



Task Allocation for Minimizing Cost of Distributed Computing Systems Using Genetic Algorithms

Ahmed Younes. Hamed

Department of Computer Science,
Faculty of Science, Sohag University,
Egypt.

Abstract --The problem of the task allocation in distributed computing system is to need to allocate a number of tasks to different processors for execution. In this paper we present a genetic algorithm, considering DCS with heterogeneous processors in order to achieve optimal cost by allocating the tasks to the processors, in such a way that the allocated load on each processor is balanced. The algorithm is based on the execution costs of a task running on different processors and the task communication cost between two tasks to obtain the optimal solution. We assess the proposed algorithm by comparing to a heuristic approach. The experimental results illustrate the effectiveness of this algorithm over conventional algorithms.

Keywords - Distributed computing systems, Genetic Algorithms, Task Allocations, execution costs, Communication costs.

I. INTRODUCTION

A distributed computing system (DCS) consists of a set of multiple processors interconnected by communication links. A very common interesting problem in DCS is the task allocation. This problem deals with finding an optimal allocation of tasks to the processors so that the system cost (i.e. the sum of execution cost and communication cost) is minimized without violating any of the system constraints [1]. In DCS, an allocation policy may be either static or dynamic, depending upon the time at which the allocation decisions are made. In a static task allocation, the information regarding the tasks and processor attributes is assumed to be known in advance, before the execution of the tasks [2]. We shall be considering static task allocation policy in this paper. Task allocation problem is known to be NP-hard problem in complexity, when we required an optimal solution to this problem. The easiest way to finding an optimal solution to this problem is an exhaustive enumerative approach. But it is impractical, because there are n^m ways for allocating m -tasks to n -processors [1].

In the last years, distributed computing systems have become a key platform for the execution of heterogeneous applications. The major problem encountered when programming such a system is the problem of tasks allocation. A best allocation of tasks leads to a best balancing of the system, [21]

Many research efforts on the task allocation problem have been identified in the past with the main concern on the performance measures such as minimizing the total sum of execution and communication costs [1, 3, 4] or minimizing the program turnaround time [5, 6], the maximization of the system reliability [7-14].

The configuration of a distributed computing system (DCS) involves a set of cooperating processors communicating over the communication links, [22]. The author presents a hybrid particle swarm optimization to deal with the task allocation problem in which multiple objectives are considered and a set of resource constraints are imposed.

A large number of techniques to task allocation in DCSs have been reported in [1-3], [8-14]. They can be broadly classified into three categories: graph theoretic technique [8, 9], integer programming technique [18-19] and heuristic technique [1-3], [15-16]. Graph theoretic and integer programming techniques yields an optimal solution at all the times. But these techniques are restricted to the small size problems. If the problem size is very large, it is necessary to use the heuristic technique to get near optimal solutions. The choice of a particular technique depends on the structure of the problem [10].

In this paper, we present a genetic algorithm for task allocation that will find a near optimal solution to the problem. The proposed algorithm tries to minimize the processor execution cost and inter processor communication. The approach will minimize the system cost more than other approaches.

The rest of this paper is organized as follows: section 2 describes the task allocation problem for minimizing the overall system cost; section 3 shows the techniques of the proposed genetic algorithm; section 4 presents the steps of the proposed algorithm; section 5 gives an implementation of the proposed algorithm. In the last section concludes the paper.

Notations

P : the set of processors in DCS.

n : the number of processors.

m: the number of tasks forming a program.

t_i : i^{th} task of the given program.

P_k : k^{th} processor in P.

x_{ik} : the decision variable such that $x_{ik} = 1$, if i^{th} task is allocated to k^{th} processor, $x_{ik} = 0$, otherwise.
 ec_{ik} : incurred execution cost (EC), if i^{th} task is executed on k^{th} processor.
 cc_{ij} : incurred inter task communication cost between task t^i and t^j , if they are executed on separate processors.
 ECM : execution cost matrix.
 ITCCM : inter task communication cost.
 gen: the generation counter.
 Pm: the GA mutation rate.
 Pc : the GA crossover rate.
 maxgen: the required number of generations
 Pop_size: the population size.

Assumptions

To allocate the tasks of a parallel program to processors in DCS, we have been made the following assumptions:

- The processors involved in the DCS are heterogeneous and do not have any particular interconnection structure.
- The parallel program is assumed to be the collection of m - tasks that are free in general, which are to be executed on a set of n - processors having different processor attributes.
- Once the tasks are allocated to the processors they reside on those processors until the execution of the program is completed. Whenever a group of tasks is assigned to the processor, the inter task communication cost (ITCC) between them is zero.
- It is also assumed that the number of tasks to be allocated is more than the number of processors ($m \gg n$) as in real life situation.

II. The PROBLEM DESCRIPTION

The problem in this paper is concerned with an optimal allocation of the tasks of a parallel application on to the processors in DCS. An optimal allocation is one that minimizes the system cost function subject to the system constraints. In this paper, we have considered a distributed computing system made up by two sets, $P = \{P_1, P_2, \dots, P_n\}$ of heterogeneous processors, interconnected by communication links and $T = \{t_1, t_2, \dots, t_m\}$ of program tasks, which collectively form a common goal [2].

The execution costs of a task running on different processors are different and it is given in the form of a matrix of order $m \times n$, named as execution cost matrix ECM. Similarly, the inter task communication cost between two tasks is given in the form of a symmetric matrix named as inter task communication cost matrix ITCCM, of order $m \times m$, [20].

A. Execution Cost (EC)

The execution cost ec_{ik} of a task t_i , running on a processor P_k is the amount of the total cost needed for the execution of t_i on that processor during the execution process. If a task is not executable on a particular processor, the corresponding execution cost is taken to be infinite (∞).

B. Communication Cost (CC)

The communication cost cc_{ij} ; incurred due to the inter task communication is the amount of total cost needed for exchanging data between t_i and t_j residing at separate processor during the execution process. If two tasks executed on the same processor then $cc_{ij} = 0$.

C. The System Cost

Given a task allocation $X = \{x_{ik}\}$, $i=1, 2, \dots, m$, $k=1, 2, 3, \dots, n$, the execution cost of all processors can be computed by the following equation:

$$PEC(X) = \sum_{k=1}^n \sum_{i=1}^m ec_{ik} x_{ik} \quad (1)$$

The inter processor communication cost for all processors can be computed as follows:

$$IPEC(X) = \sum_{k=1}^n \sum_{i=1}^m \sum_{j > i, j \neq k} ec_{ik} x_{ik} \cdot x_{jb} \quad (2)$$

The system cost which is defined as the sum of the execution and communication costs is computed as follows:

$$C(X) = PEC(X) + IPEC(X) \quad (3)$$

The purpose of defining the above function is to allocate each of the m tasks to one of the n processors such that the overall system cost is minimized.

That is $Min \{C(X) = PEC(X) + IPEC(X)\}$

III. The PROPOSED GENETIC ALGORITHM (GA)

To solve the problem of task allocation in DCS via GAs, it is necessary to find a mapping of a potential candidate for a solution onto a sequence of binary digits, the so called chromosome. In the proposed genetic algorithm, we consider the four components: (1) an encoding method that is a genetic representation (genotype) of solutions to the program. (2) A way to create an initial population of chromosomes, (3) the objective function (4) the genetic operators (crossover and mutation) that alter the genetic composition of offspring during reproduction.

A. Encoding Method

In our case, however, it is more efficient to represent chromosomes as strings of integers. The length of the chromosomes is given by the number of tasks that should be allocated. Every gene in the chromosome represents the processor where the task is running on. Fig. 1 gives an exemplary mapping of n tasks on m processors.

t_1	t_2	t_3	t_4	t_5	t_6	t_m
P_1	P_2	P_1	P_n	P_1	P_2	P_n

Fig. 1 The task allocation in the form of chromosome

B. Initial Population

The initial population is generated according to the following steps:

- A chromosome x in the initial population can be generated as shown in Fig 1.
- The chromosome must be contain only m none zero element
- The chromosome must be containing all numbers of the processors as shown in Fig. 1.
- Repeat steps 1 to 4 to generate pop_size number of chromosomes.

C. The Objective Function

That is find a task allocation X such that the overall system cost is minimized

$$Min \{C(X) = PEC(X) + IPEC(X)\}$$

D. The Genetic Operations

1) The Crossover Operation

The crossover operation is used to breed a child from two parents by one cut point. The crossover operation will perform if the crossover ratio ($P_c \geq 0.95$) is verified. The cut point is selected randomly. The crossover operation is performed as follows:

- Select two chromosomes randomly from the current population.
- Randomly select the cut point
- Fill the components of the chromosome
 - a. By taking the components of the first chromosome (from the first gene to the cut point) and fill up to the child.
 - b. Also, tacking the components of the second chromosome (from the cut point+1 to the last gene) and fill up to the child.

The offspring generated by crossover operation is shown in Fig. 2.

				↓ Cut point					
Parent	2	2	1	3	4	3	2	1	4
Child	2	2	1	3	3	3	4	2	1
Parent	1	1	2	1	3	3	4	2	1

Fig. 2 The crossover operation

2) The Mutation Operation

The mutation operation is performed on bit-by-bit basis. In the proposed approach, the mutation operation will perform if the mutation ratio (Pm) is verified. The mutation ratio, Pm in this approach will be 0.2 and is estimated randomly. The point to be mutated is selected randomly. The offspring generated by mutation operation is shown in Fig. 3.

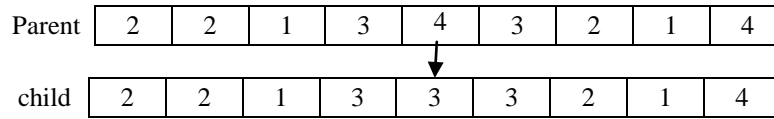


Fig. 3 The mutation operation.

IV. The PROPOSED ALGORITHM

The following algorithm explains how we can use the above assumptions and proposed functions to find a task allocation X such that the overall system cost is minimized.

Algorithm 1: Genetic algorithm for minimizing cost

1. Input : Set the parameters: pop_size, maxgen, Pm, Pc ,gen = 0, GN.
 2. Steps:
 3. Generate the initial population as in section 7.1.
 4. gen←1.
 5. While (gen <= maxgen) do
 6. P ← 1
 7. While (p <= pop_size) do
 8. Genetic operations
 - Select two chromosomes from the parent population randomly
 - Apply crossover according to pc (pc >=0.9).
 - Mutate the new child according to Pm parameter (pm<=0.2).
 9. Compute the total cost of the new child C(x) according to Eq. 3.
 10. Save this child as a candidate solution.
 11. P ← p+1.
 12. End do
 13. Set gen =gen + 1
 14. End do
 15. Compare among all solutions to obtain the best solution
-

V. CASE STUDY

In this section we show the effectiveness of the above algorithm by applying it on two examples as follows:

Example 1:

We consider a typical program made up by 9- executable tasks {t1, t2, t3, t4, t5, t6, t7, t8, t9} to be executed on the DCS having three processors {P1, P2, P3}. We have taken the execution cost of each task on different processors and inter task communication cost between the tasks in the form of matrices ECM and ITCCM respectively. Both the matrices have been given in Table 1 and Table 2 respectively, [20].

The experimental environment is a 2.93 GHz PC with 4 GB RAM. The parameters setting in this algorithm are: pop_size = 20, Pm<= 0.1, Pc=0.95, maxgen =5.

Table 1: Execution Cost Matrix

Processor	P ₁	P ₂	P ₃
Task			
t ₁	174	176	110
t ₂	95	15	134
t ₃	196	79	156
t ₄	148	215	143
t ₅	44	234	122
t ₆	241	225	27

t ₇	12	28	192
t ₈	215	13	122
t ₉	211	11	208

Table 2: The Inter Task Communication Cost Matrix

Tasks	t ₁	t ₂	t ₃	t ₄	t ₅	t ₆	t ₇	t ₈	t ₉
t ₁	0	8	10	4	0	3	4	0	0
t ₂	8	0	7	0	0	0	0	3	0
t ₃	10	7	0	1	0	0	0	0	0
t ₄	4	0	1	0	6	0	0	8	0
t ₅	0	0	0	6	0	0	0	12	0
t ₆	3	0	0	0	0	0	0	0	12
t ₇	4	0	0	0	0	0	0	3	10
t ₈	0	3	0	8	12	0	3	0	5
t ₉	0	0	0	0	0	12	10	5	0

The following chromosome shows the optimal solution of the above example.

t ₁	t ₂	t ₃	t ₄	t ₅	t ₆	t ₇	t ₈	t ₉
3	2	2	1	3	3	1	2	2

Fig. 4 The optimal solution of the example 1

By another way, Table 3 shows an optimal allocation of tasks to processors in DCS.

Table 3: An Optimal Allocation Of Tasks Processors in DCS.

Optimal allocation		System's Cost
Tasks	Processors	
t ₄ , t ₇	P ₁	459
t ₂ , t ₃ , t ₈ , t ₉	P ₂	
t ₁ , t ₅ , t ₆	P ₃	

Table 4 shows the comparison between the proposed genetic algorithm and the algorithm of [20].

Table 4: Comparison Between The Proposed Genetic Algorithm and The Algorithm of, Yadav, [20]

Proposed Algorithm			P. K., Yadav algorithm, [20]		
Tasks	Processors	System's Cost	Tasks	Processors	System's Cost
t ₄ , t ₇	P ₁	459	t ₅ , t ₇	P ₁	528
t ₂ , t ₃ , t ₈ , t ₉	P ₂		t ₂ , t ₃ , t ₈ , t ₉	P ₂	
t ₁ , t ₅ , t ₆	P ₃		t ₁ , t ₄ , t ₆	P ₃	

Example 2:

In this example, we have consider a DCS consists of three processors $P = \{P_1, P_2, P_3\}$ and a typical program made up by 4- executable tasks $T = \{t_1, t_2, t_3, t_4\}$, [19]. Table 5 and Table 6 shows, the execution cost of each task on processors and inter task communication cost between the tasks in the form of matrices ECM and ITCCM respectively.

Table 5: Execution Cost Matrix

Processor	P ₁	P ₂	P ₃
task			
t ₁	9	2	6
t ₂	3	8	7
t ₃	7	10	3
t ₄	3	4	9

Table 6: The Inter Task Communication Cost Matrix

Tasks	t ₁	t ₂	t ₃	t ₄
t ₁	0	1	4	6
t ₂	1	0	2	0
t ₃	4	2	0	8
t ₄	6	0	8	0

The following chromosome shows the optimal solution of the above example.

t ₁	t ₂	t ₃	t ₄
2	1	3	3

Fig. 5 The optimal solution of the example 2

The results obtained with the proposed algorithm, the algorithm presented in [20] and the algorithm presented in [19], for this example has been given below in Table 7.

Table 7: Comparison Between The Proposed Genetic Algorithm, The Algorithm Of, Yadav, [20] and Kumar [19]

Proposed Algorithm			P. K., Yadav algorithm, [20]			H. Kumar algorithm, [19]		
Tasks	Processors	System's Cost	Tasks	Processors	System's Cost	Tasks	Processors	System's Cost
t ₂	P ₁	11	t ₂	P ₁	24	t ₂	P ₁	27
t ₁	P ₂		Nil	P ₂		t _{1, t₄}	P ₂	
t _{3, t₄}	P ₃		t _{1, t₄, t₃}	P ₃		t ₃	P ₃	

Table 8 and Fig. 6 show the effect of varying the crossover probability.

Table 8: Effect Of Crossover On Task Allocation Problem

Pc	Pm	Allocation of 9 tasks on 3 Processor	Cost
----	----	--------------------------------------	------

≥ 0.7	≤ 0.1	3	2	2	1	3	3	1	2	2	459
≥ 0.9	≤ 0.1	2	2	2	3	3	1	1	2	2	598

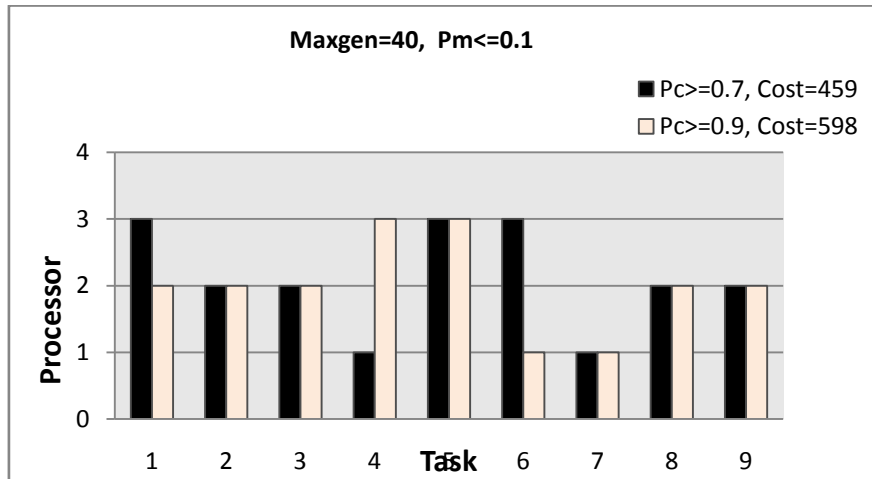


Fig. 6 Study the effect of the crossover probability on task allocation problem

Table 9 and Fig. 7 show the effect of varying the crossover probability

Table 9: Effect Of Mutation On Task Allocation Problem

Pc	Pm	Allocation of 9 tasks on 3 Processor									Cost
≥ 0.1	≤ 0.1	3	2	2	1	3	3	1	2	2	459
≥ 0.05	≤ 0.1	2	3	1	2	1	3	2	3	1	770

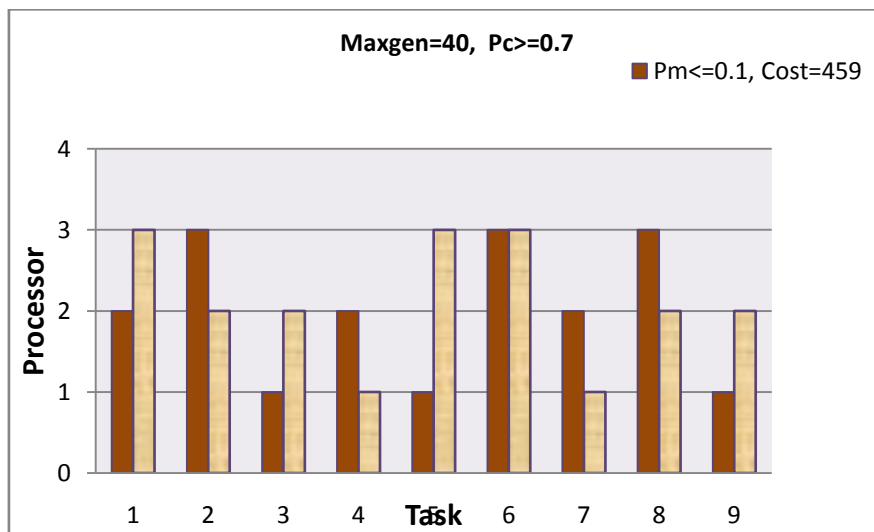


Fig. 7 Study the effect of the mutation probability on task allocation problem

VI. CONCLUSION

In this paper, we have considered the problem of task allocation in DCS. But, task allocation problem is known to be NP- hard problem in complexity, when we required an optimal solution to this problem. Therefore, we have proposed a genetic algorithm, which finds near optimal system cost for the DCS, having arbitrary structure of processors. We have used static task allocation policy to achieve this objective. One of the best options to minimize the system cost is the minimization of IPC. Therefore, the proposed genetic algorithm tries to allocate the tasks to the processors to minimize the system cost. The performance of the proposed algorithm is compared with [20] in example1 and with [20] and [19] in example 2. The run time complexity of the proposed algorithm is $O(mn+n^2m^2)$,

REFERENCES

- [1] Ajith Tom P. and C. Siva Ram Murthy, *An Improved Algorithm For Module Allocation in Distributed Computing Systems*, Journal of Parallel and Distributed Computing Systems,” 42 (1997),pp. 82-90.
- [2] Sagar G. and A.K. Sarje, *Task Allocation Model for Distributed System*, Int. J. Systems Sci. Vol. 22, 9(1991). pp. 1671- 1678.
- [3] Kafil M. and I. Ahmad, *Optimal Task Assignment in Heterogeneous Computing Systems*, 0-8186- 7879- 8/97\$ 10.00 □ 1997 IEEE
- [4] Cheol-Hoon Lee, Dongmyun Lee and Myunghwan Kim, “*Optimal Task Assignment in Linear Array Networks*, IEEE Transactions on Computers, Vol.41, 7(1997).
- [5] Imtiaz Ahmad Muhammad K.Dhodhi and Arif Ghafoor, *Task Assignment in Distributed Computing Systems*, IEEE Concurrency (1995), pp.49-53.
- [6] Elsadek A.A. Elsadek & B. E. Wells, *A heuristic model for task allocation in heterogeneous distributed computing systems*, International journal of computers and there applications, Vol.6, No.1, 1999. Pp. 0-35.
- [7] Sol M.Shatz, Jia-Ping Wang, and Masanori Goto, *Task Allocation For Maximizing Reliability of Distributed Computer Systems*, IEEE Transactions on Computers, Vol.41.9 (1992).
- [8] S.Kartik and C.Siva Ram Murthy, *Task Allocation Algorithms for Maximizing Reliability of Distributed Computing System*, IEEE Transactions on computers, Vol.46,6 (1997).
- [9] D.J. Chen et al., *A Heuristic Algorithm for the Reliability- Oriented File Assignment in a Distributed Computing System*,” Computers Math. Applic. Vol. 29.10(1995), pp. 85- 104,
- [10] Peng-Yeng Yin, Shih-Sheng Yu, Pei-Pei Wang, Yi-Te Wang, *Task Allocation for maximizing Reliability of a Distributed System using Hybrid Particle Swarm Optimization*, The Journal of Systems and Software, 80(2007).pp. 724-735.
- [11] Santhanam Srinivasan and Niraj K.Jha, *Safety and Reliability Drivan Task Allocation in Distributed Systems*, IEEE transactions on Parallel and Distributed Systems, Vol.10. 3(1999).
- [12] D.P.Vidayarthi and A.K.Tripathi, *Maximizing reliability of Distributed Computing System with Task allocation using Simple Genetic Algorithm*, Journal of System Architecture, 47(2001), pp.549-559.
- [13] Pradeep Kumar Yadav, M.P. Singh and Kuldeep Sharma, *Task Allocation Model for Reliability and Cost optimization in Distributed Computing System*, International Journal of modeling, simulation and scientific computations, vol-2, 2(2011), pp. 1-19.
- [14] Qin-Ma Kng, Hong He, Hui- Min Song, Rong Deng, *Task allocation for maximizing Reliability of distributed Computing System using Honeybee mating Optimization*, The Journal of Systems and software, Vol.83, 2010.
- [15] Lo V.M., *Heuristic Algorithms for Task assignment in distributed systems*, IEEE Transactions on computers, Vol.37. No. 11, pp. 1384- 1397, November 1988.
- [16] K. Kfe, “*Heuristic Models of Task Assignment Scheduling in Distributed Systems*,” Computer, Vol. 15, pp. 50- 56, June 1982.
- [17] W.W.Chu, Leslie J.Holloway, Min-Tsung Lan, and Kemal Kfe, *Task Allocation in Distributed Data Processing*, IEEE Concurrency, November 1980.pp.57-69.
- [18] C. C. Shen and W.H. Tasi, *A Graph Matching Approach to Optimal Task Assignment in Distributed Computing Systems Using a Minimax Criterion*, IEEE Transactions on Computers, Vol. C- 34, 3(1985).
- [19] H. Kumar et al., *A Task Allocation Model for Distributed Data Network*, Journal of Mathematical Sciences, Vol.1, 4(2006).pp.379-392.
- [20] Yadav P. L., M. P. Singh, and Kuldeep Shama, *An Optimal Task Allocation Model for System Cost analysis in Hetrogeneous Distributed Computing Systems: A Heuristic Approach*, International Journal of Computer Applications, Vol. 28, No. 4, August 2011.
- [21] Mostaph Zbakh, Mohamed Dafir El Kettani, *A Task Allocation Algorithm For Distributed Systems*, Journal of Theoretical and Applied Information Technology, Vol. 33 No. 1, 2011.
- [22] Peng-Yeng Yin, Shih-Sheng Yu, Pei-Pei Wang, Yi-Te Wang, *Multi-objective task allocation in distributed computing systems by hybrid particle swarm optimization*, Applied Mathematics and computation, pp.407-420, 2007.