*Journal of Advanced Computer Science & Technology*, 4 (1) (2015) 75-83
*www.sciencepubco.com/index.php/JACST*
*©Science Publishing Corporation*
*doi: 10.14419/jacst.v4i1.4162*
*Research Paper*

# The distributed parallel genetic algorithm on the ad hoc network

**Nima Afifi Kisomi [1]*, Hassan Tavakoli [2]**

[1] *MSc Student, Guilan University, Rasht Branch, Iran*
[2] *Assistant Professor in Department of Electrical Engineering, Faculty of Technology & Engineering, Guilan University, Rasht Branch, Iran*
**Corresponding author E-mail: afifi@msc.guilan.ac.ir*



## Abstract

Today, mobile computing is one of the important issues in computer and network sciences. Using the processing power of mobile devices purposefully for solving complex issues is one of the research fields for researchers. One of the important issues in the optimization which needs a high processing power for finding the best possible answer is travelling salesman problem. In this paper, by providing a method based on the distributed parallel genetic algorithm on a number of mobile nodes in the Ad Hoc network, it was attempted to increase the speed of finding the best answer for the travelling salesman algorithm.

***Keywords***: *Ad Hoc Networks; Distributed Algorithm; Parallel Computations; Genetic Algorithm; Optimization.*

## 1. Introduction

Today, mobile computation means using the processing power of mobile devices for doing computational affairs. Mobile computations have a limited power which mobile distributed computations can be used for their solution in which the multi-user computing power is used. The mobile distributed computations divide into two categories of "with center" and "without center". In mobile distributed computations with center, issues such as the possibility of error and distance are posed [1]. For mobile distributed computations without center, a strategy such as using Ad Hoc networks can be presented. Mobile computations make it possible to use a computing device, even when it is changing its location; therefore, the transportability is one of the aspects of mobile computations. This method also makes it possible to use the computation ability without center and or the predefined connection in a network for information dissemination and or sharing [3].
Today, with the increasing expansion of smartphones, tablets and powerful mobile computers, their potential computing power can be actualized using the mobile processing, and they can be used for wireless computations. With the combination of mobile computations and mobile Ad Hoc networks, a subject is formed called "mobile computations in Ad Hoc networks" which can use this computing power [2], [3].

**Fig. 1:** The Overview of Mobile Devices Communication

In this paper, we provide a method for solving the travelling salesman problem using the genetic algorithm distributed on a set of mobile nodes in parallel and prove the reduction of total time for finding the best answer in case of increasing processing nodes. In our proposed method, the problem parameters are distributed on processing nodes based on the processing power of each node and each node accelerates the process of finding the problem's best answer by computing the part it is assigned to and returning the reply to the distributed node.

## 2. The travelling salesman problem

The travelling salesman problem is one of the very important and widely used problems in computer sciences and operations research. This problem is in a standard form in mathematics as finding a Hamiltonian Cycle with the minimum weight in an undirected weighted graph $G = (V, E)$ with n vertices in which vertices indicate cities. Edges show intercity paths and weight shows intercity distance [7], [12]. The number of vertices in a complete graph equalsm $= n \times (n-1)/2$. The final method for solving the travelling salesman problem includes passing all possible paths, evaluating the relating travel distance and finding the travel with the minimum distance. The total number of passable paths for n city equals n!. consequently, for large values, finding the cost of all travels is very time consuming or sometimes impossible [7]. The parallel process will be very useful in reducing the computing time of this problem [6].

## 3. The parallel genetic algorithm

The genetic algorithm is a powerful technique used for solving different problems. This algorithm was inspired by the theory of human evolution in which a society with constant individuals, individuals with high fitness, is selected and this process continues until creating a society with optimal answers for the problem corresponding to that society [4]. In the genetic algorithm, first, we produce several answers randomly or based on a special algorithm. This set of answers is called the initial population. Each answer is called a chromosome. Then, after selecting better chromosomes, using the genetic algorithm operators, chromosomes are combined and mutated. Finally, we combine the current population with the new one which is resulted from the combination and mutation in chromosomes [4]. [6].
In fig. 2, the standard genetic algorithm flowchart is shown.
One of the problems of the genetic algorithm is the high volume of information and time for processing and reinforcing answers and accessing an almost optimal answer in algorithms [5]. The nature of the genetic algorithm is such that there is a kind of implicit parallelization in it. For example, the computation of individuals' fitness can be done independent from the society, and its implementation in parallel is simpler and more efficient [4], [6]. Parallel genetic algorithms are better than sequential genetic algorithms in terms of higher acceleration and greater scalability. The parallelization of genetic algorithms leads to the improvement of genetic algorithms' performance in terms of efficiency and speed [4]. The parallel genetic algorithm has different categories one of the best known of which is master-slave model used in the paper. From among other categories, the coarse-grained and fine-grained model can be mentioned [6]. In the master-slave model, the society is maintained in the master and slaves are used for the evaluation of fitness. This model does not have any default architecture and its implementation on different types of architectures is possible; therefore, it can be used in Ad Hoc networks [4], [6], [8], [9]. In this model, the number of individuals assigned to each processor for computing the fitness is fixed, but in some applications, there is a need for balancing the computational load between processors [6].
The relation for obtaining the acceleration is as follows:

$$S = \sqrt{\frac{n\,T_f}{T_c}} \tag{1}$$

In this relation, n is the number of individuals in the society, $T_f$ is the required time for computing the fitness of each individual and $T_c$ is the required time for sending information from the master to the slave. Fig.3 presents the overview of the master-slave model [6].

## 4. Ad Hoc networks

Wireless Ad Hoc networks include a set of distributed nodes which are wirelessly connected to each other [1]. Each node connects to other nodes directly and without any center. Nodes in this network continually change their position; so, the most important feature of these networks is the existence of a variable and dynamic topology resulted from nodes mobility [8], [9].

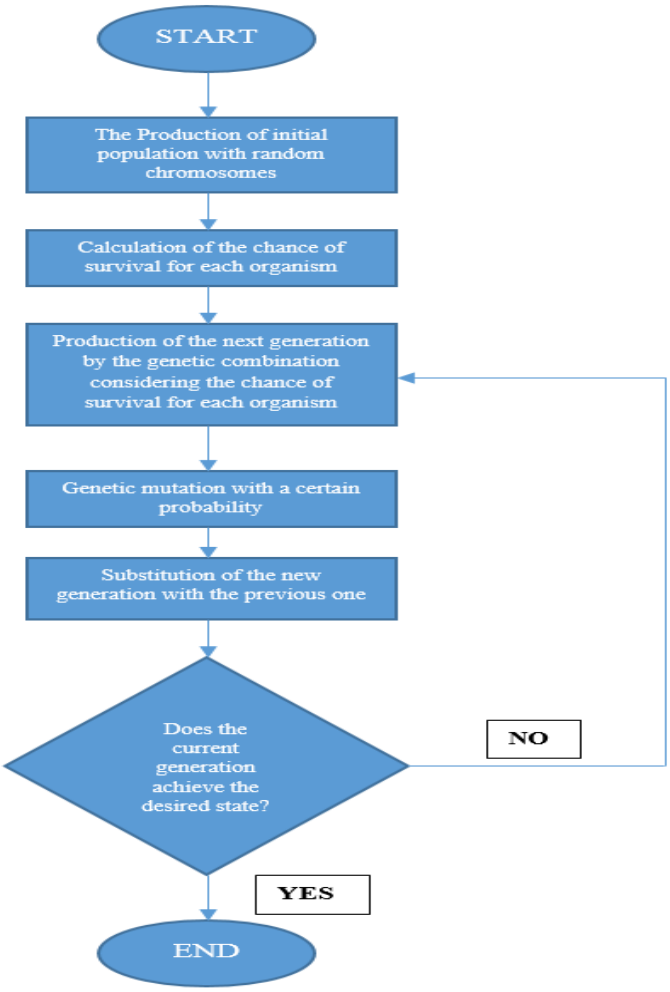


**Fig. 2:** The Genetic Algorithm Flowchart

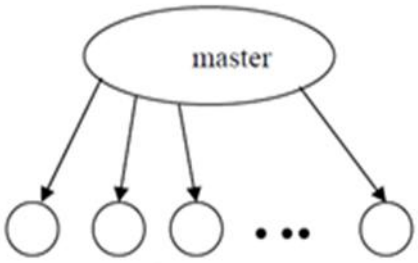


**Fig. 3:** Master-Salve Model

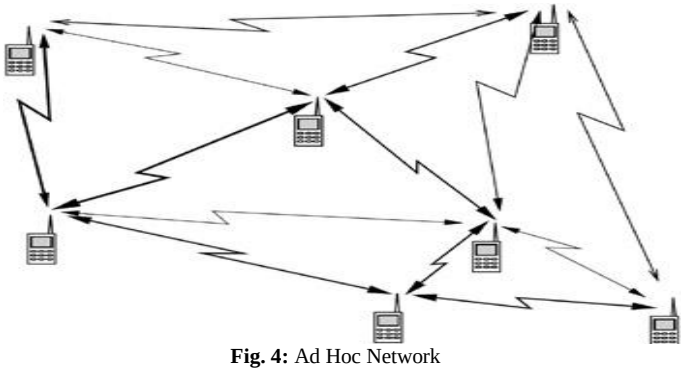

**Fig. 4:** Ad Hoc Network

A mobile Ad Hoc network is a non-infrastructure network of mobile devices which are connected wirelessly. Nodes in this network are autonomous, i.e. any device can freely move in the network [8]. Mobile Ad Hoc networks are better than conventional wireless networks because of the simple development of these networks, their development speed and lack of dependency on a constant structure, and as mentioned before, they are a proper platform for implementing the master-slave model [6], [8]. Generally, when an infrastructure is not accessible and the creation of an infrastructure is not practical and affordable, the use of an Ad Hoc network is useful [1], [9].

### 4.1. Computation in Ad Hoc networks

One of the challenges of Ad Hoc networks is the way of problem distribution among network's nodes because network nodes do not have the previous knowledge about the network's topology where there are located; so, they have to discover the destination in the network for communicating with other nodes [9]. But, here it is assumed that the master is communicating with some nodes which are used as slaves and ready to receive the problem from the master for processing and returning the answer to the master. Accordingly, it is better to distribute the genetic algorithm in parallel on the network nodes.

## 5. The proposed method

In this method, the main objective is to solve the travelling salesman problem which is a problem for finding a Hamilton Cycle with the minimum length using the genetic algorithm. When the travelling salesman problem is modeled mathematically, it turns into the Hamilton Cycle in mathematics and graphs theory and then will be ready for being solved [7].
Fig. 5 shows the cities' distribution in the travelling salesman problem under the process, including 150 cities and cities are related and randomly presented for simulating the efficiency of the proposed method in this article.

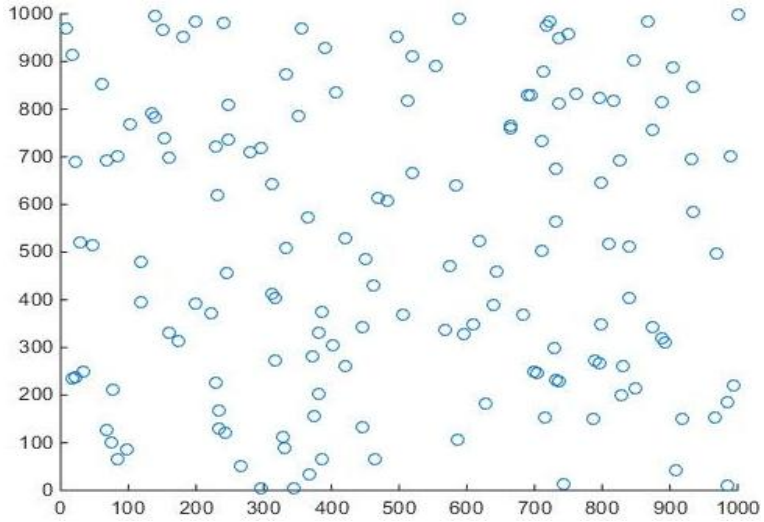


**Fig. 5:** Cities' Distribution in the Travelling Salesman Problem under Study

In our proposed method, the problem parameters are distributed based on the processing power of each node among the existing processing nodes and the standard genetic algorithm is performed on each node. After receiving the problem parameters from the master, each node is numbered by a label granted from the master. This number is notified to other existing processing nodes. Therefore, each node not only knows its number, but also other nodes' numbers. It leads to the simplicity of immigration among subpopulations of each node. Nodes can exchange some individuals for preventing from being trapped in the local optimum. This exchange called immigration leads to the sharing of genetic parameters by processing nodes. In our proposed method, the immigration relationship among nodes is as follows:
Assume that each slave node is $S_0, S_1, \ldots S_{n\,Slave-1}$. Each $S_i$ selects the best individual in its population in terms of fitness and sends it to the substitute $S_{(i+1) \bmod nslaves}$. So, the last node will send its best individual to $S_0$. As a result, each $S_i$ receives individuals from $S_{(i-1) \bmod nslaves}$ and substitutes its worst individual in terms of fitness with the individual it received. After several immigrations, each slave has the best individuals found by other processing nodes.
For distributing the problem parameters, each node notifies its processing power to the master node with $PP_i$ (processing power) parameter.

$$PP_i = CPU \times RAM \times CATCH \quad (2)$$

For dividing the problem parameters on slave nodes, the master node works as follows which each slave is computed through the relation $q_i = \left[\frac{pp_i}{\sum pp_i}\right]^{+-}$ and problem parameters are assigned to each slave based on the weight $q_i$ obtained. So:

$$[x]^{+-} = \begin{cases} [x]+1 & \text{if } x-[x] > 0.5 \\ [x] & \text{if } x-[x] < 0.5 \end{cases} \quad (3)$$

The general puzzle of distributing problem parameters and returning of the answer to the master for studying the desirability of the answer is shown in fig. 6.

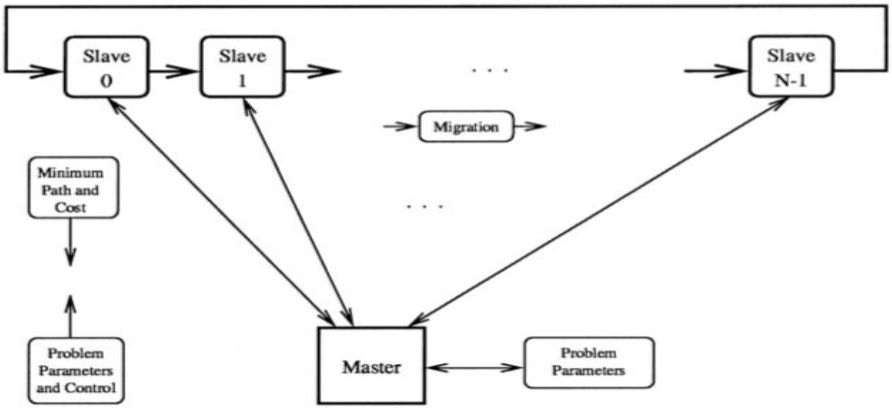


**Fig. 6:** The Problem Distribution Architecture by the Master on Slaves for Achieving an Optimal Answer

In a graph $G = (V, E)$ of n vertices, a Hamiltonian Cycle indicates a sequence of n vertices creating the cycle without repeating any vertex. In general, any Hamiltonian Cycle indicates one of n! possible permutations of graph $G = (V, E)$ vertices. We assume that graph $G = (V, E)$ shows the problem and $V = \{v_0, v_1, \ldots \ldots v_{n-1}\}$. Any individual in population is a Hamiltonian Cycle which is shown as follows:

$$hc_k \ (k = 1,2, \ldots\ldots, n!) \quad (4)$$

Therefore, as all possible permutations of elements, V equals $V_n = \{p_1, p_2, \ldots \ldots p_{n!}\}$. Now, the fitness function is defined as follows:

$$f(hc_k) = \sum_{i=0}^{n-1} \text{Cost}\left(v_{pk(i)}, v_{pk((i+1)\text{mod}n)}\right) \quad (5)$$

$\forall\, k \in [1, n!]$,
Where,

$$hc_k = \langle v_{pk(0)}, v_{pk(1)}, \ldots\ldots, v_{pk(n-1)} \rangle \quad (6)$$

cost $(v_i, v_j)$ Is the distance between cities $v_i$ and $v_j$; so, $F = hc_k$ can be the distance from the Hamiltonian Cycleh $c_k$.
Any slave receives problem parameters from the master, produces a random subpopulation and begins to perform the standard genetic algorithm among subpopulations. Individuals in the society of Hamiltonian Cycle are in the form of a sequence of vertices which make the mutation and crossover simpler and more efficient. In any generation and after the mutation and crossover operations were done by the slave, each slave sends the minimum cost of the Hamiltonian Cycle in its generation to the master. Based on the received results, the master decides whether the desired result is obtained or another generation must be computed. In our proposed method known as the steady state distributed parallel genetic algorithm, children are not substituted with their parents, but substitutes are members of population with the minimum fitness [10]. The crossover operation is done for each generation. In this article, OX crossover operator is used for the crossover operation. This crossover method is one of the crossover operators which maintain the relative order for which:

1) We select two parents randomly.
2) For producing the first child, we copy a random subsequence from one of the parents and insert it in the first child.
3) For filling other cities in the first child, we copy the second child.
4) For producing the next child, we change the role of parents.

In our proposed method, the highest time acceleration happens which the crossover rate is adjusted on 50%. Otherwise, the time of finding the best answer for the travelling salesman problem is significantly increased. In fig. 7, we observe the comparison among 4 acceleration rates of 50%, 70%, 80% and 90%.

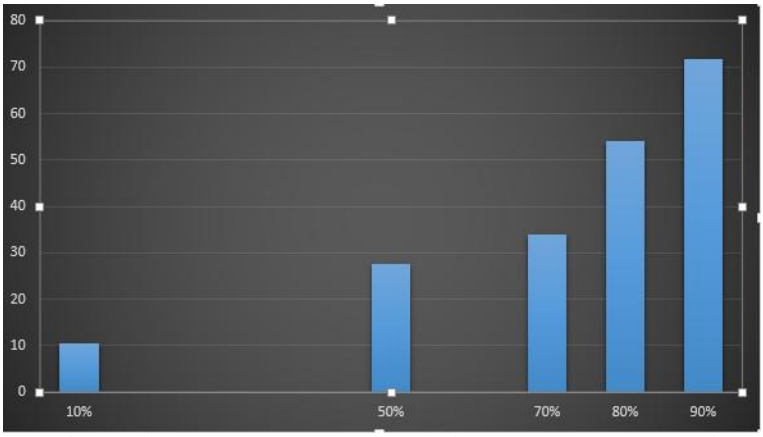


**Fig. 7:** The Increase of Time of Finding the Best Answer in Case of the Indiscriminate Increase of the Crossover Rate

As it is obvious in fig. 7, when the crossover rate is more than 50%, the time of finding the best answer is significantly increased.

For the design proposed in the article, the following flowchart can be presented:

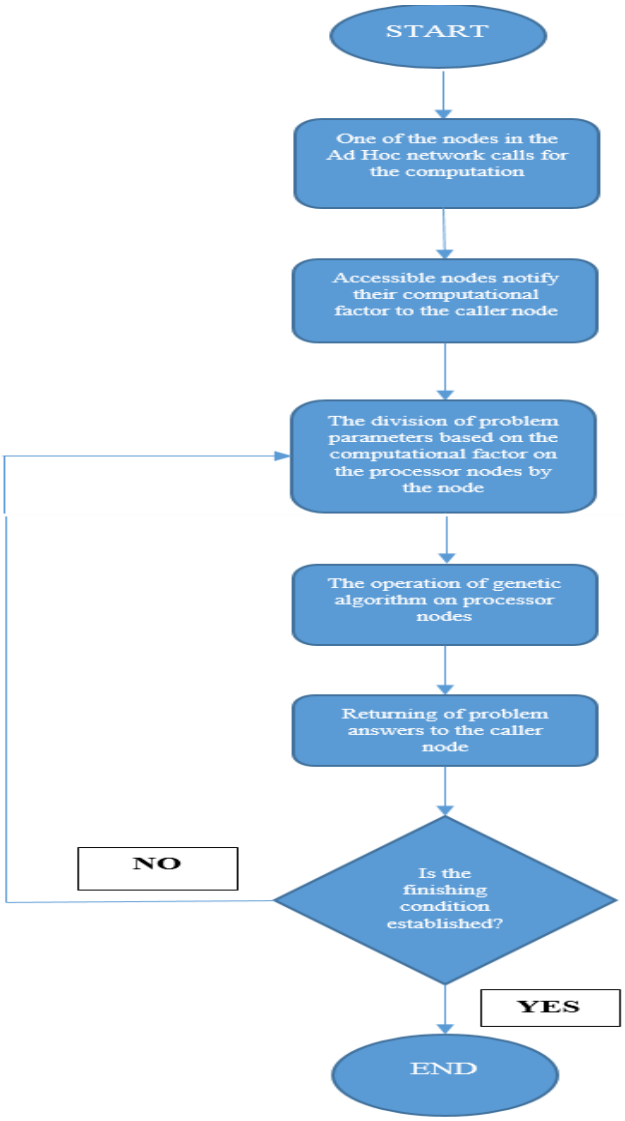


**Fig. 8:** The Proposed Method Flowchart

Based on the flowchart and the explained method, the JAVA pseudo code of the proposed method for the master node is as follows:

```
Broadcast_calculation_request()
Get_processing_power_from_slaves()
Devide_parametr_for_slaves()
Multicast_problem_parametrs()
While
    (min_cost > thereshold)^
    (not_enough_iterations) do
    Receive_min_tour_from_all_slaves()
    Compute_min_tour()
If
    Sol_found() then send_stop_to_slaves()
    Fi
Od
```

In addition, the pseudo code of the proposed method for the master nodes is as follows:

```
Receive_calculation_request_from_master();
Send_processing_power_to_master();
Get_problem_parametrs_from_master();
While
    (master_says_continue)^
    (not_enough_iterations) do
    Compute_cost_for_every_tour();
    Select_tour_to_keep_next_generation();
    Select_tours_to_crossover();
    Generate_offsprings();
      If (needed) then
          Mutate_random_tours();
          Migrate_best_tours();
Fi
Compute_minimum_tour();
Send_minimum_tour_to_master();
Wait_for_answer_from_master();
od
```

As it is observed in fig. 9, the simulation results show that with the increase in number of clusters from 1 to 4, the answer of the distributed genetic algorithm is accompanied with a uniform and significant reduction. This special feature of the master-slave distribution granted a unique and desired ability to the distributed genetic algorithm in order to have the effective implementation ability on structures without centers such as Ad Hoc networks.

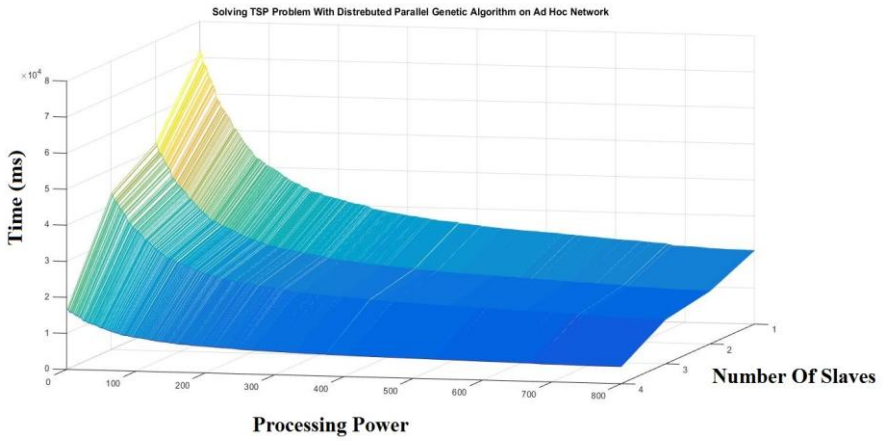


**Fig. 9:** The Speed of Finding the Best Answer for the Travelling Salesman Problem Based on the Proposed Method

For a better understanding of the improvement through the proposed method, the speed of finding the best answer for the problem is separately shown for slave nodes in fig. 10. Note that there is a great difference in the speed of finding an optimal answer by the proposed algorithm so that it leads to very high amount of savings in time; cost and energy, especially as processing nodes in Ad Hoc networks face the limitation of power.

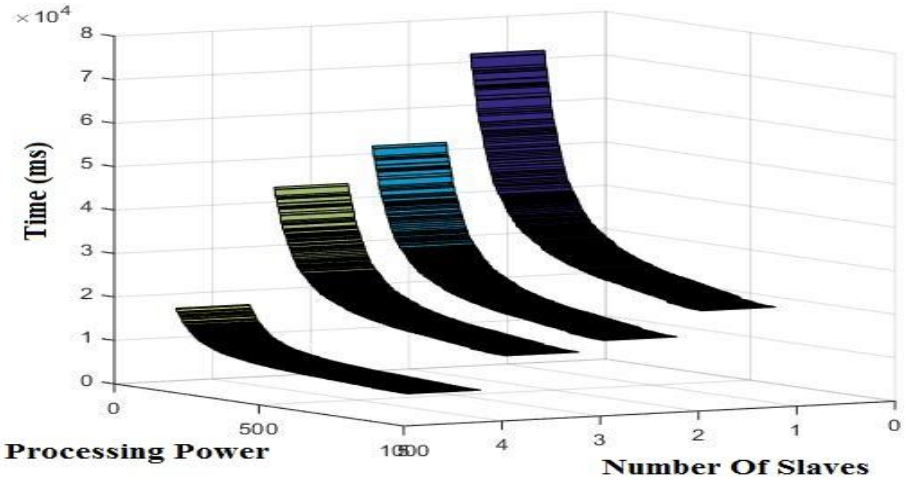


**Fig. 10:** The Processing Power of Nodes in Finding the Best Answer for the Problem Based on the Proposed Method Separately

# 6. Conclusion

In this article, after studying important issues such as the genetic algorithm and travelling salesman, it was attempted to solve the travelling salesman problem using the property of the master-slave distributed genetic algorithm in Ad Hoc networks. Results show the efficiency of the aforementioned method and its ability in Ad Hoc networks which significantly improves the speed of finding the best answer for the travelling salesman problem. The distribution of problem parameters by our proposed method which is proportional to the processing power of each slave node underlies the acceleration in solving problems and finding the optimal answer for the problem. The acceleration of the process of finding answer leads to the savings in energy and overcoming one of today's challenges in Ad Hoc networks, i.e. the limitation of nodes' power.

# References

[1] Mario Gerla, *"Ad Hoc Networks",* Springer, "Ad hoc Networks Technologies and Protocols ", Chapter 3, PP. 1-22, 2005.

[2] N. Fernando, S.W. Loke, W. Rahayu, *" Mobile cloud computing: A survey "*, Future Generation Computer Systems, Vol. 29, Issue 1, pp. 84-106, Jan. 2013. http://dx.doi.org/10.1016/j.future.2012.05.023.

[3] L. Kulik, *" Mobile Computing Systems Programming: A Graduate Distributed Computing Course "*, Distributed Systems Online, IEEE, Vol. 8, Issue 5, May 2007.

[4] M. Gorges-Schleuter,*" Explicit parallelism of Genetic Algorithms through population structures "*, Parallel Problem Solving from Nature, pp. 150–159, 1991.
[5] T.Starkweather, D.Whitley, K. Mathias, *"Optimization using distributed Genetic Algorithms "*, Parallel Problem Solving from Nature, 1991.
[6] Harun Raşit Er, Nadia Erdoğan ,*" Parallel Genetic Algorithm to Solve Traveling Salesman Problem on MapReduce Framework using Hadoop Cluster "*, International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, pp. 380-386, 2013.
[7] Saloni Gupta, Poonam Panwar, *"Solving Travelling Salesman Problem Using Genetic Algorithm "*, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue 6, Jun. 2013.
[8] Mohandas, G. Silas, S., Sam, S., *" Survey on routing protocols on mobile adhoc networks "*, Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), International Multi-Conference, pp. 514-517, march 2013.
[9] Krishna Gorantala, *"Routing Protocols in Mobile Ad hoc Networks "*, Master Thesis in Computing Science, Umea University, June 2006.
[10] ChetanChudasama, S. M. Shah and Mahesh Panchal, *"Comparison of Parents Selection Methods of Genetic Algorithm for TSP "*, International Conference on Computer Communication and Networks (CSI- COMNET), 2011.
[11] Dwivedi, TarunaChauhan,SanuSaxena and PrincieAgrawal, *" Travelling Salesman Problem using Genetic Algorithm "*, International Journal of Computer Applications(IJCA), pp. 25-30, 2012.
[12] Gábor Renner, Anikó Ekárt, *" Genetic algorithms in computer aided design "*, Computer-Aided Design, Vol. 35, Issue 8, pp. 709-726, July 2003. http://dx.doi.org/10.1016/S0010-4485(03)00003-4.