Review on Parallel Evolutionary Computing and Introduce Three General Framework to Parallelize All EC Algorithms

Amin Majd
Aras International Campus
Tabriz University
Tabriz, Iran
AminMajd89@ms.tabrizu.ac.ir

Shahriar Lotfi
Computer science department
Tabriz University
Tabriz, Iran
Shahriar_Lotfi@tabrizu.ac.ir

Golnaz Sahebi
Aras International Campus
Tabriz University
Tabriz, Iran
Sahebi-Golnnaz90@ms.tabrizu.ac.ir

Abstract— **Optimization and solving NP-hard problem are very important and Evolutionary Computing methods are useful and popular. There are different types of EC methods that most of them are serial and some other one has parallel implementation. In first step we want review some parallel implementation of EC methods and in second step we introduce three general frame work to parallelize all serial EC methods.**

Keywords-component; parallel Genetic Algorithms; parallel PSO; Parallel Ant Optimization Colony; Parallel Bee Colony Optimization; Parallel Memetic Algorithm.

I. INTRODUCTION

Optimization algorithms can be classified in heuristic and metaheuristic. In the class of heuristic algorithms, there are construction and improvement algorithms. Meta heuristic algorithms may manage a chain or flow of executions of classical heuristics, e.g. Tabu Search, Simulated Annealing, Genetic and Memetic Algorithms.

Computability is a big problem for researchers and some problems same NP-Hard problems aren't a good solution that find best answer in limited time. There are different methods to solve them but EC are the best and more popular than them. There are different type of EC that are useful for different problems, for example Genetic algorithms are an old method for discrete problems and use some regular behavior in humans body and use some characteristics of them and help researchers that optimize their solutions. Other one is PSO that mimic behavior of birds when migrate to other place.

EC methods are successful to solve different problems but there are some weak points that are important reasons of bad results. For example in some problems that have a big search space is impossible that my algorithms converge to local optimum results and we can improve results only with increasing of initial population. Other problems are speeds of algorithms and sometimes find answers after a long time. Parallel algorithms can help me to improved quality and times of results.

In last year's researchers use some parallel EC methods to obtain good results, For example Parallel Genetic, Parallel PSO, Parallel ABC, Parallel BEE and Parallel Memetic are most popular than other methods. Each one implemented with different technique and different equipment and hardware. For example Parallel Genetic Algorithms implements in four ways, Master-Slave Genetic Algorithms, Corse Grain Genetic Algorithms (Multi-populations Genetic Algorithms), Fine Grain Genetic Algorithms and Hybrid Genetic Algorithms that we will discuss about them in next chapter.

There are some Parallel methods but there are some EC methods that haven't parallel implementation, for example ICA is a new and efficient that is useful for continuous problems. In this Article we want to create a parallel implementation of ICA and compare with serial method.

II. PERIVIOS WORKS

In this chapter we will review some parallel EC methods and try to find some common characteristic between them and use some of them in our implementation.

A. Parallel Genetic Algorithms

Genetic algorithms are population base methods that use some regular behavior of human body; each GAs has an initial population and does some operations frequently that are selection operation, crossover operation, mutation operation and replacement operation. All operation repeats until give a good result or end certain generation. A GAs finds a good results when have good selection pressure, so multi-population methods are useful because we can have bigger memory with use several processors and their memories. In multi-population methods there are several processors an each one have independent populations and each processor run a simple GAs, after certain generations all processor will stop and send some chromosome (Migration) with certain strategy for example best or worse, and share results of solutions together. In this method there are some important parameters, migration rate, migration gap, topologies. Master-Slave method is other methods.

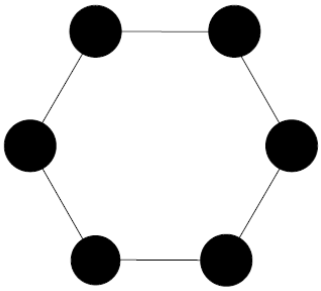

Figure 1. Multi-population schema [3]

In Master-Salve one processor is master and does all important operations of GAs same crossover, mutation, replacement and selection and other processor that called slave only evaluate fitness function and send back results to master processor. This method can be implementing synchronous and asynchronous. Synchronous methods behavior is same as simple GAs but with better runtime.

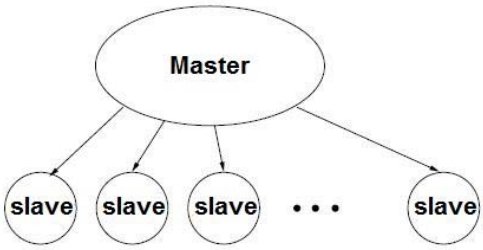

Figure 2. Master-Slave schema [3]

Fine grain are suitable method for parallel computing with massive processors and each processor can communicate to neighborhood processors and each individual can recombination with each individual on neighborhood processors. Speed of this methods are good but isn't economical.

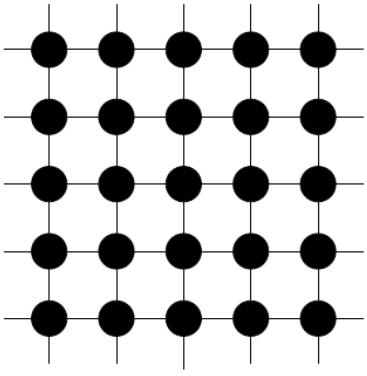

Figure 3. Fine grain schema [3]

Hybrid GAs are compound method and create from two levels, upper level use multi-population method and in lower level use each one of multi-population or master-slave or fine grain. This method is more efficient and faster than other methods because in this method we use capability of each method alone.

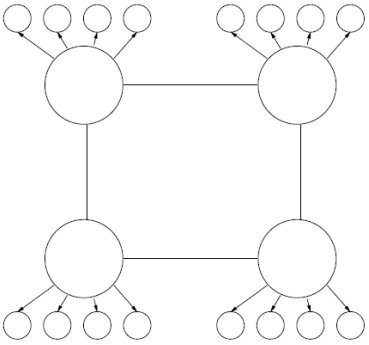

Figure 4. Hybrid schema (high level is multi-population and low level is master-slave) [3]

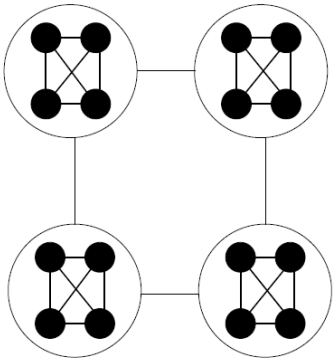

Figure 5. Hybrid schema (high level and low level are multi-population) [3]

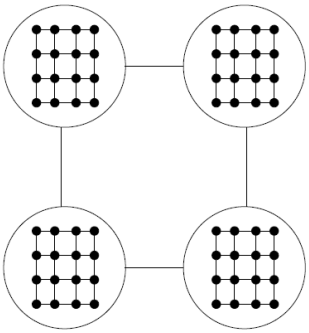

Figure 6. Hybrid schema (high level is multi-population and low level is fine grain) [3]

B. Parallel Ant Colony Optimization

Now we want to review on parallel ACO categories. There are two important methods to implement parallel ACO, the first one is PACO and other one is PACO-CGD.

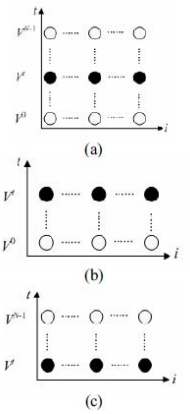

Figure 7. Graph structure [4]

PACO method is a multi-population method that each processor has an independent population, each processor run serial ACO independently and after a certain frequency sends some useful information to other processors and the name of this operation is migration. This method is useful but there are some weak points and researchers can eliminate them.

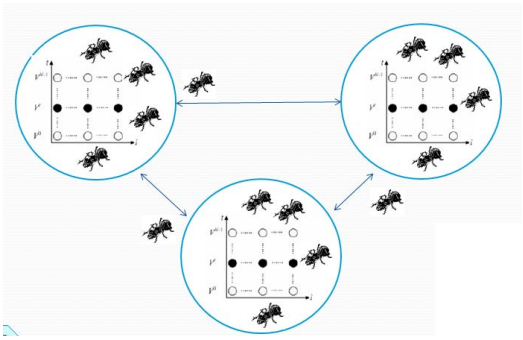

Figure 8. PACO schema

In PACO-CGD constructor graph decompose on smaller part and each part send to each processors and each one can run ACO method on each part and the speed of this method is better than PACO method.

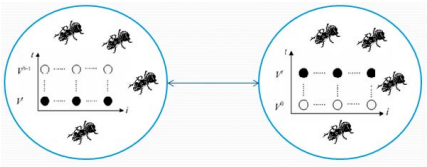

Figure 9. PACO-CGD schema

C. Parallel Artificial Bee Colony Algorithm

Other methods that we want review it is parallel implementation of BEE. There are three categories for parallel BEE, master-slave, multi-population and hybrid method.

Master-slave method is same master-slave GAs and one processor is master and run important operations and other processors only evaluate in individuals.

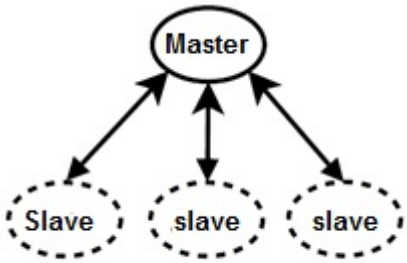

Figure 10. Master-Slave schema [5]

Multi-population method is same other course grain methods and each processor has a independence population and run serial ABC method on their population. Migration operation is available and important parameters are migration gap, migration rate and network topology.

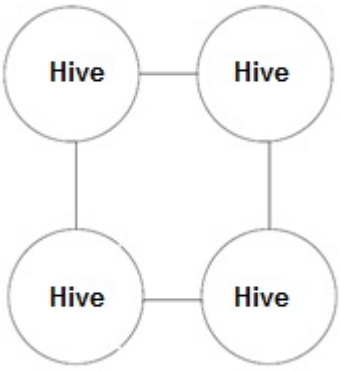

Figure 11. Multi-population schema [5]

Hybrid methods are a mix method that in high level use multi-population method and in low level use master-slave methods and work same hybrid GAs.

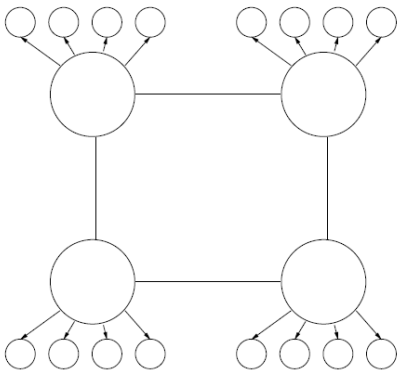

Figure 12. Hybrid method schema [5]

D. Parallel Particle Swarm Optimization

PSO is a successes method on continuous problems that parallel implementation of it is very useful and implemented on two methods, MPSO and MRPSO.

MPSO is a multi-population method that works like other multi-population methods and each processor has different

population and each one run simple PSO on their population and migration operation is available. MPSO was a succeed method that contained good results on different problems.

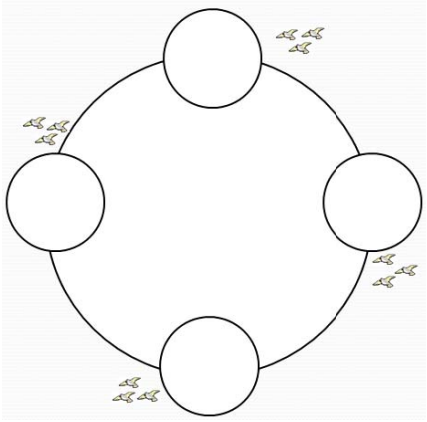

Figure 13. MPSO method schema

MRPSO is a improve method of MPSO and add an extra component of MPSO that called repulsive. Repulsive component in each processor try to give good population diversity. The particles that migrate in the even swarms should be as different as possible to the particles already contained in those swarms, High degree of diversity in EC methods are very useful and help me to contain a good result.

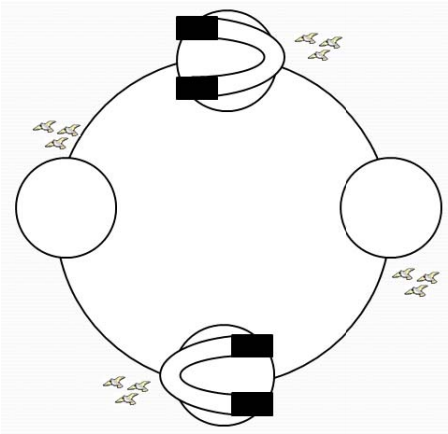

Figure 14. MRPSO method schema

E. Parallel Memetic Algorithms

Memetic Algorithms are population-based heuristic search approaches for optimization problems similar to genetic algorithms (GAs). GAs, however, rely on the concept of biological evolution, but MAs, in contrast and mimic cultural evolution. Parallel memetic algorithms implemented in coarse grain methods that called PARME.

PARME is a multi-population method that in each processor is a dependence population and each processor run memetic algorithm dependently. There is a big difference between this method and other methods like parallel genetic algorithms. One of processor called master and this processor control behavior of other processors and creates an operation table in each iteration and sends it to all processors. This operation table has value of important parameters on memetic algorithms that each iteration will change.

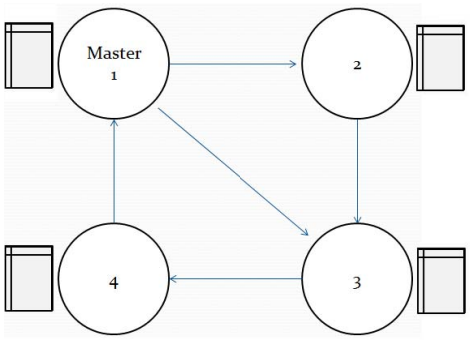

Figure 15. PARME method schema

III. THREE GENERAL FRAME WORKS TO PARALLELIZE EC METHODS

In this paper we review five parallel EC methods and describe all important technique of them, we can find some common technique that repeated on all parallel methods.

Multi-population methods are one of common techniques that are more popular than other methods. Master-Slave methods are other methods that used in different parallel implementation. Hybrid methods are other methods that are common on more parallel implementation that are faster than others method.

We want introduce three General frame works to parallelize all other EC methods that we categories them on Multi-Population methods, Master-Slave methods and Hybrid methods that describe them on next chapters.

TABLE I. TABLE OF OPERATIONS AND SYMBOLS

Title of operation	Symbols
Run serial EC methods dependency	
Independent initial population	
Run selection, exploration, exploitation and replacement operation	
Manage algorithm	
Evaluate fitness function	
Migration	

A. Multi-Population method

All EC methods are population base and selecting pressure is very important to obtain a good solution. Memory and speed of processor are limit and we can select limited numbers of initial population and run EC algorithms on them. This problem can converge to best local optimum solutions. We can solve this problem with using Multi-Population methods to increase number of initial population and increase selecting pressure. It is a logical technique that can increase probability to finding best solution.

In multi-population method we have two or more processors and we should create dependence population in each processor and run serial EC method on each processor. In

this method after certain iteration each processor sends some important elements to other processor and they should replace on some elements on destination processor. This operation is migration and there are different policy to select elements to migration and different policy to replacement element on destination processor. Selecting policy are the best or random, in the best policy processor selects best elements to migration and sends them to other processor and in random policy processor select some elements randomly. Replacement policy divided to three categories, the best, the worse and random. In the best method destination processors select the best elements to replacement, in the worse select the worse and in random method select some elements randomly to replacement.We should select some different important parameters like connection topology, migration rate and migration gap. Migration rate is the number of elements that transfer to other processors and migration gap is distance between to migration. Connection topology is dependent to two migration rate and migration gap for example if we want have short migration gap with high degree of migration rate we should use full connection topology and if we want have spars connection topology we should use a algorithms with long migration gap and low degree of migration rate.

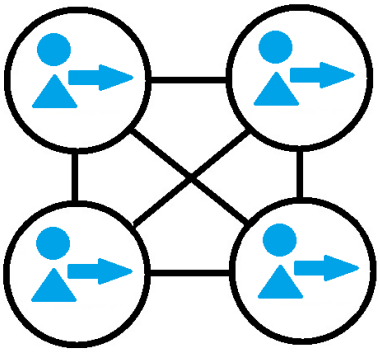

Figure 16. General framework of Multi-population method

B. *Master-Slave method*

Other method to solve other problems of some EC methods is complexity of evaluation of fitness function. This cost is different in different problems and can create some delay to solving problems with serial methods, we can divide them to several groups and send each one to a processor and all processors evaluate them in same time and send results to a main processor. This way is like master-slave methods.

In master-slave methods we have some different processors and we divide all population to different groups and send each one to a processor and each one evaluate fitness function of each elements. In this method a processor is master and does all important operation of EC method and manages parallel methods. Master processor divide all elements to some different groups and send them to other processors and each one evaluate fitness function and send back results to master processor. These processors called slave processor and only evaluate fitness function of their group.

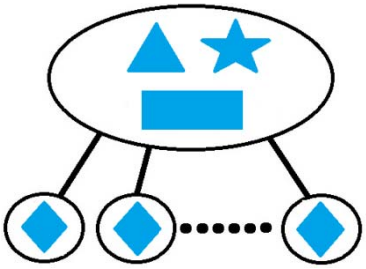

Figure 17. General framework of Master-Slave method

Master processor does all important operations of EC methods like create population, selection, exploitation, exploration and replacement. This method can implement synchronous and asynchronous. Behavior of synchronous method is like serial methods but in asynchronous method has different behavior of serial methods.

Master–slave method is success when we use share memory and each processors can use memory independently.

C. *Hybrid method*

Hybrid method is compound method that creates with mixture of two methods. We select two methods that use all benefit of them to contain better results. In our hybrid method we use two methods in two level, high level and low level. In high level we use multi-population methods that use its benefit (high selection pressure) and in low level we use master-slave method that increase speeds of algorithms and help to evaluate fitness function faster than multi-population method. Implementation of this method is more complex than each other methods alone but we can obtain better speed and efficiency.

In this methods each processor in high level are master processors and other processors in low level are slave. Each master processor runs all operation of serial EC method and slaves' processors only evaluate fitness function. After a certain repetition all master processors halt their action and send some of its elements to other master processor (migration). Migration has different policy to selection and replacement. Selection policy is categorizes to two way, the best and random and replacement policy are categorize to three way, the best, the worse and random. In the best policy we select the best elements to selection or replacement and in the worse policy we select the worse elements to selection or replacement and in random policy we select some elements randomly.

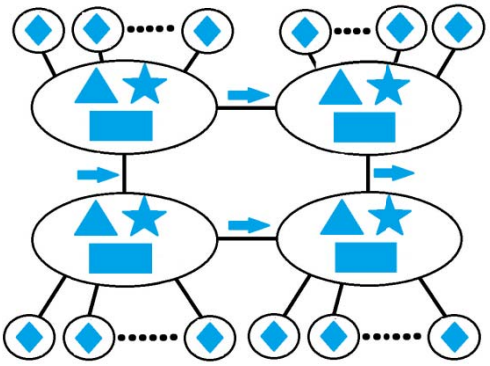

Figure 18. General framework of Hybrid method

IV. CONCLOUSION

In this paper we review some parallel implementation of EC methods and find some command behavior of them and developed them to three general frameworks to parallelize all other serial EC methods.

Each one of them has different benefit that we can choose one of them. Multi-population methods are useful when we have a problem with a big search space and we can increase selection pressure. Master-slave methods are useful when we have a problem with a very complex fitness function and we can decrease time of algorithm. Hybrid methods are useful when we have a problem with a big search space and complex fitness function.

We implement these methods to parallelize Imperialistic Competitive Algorithms and obtain good results in different problems and benchmarks. I will parallelize COCOO method in future works.

REFERENCES

[1] H. Lotfi, A. Boroumandnia and Sh. Lotfi , "Task Graph Scheduling in Multiprocessor Systems Using a Coarse Grained Genetic Algorithm," 2nd International Conference on Computer Technology and Development, IEEE, 2010.

[2] H. Lotfi, Sh. Lotfi and A. Boroumandnia , "Task Graph Scheduling in Multiprocessor Systems Using a Two Population Genetic Algorithm, " International Conference on Software and Computing Technology, IEEE, 2010.

[3] E. Cantú-Paz, "A Survey of Parallel Genetic Algorithms," Department of Computer Science and Illinois Genetic Algorithms Laboratory University of Illinois at Urbana-Champaign, 1997.

[4] H. Liu, P. Li and Y. Wen , " Parallel Ant Colony Optimization Algorithm," World Congress on Intelligent Control and Automation, China, June, 2006.

[5] R. Parpinelli, C. Benitez and S. Lopes , " Parallel Approaches for the Artificial Bee Colony Algorithm," Santa Catarina State University (UDESC) - Joinville - SC – Brazil, 2011.

[6] L. Vanneschi, D. Codecasa and G. Mauri , " A Comparative Study of Four Parallel and Distributed PSO Methods," Systems and Communication (D.I.S.Co.), University of Milano-Bicocca, Milan, ITALY, 2012.

[7] J. Digalakis and K. Margaritis , " A Parallel Memetic Algorithm for Solving Optimization Problems, " 4th Metaheuristics International Conference, Parallel Distributed Processing Laboratory, Greece, 2001.

[8] H. Narasimhan , " Parallel Artificial Bee Colony (PABC) Algorithm, Department of Computer Science and Engineering College of Engineering, Anna University, IEEE, 2009.