International Journal of Information & Network Security (IJINS)
Vol.2, No.1, February 2013, pp. 109~117
ISSN: 2089-3299

Solving the File Allocation Problem in the Distributed Networks by using Genetic Algorithms

M. H. Abd El-Aziz*, A. Younes, M. R. Hassan***, H. Abdo*****
*Faculty of Computer and Information Science Information Systems Department, Ain Shams University, Egypt.
**Faculty of Science, Computer Science Department Sohag University, Egypt.
***Faculty of Science, Mathematics Dept, Computer Science Branch, South Valley University, Aswan, Egypt

Article Info

Article history:

Received Aug 16th, 2012
Revised Oct 15th, 2012
Accepted Nov 8th, 2012

Keyword:

Genetic Algorithms
Computer Networks
Average Distributed - Program Throughput
File Allocation Problem

ABSTRACT

Average Distributed Program Throughput (ADPT) of the Distributed Computing System (DCS) depends mainly on the allocation of various resources. One of the important resources to be allocated on a DCS is various files. In this paper, we propose an approach that uses genetic algorithms to determine the optimal file allocation on the DCS that maximizes the ADPT with the constraint that the total number of copies of each file on a DCS must be equal to or less than the specified value. The algorithm has been applied on different network examples taken from literature; the results show that the algorithm is efficient to obtain better results.

Corresponding Author:

M. R. Hassan,
Faculty of Science, Mathematics Department,
Computer Science Branch,
South Valley University, Aswan, Egypt
E-mail: m_r_hassan73@yahoo.com

1. INTRODUCTION

A typical DCS consists of processing Element (PE), memory units, data files and programs as its resources. These resources are interconnected via a communication network that dictates how information could flow between PEs. Programs residing on some PEs can run using data files at other PEs as well. For successful execution of a program, it is essential that PE containing the program and other PEs that have the required data files, and communication links between them must be operational. Genetic algorithms (GAs) have been applied to various problems in the computer network design ([1-6]). Ahuja [7] developed a genetic algorithm to solve the capacity allocation problem of a given network topology such that the performance based reliability is maximized. Kumar et al. [8] developed a distributed genetic algorithm to optimize the performance and reliability for distributed computing systems under a given budget constrain. Kumar and Ahuja [9] developed a performance based reliability oriented file and capacity allocation scheme for distributed systems. Abd El-Aziz et al. [10] presented a GA for optimizing the Average Distributed Program Throughput (ADPT) and the total link capacity of a given network topology. The algorithm used GA to find the optimal set of capacities that maximize ADPT.

In this paper, we propose a genetic algorithm to determine the optimal file allocation on the DCS that maximizes the ADPT with the constraint that the total number of copies of each file on a DCS must be equal to or less than the specified value.

The rest of the paper is organized as follows; the description of the file allocation problem given in Section II. The Basic Components of the proposed genetic algorithm is given in Section III. Section IV presents the whole algorithm. Experimental results are given in Section V. Section VI illustrates the conclusion.

2. PROBLEM FORMULATION AND DESCRIPTION

2.1. Notation

L is the number of links in the network.
K is the number of options for the link capacities.
C_i is capacity of link i
α total traffic delivered**.**
α_{p_i} Average traffic requirement when program p_i is executed in a distributed environment.
ADPT is the Average Distributed Program Throughput.
N is the number of nodes.
C_{sum} is the total sum of C_i.
C_{max} is the maximum permissible system capacity.
Pop_size is the population size.
Max_gen is the maximum number of generations.
Pm is the GA mutation rate.
Pc is the GA crossover rate.
$MFST_j$ jth Minimum File Spanning Tree- is defined as the smallest subgraph of G that has required data files for the execution of a program.
$CMFST_j$ the Capacity of $MFST_j$.
$WMFST_j$ Weight for $MFST_j$.
FN_i the set of files (F's) needed to execute PRG.

L Total number of links in **DCS.**

FN Total number of files in DCS.

NF_i Total number of copies of file Fi on the DCS

$MaxNF_i$ The maximum number of copies of file Fi allowed on the DCS

ADPT(Pk) Average Distributed Program Throughput of program **Pk.**
x_{ij} Represents that the file F_i is located at node N_j.

2.2. Problem Formulation

The problem has been to determine the best allocation of files on nodes of DCS Such that the Average Distributed Program Throughput of program P_k (ADPT (Pk)) is maximized.

The mathematical formulation is:

$$\textbf{Max ADPT}(P_k)$$

S.T.

$$NF_i = \sum_{j=1}^{n} x_{ij} \leq MaxFN_i$$

I.e. the total number of copies of each file F_i does not exceed the maximum number of copies of file Fi allowed on the DCS.

3. THE GENETIC ALGORITHM

In the following subsections we will describe the components of the proposed GA.

3.1. Chromosomal Representation

If the network has n nodes and the number of files equals to nn, then the chromosome X has n x nn fields. Each field x_{ij} represents that the file F_i allocating on node j.

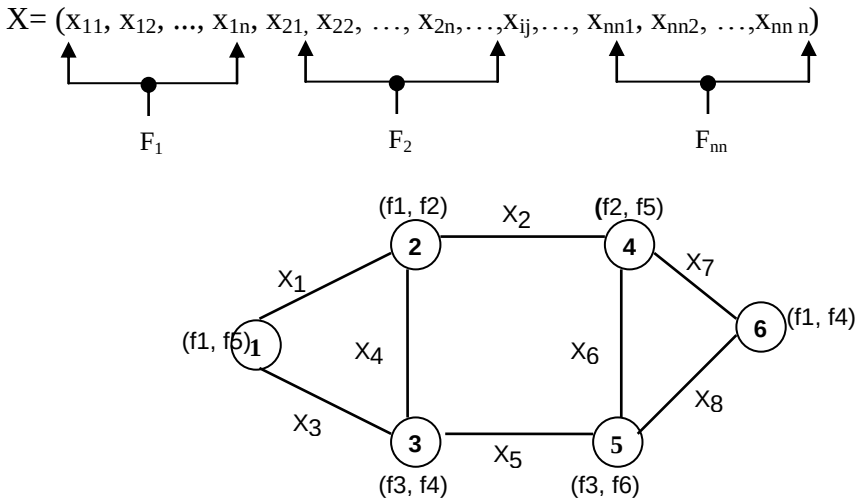

Figure 1. Computer Network Topology.

The network in figure 1 has 6 nodes and the number of files equals to 6, then the chromosome x will be in the form:

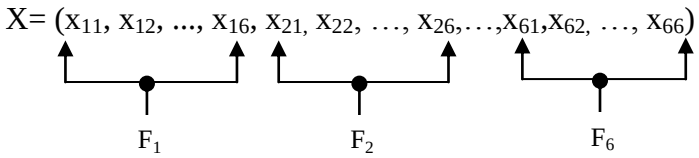

Figure 2. Chromosome of the network in figure 4.1

3.2. Initial Population

The initial population is generated according to the following steps:

Step 1: Randomly generated a chromosome X in the initial population in the form:
$X= (x_{11}, x_{12}, ..., x_{1n}, x_{21}, \ldots, x_{2n},\ldots, x_{nn1}, x_{nn2}, \ldots,x_{nn\,n})$
where $x_{ij} \in \{0,1\}$.

Step 2: Calculate the number of copies for each file F_i, NF_i.

Step 3: If NF_i of the generated chromosome in step 1 is greater than $MaxNF_i$ discard and go to step1.

Step 4: Repeat steps 1 to 3 to generate pop_size chromosomes.

3.3. The Objective Function

Find the best allocation of files on nodes of DCS Such that ADPT (P_k) is maximized by examining all possible cases for the distribution of files**.**

3.4. Genetic Crossover Operation

In the proposed GA, one-cut point crossover is used to breed two offsprings (two new chromosomes) from two parents selected randomly according to pc value. In particular, an integer value is randomly generated in the range (0, nn x n) as shown in figure 3.

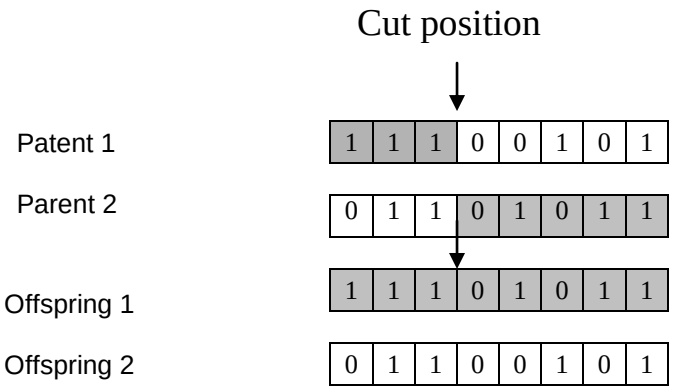

Figure 3. Cut point Crossover

3.5. Genetic Mutation Operation

A child undergoes mutation according to the mutation probability P_m and the mutation probability for each vector component P_m.

Step 1: Generate a random number r_1, $r_1 \in [0, 1]$.
Step 2: If $r_1 < P_m$, the chromosome is chosen to mutate and go
to step 3;
otherwise skip this chromosome.
Step 3: For each component of the child do:
Begin;
generate a random number r_2, $r_2 \in [0,1]$.
If $r_2 < P_m$ then mutate this component as follows:
If $x_{ij} = 1$, then $x_{ij} = 0$ and vice versa.

Else
skip this component.
End do.

Figure 4 shows an example of performing the mutation operation on a given chromosome.

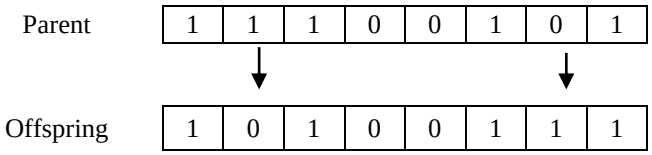

Figure 4. Mutation operator

4. THE PROPOSED GA

This section presents the proposed GA for solving the file allocation problem, described in section 4.3. The steps of this algorithm are as follows:

Step 1: Set the parameters: pop_size, max_gen, P_m, P_c, $MaxNF_i$, and set gen=0.
Step 2: Generate the initial population according to section 4.3.2.
Step 3: To obtain two new childs select two chromosomes from the current Population according to p_c. Apply crossover, and then mutate the two new Childs according to p_m parameter.
Step 4: Evaluate the two new childs as follows:
Step 4.1 Compute NF_i & $ADPT(P_k)$ for each one.
Step 4.2 For each child:

If $NF_i \leq MaxNF_i$ then increase pop_size.
Else discard this child.

Step 5: Repeat Steps 3 to 4 to generate pop_size chromosomes.
Step 6: Set gen =gen + 1.
Step 7: If gen > max_gen then goto step 8. Else goto step 3 to find a new generation.
Step 8: Print out the obtained results and end the algorithm.

5. EXPERIMENTAL RESULTS

In this section, we present the results of applying the proposed algorithm to sample networks taken from literature.

5.1 The Results of Case 1 And Case 2

In this section, we study a sample network of 6 nodes and 8 links with 6 files, given in Figure 1. For each case, Table 1 shows the capacity value and the corresponding probability for each link. The best value of the ADPT and the corresponding generation number are shown in Table 2.

Table 1. Informations of case 1 and case 2

Case no.	The values of N and L	Capacity values of each link	The corresponding Probabilities
1	N =6 L =8	40,30,40,25,35,25, 40,50	0.90, 0.85, 0.90, 0.80, 0.90, 0.80, 0.95, 0.90
2	N =6 L =8	40,35,40,35,35,30, 60,60	0.90, 0.85, 0.90, 0.80, 0.90, 0.80, 0.95, 0.90

Table 2. The Results of case 1 and case 2

	Case Studied	Generation number	ADPT	The proposed chromosome X
The program p1 start on node 1. The required files for execution are F1, F2, and F3	**Case 1**	**2**	**0.9877**	11001010100001010 00100110010000010
	Case 2	**3**	**0.9877**	11001010100001010 00100110010000010
The program p1 start on node 3. The required files for execution are F1, F2, and F3	**Case 1**	**2**	**0.9962**	11001010100001010 00100110010000010
	Case 2	**3**	**0.9962**	11001010100001010 00100110010000010

5.2. The Results of Case 3 And Case 4

In this section, we study two sample networks of 5 nodes and 7 links shown in figure 5 and 7 nodes and 10 links shown in Figure 6. For each case, Table 3 shows the capacity value and the corresponding probability for each link, furthermore the number of nodes and links. The best value of the ADPT and the corresponding generation number are shown in Table 4.

Table 3. Informations of cases 3 and 4

Case no.	The values of N and L	Capacity values of each link	The corresponding Probabilities
3	N =5 L =7	25,30,35,25,40, 30,50	0.90, 0.90, 0.90, 0.90, 0.90,0.90, 0.90
4	N =7 L =10	25,30,35,25,40, 30,50,40,25,50	0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90

Table 4: The Results of case 3 and 4

	Case Studied	Generation number	ADPT	The proposed chromosome X
The program p1 start on node 1. The required files for execution are F1, F2, and F3	**Case 3**	**454**	**0.9988**	1100101010 0010 1 1001 00001
	Case 4	**1000**	**0.9988**	1100000010101001010 1 0010010000100100001 00
The program p1 start on node 3. The required files for execution are F1, F2, and F3	**Case 3**	**454**	**0.9900**	1100101010 0010 1 1001 00001
	Case 4	**1000**	**0.9999**	1100000010101001010 1 0010010000100100001 00

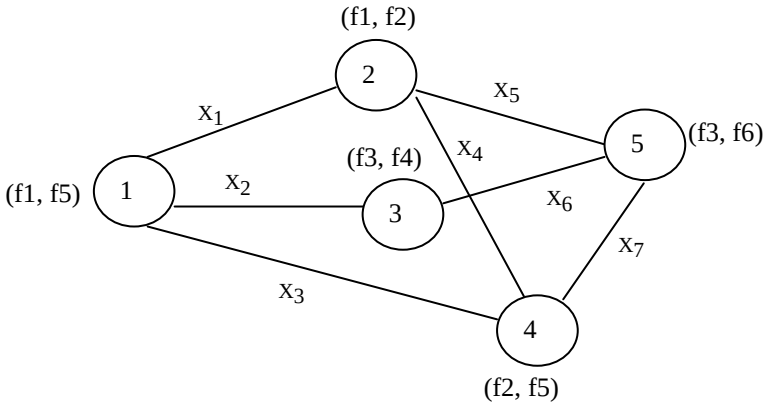

Figure 5. Network Topology

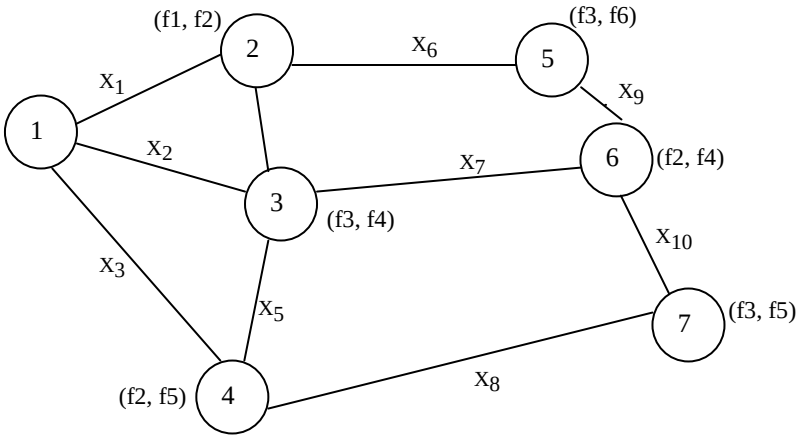

Figure 6. Network Topology

5.3. The Results of Case 5 and Case 6

In this section, we study two sample networks of 4 nodes and 5 links shown in figure 7 and 10 nodes and 19 links shown in figure 4.8. For each case, Table 5 shows the capacity value and the corresponding probability for each link, furthermore the number of nodes and links. The best value of the ADPT and the corresponding generation number are shown in Table 6.

Table 5. Informations of case 5 and case 6

Case no.	The values of N and L	Capacity values of each link	The corresponding Probabilities
5	N =4 L =5	25, 30, 25, 30, 40	0.90, 0.90, 0.90, 0.90, 0.90
6	N =10 L =19	30,35,30,40,45,30,35,40, 45,35,30,40,35,45,40,30, 40,45,35	0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90

Table 6. The Results of case 5 and case 6

	Case Studied	Generation number	ADPT	The proposed chromosome X
The program p1 start on node 1. The required files for execution are F1, F2, and F3	**Case 5**	**6**	**0.9900**	1010100101010010
	Case 6	**7**	**0.9971**	1000000000100010010001010100001011001101 0
The program p1 start on node 3. The required files for execution are F1, F2, and F3	**Case 5**	**6**	**0.9990**	1010100101010010
	Case 6	**7**	**0.9960**	1000000000100010010001010100001011001101 0

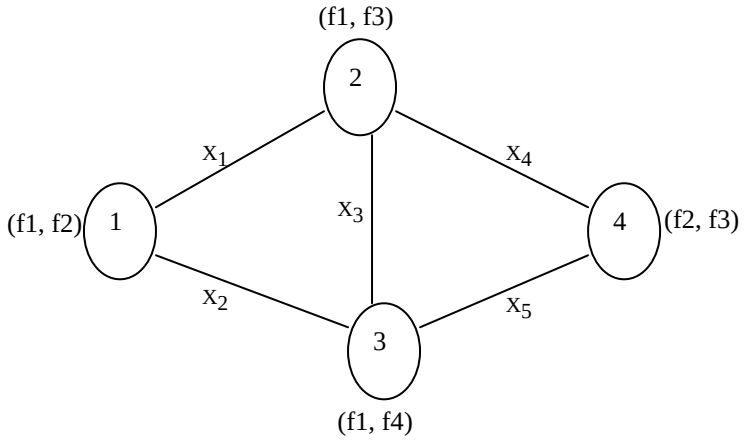

Figure 7: Network Topology

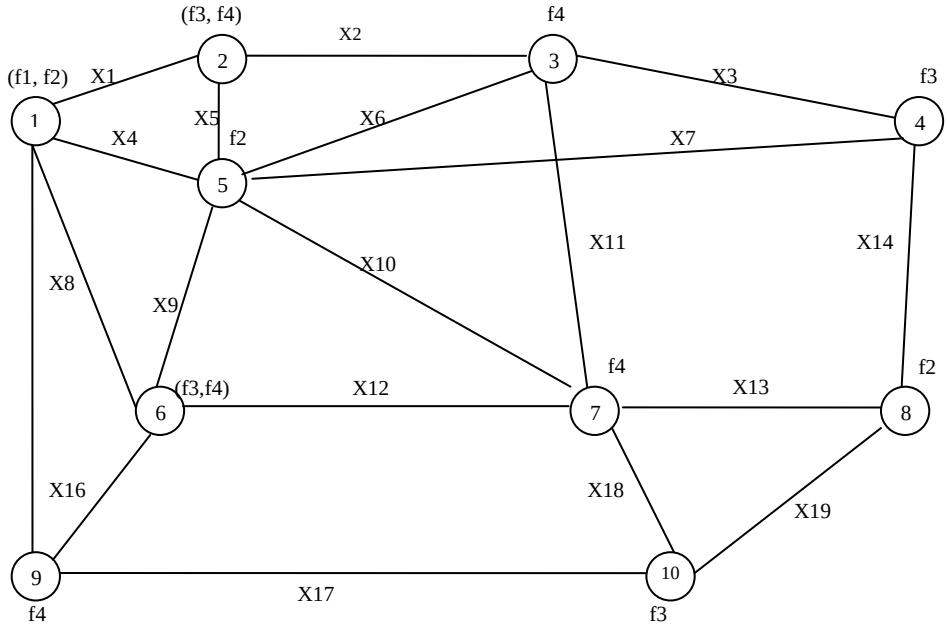

Figure 8. Network Topology

5.4 The Results of Case 7

In case 7, we study a sample network of 6 nodes and 8 links with 6 files, given in Figure 4.9, the information of this case shown in Table 4.8. The best value of the ADPT and the corresponding generation number are shown in Table 4.9.

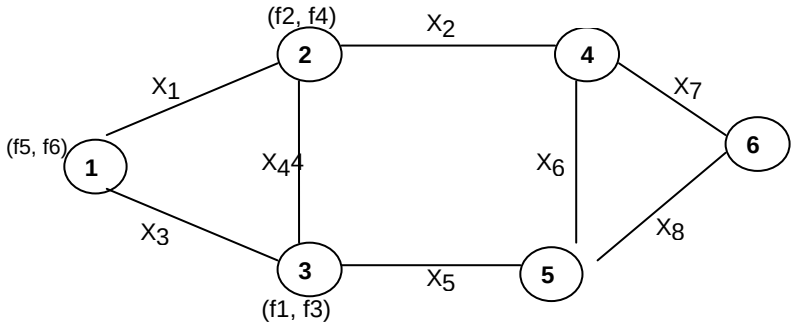

Figure 9. Network Topology

Table 7: Informations of case 7

Case no.	The values of N and L	Capacity values of each link	The corresponding Probabilities
7	N =6 L =8	120,60,120,120, 60,60,30,30	0.90, 0.90, 0.90, 0.90, 0.90, 0.90, 0.90 ,0.90

Table 8. The Results of case 7

	Case Studied	Generation number	ADPT	The proposed chromosome X
The program p1 start on node 1 and need F1, F2, and F3	**Case 7**	**7**	**0.9882**	001000010000001000 010000100000100000
The program p1 start on node 3 and need F1, F2, and F3	**Case 7**	**7**	**0.9961**	001000010000001000 010000100000100000

6. Discussion and Comparison

In comparison with the algorithm presented in [9], the algorithm in [9] used a genetic algorithm to solve both the file and capacity allocation problems. The algorithm applied on the network example studied in 5.4 and the ADPT was 0.9903 in the case of the program P start on node 3 and need F1, F2, and F3. In this paper ADPT is 0.9961 when applying the presented algorithm in this paper. So, the algorithm not only obtained the better solutions but also applied on networks that have large nuber of nodes.

7. Conclusion

In this paper, we presented a genetic algorithm to solve the file allocation problem. The proposed genetic algorithm is used to determine the optimal file allocation on the Distributed Computing Systems(DCS) that maximizes the ADPT with the constraint that the total number of copies of each file on a DCS must be equal to or less than the specified value.

REFERENCES

[1] Altiparmark Fulya, Dengiz Berna and Smith Alice E.,"Reliability optimization of computer communication networks using genetic algorithms", Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics-Intelligent Systems For Humans In A Cyberworld, SMC'98, Hyatt Regency La Jolla, San Diego, California, USA, October 11-14, pp. 4676-4681, 1998.

[2] Coit David W. and Smith Alice E., "Use of a genetic algorithm to optimize a combinatorial reliability design problem", Proceeding of the Third IIE Research Conference, 467-472, 1994.

[3] Coit David W. and Smith Alice E., "Penalty guided genetic search for reliability design optimization", Accepted to Computers and Industrial Engineering, Special Issue on Genetic Algorithms Vol. 30(4): 1996.

[4] Dengiz Berna, Altiparmark Fulya and Smith Alice E., "A genetic algorithm approach to optimal topological design of all terminal networks", Intelligent Engineering Systems Through Artificial Neural Network, Vol.5, pp. 405-410, 1995.

[5] Dengiz Berna, Altiparmak Fulya, Smith Alice E., "Local search genetic algorithm for optimization of highly reliable communications networks", IEEE Transactions on Evolutionary Computation, Vol. 1, pp. 179-188, 1997.

[6] Dengiz Berna, Altiparmark Fulya and Smith Alice E., "Genetic algorithms design of networks considering all-terminal reliability", The Sixth Industerial Engineering Research Conference Proceedings IERC'97, Miami Beach, Florida, USA, May 17-18, pp. 30-35, 1997.

[7] Ahuja S. P. "Performance based reliability Optimization for computer networks". Engineering the New Century Conference Proceeding-IEEE Southeastcon. IEEE, Piscataway, NJ, USA, 97CB36044, 1997

[8] Kumar et al. "Reliability and Performance Optimization for distributed computing system". Computers and Communications, IEEE, 1998. ISCC '98.Proceedings. Third IEEE Symposium on, 30 Jun-2 Jul 1998

[9] Kumar A. and Ahuja S. P. "Performance & Reliability Oriented CombinedFile, Capacity Allocation on Distributed Systems". Computers and Communications, IEEE, 13th Annual International Phoenix Conference on volume, Issue, 12-15 Apr 1994

[10] Abd El-Aziz et al. "Optimizing the Average Distributed Program Throughput (ADPT) by Using Genetic Algorithms". International Journal of Intelligent Computing and Information Science, Vol. 10(1), pp.1-12, 2010.