Parallel Agent Oriented Genetic Algorithm

Dr. Afaf B. Al-Kaddo \ University of Baghdad\ College of Education for Women

Computer Science Department\ **E-mail:** afafbad@yahoo.com

Abstract

Genetic Algorithms (GAs) can be used to solve difficult problems in many disciplines. The performance of GA depends on computer power. Parallel Genetic Algorithms (PGAs) are parallel implementations of GAs can provide better performance and scalability and can be implemented on networks of heterogeneous computers. Efficiency of GAs depends on crossover and mutation rates and it is difficult to adjust those parameters manually. This paper used multi agent techniques which combine existing GA and PGAs with distributed environment. The paper shows the efficiency of the parallel computation of the travelling salesman problem using the genetic approach on a multicomputer cluster. The master/slaves paradigm is applied. Performance has been made on the basis of MPI-based parallel program implementation. This algorithm is tested by the Traveling Salesman Problem (TSP). Result shows that, using of the parallel techniques can reduce the communication between different nods, therefore speed up the traditional genetic algorithm search process.

المستخلص

الخوارزميات الجينية (GAs) يمكن أن تستخدم في حل المشاكل الصعبة في العديد من التخصصات. أداء الخوارزمية الجينية يعتمد على قوة جهاز الكمبيوتر. الخوارزميات الجينية المتوازية (PGAs) هي تطبيقات موازية من الخوارزميات الجينية (GAs) التي يمكن أن توفر مكاسب كبيرة من حيث الأداء والتدرجية، ويمكن تنفيذها على شبكات من أجهزة الكمبيوتر غير المتجانسة. كفاءة الخوارزميات الجينية تعتمد على معدل التبادل ومعدل الطفرات وأنه من الصعب ضبط تلك المعاملات يدويا. هذه الورقة استخدمت تقنيات الوكيل المتعدد التي تجمع بين GA وPGAs مع مخططات بيئة موزعة. الورقة تبحث في كفاءة الحوسبة المتوازية لمشكلة البائع المتجول باستخدام النهج الوراثية على multicomputer. اجري التحليل الفني للتطبيق المتوازي بتطبيق نموذج السيد / المسود . تم تخمين الأداء على أساس تنفيذ البرنامج الموازي القائم على MPI. تم اختبار هذه الخوارزمية على مشكلة البائع المتجول (TSP). اظهرت النتيجة انه باستخدام تقنيات متوازية يمكن تقليل الاتصال بين الأجهزة المختلفة وبالتالي تسريع عملية البحث التقليدية للخوارزمية الجينية .

Key words: TSP, Genetic Algorithms, MPI, Software Agents, Parallel Genetic Algorithm, Modeling and Simulation

1. Introduction

GAs is expected as a promising method which can be applied to many areas of problems. Many GAs for automatically adjusting crossover and mutation rates have been proposed because it is difficult to adjust those parameters manually and efficiency of GAs depends on these parameters. [1, 2, 3 and 4]. An algorithm which combines GA techniques and PGA with distributed environment has been proposed. This algorithm can adjust crossover and mutation rates while finding solutions. The phases of GAs are reproduction, mutation, fitness evaluation and selection. *Reproduction* combines two or more parent individuals to obtain one or more offspring. *Mutation* is the process to produce a new version by changing genetic randomly and is applied to one individual . *Fitness evaluation* in this stage the quality of an individual is evaluated. *Selection* is used to decide which individuals should be used to produce new search points. Sequential GAs has many problems with some form of PGA:

- Some parallel form of GA is necessary because in some kind of problems, the population is very large and requires large space of memory to store each individual therefore using a single machine to run an application efficiently its impossible .

- Using parallel processing will reduce fitness evaluation time-consuming.

- PGAs less to trapped by low-quality subspaces because it can search in parallel different subspaces of the search space.

PGAs provide better performance than single population-based algorithms, because different populations evolve in different directions [2]. For this reason PGAs are not only an expansion of the GA sequential model, and because PGAs search the space of solutions differently they represent a new class of algorithms. PGAs allow theoretical analyses which are not harder than those for sequential GAs [4, 5]. A new kind of distributed PGA using MPI (Message Passing Interface) is used in this paper. Distributes environment offers a low cost, high performance approach to processing large amounts of data.

2. Parallel Genetic Algorithm (PGA)

PGAs are modifications made to the GAs to reduce time consuming and making them more efficient [6]. GA doesn't used all resources available in a multi processors computer because it is a serial algorithm, the PGA will improve this weakness by parallelizing the GA process. To parallelize a GA the process is partitioned, and split a long serial step into shorter sub-process. This paper used Master-Slave parallelization. Creating **n** sub-groups, and calculates all sub-groups simultaneously. The algorithm uses a single population and/or the genetic operators are performed in parallel. Each individual may compete and mate with any other because the selection and mating are done globally. The evaluation of the fitness function operation is parallelized, because no need to communicate during this stage. The master stores the population and the slaves evaluate the fitness and apply mutation. This algorithm must wait until all the subgroups are finished in order to move to the next step.

3. Multi Agent Oriented Genetic Algorithm [7]

It is algorithm consisting of GAs with distributed environment and GA. In AOGA multiple agents are used and each agent run a GA with low computation costs, and give different vectors to agents. Those agents find a solution by exchanging immigrants. This system is one of GA with distributed environment scheme.

The number of parameter vectors in AOGA is equal to the number of agents can be obtained with one GA generation; therefore, computation costs of parameter can be reduced. In AOGA, a parameter vector (number of individuals, mutation rate, crossover rate, linear scaling coefficient) is given to each agent. Each agent finds a solution using the algorithm in Fig.1. AOGA finds a solution using multiple agents, and evaluates exploration efficiency of those agents. AOGA's algorithm is shown in Fig.2. The division of an exploration stage into several units called *era* so that different agents find solutions in multiple eras [8].

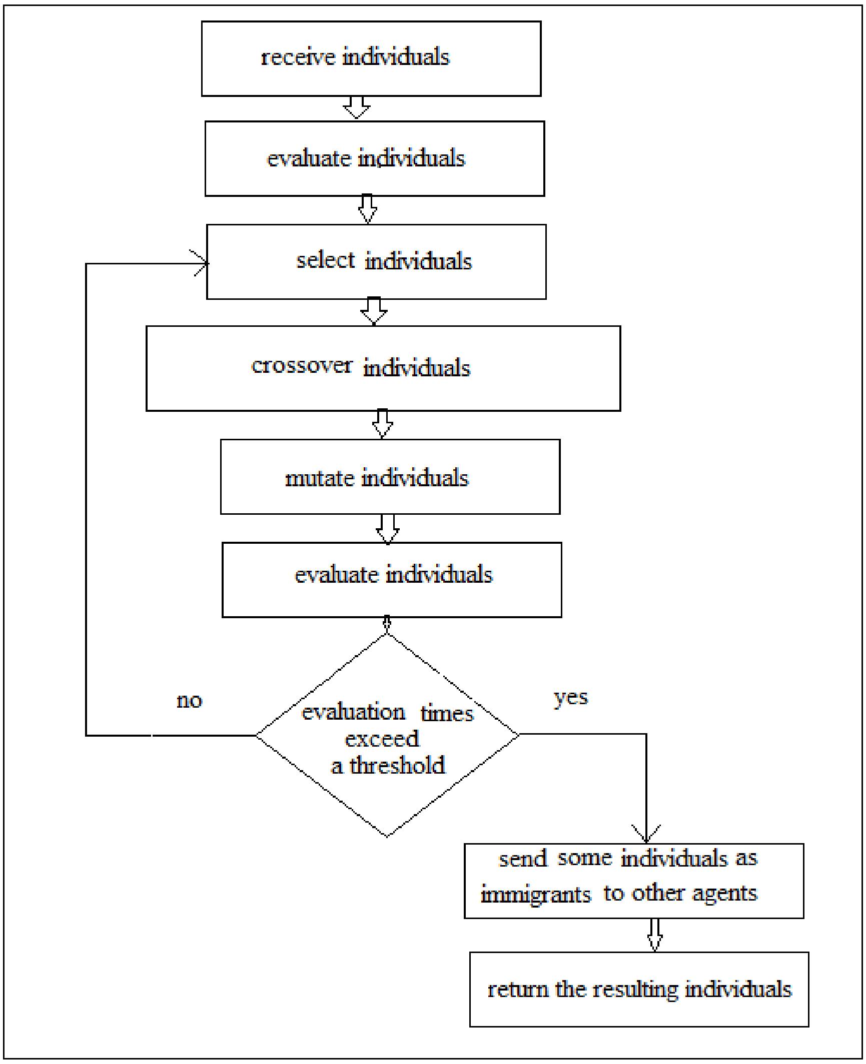

Figure 1: AOGA algorithm in each agent

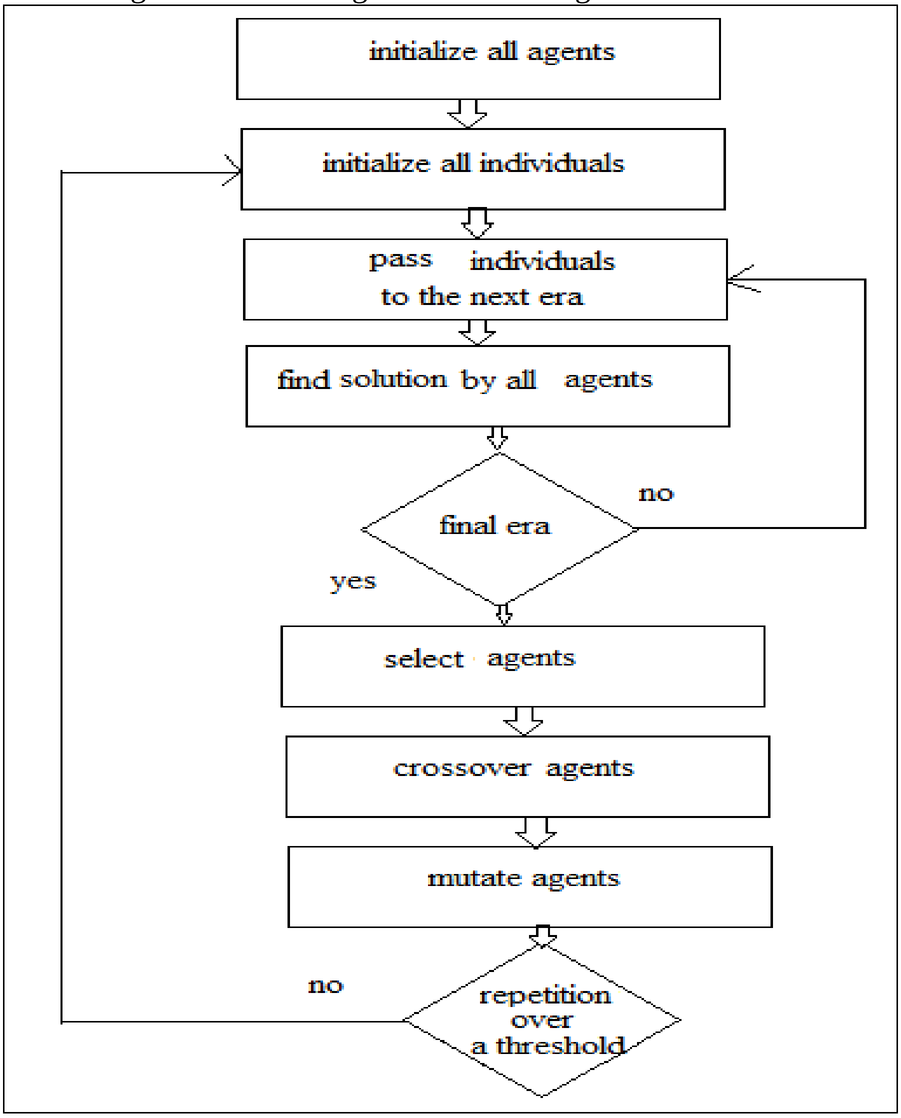

Figure 2: AOGA algorithm as the whole system

The parameters (evaluation times per each generation, the number of eras per each generation, the number of agents per each era) are used in addition to the parameters used in the GA. The parameter vector (mutation rate, crossover rates, linear scheduling coefficient, number of evaluations per each generation within each agent) for each agent is given.

4. Parallel Genetic Algorithm Based on MPI Environment

MPI (Message Passing Interface) is a user interface to write parallel programs base on message passing. MPI can be installed in the Windows NT/2000 platform, the MPI library can be called directly without any modification in C, C++ and Fortran and other programming languages. In master-slave MPI program: the main process its task is to find the best individual and send it to every other sub-processes. The sub-processes performed GA tasks and find the best individual, passed them to the master. Figure 3 explains the implementation process of the master slave PGA.

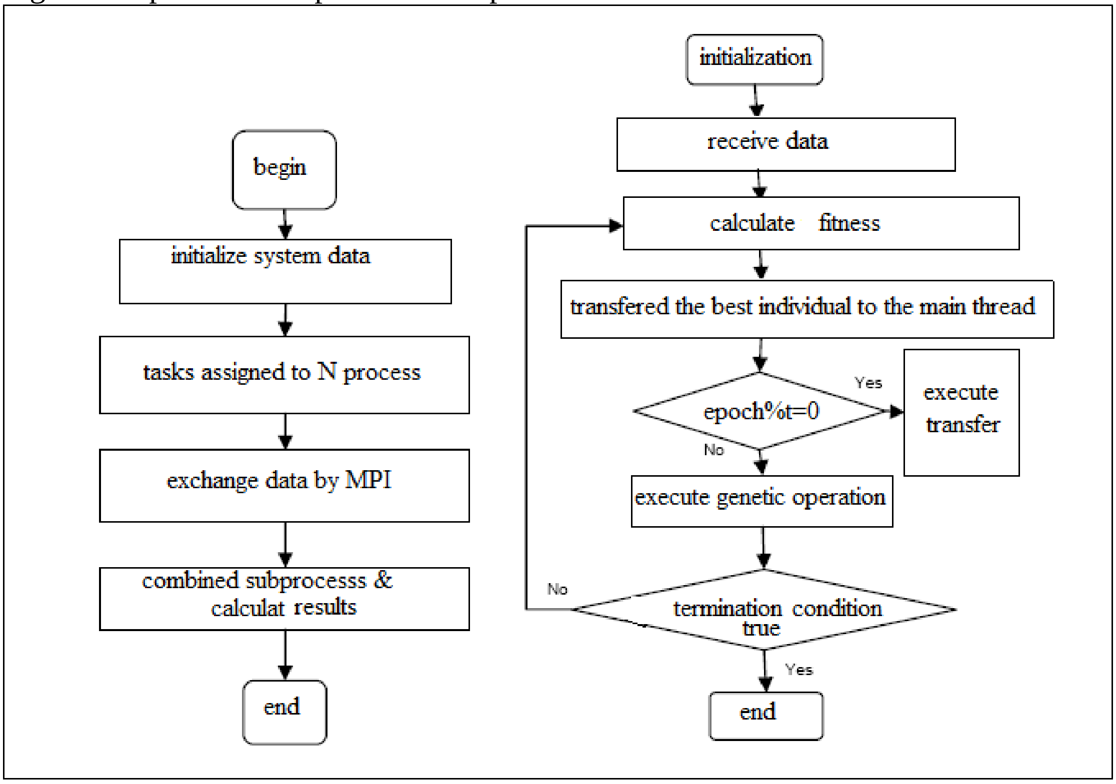

Figure 3: Implementation process of the master slave PGA

5. Parallel Environment's Structures Based on MPICH

MPI is a messaging standard for the development of distributed storage parallel computer [7, 9]. The purpose of a platform-independent message transfer library standards provided by MPI is to achieve the portability of the program, and it supports PC, workstations and almost all parallel machines. The programs written by MPI can be used by all operating system platform without any modification, the MPI environment initialization and end process as follows: before calling the MPI routines, each process should be executed MPI_INIT, call the MPI_COMM_SIZE to get the size of the default group, then call the MPI_COMM_SIZE to get the logical number of the invoking process in the default group. After completion, every process can send or receive the messages of other nodes according to the necessary needs. The MPI_SEND and MPI_RECV are used for this purpose. Finally, called MPI_FINALIZE when no need to call any MPI routines.

6. Experimental results

Traveling Salesman Problem (TSP) carried out to measure the accuracy and efficiency of the algorithm. The comparison with the GA and PGA show that the PGA improved the convergence speed of the GA and reduced the given precision time-consuming as shown in figure 4.

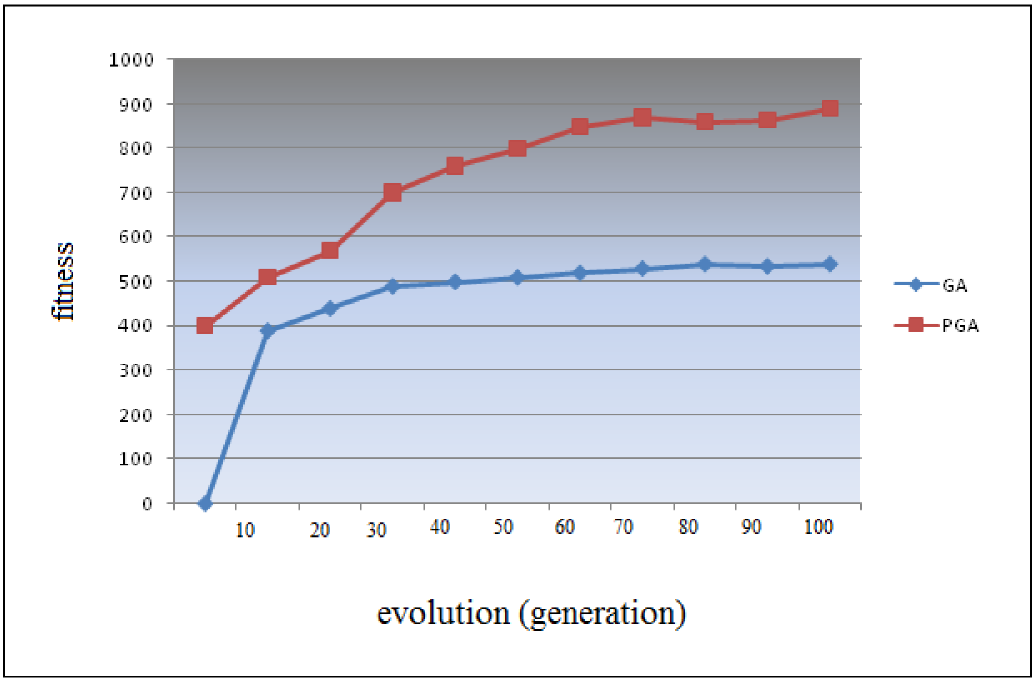

Figure 4: TSP individual fitness

7. Conclusion

AOGA based on multi agent merges the traditional GA techniques and PGA with distributed scheme. The algorithm is applied to solving the Travelling Salesman problem. The experimental results had shown that: using MPI as a parallel programming environment can reduce the increase access conflicts when the number of processor is large, thus speeding up the convergence rate of the parallel genetic algorithm.

References

[1] T. Black, "*Self-adaptation in genetic algorithms*", In F.J.Varela, P. B., editor, Proceedings of 1st European Conference on Artificial Life, pp. 263–271, 1992.

[2] F. Espinoza, B. S. Minsker, and D. Goldberg, "*A Self Adaptive Hybrid Genetic Algorithm*", Proceedings of the Genetic and Evolutionary Computation Conference, ,San Francisco, Morgan Kaufmann Publishers, 2001.

[3] E. Kee, S. Airey and W. Cye, "*An Adaptive Genetic Algorithm*", Proceedings of the Genetic and Evolutionary Computation Conference, pp 391–397, 2001.

[4] T. Krink and R. K. Ursem, "*Parameter Control Using the Agent Based Patchwork Model*", Proceedings of the Congress on Evolutionary Computation, pp. 77–83, 2000.

[5] M. Miki, K. Hiroyasu, M. Kaneko and I. Hatanaka, "*A Parallel Genetic Algorithm with Distributed Environment Scheme*", IEEE Proceedings of Systems, Man and Cybernetics Conference SMC'99, pp. 695–700, 1999.

[6] Mariusz, Nowostawski. Ricardo, Poli. Parallel Genetic Algorithm Taxonomy, KES’99, May 1999.

[7] Jing-Jun Zhang[1], Wen-Juan Liu[2], Guang-Yuan Liu[2], "*Parallel Genetic Algorithm Based on the MPI Environment*",[1]Department of Scientific Research, Hebei University of Engineering, Handan, China, [2]School of Information and Electrical Engineering, Hebei University of Engineering, Handan, China, TELKOMNIKA, Vol.10, No.7, November 2012, pp. 1708~1715.

[8] Yoshihiro Murata, Naoki Shibata, Keiichi Yasumoto and Minoru Ito, "*Agent Oriented Self Adaptive Genetic Algorithm*", Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5, Takayama, Ikoma, Nara 630-0192, Japan, 2003.

[9] Ben-jun Guo,Peng Wang. *"Cloud computing model based on MPI ",* Computer Engineering, 2009, 35(24), 84–87.