorting networks. The Hillis proposal consisted in a massively parallel GA with two independent populations, which evolve according to an standard cGA. In one population, the *hosts* represent sorting networks, meanwhile in the other population the *parasites* represent test cases. The fitness of the sorting networks is given by measuring how well they sort the test cases provided by the *parasites* at the same grid location. Conversely, the *parasites* are scored according to how well they find flaws in the corresponding sorting networks (in the same location of the grid of the *hosts*). In this way, the algorithm evolves for finding the solution to the problem, as in the population of the *parasites* the individuals evolve to more difficult test cases, meanwhile in the other population the sorting networks evolve to solve these more difficult study cases each time.

Another interesting co-evolutionary variation of the cGA model is the *cellular programming algorithm* by Sipper [230]. The cellular programming has been widely used for evolving cellular automata in order to make computational tasks, and it is based in the co-evolutionary topology of the cellular automata neighbor rules.

For ending this section, we briefly present the main existing works where cGAs have been applied on dynamical optimization problems. Some examples are the work of Kirley and Green [148], the previously mentioned CGAD [147] (also from Kirley), or the comparative study between the efficiency of the panmictic GAs (stationary state and generational) and their equivalent cellular model performed by Alba and Saucedo [24], from which we can conclude that, generally speaking, it is the best of the three algorithms (and also the stationary state panmictic GA in some cases).

Although we can find some proposals in the literature of cEAs applied to the multi-objective field, there only exist two orthodox models, namely cMOGA [18] and MOCell [195, 194]. In this book we present these two models of multi-objective cGA, which are adaptations of the cGA model, and we apply them for solving both a complex optimization problem from the telecommunications field, and also a wide set of problems from academical benchmarks. We also present in this work a new cGA model with hierarchial population [139], called HcGA, where the exploitation of the best solutions is promoted, maintaining the diversity in the population simultaneously.

## 2.5 Parallel Models of cEAs

As it has been previously commented, cEAs were initially developed in massively parallel machines, although there have also merged some other models more appropriate for the currently existing distributed architectures. In Table 2.2 it is shown a summary of the main existing parallel cEAs in the literature.

Some examples of cGAs developed on SIMD machines are those studied by Manderick and Spiessens [176] (later improved in [232]), Mühlenbein [185, 186], Gorges-Schleuter [115], Collins [50] and Davidor [58], where some individuals are located in a grid, restricting the selection and the recombination

**Table 2.2.** Brief summary of the main existing parallel cEAs

| Algorithm | Reference | | Model |
|---|---|---|---|
| Manderick & Spiessens | [176] | (1989) | Parallel cGA on SIMD machines |
| ECO-GA | [58] | (1991) | Neighborhood of 8 individuals. Two offsprings per step |
| HSDGA | [256] | (1992) | Fine and coarse grained hierarchical GA |
| fgpGA | [35] | (1993) | cGA with two individuals per processor |
| GAME | [235] | (1993) | Generic library for constructing parallel models |
| PEGAsuS | [214] | (1993) | Fine and coarse grained for MIMD |
| LICE | [233] | (1994) | Cellular model of evolutionary strategy |
| RPL2 | [239] | (1994) | Fine and coarse grained; very flexible |
| Juille & Pollack | [141] | (1996) | Cellular model of genetic programming |
| ASPARAGOS | [117] | (1997) | Asynchronous. Local search applied if no improvement |
| dcGA | [53] | (1998) | Cellular or steady state islands models |
| Gorges-Schleuter | [119] | (1999) | Cellular model of evolutionary strategy |
| CAGE | [92] | (2001) | Cellular model of genetic programming |
| MALLBA | [81] | (2002) | Generic library for constructing parallel models in C++ |
| Combined cGA | [192] | (2003) | Population composed by some cellular sub-populations |
| ParadisEO | [42] | (2004) | Generic library for constructing parallel models in C++ |
| Weiner et al. | [257] | (2004) | Cellular ES with a variable neighborhood structure |
| Meta-cGA | [172] | (2005) | Parallel cGA for local area networks using MALLBA |
| PEGA | [76] | (2007) | Island distributed cGA (for *grid computing*) |

to small neighborhoods in the grid. ASPARAGOS, the model of Mühlenbein and Gorges-Schleuter, was implemented on a *transputers* network, with the population structured in a cyclic stair. Later it evolved including new structures and matching mechanisms [117] until it was constituted as an effective optimization tool [118].

We also would like to remark the works of Talbi and Bessière [242], where the use of small neighborhoods is studied, and the one by Baluja [35], where three models of cGAs and a GA distributed in islands are analyzed on a MasPar MP-1, obtaining as a result the best behavior of the cellular models. Though, Gordon and Whitley presented in [114] a study comparing a cGA to a coarse grained GA, being the results of the latter slightly better. In [152] a comparison between some cGAs and the equivalent sequential GA is presented, clearly showing the advantages of using the cellular model. We can find in [27] a more exhaustive comparison than the previous ones between a cGA, two panmictic GAs (steady state and generational GAs), and a GA distributed in an island model in terms of the temporal complexity, the selection pressure, the efficacy, and the efficiency, among others issues. The authors conclude the existence of an important superiority of the structured algorithms (cellular and island models) according to the non structured ones (the two panmictic GAs).

In 1993, Maruyama et al. proposed in [177] a version of a cGA on a system of machines in a local area network. In this algorithm, called DPGA, an individual is located in each processor and, in order to reduce the communication to the minimum, in each generation each processor sends a copy of its individual to another randomly chosen processor. In each processor there exists a list of *suspended* individuals, where the individuals are located when they arrive from other processors. When applying the genetic operators in

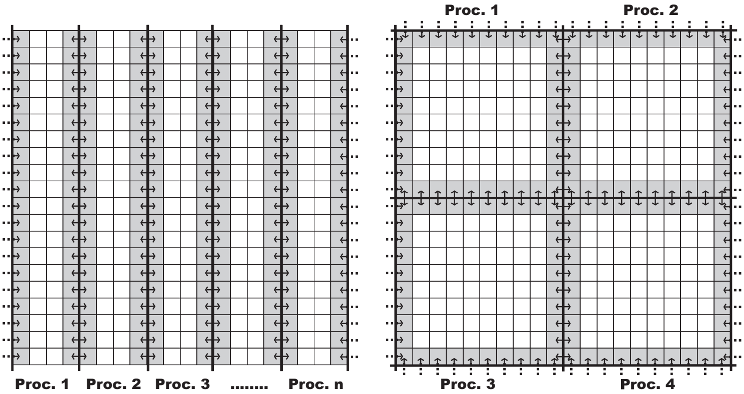

**Fig. 2.1.** CAGE (left) and the combined parallel model of cGA (right)

each processor, this list of *suspended* individuals behaves as the neighborhood. This model is compared to APGA, an asynchronous cGA proposed by Maruyama et al. [178], a sequential GA, and an specialized heuristic for the tackled problem. As a conclusion, the authors remark that DPGA shows a similar efficiency to the equivalent sequential algorithm.

There also exist more modern parallel cGA models, which work on connected computers in local area networks. These models should be designed for reducing the communications to the minimum as, due to their own characteristics, the cellular models need a high number of communications.

In this frame, Nakashima et al. propose in [191] a *combined cGA* where there exist some sub-populations with evolving cellular structure, and interacting through their borders. A graph of this model can be seen in the right part of Fig. 2.1. In a later work [192], the authors propose some parameterizations with different number of sub-populations, ways of replacement, and the topology of the sub-population, and they analyze the results. The authors used this model in a sequential machine, but it is directly extrapolated to a parallel model, where each processor contains one of the sub-populations.

Folino et al. propose in [93] CAGE, a parallel GP. In CAGE, the population is structured in a bidimensional toroidal grid, and it is divided in groups of columns which constitute sub-populations (see the graph on the left in Fig. 2.1). In this way, the number of messages to send is reduced according to other models which divide the population in two dimensions (axis $x$ and $y$). In CAGE, each processor contains a determined number of columns which evolve, and at the end of the generation the two columns in the borders are sent to the neighbor processors, so that they can use these individuals as neighbors of the individuals located in the limits of its sub-population.

Finally, there exist some generic programming frameworks of parallel algorithms which offer eases for implementing any kind of parallel algorithm, including the considered cellular models. Some of these frameworks are GAME [235], ParadisEO [42] or MALLBA [81].

To end this section, we remark the two parallel models of cGAs studied in this book. In Chap. 8 we present the meta-cGA, which was developed using the MALLBA framework, and PEGA, a new cellular GA distributed in islands which can be executed in local area network environments or in *computational grids*. PEGA was applied to the largest existing instances of the VRP problem, contributing to the state of the art with some new solutions.

## 2.6 Conclusions

In this chapter we explored most of the existing works in the field of cellular evolutionary algorithms. The analyzed issues include both the main publications in the field and the most recent trends which are currently emerging. This study allows us to acquire some (necessary) knowledge of the domain of cEAs to understand open research lines like hybridization with local search, multi-objective optimization, necessity of theoretical models, etc.