SOLVING NON-LINEAR OPTIMIZATION PROBLEMS USING PARALLEL GENETIC ALGORITHM

Ali. T. Al-Oqaily
Information Technology Faculty
Ajloun National University (ANU)
Ajloun, Jordan
Oqili83@yahoo.com

Ghazi Shakah
Information Technology Faculty
Ajloun National University (ANU)
Ajloun Jordan
warsaw2000@yahoo.com

Abstract: In last few decades, a growing interest in the domain of evolutionary algorithms has been observed due to its performance in discover the optimal solutions for the complex problems. The Genetic Algorithm (GA) is one of the most used evolutionary algorithms that attract the researchers' interests in many fields such as the physics and mathematics. GA can provide optimal solution for the problems of complex environments i.e. polarize environments. Like other evolutionary algorithms, the execution time of GA is relatively long, whereby the optimization processes could consume many hours. This study aims to improve the optimization accuracy and reduce the execution time of traditional GA. The parallel GA is proposed to conduct the optimization processes through distributed machines or processors (network of processes). The original complex problem is divided into sub-small areas, and each sub area is optimized by sequential GA that applied in each processor in the network. The final solution is collected from all processors in the network in order to decide the best final solution. To evaluate the proposed parallel genetic algorithm, two complex problems are optimized; De Jong's and Ackeleys path functions. For testing purpose, a network of four processors is constructed using MATLAB toolbox distributed system toolbox. The significances results show that proposed parallel GA give better results with less execution time than the traditional or sequential GA. The contribution of this study is the segmentation of large and complex area into small area and optimizes the solutions of the small areas using network of processors. This approach simplifies the optimized problems, reduce the execution time, and give better chances to discover the optimal solutions.

Keywords: GA, Parallel, Non-linear problems, optimization, distribution.

I. INTRODUCTION

The idea of GA is initiated by John Holland along with his students in 1960', and their main goal was to simulate some phenomenon that occur in the nature to adapt it as a computer system for some problems like searching taxonomy and big data management [1]. Later on, the based architecture of GA as evolutionary algorithm was formulated by Holland' team. The GA is applied for optimization purposes such as discover best records, low points, and high points. Also, GA can be utilized in many applications such as searching for sorted data, and search for free paths of network. The main benefit of the GA is the ability to discover the optimal solution from large dataset or population. Hence, GA can be applied effectively in the complex optimization problems of big data. Another advantage of the GA is the simplicity of optimization processes comparing with other evolutionary algorithms [2].

Basically, there are five main steps of GA algorithms, which are [2]: i) initialization, in this process the main area/population of the problem is represented as set of chromosomes, and each all chromosomes are fitted to same size; ii) selecting, in this process the set of initiated chromosomes of fitness value (best values) are selected, and each two chromosome form the selected sample are considered as parents that connected together; iii) reproduction, in this process, the crossover between the parents will conducted in order to assure the covering of most best solutions; iv) replacement, after crossover process, the value of each child chromosome will be mutated to cover the possible surrounded values in the area; v) termination, all previous processes will be repeated until reach the optimal solution or achieve the termination condition (i.e. 20 loops).

Simply put, GA initiates the nearest value to the optimal solution before start discovering better solutions through the crossovers between the initial solutions. The surrounded better values of the best solutions would b discovered using mutation processes. The iteration of the processes will be loop for to discover better solution. Once the processes are terminated the optimal solution will be retrieved. For example, in low point optimization, the fitness solution is zero. Thus, GA will look in the population for zero value or the nearest value to zero.

Although, GA is effective algorithm to optimize the best solutions of complex problems, the execution time of this algorithm stills an issue [3]. GA takes long execution time comparing with other optimization algorithms due to sequential processes of GA that conducted on large population. Therefore, the Parallel Genetic Algorithm (PGA) could be effective approach to address the issue of execution time of GA. On the other hand, PGA could provide more focused or accurate solutions more than the sequential GA through divide the main complex problem as sub-smaller problems [4].

The PGA is based on distributed computing, which a network of processors involve the solutions optimization using GA [1]. The large population of the main problem is divided as smaller population called islands. Each processor in the network is responsible for conducting GA on one or more islands, and after every round of GA, the processors swap specific percentage of local generated population in the island with other global population in other islands. Hence the execution time would reduced through processing collaboration between many processors, and the optimization results would be enhanced through focused and structured searching in the sub-population.

This study aims to utilize the PGA in complex non-linear problems in order to evaluate the execution time and results accuracy of PGA comparing with traditional GA. It is expected to record significances enhancement in the optimization time and results using the PGA. The next section presents some related works in the domain of GA and PGA. The third section explains the research methods. The fourth section discusses the experimental results. Lastly, the fifth section provides the conclusion and some future works.

II. RELATED WORKS

The PGA was utilized by many researchers in various domains. The study of Sahingoz [5] focused on utilize the PGA to optimize the most useful air path for the planes. The optimal solution is to find the path of low constraints such as low traffic, and avoid the high points in the path (i.e. mountains). The path area is divided into smaller connected paths, and each path represents a sub-population of the overall area. The PGA records accuracy of about 91% nearest to free constraints path (optimal solution).

Roberge et al. [6] focused on the GA implementation in the electromagnetic inverting from high voltage (DC) sources to low voltages (AC) destination. The inverting processes would be accomplished through multilevel inverters with using many DCs to handle the minimum required voltage for several AC applications. Thus, the optimal inverting angles should be computed based on the AC applications in order to decide the inverting path via DCs invertors. Roberge et al. [6] argued that the sequential computation of GA cannot easily used to calculate the optimal switching angels in real time, which delay the multilevel inverting processes of electromagnetic waves. For this reason, the PGA is proposed to divide the various DC sources into many islands to find the optimal switching angels in faster time. The significant results show that PGA approach is 469× faster than the sequential GA.

Nandy et al. [3] mentioned that the placement of sensors devices in large area is a challenge due to difficulty of deiced useful placement structure to cover the whole area using the lowest number of sensors. in addition to areas spaces, the nature of the area zones is additional parameter that increase the difficulty of sensors placement structure. To address these challenges, Nandy et al. [3] adopted the PGA approach to optimize the low number of sensors that can be placed to cover the large areas of various zones nature. The original area is divided into smaller-areas, and the optimal placement structure of each sub-area is optimized using sequential GA based on the area space and nature. The integration between the local computations is conducted to mutate the placement structure of the global area. Finally, the optimized placement structure is compared with the manual structure of sensors placement. The results indicated that the PGA approach provided effective placement structure with less number of used sensors comparing with the manual placement strategies.

Lei et al. [7] adopted the PGA to identify the aquifer zones in large area based on the zones structure (minimum points in the area) and the hydrogeological parameters. The original area is divided into multi-smaller area, before apply the sequential GA to guess the optimal aquifer zone (minimum points in the area). The optimized aquifer zone is confirmed through assure that the levels of surrounded zones are increase continually decreases in structured pattern to arrive the minimum point (aquifer zone). The proposed approach is tested and compared with real aquifer cases in Los Alamos National Laboratory. The experimental results show that the PGA is effective to find the aquifer zones in large area.

Moumen et al. [4] adopted the PGA approach to enhance the solutions of hard nonlinear problems such as Traveling Salesman Problems (TSP). Problems such as TSP is controlled by many parameters and conditions such as find the shortest path, not visit any previous visited node, and visit all nodes. TSP involves the implementations of mathematic searching processes in computer systems. PGA is adopted to address the TSP through divide the searching paths into sub-paths, and in each path, the shortest travel path for all nodes is optimized using the sequential GA. Therefore, the optimal solutions of local paths are integrated to represent the optimal solution of the global area paths. The result of Moumen et al. [4] study shows that the PGA is effective to find the optimal solutions of complex nonlinear problems such as TSP.

As well as the above studies, PGA approach was used to find the optimal clustering design in for big data network [8], find the optimal scheduling of team jobs [9], optimize the design of water distribution system in large areas, and find the possible parameters that cause the breast cancer [10, 11].

Although there many studies were conducted on the PGA implementations, there is lack in the conducted study on the contribution of PGA in speed up the execution time of the optimization processes. Most of the conducted studies were focused on the results effectiveness factor. Thus, this study will try to clarify the contribution of PGA based on two factors, the processing speed and the effectiveness of the optimized solutions. The next section presents the research methodology of this study.

III. RESEARCH METHODOLOGY

In this study, there are two complex functions were selected from GEATbx dataset to evaluate the PGA performance comparing with sequential GA. The first one is De Jong's function and the second one is Ackley's function (Figure 1). These functions represent complex nonlinear problem in many fields that focus on find the optimal points (minimum or maximum) in complex or polarized environment.

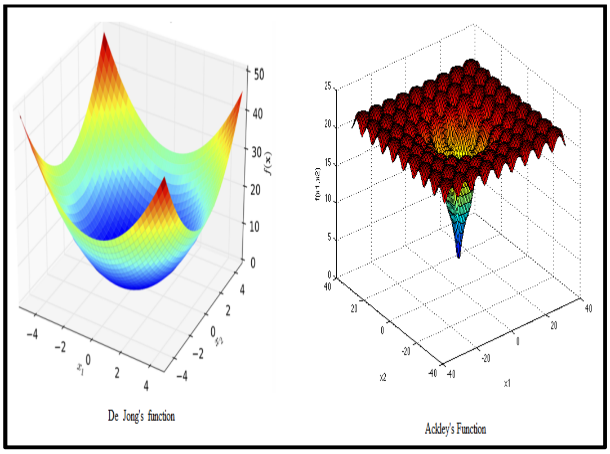

Figure 1: Study Problems

In this study, the De Jong's function is used to find the optimal solution nearest to low value (i.e. zero), and the following formula (1) represents the De Jong's function:

f(x)= sum(x(i)^2), i=1:n, 0<=x(i)<=5.12 (1)

On the other hand, the Ackley's Path function is used to find the optimal values nearest to mid-point of the function environment (i.e. zero between -40 to 40), and the following formula (2) represents the Ackley's Path function.

f(x)=-a·exp(-b·sqrt(1/n·sum(x(i)^2)))-exp(1/n·sum(cos(c·x(i))))+a+exp(1) a=20; b=0.2;c=2·pi; i=1:n; -32.768<=x(i)<=32.768 (2)

For effective experiment , the optimization processes of the two selected function is conducted using sequential and parallel GA in order to do a comparison between these two approach based on accuracy and speed factors. For this purpose, A MATLAB toolbox called distributed system is utilized. There are three utilization scenarios are conducted which are as the following:

- Sequential GA using 1 central processor of 1.GH speed, 512 RAM, and 2GB virtual Memory.
- PGA using 1 central processor and four virtual slaves' processors of the same specifications in the first scenario.
- PGA using 1 central processor and eight virtual slaves' processors based on the same specifications in the first scenario.
- PGA using real network of 1 master processor and eight virtual processors based on the same specification of the above scenarios. The connection is based on Microsoft server and client operating system of LAN transfer speed (100MBPS).

The GA and PGA configuration are set as 1000 initial population, 50 generation, roulette wheel selection, 0.8 amount of crossover, 0.2 amount of mutation, and possible to migrate the optimized fitness solution in case of PGA. The accuracy results and execution time of all scenarios are recorded and compared Excel Microsoft application.

IV. DISCUSSION OF THE RESULTS

Figure1 illustrates the average optimal results of De Jong's function based on the four scenarios of this study. It clearly that the four scenarios are produced instable solutions in the earlier generations, but in last generations, the real network of PGA is provide the best solutions near to optimal value (0.0002), followed by the 8 virtual slaves (0.0015), then four slave machine (0.0186), lastly the sequential GA (0.023). This reflects the effectiveness of PGA accuracy comparing with the sequential GA.

On the other hand, the execution time of the sequential GA is about 274 seconds comparing with 238 second of four slaves PGA, 182 seconds of eight slaves PGA, and 102 second of real PGA network. This indicates the significance time saving of PGA approach comparing with the sequential.

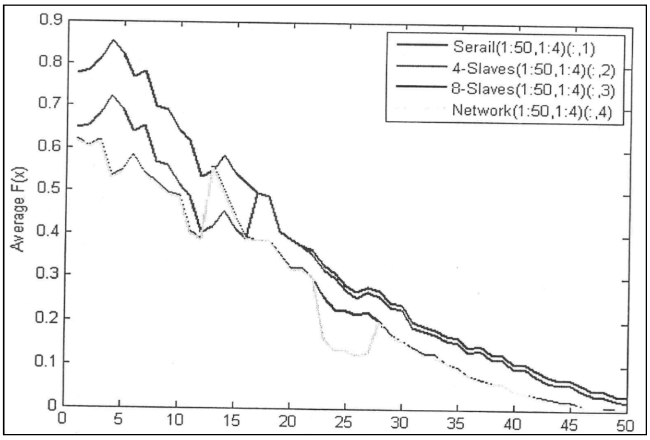

Fig . 2: Average results of De Jong's Function

Figure 3 illustrates the average optimal results of Ackley's path function based on the four scenarios of this study. It difficult to finds the optimal solution of the mid-point (zero) due to high level of Ackley's complexity. However, the real network of PGA records the best optimal solution (2. 98) followed by the 8 virtual slaves (2.992), then four slave machines (2.995), lastly the sequential GA (3.245). This indicates the effectiveness of PGA accuracy in high complex problems comparing with the sequential GA.

On the other hand, the execution time of the sequential GA is about 870 seconds comparing with 487 second of four slaves PGA, 376 seconds of eight slaves PGA, and 272 second of real PGA network. This indicates the significance time saving of PGA approach comparing with the sequential GA.

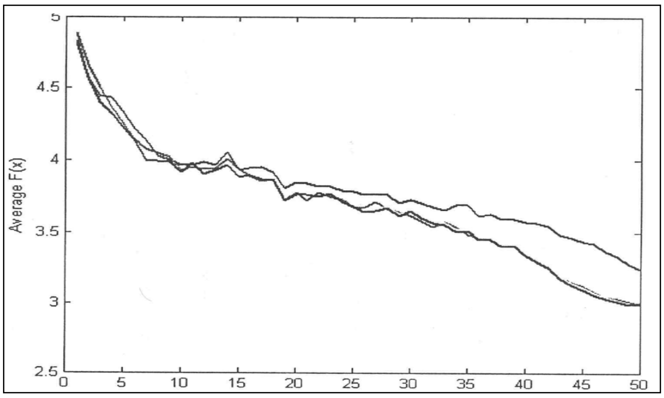

Fig. 3: Average results of Ackley's path function

Based on the experimental results, it can be noticed that the PGA approach is better than the sequential GA to handle the optimization of the complex problems either in execution time and results accuracy. The difference between accuracy and execution time of PGA and GA based on the Ackley's path function is more significance that the results of De Jong's function. This because the Ackley's path functions is more complex than the De Jong's function. Hence, it is recommended utilizing the PGA rather than GA in the complex problems, especially in the much complexity level of the problems.

V. CONCLUSION AND FUTURE WORKS

The GA one of the most algorithms for the optimization purposes due to clearness, effectiveness, and simple implementations of this algorithm. However, the accuracy and the execution time of GA in complex problems still and issue. The PGA approach is proposed by many researchers in various domains to enhance the optimization accuracy and reduce the execution time of sequential GA. This study conducts comparison experiment between GA and PGA approach to understand the significance conurbation of PGA in the optimization accuracy and execution time in complex problems. For this, purpose, two complex functions are selected; Ackley's path function, and De Jong's function. The significance results show that the PGA is more accurate and faster than the GA, especially when the complexity level of the problems is increase. In the future, higher complex problems of real environment (such as time scheduling of big projects) will be optimized using PGA approach.

VI. REFERENCES

[1] BE, R. M. (2003). A Generic Parallel Genetic Algorithm.

[2] Razali, N. M., & Geraghty, J. (2011, July). Genetic algorithm performance with different selection strategies in solving TSP. In *Proceedings of the world congress on engineering* (Vol. 2, pp. 1134-1139). Hong Kong: International Association of Engineers.

[3] Nandy, A., Chakraborty, D., & Shah, M. S. (2016). Optimal Sensors/Actuators Placement in Smart Structure Using Island Model Parallel Genetic Algorithm. International Journal of Computational Methods, 1840018.

[4] Moumen, Y., Abdoun, O., & Daanoun, A. (2017, November). Parallel approach for genetic algorithm to solve the Asymmetric Traveling Salesman Problems. In Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems (p. 24). ACM.

[5] Sahingoz, O. K. (2014). Generation of bezier curve-based flyable trajectories for multi-UAV systems with parallel genetic algorithm. Journal of Intelligent & Robotic Systems, 74(1-2), 499-511.

[6] Roberge, V., Tarbouchi, M., & Okou, F. (2014). Strategies to accelerate harmonic minimization in multilevel inverters using a parallel genetic algorithm on graphical processing unit. IEEE transactions on power electronics, 29(10), 5087-5090.

[7] Lei, H., Lu, Z., Vesselinov, V. V., & Ye, M. (2017, December). Identification of Arbitrary Zonation in Groundwater Parameters using the Level Set Method and a Parallel Genetic Algorithm. In AGU Fall Meeting Abstracts.

[8] Shayeghi, A., Götz, D., Davis, J. B. A., Schäfer, R., & Johnston, R. L. (2015). Pool-BCGA: a parallelised generation-free genetic algorithm for the ab initio global optimisation of nanoalloy clusters. Physical Chemistry Chemical Physics, 17(3), 2104-2112.

[9] Asadzadeh, L. (2015). A local search genetic algorithm for the job shop scheduling problem with intelligent agents. Computers & Industrial Engineering, 85, 376-383.

[10] Xu, H., Chen, T., Lv, J., & Guo, J. (2016). A combined parallel genetic algorithm and support vector machine model for breast cancer detection. Journal of Computational Methods in Sciences and Engineering, 16(4), 773-785.

[11] Bi, W., Dandy, G. C., & Maier, H. R. (2015). Improved genetic algorithm optimization of water distribution system design by incorporating domain knowledge. Environmental Modelling & Software, 69, 370-381