

REVIEW OF PARALLEL GENETIC ALGORITHM BASED ON COMPUTING PARADIGM AND DIVERSITY IN SEARCH SPACE

A. J. Umbarkar¹ and M. S. Joshi²

¹Department of Information Technology, Walchand College of Engineering, India

E-mail: anantumbarkar@rediffmail.com

²Department of Computer Science and Engineering, Jawaharlal Nehru Engineering College, India

E-mail: madhuris.joshi@gmail.com

Abstract

Genetic Algorithm (GA), a stochastic optimization technique, doesn't ensure optimal solution every time. Nowadays there is a need to improve the performance of each and every application so that the time required for obtaining quality solution can be minimized. This paper gives a brief overview of theoretical advances and computing trends, particularly population diversity in PGA (Parallel GA) and provides information about how various authors, researchers, scientists have parallelized GA over various parallel computing paradigms viz. Cluster, MPP (Massively Parallel Processing), GPGPU (General purpose Graphics Processing Units), Grid, Cloud, Multicore/HPC to ensure more optimal solution every time with efficacy and efficiency.

Keywords:

Genetic Algorithm (GA), Parallel GA (PGA), General Purpose Graphics Processing Unit (GPGPU), Massively Parallel Processor (MPP), Population Diversity, Cloud, Grid, Cluster, HPC

1. INTRODUCTION

The GA is one of the most important soft computing tools used for solving many optimization problems. Being stochastic (soft computing tool) in nature, it is non-deterministic process. Therefore solution evolved using GA may or may not be optimized i.e. it doesn't ensure optimal solution every time. This is called as premature convergence problem (converge too early giving suboptimal solution). Besides this there are various problems associated with GA viz. diversity problem (either increase population size or have more than one population to get more diversified search space to find global optimum solution), population size problem (increase in population size may deteriorate the performance of GA), curse of dimensionality problem (increases in dimension of search space deteriorates performance of GA).

Also, there are many real-life problems which when run on serial machines may take days or weeks of computing time to solve. Although, by using finite number of computing resources in parallel, one cannot lower the intrinsic time complexity of a problem but parallelism often reduces time to reasonable levels. This is useful (or important) in industrial or commercial setting where the time to get the solution is critical for decision making.

To address these problems efficiently, one can take advantage of parallel computing environment viz. Cluster, MPP, GPGPU, Grid, Cloud, Multicore or HPC and can exploit their functionality to solve GA effectively with more speed up. Apart from this diversity - where extra population (called as reserve population) is used to provide additional diversity to the main

population through crossbreeding - also helps to get the optimal solution efficiently.

This paper provides brief literature review of implementation of PGA on various parallel computing paradigms. In the remnant of this paper we briefly introduce various parallel computing paradigms available. Thereafter we discuss recent work, with relevance to diversity in PGA, carried out by various researchers over different parallel computing paradigm. Finally, we discuss the finding of this review work.

2. PGA OVER COMPUTING PARADIGMS

Literature survey given in [1], [2] are on Advances, Computing Trends, application and Perspective of Parallel Genetic Algorithm (PGA) [3]. In computing trends the important issues are architecture of computing paradigm, OS, topologies and programming language, facilitated with set of special system calls or libraries like - Linda, OpenMP^{a,h} (Open Multiprocessing), HPF (High Performance Fortran), Parallel C, Java using communication libraries, MPI (Message-Passing Interface)^b, Express MPI, PVM (Parallel Virtual Machine), POSIX threads, C sharp, CUDA^c (Compute Unified Device Architecture (CUDA), Brooks^d, OpenCL^e (Open Computing Language), OpenGL^f (Open Graphics Library) and Java threads on SMP (Symmetric multiprocessing) machines, available to exploit the functionality. Some parallel programming options are available in footnote^g.

The performance of GA is optimized by parallelizing it. The Way in which GAs/PGAs are parallelized depends upon following parameters [4]:

- Population Initialization
- Single or multiple subpopulation(s)
- Migration policy, migration rate
- Method of selection
- Fitness evaluation
- Crossover
- Mutation
- Survival selection method

a. <https://computing.llnl.gov/tutorials/openMP/>

b. <http://www.lam-mpi.org/>

c. <https://developer.nvidia.com/category/zone/cuda-zone>

d. <http://graphics.stanford.edu/projects/brookgpu/>

e. <https://developer.nvidia.com/opencv>

f. <https://www.opengl.org/>

g. <http://wotug.ukc.ac.uk/parallel/>

h. <http://www.openmp.org>

Depending on how each of these elements is implemented, there are different methods of PGAs viz. Master-Slave parallelization (distributed fitness evaluation)- with two versions a) Synchronous b) Asynchronous, Static subpopulations with migration, Static overlapping subpopulations (without migration), Massively parallel GAs, Dynamic demes (dynamic overlapping subpopulations), Parallel steady-state GAs, Parallel messy GAs, Hybrid methods (e.g. static subpopulations with migration, with distributed fitness evaluation within each subpopulation).

The computing paradigms considered for survey are Cluster, MPP, Grid, GPGPU, cloud computing and Multicore/ HPC.

2.1 GA OVER CLUSTER AND MPP

Cluster (a set of loosely connected or tightly connected computers) is generally deployed for improving the performance and availability over that of a single computer. It is much more cost-effective than single computers of comparable speed or availability.

Most of the existing parallel implementations of PGAs use either cluster or MPP (large number of processors (or computers) to perform a set of coordinated computations in parallel.) for execution. A communication delay should be taken into consideration while implementing GA on cluster and MPP. Most of the worlds supercomputers are clusters and rest are MPPs [54]. MPI is very commonly used library for Clusters and MPPs environments. Cluster has various disadvantages over grid such as centralized control, less security, no direct access to distributed heterogamous resources and complex access to remote data sources. The advantage of Cluster and MPPs is, it consist of homogenous nodes, hence parallel algorithms can be easy to implement on them.

Most famous model for PGAs running over clusters is the island model [5], [6]. The [7] [8] and [9] reports the PGA implemented on cluster.

2.2 GA OVER GRID

Grid (composed of many loosely coupled computers networked together to perform large tasks) is important and quickly evolved parallel paradigms. It is widely accepted distributed form of computing [10], [11], [12], [13], [14]. MPI is very commonly used library for grid environment. It is more loosely coupled, heterogeneous, and geographically dispersed form of distributed computing than cluster. Grid has many advantages over cluster such as no centralized control, security, access to distributed heterogeneous resources, easy and reliable access to remote data sources and service to any available application.

GA on grid is called as Grid Oriented Genetic Algorithm (GOGA). This term was first introduced by [15], [16]. Coarse grained, medium grained and fine grained GAs is implemented on grid.

2.3 GA OVER GPGPU

Today, the cheapest and highly available computing power is GPU. Many PGAs are implemented on GPGPU. GAs can be parallelized over GPGPU by parallelizing the genetic operators such as random number generator, selection, crossover,

mutation. Some implementation guidelines are given in [17], [18], [19]. Some of the parallel hybrid GA [20], steady GA, binary GA, real coded GA [21], cellular GA [22] etc. are implemented over GPGPU. Many GAs / PGAs are yet to be implemented on GPGPU, so researchers have huge scope in this area and can further exploit the functionality of GAs using GPU. Based on the search space complexity of problems, it is possible to provide diversity in search space. For such problems use of GAs / PGAs on GPGPU is a good option. Many real life problems, those support the SIMD (Single Instruction Multiple Data) (Flynn's taxonomy) architecture, can be effectively solved over GPGPU. CUDA and OpenCL are programming options available for programming with GPGPU.

Problems solved by PGA on GPGPU are Medical Image Registration [23], Feature Selection, Electrical Circuit Synthesis and Data Mining [24], SAT Problems [25], Function Optimization [26], Benchmark Problems [27], [28], [29], Texture-Rendering [30], One-MAX Problem [31], Quadratic Assignment Problems [32], Non-convex Mixed Integer Non-Linear Programming (MINLP) and Non-convex Non Linear Programming (NLP) Problems [33], Cellular Automata Rules Acceleration [34], Stereo Matching [35], Data Mining[36], Drug discovery [37], Gaming Application [38], Image Matching[39], Gaming-Sudoku Solution [40] , Task Scheduling and Load Balancing Based on Workload and Case Study of Financial Option Pricing Problem [41], Multi-objective evolutionary algorithm Test Suite Minimization [42], Sketching Interface [43], Daily Activity Plans [44] etc.

2.4 GA OVER CLOUD

Today is the fast growing era of computing and supporting software for parallelization. Buying latest configuration PC or high performance system will be absolute in a year or two. Further, it is not possible to invest in all latest computing infrastructures and softwares. Therefore, cloud computing (use of computing resources such as hardware and software, which are delivered as a service over a network) is a good solution for on demand computing power (PaaS- Platform as a service) and on demand software support (SaaS – Software as a Service). The availability of computing infrastructure and software on cloud is quite easily available. Due to computational intensive and parallel nature of PGAs, it is hard to implement and optimize the performance on cloud. Developing PGA on cloud will require knowledge of GA as well as of cloud. Many issues related to cloud like security, bandwidth etc. are still under discussion.

Zhao et al., (2011) implemented PGA on prototype of cloud called Hadoop. Hadoop is an open source Cloud computing. The Function Optimization using PGA on Cloud Computing gives speed-up but with short communication delay and it is suitable to solve large dimension problems. Comparing traditional PGAs on cloud with - HPC, cluster, Grid - PGA on cloud is simple, easy to implement and easy to extend to solve larger-scale problems [45]. PGA on cloud will not be effective because of its distributed environment, as performance depends on network delay (Internet bandwidth).

To solve the compute intensive optimization problems without hiring the computing infrastructure, cloud is good option. Some of optimization problems which are solved using PGAs are Resource Scheduling [46], Scheduling HPC

Applications [47], Task Scheduling[48], [49], [50], Performance Improvement of Cloud Storage [51], Power Management in Cloud [52], Clustering composite SaaS components [53] etc.

2.5 GA OVER MULTICORE AND HPC SYSTEM

Many GAs are implemented on multicore system but does not care about their CPU/core utilization. Nowadays multicore processors are getting cheaper and are easily available. One cannot ignore their importance anymore. Multicore systems have multiple processing cores on same chip while multiprocessor systems have multiple chips inside a system. PGAs can be implemented over multicore architecture using MPI or OpenMP. MPI performs better on the distributed memory architecture while OpenMP performs better on the shared memory architecture [54]. The OpenMP, POSIX threads and Java threads are also the programming option available for programming on Multicore and HPC. PGA's parallel algorithmic design need to be considered and explored on multicore and HPC computing paradigm. The challenges of GA over HPC facing are fault tolerance, scalability, load balancing, data storage and multicore architectures.

Zheng et al. (2011) explained the PGA with perspective of architectures-multicore and many core systems. In their work, they have shown Architecture-based performance evaluation of GAs on Multi/Many-core Systems. They described PGA on thread organization, memory hierarchy, core utilization, execution time and solution quality. They considered the material, island and cellular models of PGA for solving non linear programming problem [55]. Cristea showed the conception and design of PGA on HPC and performance analysis models [56]. Zhuang et al. proposed PGA for IC chip planning on HPC [57]. Dunlop et al. uses GA on automatic benchmark tuning tool [58].

PGAs are implemented over Clusters, MPPs, Grids, GPGPU, cloud computing, Multicore and HPC. The GAs implemented over distributed environment such as cluster, MPPs, grid and cloud computing is also called PGAs, because of parallel hardware used. The design of algorithm needs to be tuned with the hardware selected for implementation. The parameters to be considered are network delay, system configuration of each node in distributed system. Network delay plays important role on the performance of algorithm in case of distributed system. The programming on such parallel hardware is called loosely coupled programming.

In case of Multicore, HPC and GPGPU, GAs on them are also called PGAs; here also the design of algorithm needs to be tuned with the hardware selected for implementation. While designing PGA on Multicore, HPC and GPGPU, the number of cores can be taken into consideration to maximize the CPU or GPU utilization. The programming on such parallel hardware is called as tightly coupled programming. No parameters like network delay, system configuration of each node need to be considered in tightly coupled system. PGAs have capacity to run in parallel on many computing paradigms but finding suitable hardware and software will give optimal solution with optimal resource utilization.

3. LITERATURE REVIEW

Recently, PGAs have received more attention from the research community due to changing computing infrastructures.

Also, PGAs have many interesting unique features like parallelizable operators, ability to search large dimension search space etc. which require in-depth analysis [59]. PGA is intended to provide following benefits [60]:

- Reduces the time to locate a solution
- Reduces the number of function evaluations
- Explores the large populations size over the parallel platforms used for running the algorithms
- Improves the quality of the solutions
- Solves the large scale, large dimensions problems with efficacy and efficiency

The GAs with multiple populations but normal selection, crossover and mutation with exchange of some individuals are also called as PGAs. The migrations rate, migration destination type were important parameters introduced in PGAs [61], [62], [63], [64]. The interdeme / interpopulation communications result in significant improvements in the quality of the solutions, which results in reductions in deme/population size and the computational effort, however with more processors, additional communication is required. Thus there is a tradeoff between increasing communication cost and decreasing computations [65], [66].

Ursem (2000) [67] proposed the Multicannical GA (MGA) which spread different population through different peaks. In the fitness landscape, a hill-valley detection procedure and a migration policy are used. The MGA was very complex and requires more tuning time.

Sefrioui and Periaux (2000) [68] proposed Hierarchical Genetic Algorithms (HGAs) with multi-layered hierarchical topology and multiple models for optimization problems.

Fernandes, Tavares and Rosa (2000) [69] proposed combined strategy of outbreeding and a varying population size. Fernández [70] (2000) experimented multipopulation parallel genetic programming. Again Fernandes, Tavares and Rosa (2001) proposed the importance of non random mating called as assortative or disassortative mating in GA. They tested algorithms by varying population size scheme on royal road problem.

Rivera (2001) [71] investigated how to implement PGAs for getting quality solutions efficiently. Rivera reviewed the state-of-the-art in PGAs, parallelization strategies, emerging implementations and relevant results.

Alba and Troya (2001) [72] proposed a common framework for studying PGAs on cluster of workstations. It shows linear and even super-linear speedup. Alba and Troya (2002) [73] bring some uniformity in comparison, and knowledge exchanges among the traditionally opposite kinds of serial and parallel GAs. They analyzed the properties of steady-state, generational and cellular GAs and extended the idea to distributed model ring of GA islands.

Giacobini (2003) [74] proposed a theoretical study of the selection pressure in asynchronous cellular (fine-grained) evolutionary algorithms (cEAs).

Xiao and Armstrong (2003) [75] proposed a model of parallel evolutionary algorithms (EAs) called a specialized island model (SIM).

Gagné (2003) [5] says that the classic master slave distribution model was superior to the island-model when

exploiting Beowulf and networks of heterogeneous workstations. They identified the key features of a good computing system for evolutionary computation viz. transparency, robustness and adaptivity.

Yang (2003) [76] proposed Primal-Dual Genetic Algorithm (PDGA) which adopted complimentary and dominance mechanism for diversity. PDGA solved complex dynamic problems efficiently than traditional SGA.

Ochoa and Jaffe (2005) [77] proposed the idea related to Assortative Mating GA (AMGA) and applied it to dynamic optimization problems. Assortative GAs (very similar to AMGA) are used to solve a dynamic knapsack problem. Again Ochoa and Jaffe (2006) [78] proposed a study on the error threshold of replication in GAs with different mating strategies.

Russell (1998) [79] investigated that the Outbreeding increases the diversity. The inbreeding decreases the genetic diversity in a population. The mating of dissimilar (not from one population) individuals, called as dissortative mating (negative assortative mating), does not give the diversity but this type of mating results in highly distinct cluster of similar genotypes. In EAs, selection pressure and genetic diversity are important parameters. EAs need to provide selection pressure and genetic diversity with delicate equilibrium between exploration and exploitation. Therefore nonrandom mating naturally came in EAs in order to provide genetic diversity and avoid premature convergence.

The crossover (recombination or mating) gives exploitation and mutation is responsible for exploration. The choice of exploitation, exploration, selection pressure and genetic diversity are of prime concern depending on the problem characteristics like static and dynamic, unimodal and multimodal, linear and non linear, separable and non separable, Scalable and non scalable and the dimensionality of the problem [80].

Most of the PGAs concentrated on multiple demes with common fitness function for all populations. But there are some PGAs, with multiple demes and multiple fitness functions, implemented on multicore/HPC, GPGPU and even on cloud which gave birth to the concept of diversity. The algorithm is said to be parallel only when its design is parallel and not because of parallel hardware or parallel programming used for implementations. But many researchers used the latter concept of parallelism.

PGAs AND DIVERSITY

Many researchers have been working on the issue of maintaining population diversity, avoiding premature convergence and improving the convergence speed. Research in the area of evolutionary algorithm (EA) is mainly categorized into following five categories- 1) Dynamic operator and parameter control, 2) Specialized selection operators, 3) Specialized variation operators 4) Memory-based algorithms and 5) Complex population structures [100].

In some of the multiple demes based PGAs, the concept of additional objective (Fitness function) is added for diversity in search space. These PGAs are parallel by design and provide diversity in search space and quality of solutions irrespective of other kind of parallelism. When attempts are made to get the quality solutions through diversity as a second objective, it is called as niching techniques. Niching techniques in GAs are

given in [81]. Niche PGAs can provide both multiple high-quality solutions for multimodal problems and techniques for improving diversity in population [2].

The literatures based on diversity as second objectives in PGAs are as follows,

In GA, the concept of reserve population with diversity as an objective, called DPGA, was experimented by Park and Ryu, (2007) [82]. The DPGA outperforms on stationary optimization problems.

Park, Choe, and Ryu (2007) [83] has proposed DPGA-ED, an improved version of binary representation based DPGA. The reserve population in DPGA-ED is different than reserve population in DPGA, as it evolves itself. The DPGA-ED outperforms on nonstationary optimization Problems. Park, Choe, and Ryu (2008) [84] has proposed DPGA2 for Nonstationary optimization Problems. The improvement in DPGA2 is that there are two reserve populations, which provide controlled inflow information to main population with survival selection. DPGA2 shows the performance without relying on prior knowledge of nonstationary optimization problems.

Junhua and Ming (2008) [85] proposed DSGA called Dual Species GA. In this algorithm one subpopulation work on local exploitation and other subpopulation work for global exploration with migration between them for optimal solution.

Fernandes and Rose (2008) [86] proposed Variable Dissortative Assortative Mating GA (VDMGA) based on mating strategies. VDMGA performs crossover when the hamming distance between the chromosomes is below a threshold value. VDMGA maintain genetic diversity and can tackle static and dynamic optimization problems. The VDMGA falls under the category of Dynamic operator and parameter control.

Park (2010) [87] has proposed binary, real valued and ordered based DPGA with adaptive diversity control for Unimodal and multimodal optimization Problems. Real valued DPGA for Unimodal functions with large D is fluctuating. Real valued DPGA for multimodal functions with large D is better. Real valued DPGA for multimodal functions with small D is fluctuating. Binary valued DPGA for Unimodal functions with large D is better. Binary valued DPGA for multimodal functions with large D is better. Binary valued DPGA for multimodal functions with small D is fluctuating. It is observed that, in DPGA, by controlling the crossbreeding probability between main and reserve population the diversity can be adjusted base on requirements of problems modality and other features.

Umbarkar and Joshi in [88] compared the performance serial DPGA with Multithreaded Parallel DPGA (MPDPGA). The performance of MPDPGA with DPGA was in terms accuracy, number of generations and execution time on multicore system.

The fitness calculation of reserve population, to add diversity in search space, is comparison of two chromosomes; usually compute by using Hamming distance or Euclidian distance, which is an additional computational overhead, results in increased search time. The main population evolution, reserve population(s) evolution and crossbreeding are the three important processes of DPGA.

Umbarkar and Joshi in [89] compared the performance of DPGA with simple GA and OpenMP GA for multimodal

function optimization. DPGA shows the performance over simple GA and programming parallel OpenMP GA.

The DPGA, DPGA-ED, DPGA2, DSGA and MPDPGA falls under the category of complex population structures, where it works on the concept of dual populations to get the optimal solution.

The other population diversity based works are - Classical EP (CEP), Improved Fast EP (IFEP) [90], Adaptive EP with Lévy Mutation (ALEP) [91], Island-model GA (IMGGA) [92], Restricted Truncation Selection (RTS) [93], Real Coded Memetic Algorithm (RCMA) with Crossover Hill Climbing (XHC) [94], Comprehensive Learning Particle Swarm Optimizer (CLPSO) [95], RCMA with Adaptive Local Search (LSRCMA) [96], Differential Evolution with Neighborhood Search (NSDE) [97], Covariance Matrix Adaptation Evolution Strategy (CMAES) [98], Diversity Guided Evolutionary Programming (DGEP) [99].

4. DISCUSSION

The computing power is increasing over the time. It is moving from loosely coupled architecture to tightly coupled architecture or hybrid architecture. This computing power can be exploited optimally by using various programming languages mentioned above. Therefore the algorithm designers have challenges in understanding the latest computing and programming architectures. So it is need of time to redesign the algorithm for new computing paradigms like cloud, HPC and GPGPU.

In case of MPP, Grid and cluster- fine grained, medium grained and coarse grained GAs are explored and superliner speedup is observed with some of the applications. The GPGPU is latest computing platform, where GPUs are used for processing Single Instruction Multiple Data (SIMD) kind of part of algorithm. It is observed that the PGAs which support SIMD architecture when implemented on GPGPU give better speedup.

GA on cloud is prototypic concept of GA on cluster. It is helpful to solve the optimization problem in which processing of big data is needed. It is observed that the GAs are used to solve the optimization problems of cloud but GAs are not implemented on cloud to solve optimization problems.

In case of HPC/Multicore systems many challenges like proper utilization of cores, memory, and cache are yet to be addressed. Selection of proper programming language, algorithmic design (based on the architecture), use of niching technique, diversity etc. also play critical role in obtaining optimal solution efficiently.

Survey shows that, PGA over parallel platforms helps in reducing the function evolution, improve the quality of solution, explore the large population size, and solve the large scale, large dimensions problems efficiently.

Some authors proposed multiple populations based PGA and the importance of communication delay in performance is recognized.

PDGA, VDMGA, DPGA, DPGA-ED, DPGA2 and MPDPGA works on the concept of diversity in population(s). DPGA has extra population with diversity as objective, results in increased in computational efforts.

The diversity based EAs are available in the literature but not tried to parallelize them on today's parallel computing paradigms such as multi-core system, GPU etc. The reasons for not being exposed for parallelization could be:

- i. The additional population or objective for diversity adds complexity to the EA.
- ii. Researchers have concentrated more on altering the EA to improve the performance rather than modifying it for implementation over parallel computing system.

ACKNOWLEDGMENT

We express our sincere thank to all the authors, whose papers, in the area of evolutionary algorithm over various parallel computing paradigms, are published in various conference proceedings and journals.

REFERENCES

- [1] Z. Konfrst, "Parallel Genetic Algorithm: Advances, Computing Trends, application and Perspective", *Proceedings of 18th International Parallel and Distributed Processing Symposium*, 2004.
- [2] E. Cantú-Paz, "Efficient and Accurate Parallel Genetic Algorithms", Springer, 2000.
- [3] E. Cantú-Paz, "A Survey of Parallel Genetic Algorithms", *Calculateurs Paralleles*, Vol. 10, No. 2, pp. 141-171, 1998.
- [4] M. Nowostawski and R. Poli, "Parallel Genetic Algorithm Taxonomy", *Proceedings of the 3rd International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, pp. 88-92, 1999.
- [5] C. Gagné, M. Parizeau and M. Dubreuil, "The Master-Slave Architecture for Evolutionary Computations Revisited", *Proceedings of the International Conference on Genetic and Evolutionary Computation*, Vol. 2724, pp. 1578-1579, 2003.
- [6] S. Lin, W. Punch and E. Goodman E, "Coarse-grain parallel genetic algorithms: categorization and new approach", *Proceedings of Sixth IEEE Symposium on Parallel and Distributed Processing*, pp. 28-37, 1994.
- [7] M. Bubak and K. Sowa, "Object-oriented implementation of parallel genetic algorithms, Chapter 16 in: Buyya R, High Performance Cluster Computing", *Programming and Applications*, Vol. 2, Prentice Hall, pp. 331-349, 1999.
- [8] G. A. Sena, D. Mergherbi and G. Isern, "Implementation of a parallel genetic algorithm on a cluster of workstations: Travelling salesman problem a case study", *Future Generation Computer Systems*, Vol. 17, No. 4, pp. 477-488, 2001.
- [9] Z. Konfrst, "Java Threads on an Experimental IBM Opteron Cluster System", Technical Report of ESP, CTU VC, Prague, pp. 6, 2004.
- [10] R. Hauser and R. Männer, "Implementation of Standard Genetic Algorithms on MIMD Machines", *Parallel Problem Solving from Nature – PPSN III*, pp. 504-513, 1994.
- [11] R. Tanese, "Distributed Genetic Algorithms", *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 434-439, 1989.

- [12] H. M. Voigt, J. Born and I. Santibanez-Koref, "Modeling and Simulation of Distributed Evolutionary Search Processes for Function Optimization", *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, pp. 373-380, 1991.
- [13] H. M. Voigt, I. Santibanez-Koref, and J. Born, "Hierarchically Structured Distributed Genetic Algorithm", *Parallel Problem Solving from Nature 2, PPSN-II*, pp. 157-166, 1992.
- [14] H. Imade, R. Morishita, I. Ono, N. Ono and M. Okamoto, "A grid-oriented genetic algorithm for estimating genetic networks by s-systems", *Proceeding of SICE 2003, Annual Conference*, Vol. 3, pp. 2750-2755, 2003.
- [15] J. Herrera, E. Huedo, R.S. Montero and I.M. Llorente, "A grid oriented genetic algorithm", *Advances in Grid Computing -EGC 2005*, pp. 315-322, 2005.
- [16] H. Imade, R. Morishita, I. Ono, N. Ono and M. Okamoto, "A grid-oriented genetic algorithm framework for bioinformatics", *New Generation Computing – Grid Systems for Life Sciences*, Vol. 22, No. 2, pp. 177-186, 2004.
- [17] M. Lazarova, "Efficiency of parallel genetic algorithm for solving N-queens problem on multicomputer platform", *Proceedings of 9th WSEAS International Conference on Evolutionary Computing*, pp. 51-56, 2008.
- [18] R. Arora, R. Tulshyan and K. Deb, "Parallelization of binary and real-coded genetic algorithm on GPU using CUDA", *Proceedings of IEEE Congress on Evolutionary Computation*, pp.1-8, 2010.
- [19] M. Oiso and T. Yasuda, K. Ohkura and Y. Matumura, "Accelerating Steady-state genetic algorithms based on CUDA architecture", *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 687-692, 2011.
- [20] M. Wong and T. Wong, "Parallel Hybrid Genetic Algorithms on Consumer-level Graphics Hardware", *Proceedings of Congress on Evolutionary Computation*, pp. 2973-2980, 2006.
- [21] M. Oiso, Y. Matsumura, T. Yasuda and K. Ohkura, "Implementing genetic algorithms to CUDA environment using data parallelization", *Technical Gazette*, Vol. 18, No. 4, 2011.
- [22] P. Vidal and E. Alba, "A multi-GPU implementation of a cellular genetic algorithm", *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1-7, 2010.
- [23] W. Withayachumnankul, B. Laksanapanai and C. Pintavirooj, "Hardware-accelerated Objective Function Evaluation for Medical Image Registration", *Proceedings of IEEE Region 10 Conference TENCON*, pp. 419-422, 2004.
- [24] M. Wong, T. Wong and K. Fok, "Parallel Evolutionary Algorithms on Graphics Processing Unit", *Proceedings of IEEE Congress on Evolutionary Computation*, Vol. 3, pp. 2286-2293, 2005.
- [25] Z. Luo and H. Liu, "Cellular Genetic Algorithms and Local Search for 3-SAT Problem on Graphic Hardware", *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 2988-2992, 2006.
- [26] M. Wong and T. Wong, "Parallel Hybrid Genetic Algorithms on Consumer-level Graphics Hardware", *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2972-2980, 2006
- [27] D. Robillard, V. Marion and C. Fonlupt, "High Performance Genetic Programming on GPU", *Proceedings of the workshop on Bio-inspired algorithms for distributed systems*, pp. 85-94, 2009.
- [28] M. Wong and T. Wong, "Implementation of Parallel Genetic Algorithms on Graphics Processing Units", *Intelligent and Evolutionary Systems, Studies in Computational Intelligence*, Vol. 187, pp. 197-216, 2009.
- [29] M. Yoshimi, Y. Kurano, M. Miki and T. Hiroyasu, "An Implementation and Evaluation of CUDA-based GPGPU Framework by Genetic Algorithms", *International Journal of Computer Science and Network Security*, Vol. 10, No. 12, pp. 29-37, 2010.
- [30] J. Li, X. Wang, R. He and Z. Chi, "An Efficient Fine-grained Parallel Genetic Algorithm Based on GPU-Accelerated", *IFIP International Conference on Network and Parallel Computing Workshops*, pp. 855-862, 2007.
- [31] S. Debattisti, N. Marlat, L. Mussi and S. Cagnoni, "Implementation of a Simple Genetic Algorithm within the CUDA Architecture", *Proceedings of the GECCO workshop on Computational Intelligence on Consumer Games and Graphic Hardware*, 2009.
- [32] S. Tsutsui and N. Fujimoto, "Solving Quadratic Assignment Problems by Genetic Algorithms with GPU Computation: A Case Study", *Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pp. 2523-2530, 2009.
- [33] A. Munawar, M. Wahib, M. Munetomo and K. Akama, "Advanced Genetic Algorithm to Solve MINLP Problems over GPU", *IEEE Congress on Evolutionary Computation*, pp. 318-325, 2011.
- [34] L. Zaloudek, L. Sekanina and V. Simek, "GPU Accelerators for Evolvable Cellular Automata", *Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*, pp. 533-537, 2009.
- [35] Dong-Hu Nie, Kyu-Phil Han and Heng-Suk Lee, "Stereo Matching Algorithm using Population-based Incremental Learning on GPU", *Proceedings of International Workshop on Intelligent Systems and Applications*, pp. 1-4, 2009.
- [36] M. Wong and G. Cui, "Data Mining using Parallel Multi-objective Evolutionary Algorithms on Graphics Hardware", *IEEE Congress on Evolutionary Computation*, pp. 1-8, 2010.
- [37] S. Kannan and R. Ganji, "Porting Autodock to CUDA", *IEEE Congress on Evolutionary Computation*, pp. 1-8, 2010.
- [38] C. Salwala, V. Kotrajaras and P. Horkaew, "Improving Performance for Emergent Environments Parameter Tuning and Simulation in Games Using GPU", *Proceedings of 3rd IEEE International Conference on Computer Science and Information Technology*, Vol. 2, pp. 37-41, 2010.
- [39] Y. Ke, Y. Li and D. Li, "Image Matching using Genetic Algorithm on GPU", *Proceedings of International Conference on Control, Automation and Systems Engineering*, pp. 1-4, 2011.
- [40] Y. Sato, N. Hasegawa and M. Sato, "GPU Acceleration for Sudoku Solution with Genetic Operations", *IEEE Congress on Evolutionary Computation*, pp. 296-303, 2011.
- [41] R. Bogdanski, L. Peter, T. Becker and X. Yao, "Improving Scheduling Techniques in Heterogeneous Systems with

- Dynamic, On-Line Optimisations”, *Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 496-501, 2011.
- [42] S. Yoo, M. Harman and S. Ur, “Highly scalable multi objective test suite minimization using graphics cards”, *Proceedings of the Third International conference on Search based software engineering*, pp. 219-236, 2011.
- [43] J. Lin, T. Igarashi, J. Mitani, M. Liao and Y. He, “A Sketching Interface for Sitting Pose Design in the Virtual Environment”, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 18, No. 11, pp. 1979-1991, 2012.
- [44] K. Wang and Z. Shen, “A GPU-Based Parallel Genetic Algorithm for Generating Daily Activity Plans”, *IEEE Transactions on Intelligent Transaction Systems*, Vol. 13, No. 3, pp. 1474-1480, 2012.
- [45] J. F. Zhao, W. H. Zeng, G. M. Li and M. Liu, “Simple Parallel Genetic Algorithm Using Cloud Computing”, *Applied Mechanics and Materials*, Vol. 121–126, pp. 4151-4155, 2011.
- [46] Z. Zheng, R. Wang, H. Zhong and X. Zhang, “An Approach for Cloud Resource Scheduling Based on Parallel Genetic Algorithm”, *Proceedings of 3rd International Conference on Computer Research and Development*, Vol. 2, pp. 444–447, 2011.
- [47] Y. Kessaci, N. Melab and E. Talbi, “A Pareto-based GA for Scheduling HPC Applications on Distributed Cloud Infrastructures”, *Proceedings of International conference on High Performance Computing and Simulation*, pp. 456 – 462, 2011.
- [48] E. Maria Mocanu, M. Florea, M.I. Andreica and N. Țăpuș, “Cloud Computing – Task Scheduling based on Genetic Algorithms”, *Proceedings of IEEE International Systems Conference*, pp. 1-6, 2012.
- [49] S. Kaur and A. Verma, “An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment”, *International Journal of Information Technology and Computer Science*, Vol. 4, No. 10, pp. 74-79, 2012.
- [50] Y. Chang-tian and Y. Jiong, “Energy-aware Genetic Algorithms for Task Scheduling in Cloud Computing”, *Proceedings of Seventh ChinaGrid Annual Conference*, pp. 43-48, 2012.
- [51] K. Jindarak and P. Uthayopas, “Performance Improvement of Cloud Storage using a Genetic Algorithm based Placement”, *Proceedings of Eighth International Joint Conference on Computer Science and Software Engineering*, pp. 54-57, 2011.
- [52] Z. Xiong, Z. Zhang, H. Kong and D. Zou, “Genetic Algorithm-based Power Management in Cloud Platform”, *Proceedings of International Conference on Internet Technology and Applications*, pp. 1-4, 2011.
- [53] M. Yusoh, Z. Izzah and T. Maolin, “Clustering composite SaaS components in cloud computing using a grouping genetic algorithm”, *IEEE Congress on Evolutionary Computation*, pp. 1-8, 2012.
- [54] A. Munawar, M. Wahib, M. Munetomo and K. Akama, “A Survey: Genetic Algorithms and the Fast Evolving World of Parallel Computing”, *Proceedings of 10th IEEE International Conference on High Performance Computing and Communications*, pp. 897-902, 2008.
- [55] L. Zheng, Y. Lu, M. Ding, Y. Shen, M. Guo and S. Guo, “Architecture-based Performance Evaluation of Genetic Algorithms on Multi/Many-core Systems”, *Proceedings of 14th IEEE International Conference on Computational Science and Engineering*, pp. 321-334, 2011.
- [56] V. Cristea, “Conception and design of parallel and distributed applications”, *Proceedings of the Romanian academy, Series A*, Vol. 5, No. 1, pp. 1-8, 2004.
- [57] W. Zhuang, F. Hanyang, S. Zhaoxuan and D. Rajesh, “HPC application in DSM/VDSM IC chip planning”, *Proceedings of The Fourth International Conference on High Performance Computing in the Asia-Pacific Region*, Vol. 2, pp. 1125 – 1131, 2000.
- [58] D. Dunlop, S. Varrette and P. Bouvry, “On the Use of a Genetic Algorithm in High Performance Computer Benchmark Tuning”, *Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, pp. 105-113, 2008.
- [59] V. S. Gordon and D. Whitley, “Serial and parallel genetic algorithms as function optimizers”, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 177–183, 1993.
- [60] W. E. Hart, S. Baden, R.K. Belew and S. Kohn, “Analysis of the numerical effects of parallelism on a parallel genetic algorithm”, *Proceedings of the 10th International Parallel Processing Symposium*, pp. 606-612, 1996.
- [61] E. Alba and J. M. Troya, “Influence of the Migration Policy in Parallel Distributed GAs with Structured and Panmictic Populations”, *Applied Intelligence*, Vol. 12, No. 3, pp. 163-181, 2000.
- [62] E. Cant’u-Paz, “Migration Policies and Takeover Times in Parallel Genetic Algorithms”, *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 775, 1999.
- [63] E. Cant’u-Paz, “Migration policies, selection pressure, and parallel evolutionary algorithms”, *Journal of Heuristics*, Vol. 7, No. 4, pp. 311-334, 2001.
- [64] E. Cant’u-Paz, “Topologies, migration rates, and multi-population parallel genetic algorithms”, *Proceedings of the Genetic and Evolutionary Computation Conference*, Vol. 1, pp. 91-98, 1999.
- [65] T. G. Crainic and M. Toulouse, “Parallel Meta-Heuristics”, *Handbook of Metaheuristics, International Series in Operations Research & Management Science*, Vol. 146, pp. 497-541, 2010.
- [66] V. Cung, L. S. Martins, C. C. Ribeiro and C. Roucairol, “Strategies for the Parallel Implementation of Metaheuristics”, *Essays and Surveys in Metaheuristics*, Vol. 15, pp. 263-308, 2002.
- [67] R. K. Ursem, “Multinational Gas: Multimodal optimization techniques in dynamic Environments”, *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 19-26, 2000.
- [68] M. Sefrioui and J. Periaux, “A Hierarchical Genetic Algorithm Using Multiple Models for Optimization”, *Proceedings of Parallel Problem Solving from Nature VI*, Vol. 1917, pp. 879-888, 2000.
- [69] C. Fernandes, R. Tavares and A.C. Rosa, “niGAVaPS—outbreeding in genetic algorithms”, *Proceedings of ACM symposium on Applied Computing*, Vol. 1, pp. 477–482, 2000.

- [70] C. Fernandes and A. C. Rosa, "A study on non-random mating and varying population size in genetic algorithms using a royal road function", *Proceedings of the Congress on Evolutionary Computation*, Vol. 1, pp. 60–66, 2001.
- [71] W. Rivera, "Scalable Parallel Genetic Algorithms", *Artificial Intelligence Review*, Vol. 16, No. 2, pp. 153-168, 2001.
- [72] E. Alba and J. M. Troya, "Analyzing synchronous and asynchronous parallel distributed genetic algorithms", *Future Generation Computer System – Special issue on bio-impaired solutions to parallel processing problems*, Vol. 17, No. 4, pp. 451-465, 2001.
- [73] E. Alba and J. M. Troya, "Improving flexibility and efficiency by adding parallelism to genetic algorithms", *Statistics and Computing*, Vol. 12, No. 2, pp. 91-114, 2002.
- [74] M. Giacobini, E. Alba and M. Tomassini, "Selection Intensity in Asynchronous Cellular Evolutionary Algorithms", *Proceedings of the Genetic and Evolutionary Computation Conference*, Vol. 2723, pp. 955-966, 2003.
- [75] N. Xiao and M. Amstron, "A Specialized Island Model and Its Application in Multiobjective Optimization", *Proceedings of the Genetic and Evolutionary Computation Conference*, Vol. 2723, pp. 1530-1540, 2003.
- [76] S. Yang, "PGDA: The primal dual genetic algorithm", *Design and Application of Hybrid Intelligent Systems*, pp. 214-223, 2003.
- [77] G. Ochoa, C. Madler-Kron, R. Rodriguez and K. Jaffe, "Assortative mating in genetic algorithms for dynamic problems", *Proceedings of the 3rd European conference on Evolutionary Computing*, pp. 617–622, 2005.
- [78] G. Ochoa and K. Jaffe, "Assortative mating drastically alters the magnitude of error thresholds", *Proceedings of 9th International conference on Parallel Problem Solving from Nature*, Vol. 4193, pp. 890–899, 2006.
- [79] Peter. J. Russel, "*Genetics*", 5th Edition, Benjamin-Cummings Publications, 1998.
- [80] P. M. Todd and G. F. Miller, "On the sympatric origin of species: mercurial mating in the quicksilver model", *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 547–554, 1991.
- [81] Xinjie Yu and Mitsuo Gen, "*Introduction to Evolutionary Algorithms*", Series: Decision Engineering, Springer-Verlag, pp.165 - 191, 2010.
- [82] T. Park and K. R. Ryu, "A dual population genetic algorithm with evolving diversity", *IEEE Congress on Evolutionary Computation*, pp. 3516 – 3522, 2007.
- [83] T. Park, R. Choe and K. R. Ryu, "Adjusting population distance for the dual-population genetic algorithm", *Proceedings of the 20th Australian joint Conference on Advances in Artificial Intelligence*, pp. 171–180, 2007.
- [84] T. Park, R. Choe and K. R. Ryu, "Dual-population Genetic Algorithm for Non-stationary Optimization", *Proceedings of Genetic and Evolutionary Computation Conference*, pp. 1025 - 1032, 2008.
- [85] Li. Junhua, Li. Ming and Li. Junhua, "Genetic Algorithm with Dual Species", *Proceedings of the IEEE International Conference on Automation and Logistics*, pp. 2572 - 2575, 2008.
- [86] Carlos Fernandes and Agostinho C. Rosa, "Self-Adjusting the Intensity of Assortative Mating in Genetic Algorithms", *Journal of Soft Computing*, Vol. 12, No. 10, pp. 955 - 979, 2008.
- [87] T. Park and K. R. Ryu, "A Dual-Population Genetic Algorithm for Adaptive diversity control", *IEEE Transactions on Evolutionary Computation*, Vol. 14, No. 6, pp. 865–884, 2010.
- [88] A. J. Umbarkar and M. S. Joshi, "Serial DPGA versus Parallel Multithreaded DPGA: Threading Aspects", *Proceedings of the International Conference on Soft Computing for Problem Solving, Advances in Intelligent and Soft Computing*, Vol. 130, pp. 39 – 50, 2011.
- [89] A. J. Umbarkar and M. S. Joshi, "Dual Population Genetic Algorithm (GA) versus OpenMP GA for Multimodal Function Optimization", *International Journal of Computer Applications*, Vol. 64, No. 19, pp. 29 - 36, 2013.
- [90] Xin Yao, Yong Liu and Guangming Lin, "Evolutionary programming made faster", *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, pp. 82 – 102, 1999.
- [91] Chang-Yong Lee and Xin Yao, "Evolutionary programming using mutations based on the Lévy probability distribution", *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 1, pp. 1 – 13, 2004.
- [92] Thomas Baeck, D.B. Fogel and Z. Michalewicz, "*Handbook of Evolutionary Computation*", 1st Edition, Taylor & Francis, 1997.
- [93] Georges Harick, "Finding multimodal solutions using restricted tournament selection", *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 24 – 31, 1995.
- [94] M. Lozano, F. Herrera, N. Krasnogor and D. Molina, "Real-coded memetic algorithms with crossover hill-climbing", *Evolutionary Computation*, Vol. 12, No. 3, pp. 273 – 302, 2004.
- [95] J. J. Liang, A. K. Qin., P. N. Suganthan and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, pp. 281 – 295, 2006.
- [96] D. Molina, "Adaptive local search parameters for real coded memetic algorithms", *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, pp. 888 – 895, 2005.
- [97] Z. Yang, J. He and X. Yao, "Making a difference to differential evolution", *Advances in Metaheuristics for Hard Optimization*, pp. 415-432, 2007.
- [98] N. Hansen, S.D. Muller and P. Koumoutsakos, "Reducing the time complexity of the de-randomized evolution strategy with covariance matrix adaptation (CMA-ES)", *Evolutionary Computation*, Vol. 11, No. 1, pp. 1 – 18, 2003.
- [99] Mohammad Shafiul Alama, Mohammad Monirul Islama, Xin Yaob and Kazuyuki Murase, "Diversity Guided Evolutionary Programming: A novel approach for continuous optimization", *Applied Soft Computing*, Vol. 12, No. 6, pp. 1693 - 1707, 2012.