J. Parallel Distrib. Comput. 73 (2013) 122–136

Contents lists available at SciVerse ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Dynamic Fault-Tolerant three-dimensional cellular genetic algorithms

Asmaa Al-Naqi *, Ahmet T. Erdogan, Tughrul Arslan

School of Engineering, The University of Edinburgh, Kings Building, Mayfield Road, EH9 3JL, Edinburgh, Scotland, UK

ARTICLE INFO

Article history:
Received 13 September 2011
Received in revised form
28 June 2012
Accepted 5 September 2012
Available online 27 September 2012

Keywords:
Cellular genetic algorithms (cGAs)
Fault tolerance
Evolutionary algorithms (EAs)
Dynamic adaptation
Migration

ABSTRACT

This paper proposes a new dynamic and algorithm-based approach to achieve fault tolerance using 3D cellular genetic algorithms (Dynamic Fault-Tolerant 3D-cGA). The proposed algorithm is an improved version of our previous algorithm (Fault-Tolerant 3D-cGA) that introduces and utilizes a dynamic adaptation feature to achieve further improvement. In Dynamic Fault-Tolerant 3D-cGA, faulty individuals are isolated and the maximum number of fitness evaluations is recalculated to adapt to faults encountered. To improve the performance of the algorithm, a mitigation technique is integrated into our algorithm by introducing an explicit migration operator. A benchmark of well-known real-world and test problems is used to test the effectiveness of the algorithm in order to investigate the influence of adaptation schemes and migration on algorithm performance. Faulty critical system data is tackled in conjunction with various fault ratios. To illustrate the improvement achieved, Dynamic Fault-Tolerant 3D-cGA is compared with Fault-Tolerant 3D-cGA, the previously proposed algorithm. The overall results demonstrate the ability of Dynamic Fault-Tolerant 3D-cGA to maintain system's functionality despite an increasing number of faults with up to 40% of processing elements (PEs), and clearly illustrate the importance of migration. Significant improvements in the performance of the algorithm, measured as efficiency, efficacy, and speed, are achieved, especially when migration is employed.

1. Introduction

Evolutionary Algorithms (EAs) have been widely employed for different applications. This paper targets EAs' application to the design of Fault-Tolerant systems. The considerable reduction in feature sizes of electronic circuits increases the possibility of transient errors to occur; in particular, single event upsets (SEUs) [33]. For this reason, there is high demand for implementing efficient and reliable high-performance systems that can quickly adapt to different failures. EAs are powerful optimization techniques that have shown success and universal applicability in different fields. They have proved their ability to solve problems of diverse complexities, along with considerable gains in terms of reliability, scalability, and efficiency due to their inherent parallelism. They work over a population of potential solutions (individuals) and apply simple stochastic operators that push the population towards better solutions. They are classified into either serial or parallel search techniques according to the way individuals communicate with each other [30]. Serial versions use a single population, while parallel versions use a structured population. There are several parallel models; the two most popular models are distributed (dEAs) and cellular (cEAs) evolutionary algorithms. In dEAs, the population is divided into multiple subpopulations (islands), where each of them is assigned to a processing element (PE). In this model, each island is evolved independently, and interactions between individuals occur by exchanging a number of individuals among islands (i.e., migration). On the other hand, in cEAs the population is distributed over PEs arranged as n-dimensional grid with wrapped edges (i.e., toroidal). In addition, each individual is assigned to a PE (or cell), and interactions among individuals occur through their defined local neighborhoods, where an implicit migration is applied due to the overlapped neighborhoods. The spatial arrangement of the population leads to better exploration and exploitation trade-off, which is the key for determining the behavior of the algorithm. The exploration is provided through the overlapped neighborhoods, and aims at enhancing the diversity of the population and avoiding premature stagnation. In contrast, the exploitation targets each neighborhood to improve the quality of the solutions; and can be controlled through genetic operators. Inappropriate balance between exploration and exploitation leads to inefficient search. There are several studies for controlling exploration/exploitation trade-off. Giacobini et al. [14], Alba and Troya [3], and Alba and Dorronsoro [1] showed that the balance between exploration and exploitation can be achieved through changing the size and shape of the neighborhoods and/or the grid's shape. In later works, Simoncini et al. [27,28] proposed new selection techniques, namely, anisotropic and centric selections for tuning the exploration/exploitation trade-off. Based on the latter, another adaptive cEA was proposed in [4].

* Corresponding author.
E-mail addresses: A.Al-Naqi@ed.ac.uk (A. Al-Naqi), Ahmet.Erdogan@ed.ac.uk (A.T. Erdogan), T.Arslan@ed.ac.uk (T. Arslan).

0743-7315/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2012.09.011

The grid's topology also plays an important role in determining the performance of the algorithm. Research works surrounding cGAs are commonly concerned with their implementation on one- or two-dimensional grid topology. In this study, a 3D cubic topology is utilized. A cubic topology allows good solutions to spread quickly to all PEs due to its shorter diameter [10], as well as diverse degrees of exploration and exploitation. In past studies [9,19], a 3D architecture was utilized and investigated; the overall results showed improvements in the performance of the algorithm when compared with smaller grid dimensions. A further reason for using a 3D topology is its amenability to be implemented with new advanced custom silicon chip technologies to achieve added significant benefits, such as fast operation, reduction in power consumption, new design possibilities, heterogeneous integration, circuit security, and wide bandwidth [12].

In this study, a cGA is employed because it is one of well-known cEA techniques. cEAs and cGAs in particular showed not only their ability to solve problems of different complexities, but also exhibit speedup in computation time, robustness, ability to escape local optima, and therefore achieve high efficiency and efficacy [2]. Consequently, cGAs' suitability for real-time applications increases.

This work aims to introduce a new dynamic 3D-cGA, which is tolerant to SEU faults targeting PE's registers that correspond to critical data (in this study, fitness score registers). Given the fact that there are various fault scenarios, the most critical fault scenario (which will be defined and discussed later) is tackled in this study. The basis of our design was proposed in our previous study [5], which is improved by introducing a dynamic adaptation feature in this study.

The first objective of this study is to maintain system reliability even with a growing number of faulty registers or PEs (recall that in a cGA each PE holds one individual, therefore comprises one fitness score register). The other objective is to improve the performance of the algorithm by mitigating the impact of faults.

The paper is structured as follows. Section 2 presents an introduction to fault tolerance. The canonical model for cGAs implemented on a 3D architecture is discussed in Section 3. Dynamic Fault-Tolerant 3D-cGA is proposed in Section 4. The benchmark problems selected to evaluate and analyze the proposed algorithm are described in Section 5. Section 6 presents the parameters used in experiments and the results obtained. Finally, concluding remarks are given in Section 7.

2. Fault tolerance

This section presents the main causes of system failures considered in this study. Due to the significant miniaturization of electronic systems and their use in hostile environments, such as space, they are subjected to various anomalies, such as plasma and radiation, among others [32]. Such anomalies have major effects on electronic systems, which result in different types of failures. In this study, attention is paid to radiation effects because the radiation is the main contributor to failure (45%), with SEUs (also known as transient errors) having the highest impact of all possible radiation effects (80%) [32]. Recently, radiation-induced SEUs have also been observed at ground level due to the considerable reduction in feature sizes of electronic circuits and increase in functional complexity and sensitivity [25].

SEUs occur as single-bit or multiple-bit flips in memory or data registers due to the passage of one or more energetic radiation particles [18]. SEUs, or soft errors, do not cause permanent damage to a system's functionality, and can be handled by Fault-Tolerant techniques.

Fault-Tolerant techniques are classified into hardware or software fault tolerance, or a combination of both. The most commonly used hardware technique to mitigate SEUs is Triple Modular Redundancy (TMR). However, TMR is very area-extensive and cannot cope with all errors that occur. SEU hardware Fault-Tolerant techniques can rapidly detect and recover faults; however they incur overhead, which increases the cost and complexity of systems. Further, in general, hardware techniques cannot handle all types of random and multiple-bit errors caused by potential transients. These types of errors, specifically SEUs, cause functional impacts (software faults), rather than physical impacts. Consequently, many error-coding techniques have been proposed to solve the above-mentioned problems. These are seldom implemented due to their complexity.

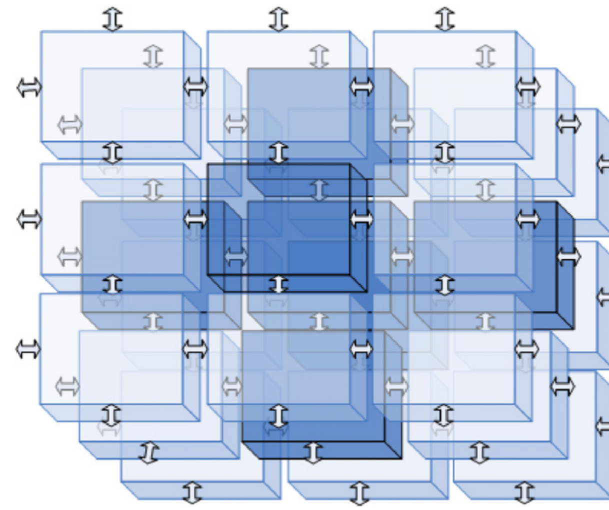

Fig. 1. 3D toroidal grid (cubic) topology when implemented in cGA.

Nowadays, Fault-Tolerant techniques to mitigate SEUs are being intensely researched, not only for aerospace applications, but also for terrestrial applications. Gong et al. [16] proposed a hardware approach for tolerance to single event effects, a general class of SEUs, where two new structures were presented and compared with the traditional TMR. Conversely, Singh et al. [29] presented a software approach to SEU tolerance that combined several techniques, such as checkpoint and TMR. In addition, several studies to explore the ability of cGAs to tackle SEUs have been conducted. Research studies related to the ability of a normal cGA and a parallel cGA to deal with SEUs that occur in fitness score registers were presented in [22,21], while the ability of an adaptive cGA to handle SEUs-targeted chromosomes registers was explored in [20]. In all previous studies, EAs have proved their capability and power to tackle SEUs, as well as in improving the performance of the algorithm in terms of efficacy and efficiency.

This study deals with failures caused by SEUs when targeting individuals' phenotypes, particularly when fitness scores are stuck at 'one' or 'zero'. Although other possible memory or data registers, such as chromosome and finite machine state could be also targeted, this study focuses on fitness registers due to the importance of fitness information in guiding the search.

3. Cellular genetic algorithms

In this section, we discuss and illustrate the pseudocode of the canonical model of cGAs implemented on a 3D architecture. In the next part of this section, the concept of selection intensity is discussed to illustrate its effect on the behavior of the algorithm.

In cGAs, the population is distributed over an nD toroidal grid, such that each individual is assigned to a grid's position (cell). This arrangement forces individuals to interact with their local neighbors. In this study, the population is arranged in a 3D grid with wraparound edges. A neighborhood consists of seven