A hybrid algorithm for grid task scheduling problem

Atefeh Shahkolaei¹, Hamid Jazayeriy²

¹Department of computer engineering, Islamic Azad University, Science and Research Ayatollah Amoli branch, Amol, Iran
²Babol Noshirvani University of Technology, Iran

Abstract

At the beginning the Grid computing and Grid technologies was used for scientific and technical jobs where computers were distributed geographically, connected through internet, are used in order to invent virtual supercomputers with high computing competency. The goal of inventing such a virtual supercomputer was solving complex scientific problems in less time than before. The goal of task scheduling in Grid systems is decreasing the total completion time of the all tasks with providing a timetable to perform all existing tasks. The problem of task scheduling in grid systems has large solution space and is considered as a kind of NP-Hard problems. In this paper a new method for solving task scheduling in Grid computing system is proposed by combining Genetic Algorithm with Tabu Search Algorithm. The goal of proposed algorithm is to combining the global search abilities of Genetic Algorithm with local search of Tabu Search, and creating a stable algorithm that highly guarantees the accessing of the global optima. The proposed algorithm is compared with other heuristic algorithms and experimental results showed that the proposed algorithm has high efficiency for solving task scheduling in Grid computing.

Keywords: Computational Grids, ETC Matrix, Genetic Algorithm, Tabu Search.

1. INTRODUCTION

Grid computing and its technologies creates at first for technical and scientific activities in computers where geographically distributed, connected through internet, and used to create virtual supercomputers with high computing abilities. The purpose of creating these virtual supercomputers is solving complex scientific problems in less time than before. Afterwards, Grid Computing was applied for helping and reaching improvements in meteorology, physics, medicine and other computing-intensive scientific fields. Examples of such works in large scale including optimization and data-intensive computing are studied in [1]–[7].

The current definition of Grid was introduced for first time by [8] in 1998 as following: "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."

Task scheduling in Grid Computing contains a set of tasks that should be run on a limited set of resources. The purpose of task scheduling in Grid systems is offering a time table for executing all tasks and decreasing the execution time of all tasks. The problem of task scheduling in grid systems has large solution space and is considered as a kind of NP-Hard problems [9], [10]. In the past few years, many researches were performed for solving Grid Task Scheduling problems based on GA [11]–[14] and TS [11], [16]–[18]. A hybrid GA (TS) algorithm for the problem of independent scheduling in computational grids presented in [15]. In their proposed algorithm the hybridization follows a low level approach in which GA is the main flow and TS is subordinated to it. The experimental evaluation showed that GA (TS) outperforms both GA and TS for small and medium size grid scenarios but achieves worse value for large size scenario. In this paper, the proposed algorithm utilize GA and TS together to maintain the global search abilities of Genetic Algorithm and local search of Tabu Search. In GA(TS) algorithm [15] the TS algorithm used instead of mutation operation, but in this paper, in the proposed algorithm the TS algorithm execute separately after Crossover and Mutation operations, to optimize some percent of individuals in GA population.

2. GRID TASK SCHEDULING

The Grid Task Scheduling is including of a set of tasks that should run on limited set of resources. The purpose of task scheduling in Grid systems is offering a time table for executing all tasks, so that the execution time of all tasks minimized and the computational load of all system divided between all resources equally.

For simulating grid environment, a static environment used that contained $n \times m$ matrix called ETC (expected time to compute); the model offered by [11] is used. Any row in ETC matrix shows the estimated execution times for a given task on each machine. Thus, for an arbitrary task $i$ and an arbitrary machine $j$, the entry $ETC(i,j)$ is the estimated execution time of running task $i$ on machine $j$.

In simulated model, the purpose of heuristic algorithm is minimizing the metatask execution time that is called $makespan$. Consider that the earliest time machine $j$ can complete the execution of all the tasks that have previously been assigned to it, called Machine availability time. Thus, the completion time for a new task on machine will define by relation (1):
\[ ct(t_i \cdot m_j) = mat(m_j) + ETC(t_i \cdot m_j) \]  

(1)

Where \( mat \) is machine availability time for machine \( m_j \) and \( ETC \) is the completion time for task \( t_i \) on machine \( m_j \). With having the amount of \( ct(t_i \cdot m_j) \) for each task, the metatask execution time (makespan) will calculate by relations (2).

\[ \text{makespan} = \max \left( \sum_{i=0}^{t} \sum_{j=0}^{m} ct(t_i \cdot m_j) \right) \]  

(2)

3. PROPOSED ALGORITHM

In this paper a new method for solving task scheduling in Grid computing system is proposed by combining Genetic Algorithm with Tabu Search Algorithm that will describe in this section. For displaying chromosomes a \( 1 \times t \) vector is used [15], where \( t \) is equal by the number of all available tasks for processing. Each chromosome in the proposed algorithm shows a possible mapping for all tasks of users. In the proposed algorithm parents are selected using the ranking method, and the chosen operator for generating new children is single point crossover. After generating new children the mutation operator is used for making diversity in children population. Swap and Transfer are two types of methods that selected for mutation. The main purpose of these mutations is changing tasks between different machines to process the tasks with more suitable machines.

In Population Optimizations step, some percentages of available chromosomes optimized by Tabu Search algorithm and added to the genetic population as new children. Tabu Search algorithm receive one genetic chromosome as base solution, and try to optimize the chromosome with its operators in some iterations. In the proposed Tabu Search algorithm, at first a solution will create in neighborhood of current solution. Then, the index of tasks that has a role in generating the new neighbor will locate in the tabu list, to forbid these tasks to participate in generating new neighbors. In following, if the generated neighbor be better than the best solution produced so far, the generated neighbor will select as the best solution and the structures of memory will updated. This cycle of generating neighborhood will repeat in numbers that was determined before, and finally the Tabu Search algorithm will return its best solution to Genetic Algorithm.

The proposed algorithm get terminated if no improvement observed in chromosomes in a predefine number of generation. Figure 1 shows the flowchart of the proposed algorithm.

![Flowchart of the proposed algorithm](image)

**Figure 1:** Flowchart of the proposed algorithm

4. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, it is compared with other heuristic algorithms in [11]. The proposed algorithm was implemented in C# 2010 and the tests were made on a Pentium IV 3.00GHz processor with 1.00 GB of RAM. For testing the proposed algorithm in simulated environment, 9 test data set based on different type of ETC matrices are designed, each of them consisting of 100 instances. They are labeled as Test_X_YZ_N where \( X \) is the type of ETC matrices (\( C \) - consistent, \( I \) – inconsistent and \( p \) means partially consistent), \( Y \) and \( Z \) indicate the task and machine heterogeneity. Their value could be \( H \) for maximum heterogeneity and \( L \) for the minimum heterogeneity. \( N \) is used to number the corresponding test. For example, if a test labeled as Test_C_HH_10, it means that, in this test a consistent matrices is used with maximum heterogeneity for tasks and machines. In test of the proposed algorithm, all test data sets are composed from 512 jobs and 16 machines. Various GA and TS options used are:

- Population size = 100
- Crossover probability = 0.75
Mutation probability = 0.25
Tabu Search neighborhood size = 20
tabu_tenure = 5
The proposed algorithm for any test performed 100 times and the average of fitness values is calculated by the proposed algorithm. Figure 2 shows the output of Test_C_HH_1 in Gant chart. This Chart is consisting of 16 rows and 512 colorful rectangles that any one of them is showing one of the 16 machines and 512 works in the simulated system.

![Figure 2: Gant chart of Test_C_HH_1](image)

In Figure 3 the chart of movements of best chromosome in genetic population is shown, that represents high speed of algorithm in achieving to optimal solution. In this chart tilt of curve will decrease during the time, and finally it’s stopped in local optima.

![Figure 3: Fitness of the best chromosome in Genetic Population](image)

Table 1 shows the result of several tests that obtained to optimal solutions. In Table 1, the first column presents test name, and other results such as the best iteration that optimal solution obtained, the total running time for the proposed algorithm, the time that optimal solution obtained, and fitness of the optimal solution are presented in other columns respectively.

<table>
<thead>
<tr>
<th>Test Data</th>
<th>Best Iteration</th>
<th>Run Time</th>
<th>Best time</th>
<th>Best fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test_C_HH_1</td>
<td>40115</td>
<td>785</td>
<td>767</td>
<td>7234264</td>
</tr>
<tr>
<td>Test_I_HH_1</td>
<td>69897</td>
<td>1358</td>
<td>1339</td>
<td>3103133</td>
</tr>
</tbody>
</table>
Figure 4 shows the chart of average running time for the proposed algorithm and other heuristic algorithms [11]. In Figure 4 the average running time of proposed algorithm shown with red color.

![Average Run Time of Algorithms](image)

**Figure 4:** Average run time of the proposed and other heuristic algorithms

Figure 5 shows the comparison of makespan obtained by the proposed algorithm and other heuristic algorithms [11]. As you can see in Figure 5 the proposed algorithm obtained better optimal solution in comparison with other heuristic algorithms.

![Comparison of Makespan](image)

**Figure 5:** Comparison of makespan versus other algorithms in the literature

5. CONCLUSION
In this paper a new method for solving task scheduling in Grid computing system is proposed by combining Genetic Algorithm with Tabu Search Algorithm. The goal of proposed algorithm is to minimizing the execution time of all tasks and dividing the computational load of the whole system between all resources equally. To evaluate the performance of the proposed algorithm, it is compared with other heuristic algorithms, and experimental results showed that the proposed algorithm has high performance for solving task scheduling in Grid computing. For future works, the effect of selecting other type of operations and parameters on the quality of solving problem in proposed algorithm could be examine.

References


AUTHOR

AtenaShahkolaei received her B.S. in IT Engineering from University of Ghaemshahr, Iran and received her M.S degree in Software Engineering from Science and Research University of Amol, Iran. She is a reader in Grid Computing, Cloud Computing and she has done a lot of researches about Grid Computing in recent years.

Hamid Jazayeriy is a lecturer at the faculty of electrical and computer engineering, NIT, IRAN. He received his B.Sc. of computer engineering from university of Tehran in 1996 and his master of software engineering from university of Isfahan in 2000. In 2011, he received his Ph.D. from university Putra Malaysia, UPM. He is a reader in multi-agent systems, swarm intelligence and graph theory.