An Empirical Analysis on Dimensionality in Cellular Genetic Algorithms

Alicia Morales-Reyes, Hugo Jair Escalante, Martin Letras, Rene Cumplido
Instituto Nacional de Astrofisica, Optica y Electronica
Luis Enrique Erro No. 1
Tonantzintla, Puebla, Mexico
[a.morales, hugojair, mletras, rcumplido]@inaoep.mx

ABSTRACT

Cellular or fine grained Genetic Algorithms (GAs) are a massively parallel algorithmic approach to GAs. Decentralizing their population allows alternative ways to explore and to exploit the solutions landscape. Individuals interact locally through nearby neighbours while the entire population is globally exploring the search space throughout a predefined population's topology. Having a decentralized population requires the definition of other algorithmic configuration parameters; such as shape and number of individuals within the local neighbourhood, population's topology shape and dimension, local instead of global selection criteria, among others. In this article, attention is paid to the population's topology dimension in cGAs. Several benchmark problems are assessed for 1, 2, and 3 dimensions while combining a local selection criterion that significantly affect overall selective pressure. On the other hand, currently available high performance processing platforms such as Field Programmable Gate Arrays (FPGAs) and Graphics Processing Units (GPUs) offer massively parallel fabrics. Therefore, having a strong empirical base to understand structural properties in cellular GAs would allow to combine physical properties of these platforms when designing hardware architectures to tackle difficult optimization problems where timing constraints are mandatory.

General Terms

Cellular Genetic Algorithms, Topology's Dimension, Selective Pressure.

1. INTRODUCTION

A general taxonomy for parallel Genetic Algorithms (PGAs) classifies them in coarse or distributed and fined or cellular GAs [5]. Dealing with coarse or fine grained populations or sub-populations also helps defining suitable processing platforms. Among the most used massively parallel processing platforms are Field Reconfigurable Gate Arrays (FPGAs) and Graphics Processing Units (GPUs). Both provide processing resources that can be interconnected in such a way that massively parallel algorithmic techniques could replicate their internal processing structure. In this regard, fine grained or cellular GAs (cGAs) are particularly appropriate to adapt to these fabrics while taking advantage of their implicit massive parallelism when the population is decentralized. Moreover, additional structural characteristics in cellular GAs affect the searching process and in particular these parameters can directly modify the induced selective pressure [2, 7]. This article presents an empirical analysis of a specific structural characteristic in cellular GAs, the topology's dimension. It includes an extended experimental benchmarking and an initial overview of the implication of changing dimension on the fly during the search.

This study is carried out at an algorithmic level with the purpose of leaving a strong empirical base to exploit structural characteristics in cGAs when designing hardware architectures using FPGAs or parallel programming models when using GPUs as processing platforms. Several authors have proposed different approaches to design hardware architectures to implement GAs in order to accelerate their searching process. Most of these proposals are based on centralized or panmictic GAs.

In [9] a GA based hardware architecture is proposed with variable configuration parameters for population size and selection criteria, it also offers interchangeable objective function module. Connecting several FPGA devices is optional in case computing resources are not enough. This architectural design does not exploit implicit parallelism in GAs. Zhu et. al. proposed an architecture that only evolves one individual at a time reducing memory storage; although resources usage is significantly reduced, also GAs searching capabilities are affected [20]. In [6], a centralized GA based IP core (Intellectual Property core) module is proposed. Having an independent module implementing a GA allows its integration within embedded systems that requires an optimization module for specific problem domains. Having problem independent GA based optimization engines is investigated in [15], where authors proposed to use Neural Networks (NN) as fitness function evaluators, every time a different problem is tackled the design needs to be re-synthesized. On the other hand, several studies have been made targeting cellular GAs in order to take advantage of their implicit parallelism at an algorithmic level from an architectural perspective. Dos Santos et. al. proposed a toroidal array of Processor Elements (PEs) which deal with small sub-populations each; however, toroidal connection is

GECCO'15, July 11-15, 2015, Madrid, Spain.
Copyright 2015 ACM 978-1-4503-3472-3/15/07 ...$15.00
DOI: http://dx.doi.org/10.1145/2739480.2754699 .

lost among individuals and therefore the canonical algorithmic structure of a cellular GA is significantly modified [8, 19].

Cellular GAs are massively parallel GAs which ideally can evolve one generation in few clock cycles. Normally, they imply simpler PEs which deal with one individual each. Having decentralized populations requires defining a structure to allocate individuals and from a structural perspective to limit the interaction of individuals. Therefore, structural characteristics like topology's shape and size, neighbourhood's shape and size and topology's dimension are defined. These structural properties not only affect the processing structure of the search but also the search itself. Several authors have studied structural properties in cGAs [2]. In [3], a study on the relationship between topology and neighbourhood in cGAs is carried out. Authors show the influence of these structural parameters in the searching process, specifically in the applied selective pressure. Alba et. al. also proposed to dynamically change the population's shape in order to affect the selective pressure, algorithmic performance metrics were improved [1].

Comparing the effect of local selection versus topology modification as well as studying the internal dynamic change of the topology configuration has been approached [13, 12]. Both studies show how effective is to modify structural configuration in cGAs in order to improve their performance. A double structural level criterion to affect the search locally and globally is presented [14]. Although adaptive criteria to modify structural parameters are proposed, their computational cost is high and applying constant changes result in very similar performances. Previously, an initial study on dimension was presented in [11]; however, few test problems were evaluated and therefore the empirical assessment was limited. In this article, authors continue studying cGAs structural properties, specifically dimension, in order to improve their algorithmic performance and to have a robust experimental base over a widen benchmark.

This article is organized as follows, in Section 2 cGAs structural properties are described as well as their effect in the searching process. Section 3 presents this study's experimental framework as well as algorithmic performance metrics and applied statistical tests. In Section 4, results obtained over several difficult optimization problems are presented and analysed. Finally, Section 5 draws conclusions and future research directions.

2. ALGORITHMIC STRUCTURE

Cellular GAs are massively parallel GAs with decentralized populations that are normally placed on a toroidal grid-like structure where each crossing point holds an individual or solution. Having a structure requires defining other parameters, such as the shape and size of the population's topology and the local neighbourhood, as well as their dimension. This study focuses on the effect topology's dimension has over the searching process. In Figure 1, cGAs topologies for 1, 2, and 3 dimensions are shown. Local neighbourhoods are also visible in every configuration, these are composed by individuals located to East and West in 1-D arrays; to North, East, South and West of every individual in a 2-D array and in a 3-D grid, individuals at front and back of the central one are also within the local neighbourhood.

In Algorithm 1, steps for a canonical cGAs on a 2-D topology are described. A population is randomly initialized considering one individual per crossing point. According to the targeted problem the fitness function is evaluated for every individual. Evolution starts for all individuals with a selection process. Common selection methods are tournament and roulette-wheel which are also normally applied in cGAs. However, other selection techniques have been proposed specifically for cGAs; in this study a method called anisotropic selection is applied [16, 17]. Current individuals are assumed as one of the parents, and a second one is selected from the neighbourhood. Once both parents are selected, genetic operators are applied, recombination followed by mutation. These operators are defined according to individuals representation. In this study single point crossover is applied with probability equals to 1.0 and mutation is applied with probability $\frac{1}{n}$ where n is the chromosomes length which varies according to each benchmark problem.

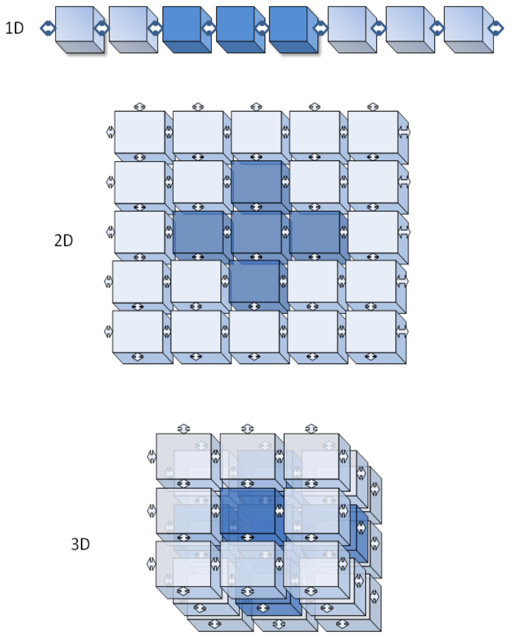

Figure 1: Different dimensions arrays in cGAs

Because cGAs carry out exploitation locally and exploration globally throughout population's topology; replacement or updating criterion normally exchange central individuals only if one of the evolved children is better or more fitted than its parent. Neighbourhoods are overlapped, thus updating criterion can be synchronous or asynchronous; in this study synchronous updating is always applied. One generation corresponds to an execution of this operations set. A number of generations is executed until the stop condition is fulfilled either after a certain number of generations or evaluations (number of fitness function evaluations) or when an error threshold is reached.

2.1 Local selection

Anisotropic selection consist in applying different selection probabilities to individuals within the local neighbour-

Algorithm 1 Canonical cGA

```
procedure cGA
    $random(x)$                                  ▷ Initialize population
    $f = calulateFit(x)$                         ▷ Calculate individuals fitness
    for $i \leftarrow 1, generations$ do
        for $j \leftarrow 1, population$ do
            $x_0, f_0$                           ▷ Current individual
            $(x_n, x_e, x_s, x_w) \leftarrow getNeigh(x_0)$
            $(f_n, f_e, f_s, f_w) \leftarrow getFit(x_n, x_e, x_s, x_w)$
            $(x') \leftarrow localSel(x_n, x_e, x_s, x_w)$
            $(x_1, x_2) \leftarrow recombine(x_0, x')$
            $(x_1', x_2') \leftarrow mutate(x_1, x_2)$
            $(f_1', f_2') \leftarrow calculateFit(x_1', x_2')$
            $(f_0') \leftarrow selBest(f_0, f_1', f_2')$
            $(x_{temp}(i, j), f_{temp}(i, j)) \leftarrow$
                    $replace(x_0', f_0', [ifBetter])$
        end for
        $x = x_{temp}, f = f_{temp}$             ▷ Synchronous updating
        if $\bar{f} <= threshold$ then
            stop
        else
            next
        end if
    end for
end procedure
```

hood according to their position. Figure 2 shows population topologies for 1, 2, and 3 dimensions. For example, in a 2-D topology, higher probabilities could be assigned to individuals located to the North and the South and lower ones to the East and the West of the central individual. Similarly, in a 3-D grid, neighbours located at North, South and front of the central individual could have higher or lower probability for selection, opposite to individuals located at East, West and back of the current individual. In a 1-D array, neighbours are located to left and right of the central individual, thus probabilities can be higher or lower depending on their direction. In all cases larger Manhattan distances from central individuals can be implemented affecting selective pressure accordingly. In anisotropic selection, an α parameter is required in order to assign selection probabilities for individuals within the neighbourhood. For example, in a 2-D array, selection probability for individuals located at North and South of central one is $P_{NS} = (1 + \alpha) \times P_{uni}$, where P_{uni} is the uniform probability for all neighbours; and $P_{EW} = (1+\alpha) \times P_{uni}$ is the selection probability for individuals located at East and West positions of central individual. Similar equations are applied to 1 and 3 dimensions while modifying uniform probability accordingly.

A concept known as take over times allows to draw applied selective pressure on specific topologies when applying a local selection method. Take over time refers to the number of generations the fittest individual of an initial population needs to conquer all positions within the grid. Figure 3 shows take over times for 1, 2, and 3 dimensions considering local neighbourhoods configurations in Figure 2. Curves in Figure 3 are calculated over a 100 experimental samples per case with very similar population size distributed on different dimensional grids. Weaker selective pressure is applied through a 1-D array while strongest selective pressure is applied on a 3-D grid. A 2-D array's selective pressure is in the middle of both cases. For $\alpha = 0.1$ (strong selective pressure), it takes approximately 210, 25 and 15 generations in a 1, 2, and 3 dimension array respectively for the best individual of an initial population to conquer the grid. In contrast, for $\alpha = 0.9$ (weaker selective pressure), it takes approximately 325, 60 and 25 generations in a 1, 2, and 3 dimension array to conquer the grid. These ranges in terms of number of generations show how exploitative or explorative a cGA would be from a structural perspective. This study focuses on dimension both of topologies and of neighbourhoods. Anisotropic selection implies structural properties when defining internal selection probabilities by individuals' location within neighbourhoods. Assessing a combination of both structures would provide a better understanding of the internal working in a cGA.

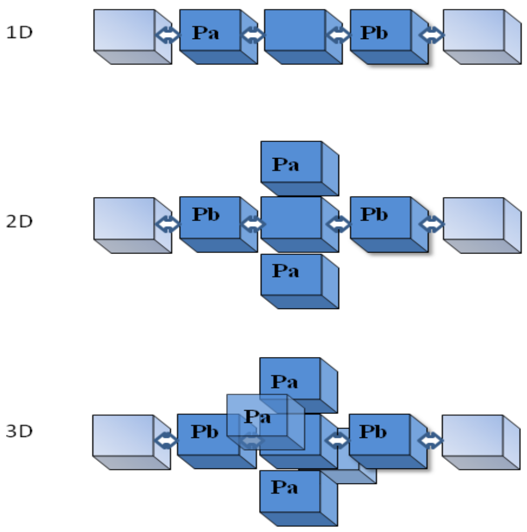

Figure 2: Local anisotropic selection

2.2 Constant dimensional change

In this study, an empirical comparison among dimensions and their effect in the algorithmic performance of cGAs is carried out. Moreover, an additional empirical set-up is also included. It consists in applying a constant change among 1, 2, and 3 dimensions. A constant population size is used and solutions are re-allocated when dimension is changed. All experimental samples start evolving on a 1-D array and after a certain number of generations, individuals are re-arranged following a 2-D square array; after another number of generations, solutions are mapped to a 3-D cubical array closing a cycle of constant dimension change. Different sizes of generations blocks are assessed. A short block of 15 generations, a medium block of 30 generations and a large block of 60 generations for most problems are specified. For the Minimum Tardy Task Problem (MTTP) blocks of 5, 10, and 15 generations are defined and for the System of Linear Equations (SLE) problem blocks of 20, 40 and 80 generations are set.

The mechanism to re-allocate individuals within a different grid dimension is straight forward. It consists in taking a number of individuals of a 1-D array and filling every row of a 2-D square grid until all individuals are allocated. Once

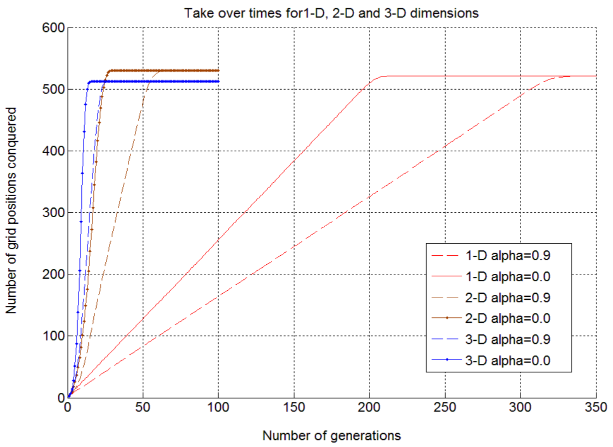

Figure 3: Take over times for 1, 2, and 3 dimensions

in a 2-D square topology, tiles of individuals are moved to a 3-D cubical topology until the whole population is migrated. Considerations about population size are needed, normally few individuals positions remain available when dimension is modified and therefore some extra-criterion to generate new or to replicate current individuals is required. In the experimental set-up, population size fits 1, 2, and 3 dimensions in order to avoid new or to replicate previous solutions. The experimental framework is detailed in the the next section.

3. EXPERIMENTAL FRAMEWORK

In order to assess the effect dimension has in cGAs algorithmic performance several test problems have been tackled. Experimental constraints are provided in this section as well as results validation using statistical tests.

3.1 Benchmark problems

Several optimization problems considered difficult for GAs have been tackled. In total, nine problems from combinatorial and continuous domains are evaluated. Continuous domain problems (Rastrigin, Langerman, Griewank and a System of Linear Equations) have been discretized. Authors acknowledge GAs' drawbacks when dealing with continuous problems in their discrete domain. However one of the aims in this study is to assess GAs' ability to solve difficult optimization problems when binary representation of solutions is used; this kind of representation would be mostly used when designing hardware architectures in order to avoid using specialized processing units within the device. On the other hand, a group of five combinatorial problems are also tackled: Error Code Correction design problem (ECC) problem, Frequency Modulation Sound (FMS) problem, Massively Multi-modal Deceptive Problem (MMDP), Minimum Tardy Task Problem (MTTP) and P-Peaks problem. Objective functions of these problems are presented in Table 1 as well as significant configuration details are included for each problem, if more details are needed, readers are referred to provided references.

3.2 Experimental constraints

The following experimental contrainst are common to all cGAs configurations for their empirical assessment.

- Solutions representation: binary
- Population size: 729 individuals
- Topology shape: square
- Topology dimension: 1-D, 2-D, 3-D
- Selection: anisotropic, $\alpha = 0.1||0.9$
- Recombination: single point, $P_c = 1.0$
- Mutation: bit flip, $P_m = \frac{1}{n}$ where n is chromosome's length
- Local replacement (within neigbhourhoods): only if better
- Global replacement (whole grid): synchronous
- Convergence condition: error threshold or maximum of 1500 generations (2000 generations for SLE)

Individuals are represented as binary strings in both continuous and combinatorial problems. A population size of 729 individuals is implemented. This number of individuals allow for having a 1-D array of 729 solutions, a 2-D square array of 27×27 and a 3-D square array of 9×9×9. Anisotropic selection (see Subsection 2.1) is configured for border values where $\alpha = 0.1$ implies strong exploitation and $\alpha = 0.9$ means promoting exploration of solutions. Genetic operations, recombination and mutation, are single point and bit flip with specific probabilities to affect solutions. Recombination is always carried out with probability ($P_c = 1.0$). Because of cGAs massive parallelism, solutions replacement require local and global criteria. Locally, current individuals are replaced only if any of their offspring is a better solution. Globally synchronous replacement is implemented, this means that individuals in one generation evolve into the next generation.

Algorithmic performance is measured in terms of efficiency and efficacy. Efficiency consists of the number of generations cGAs require to converge to a problem's solution together with the solutions quality which in this case is given by specific problems' error threshold. On the other hand, efficacy is given by the number of success experimental samples out of the total number of experiments carried out. A total of 100 experimental samples per case per problem are performed.

3.3 Statistical analysis

Results are statistically supported by an initial normality test performed on each set of experimental results regarding the convergence time. First, normality is determined by the Kolmorov-Smirnov test or the lilliefors test, both at 5% of significance. Lilliefors test is suitable when a fully-specified null distribution is unknown, contrary to the Kolmorov-Smirnov test. Once the normality of results has been established, an Analysis of Variance (ANOVA) is applied to results with normal distribution whereas Kruskal-Wallis is applied in any other case. In the next section, experimental results and their analysis are drawn.

A total of 100 experimental samples per algorithmic case per problem are carried out. In order to assess algorithmic efficacy for statistical significance, every experiment is considered as a Bernoulli trial which output is to converge or not to a problem's solution according to a stopping condition

Table 1: Benchmark problems

Problem	Fitness function	Properties
Rastrigin fn. [4]	$f\left(\vec{x}\right) = 10q + \sum_{i=1}^{q} \left(x^2 - cos\left(2\pi x_i\right)\right)$	$q = 15$, $f_T \leq 0.0005$, multi-modal, non epistatic, regular
Griewank fn. [4]	$f\left(\vec{x}\right) = 1 + \sum_{i=1}^{q} \frac{x_i^2}{4000} - \prod cos\left(\frac{x_i}{\sqrt{i}}\right)$	$q = 15$, $f_T \leq 0.0001$ Multi-modal, epistatic, regular
Langerman fn. [4]	$f\left(\vec{x}\right) = -\sum_{i=1}^{q} c_i e^{-\frac{1}{\pi}\sum_{j=1}^{D}\left(x_j - a_{ij}^2\right)} cos\left(\pi \sum_{j=1}^{D}\left(x_j - a_{ij}\right)^2\right)$	$q = 15$, $f_T \geq -1.4990$ Multi-modal, epistatic, non-regular
FMS [2]	$f\left(t\right) = a_1 \cdot sin(w_1 \cdot t \cdot \theta + a_2 \cdot sin\left(w_2 \cdot t \cdot \theta + a_3 \cdot sin\left(w_3 \cdot t \cdot \theta\right)\right))$	$q = 6$, $f_T \leq 0.0005$ Multi-modal, epistatic, non-regular
SLE[2]	$Ax = b$	$q = 10$, $f_T \leq 0.04$ Epistatic
ECC[2]	$f\left(n, M, d\right) = \frac{1}{\sum_{i=1}^{M}\sum_{j=1, i \neq j}^{M} \frac{1}{d_{ij}^2}}$	$M = 24, n = 12$, $f_T \leq 0.0674$ Multi-modal, epistatic, non-regular
MMDP[2]	$f_{MDDP}(q) = \sum_{i=1}^{k} fitness_{q_i}$	$q = 25$, k sub-problems, $f_T \geq 0.99$ Multi-modal, combinatorial
MTTP[10, 2]	$f_{MTTP}(\vec{x}) = \sum_{i=1}^{q} weight_{x_i}$	$q = 100$, $f_T \leq 0.0051$ NP-combinatorial, epistatic
P-Peaks Problem [2]	$f_{P-Peaks}(\vec{x}) = \frac{1}{Q} max_{i=1}^{P}\left(Q - Hamming\left(\vec{x}, P_i\right)\right)$	$P = 100$, $\bar{f}_T = 1.0$, non-manually tunable, multi-modal, non-regular

[18]. In a series of Bernoulli trials, successful experimental samples follow a binomial distribution with a maximum likelihood estimator calculated as the number of successful experiments divided by the total of experimental samples.

4. EXPERIMENTAL RESULTS

The main objective in this study is to compare cGAs algorithmic performances when populations evolve on 1, 2, and 3 topology dimensions. Difficult optimization problems are tackled in both continuous and combinatorial domains. Locally, anisotropic selection is applied (see Subsection 2.1). Two border values for anisotropic selection are assessed: $\alpha = 0.1$ and $\alpha = 0.9$. Having a small α value implies that strong selective pressure is induced locally; while a large α value means selective pressure is weak, therefore exploration is promoted within neighborhoods.

In Tables 2 and 3, efficiency in terms of convergence time (number of generations) and efficacy in terms of percentages of successful experimental samples are drawn. In both tables, implicit selective pressure strength induced by different dimensions in all test problems is clearly observed. A 1-D array promotes exploration and therefore takes a large number of generations to find solutions, in contrast to a 3-D cubical array where strong selective pressure modifies the search and successful experiments converge faster. At a local level through anisotropic selection, induced selective pressure also influences convergence times. However, for discretized continuous problems (Rastrigin, Griewank and Langerman functions, and SLE and FMS problems) a weak selection pressure induced by a 1-D linear array overcomes the locally induced selective pressure either strong or weak ($\alpha = 0.1$ or $\alpha = 0.9$); results do not show significant difference between convergence times and percentages of successful experiments are low. In contrast, for combinatorial problems (ECC, MMDP, MTTP and P-Peaks problems), significant difference is observed when weak or strong local selective pressure is induced.

In terms of efficacy, Table 2 shows statistically higher percentages of successful experiments for Rastrigin, Griewank, SLE and FMS problems when evolving on a 2-D topology with strong induced local selective pressure ($\alpha = 0.1$). In these cases, using a 3-D cubical array negatively affects cGAs algorithmic efficacy. On the other hand, combinatorial problems: ECC, MMDP, MTTP and P-Peaks problems achieve 100% efficacy; in this regard using a 3-D array positively affects convergence times with a significant reduction in the number of generations required to converge. A 3-D array improves both convergence time and efficacy when tackling the Langerman problem; this problem has a difficult landscape not only for its multi-modality and solutions epistasis but also for its non-regular search space. Similar results are achieved when anisotropic local selection induces weak selective pressure ($\alpha = 0.9$), see Table 3. However, efficacy is significantly improved for Rastrigin and Griewank problems when cGAs are evolving on a 2-D square topology; for Langerman, SLE and FMS problems, efficacy percentages are very similar when using a 3-D cubical grid. Combinatorial problems present highest efficacy while lower convergence times are achieved when cGAs run on a 3-D topology. For these problems applying a strong local (anisotropic $\alpha = 0.1$) and global (3-D topology) selective pressure results in shortest convergence times without affecting cGAs efficacy.

Modifying the induced selective pressure by changing structural cGAs properties like topology's shape and dimension adds certain flexibility to their searching capacity. Modifying topology's shape has been studied before by Alba et. al. in [1] with successful results. This study also presents, in

Table 2: Convergence time[1] and hit rate[2] for test problems in 1-D, 2-D and 3-D cGAs, anisotropic $\alpha = 0.1$

Problem	**1-D**	**2-D**	**3-D**	**1-2-3-D**	**1-2-3-D**	Statistical [3]
				Small gens block	Large gens block	Test
Rastrigin	1387.6 ± 74.06	262.11 ± 25.49	$\mathbf{175.15 \pm 15.87}$	$\mathbf{224.73 \pm 26.80}$	286.76 ± 26.52	*(+)*
	27%, 4.43	63%, 4.82	13%, 3.36	**66%**, 4.73	25%, 4.33	
Griewank	1255.3 ± 146.67	318.81 ± 29.64	$\mathbf{225.68 \pm 21.90}$	$\mathbf{240.30 \pm 27.56}$	293.17 ± 22.89	*(+)*
	41%, 4.91	**76%**, 4.27	16%, 3.66	67%, 4.70	26%, 4.38	
Langerman	$- - \pm - -$	550.12 ± 224.48	336.35 ± 157.20	$\mathbf{245.80 \pm 59.86}$	431.90 ± 266.22	(+)
	$- -$ %,	39%, 4.87	59%, 4.91	65%, 4.76	**67%**, 4.70	
SLE	$- - \pm - -$	631.53 ± 161.31	$\mathbf{378.76 \pm 192.54}$	$\mathbf{391.00 \pm 157.87}$	516.64 ± 342.55	*(+)*
	$- -$ %,	32%, 4.66	26%, 4.38	69%, 4.62	**75%**, 4.33	
FMS	1371.0 ± 77.5	296.93 ± 69.15	$\mathbf{269.13 \pm 133.53}$	$\mathbf{250.10 \pm 86.71}$	364.71 ± 165.33	*(+)*
	4%, 1.95	**95%**, 2.17	65%, 4.76	50%, 5.0	62%, 4.85	
ECC	455.3 ± 88.42	207.78 ± 29.41	$\mathbf{163.27 \pm 29.45}$	190.44 ± 37.79	229.28 ± 44.16	(+)
	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	
MMDP	519.19 ± 48.45	134.77 ± 5.46	106.9 ± 23.57	$\mathbf{105.86 \pm 6.29}$	144.41 ± 4.73	(+)
	100%, 0.0	100%, 0.0	100%, 0.0	99%, 0.99	100%, 0.0	
MTTP	723.93 ± 74.97	268.50 ± 19.18	$\mathbf{189.5 \pm 9.35}$	$\mathbf{196.23 \pm 13.08}$	239.42 ± 19.54	*(+)*
	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	
P-Peaks	171.48 ± 4.35	116.19 ± 2.96	$\mathbf{95.21 \pm 2.92}$	$\mathbf{96.6 \pm 3.56}$	105.48 ± 3.19	*(+)*
	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	

1 average number of generations and standard deviation are included.

2 standard deviation for the success search rate is included next to the percentage.

3 ANOVA/K-W based statistical difference: *(+)* is proved between 3-D or 1-2-3-D small block and 1-D, 2-D or 1-2-3-D large block / (+) statistical difference is proved between 1-2-3-D small block and 1-D, 2-D, 3-D or 1-2-3-D large block / (•) is not proved.

addition to directly compare cGAs performance on different topology's dimensions, an experimental base to constantly change grid's dimension. Three size blocks of generations are assessed, see Subsection 2.2. In Tables 2 and 3, results for dimension's change every small and large generations block sizes are shown; in terms of efficiency, statistical difference is proved between using a small or a large generations block when changing among 1, 2, and 3 dimensions in most cases except for SLE problem when weak selection pressure is locally applied ($\alpha = 0.9$). Another significant observation from the statistical analysis is that there is no significant difference in efficiency when applying a constant change among dimensions when using a small generations block and evolving on a 3-D topology for strong selective pressure ($\alpha = 0.1$) in most problems. In contrast, when weak selective pressure ($\alpha = 0.9$) is locally induced only three benchmark problems (Rastrigin, FMS and ECC) show this behavior. In general, it is observed that significant statistical difference exist between evolving on a 3-D topology or changing constantly among dimensions considering a small generations block and evolving on a 1-D, 2-D and constantly change among dimensions considering a large generations block.

Promoting locally solutions exploitation ($\alpha = 0.1$) shows in general no significant improvement in efficacy. It is worth mentioned, results for SLE problem where the number of successful experimental samples increases from 26% to 75%; authors acknowledge GAs limitations when tackling this problem but recognize their ability of improvement when taking advantage of cGAs structural properties. Similarly occurs when exploration is promoted locally ($\alpha = 0.9$) not only for SLE but for Langerman problem too, in both cases not only efficacy is significantly improved but also efficiency.

Overall, best efficiency results are concentrated when a 3-D topology or a constant change of dimensions with a small generations block are used. However, best efficacy is not always achieved by these two configuration cases except for combinatorial problems (ECC, MMDP, MTTP and P-Peaks) where 100% efficacy is always achieved. These results widen cGAs empirical base to show how their algorithmic ability is significantly modified and in several cases improved when taking advantage of their structural properties. From an architectural design perspective, demonstrating that in all combinatorial problems, cGAs achieve better convergence times on a 3-D topology and on constantly changing topology dimensions show at an algorithmic level a strong basis to implement cGAs on FPGAs or GPUs while taking advantage of their massively parallel fabrics.

5. CONCLUSIONS

This article presented an empirical study on the effect dimension has in algorithmic cGAs performance when it is

Table 3: Convergence time[1] and hit rate[2] for test problems in 1-D, 2-D and 3-D cGAs, anisotropic $\alpha = 0.9$

Problem	**1-D**	**2-D**	**3-D**	**1-2-3-D**	**1-2-3-D**	Statistical [3]
				Small gens block	Large gens block	Test
Rastrigin	1391.2 ± 93.46	471.59 ± 63.45	**249.87 ± 17.99**	**220.45 ± 24.53**	288.27 ± 27.42	*(+)*
	29%, 4.53	**83%**, 3.75	58%, 4.93	60%, 4.89	23%, 4.20	
Griewank	1351.8 ± 94.93	551.43 ± 80.23	298.61 ± 38.13	**266.23 ± 28.45**	295.10 ± 25.47	(+)
	8%, 2.71	**87%**, 3.36	52%, 4.99	66%, 4.73	35%, 4.76	
Langerman	− − ± − −	924.85 ± 230.19	507.35 ± 182.92	**278.12 ± 85.83**	396.82 ± 156.26	(+)
	− − %,	35%, 4.76	34%, 4.73	**68%**, 4.66	54%, 4.98	
SLE	− − ± − −	1310.4 ± 316.56	1299.5 ± 350.50	**455.26 ± 171.91**	**573.75 ± 156.91**	**(+)**
	− − %,	31%, 4.62	30%, 4.58	62%, 4.85	**64%**, 4.8	
FMS	− − ± − −	720.93 ± 217.27	**344.05 ± 120.74**	**273.50 ± 120.56**	463.28 ± 125.54	*(+)*
	− − %,	**62%**, 4.85	57%, 4.95	50%, 5.0	**62%**, 4.85	
ECC	600.98 ± 101.73	323.21 ± 37.90	**214.88 ± 33.16**	**198.14 ± 41.04**	265.25 ± 40.03	*(+)*
	99%, 0.99	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	
MMDP	757.87 ± 55.01	225.88 ± 14.6	126.59 ± 6.68	**117.4 ± 5.9**	165.16 ± 5.6	(+)
	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	
MTTP	1040.6 ± 104.49	445.03 ± 36.75	244.45 ± 18.92	**215.32 ± 14.64**	287.14 ± 22.39	(+)
	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	
P-Peaks	205.7 ± 6.23	155.51 ± 3.75	109.73 ± 2.66	**101.05 ± 2.73**	117.64 ± 2.96	(+)
	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	100%, 0.0	

1 average number of generations and standard deviation are included.

2 standard deviation for the success search rate is included next to the percentage.

3 ANOVA/K-W based statistical difference: *(+)* is proved between 3-D or 1-2-3-D small block and 1-D, 2-D or 1-2-3-D large block / (+) statistical difference is proved between 1-2-3-D small block and 1-D, 2-D, 3-D or 1-2-3-D large block / **(+)** statistical difference is proved between 1-2-3-D small and large blocks and 1-D, 2-D, 3-D / (•) is not proved.

modified. Structural properties in cGAs directly affects the induced selective pressure at a local level via the selection criterion and at a global level via the topology's dimension.

Several benchmark problems considered difficult for GAs were tackled and results showed better results when a 3-D topology or a constant change among dimensions every certain number of generations (1-2-3-D small generations block) is used. Overall, efficiency when inducing a locally strong selective pressure results in no statistical significant difference between these topologies configuration; behavior that is not repeated when weak selective pressure is applied locally. In terms of efficacy for combinatorial problems highest success search rate is achieved by different topology dimensions and after a constant dimension's change. For continuous domain, efficacy improvement is noticed for two difficult problems (Langerman and SLE) in which efficacy is improved when dimension is constantly changed.

Having a strong experimental base regarding cGAs structural properties and their searching ability makes them a suitable algorithmic optimization technique for acceleration from a processing perspective. Specifically, FPGAs have been targeted as a platform to accelerate GAs but to the best of the Authors knowledge, there is not research reported yet to approach cellular GAs structural properties as an alternative for acceleration. Another platform that offers massive parallelism are GPUs; as a future direction of this research both platforms will be targeted as processing platforms.

6. REFERENCES

[1] E. Alba and B. Dorronsoro. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. In *IEEE Transactions on Evolutionary Computation*. IEEE, 2005.

[2] E. Alba and B. Dorronsoro. Cellular genetic algorithms. In *Operations Research / Computer Science Interfaces*. Springer, 2008.

[3] E. Alba and J. Troya. Cellular evolutionary algorithms: Evaluating the influence of ratio. In *Proceeding of the 6th International Conference on Parallel Problem Solving from Nature*, pages 29–38. LNCS Springer, 2000.

[4] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[5] E. Cantu-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Genetic Algorithms and Evolutionary Computation. Kluwer Academic Publishers, 2000.

[6] P.-Y. Chen, R.-D. Chen, Y.-P. Chang, L. san Shieh, and H. Malki. Hardware implementation for a genetic algorithm. *Instrumentation and Measurement, IEEE Transactions on*, 57(4):699–705, April 2008.

[7] B. Dorronsoro and E. Alba. A simple cellular genetic algorithm for continous optimization. In *Proceedings of 2006 IEEE Congress on Evolutionary Computation*. IEEE, 2006.

[8] P. dos Santos, J. Alves, and J. Ferreira. A scalable array for cellular genetic algorithms: Tsp as case study. In *Reconfigurable Computing and FPGAs (ReConFig), 2012 International Conference on*, pages 1–6, Dec 2012.

[9] P. Fernando, S. Katkoori, D. Keymeulen, R. Zebulum, and A. Stoica. Customizable fpga ip core implementation of a general-purpose genetic algorithm engine. *Evolutionary Computation, IEEE Transactions on*, 14(1):133–149, Feb 2010.

[10] S. Khuri, T. Bäck, and J. Heitkötter. An evolutionary approach to combinatorial optimization problems. In *Proceedings of the 1994 Computer Science Conference (CSC'94)*, page (to appear). ACM Press, 1994.

[11] A. Morales-Reyes, A. Al-Naqi, A. Erdogan, and T. Arslan. Towards 3d architectures: A comparative study on cellular gas dimensionality. In *Proceedings of IEEE NASA/ESA Conference on Adaptive Hardware and Systems (AHS'09)*, page (to appear). IEEE, 2009.

[12] A. Morales-Reyes and A. Erdogan. Internal lattice reconfiguration for diversity tuning in cellular genetic algorithms. In *PLoS ONE*, page 7(7): e41279. PloS ONE, 2012.

[13] A. Morales-Reyes, A. Erdogan, and T. Arslan. Lattice reconfiguration vs. local selection criteria for diversity tuning in cellular gas. In *Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC 2010)*, pages 1 – 8. IEEE, 2010.

[14] A. Morales-reyes and A. T. Erdogan. A structure-based coarse-fine approach for diversity tuning in cellular gas. *Advances in Electrical and Computer Engineering*, 12(3):39–46, 2012.

[15] N. Nedjah and L. de Macedo Mourelle. An efficient problem-independent hardware implementation of genetic algorithms. *Neurocomput.*, 71(1-3):88–94, Dec. 2007.

[16] D. Simoncini, P. Collard, S. Verel, and M. Clergue. From cells to islands: An unified model of cellular parallel genetic algorithms. In *2006 International Conference on Cellular Automata*, page 248Öš57. Springer-Verlag, 2006.

[17] D. Simoncini, P. Collard, S. Verel, and M. Clergue. On the influence of selection operators on performances in cellular genetic algorithms. In *Proceeding of IEEE Congress on Evolutionary Computation (CEC'07)*, page 4706Öt'713. IEEE, 2007.

[18] M. Tomassini. *Spatially Structured Evolutionary Algorithms, Artificial Evolution in Space and Time*. Springer. Series: Natural Computing Series, 2005.

[19] P. Vieira dos Santos, J. Alves, and J. Canas Ferreira. A framework for hardware cellular genetic algorithms: An application to spectrum allocation in cognitive radio. In *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*, pages 1–4, Sept 2013.

[20] Z. Zhu, D. Mulvaney, and V. Chouliaras. Hardware implementation of a novel genetic algorithm. *Neurocomputing*, 71(1Öş):95 – 106, 2007. Dedicated Hardware Architectures for Intelligent Systems Advances on Neural Networks for Speech and Audio Processing.