

Evaluation and Making a Tradeoff between Load Balancing and Reliability in Grid Services using Formal Models

Zeinab Sayari

Science and Research Damavand Branch, Islamic Azad University, Tehran, Iran

Ali Harounabadi

Assistant Professor
Central Tehran Branch, Islamic Azad University, Tehran, Iran

ABSTRACT

Computational grid is a new technology aimed to facilitate sharing resources and easing common cooperation in vast areas. Resource allocation and task scheduling for achieving requested quality of service is one of the most essential topics in grid environments. Reliability and load balancing are two important parameters for grid services scheduling. Establishing high reliability and reducing task execution time in grid environment can lead to load balance reduction in grid systems. The purpose of this paper is to establish a balance between these two parameters in grid environment. At first, using colored petri net, task execution in grid environment is modeled and then based on that, the first parameter, reliability is computed. Then, using resource workload variance, resource load balancing in various task scheduling types are computed. Finally, after computing these two parameters, using multiple criteria decision making, the resource allocation type with balance between reliability and load is selected.

General Terms

Grid, Load balancing, reliability, petri net.

Keywords

Computational grid, resource management system, reliability, load balancing, colored petri nets, multiple criteria decision making.

1. INTRODUCTION

Computational grid is a hardware and software infrastructure that provides reliable, stable, vast and inexpensive access to computational resources [1]. Nowadays, grid performance is a problem. Grid scheduling aim is optimal resource allocation. Optimal subtask submission and scheduling among resources is indispensable for maximizing resource usage in grid environments. Quality of service will be considered for scheduling. It is often strived to get the maximum quality of service during subtask scheduling. Reliability and load balancing are two parameters of Quality of Service that optimizing these two lead to optimize grid performance. The purpose of this paper is to model task scheduling with the reliability in mind. Then, using variance concept and the fact that more workload variance will result to more optimal load balancing in system, load balancing factor is computed and then, using multiple attribute decision making method that is used for finding the optimal alternative and specifically, using TOPSIS¹ method, a proportional balance between reliability and workload is established. Despite classic optimization models which have a single evaluation criteria (with one

target function), in MADM², multiple evaluation criteria is used and this is the reason behind using MADM in this paper. In the second part of this paper, background is discussed. In the third part, related works in the current field is presented. In the fourth part, suggested model will be elaborated and in the fifth part, an example of the model is presented and final conclusion will come in the sixth part.

2. BACKGROUND

2.1 Grid Environment

Grid is a kind of distributed parallel system that provides the possibility of dynamic and runtime sharing of heterogeneous, autonomous geographically-distributed resources with the consideration of performance, accessibility, cost and quality of service. Task scheduling is one of the important aspects of grid computing that boosts quality of service. RMS is responsible for task scheduling and execution in grind environment. It takes subtasks and distributes them among resources. Computational grid is one of subtypes. Grid resources are connected using communication links. Link topology is determined by the connectivity structure between computers. Among well-known topologies in grid environment, star, ring, graph and compound models can be named [2]. In star topology, RMS is placed in the center of the system and all grid resources are connected to it using communication links. After submission of task to RMS by grid users, RMS is responsible for distributing subtasks among resources and taking results from them.

2.2 Colored Petri Net

Petri nets were introduced in 1962 by Carl Adam Petri [3]. Petri nets are a means for modeling and are used for modeling concurrent, parallel, dynamic and separate systems. Colored petri net supports color tokens. Colors are equivalent to data-types in programming languages and color tokens represent valid values for a data-type, or in other words, are an instance of a data-type. There are various tools for simulating and analyzing colored petri nets. The most notable and applicable one is CPN Tools [4]. CPN is an applied well-known utility for modeling, analyzing and proving CPNs.

2.3 Multiple Criteria Decision Making

Main emphasis in classical optimization models has been on having a single evaluation criteria (or target function) but researchers have moved towards multi-criteria models (MCDM) for complex decision making in recent decades. In these decision makings, multiple criteria may be used instead of single criteria. These models are categorized into two main types: multiple objective decision making models and

¹ Technique for order-Preference by Similarity to ideal Solution

² Multiple Attribute Decision Making

multiple attribute decision making models. MODM models are used for design while MADM models are used for finding optimal alternative. MADM model is formulated as decision matrix of table 1.

Table 1. Multilpe attribute matrix (D matrix)

option \ attribute	X ₁	...	X _n
A ₁	r ₁₁	...	r _{1n}
...
A _m	r _{m1}		r _{mn}

A₁, A₂, ..., A_m in decision matrix D, in order, shows m predetermined alternatives (such as buying airplanes from specific vendors). x₁, x₂, ..., x_n shows n attributes like cost, capacity, comfort ability, fame, ...for evaluating the suitability of each alternative and finally, r_{ij} elements indicate specific values from j-th attribute for i-th alternative. It is clear that x_{ij} tributes may be quantitative (such as cost) or qualitative (like comfort). One of MADM methods is TOPSIS. In this method, in addition to considering the distance of an alternative A_i from ideal point, its distance from negative ideal point is considered too. Meaning that selected alternative must have the minimum distance from ideal solution and at the same time, has the longest distance from the negative ideal solution.

3. RELATED WORKS

Up to this point, lots of work for enhancing performance in grid environment has been done. Parsa and Entezari Maleki [5] have suggested an algorithm based on queue networks that give a lower average response time while doing balanced distribution of subtasks among grid resources.

Entezari Maleki and Azgomi [6] have presented a high level model for defining subtask distribution method in grid in which reliability in grid services using color Petri networks studied. L.Li et al in [7] assumed a scheduling system based on multi-queue multi-server and accomplished a modeling and analytical methods based on stochastic Petri net. A.Afzal et al [3] suggested an algorithm to reduce program execution cost using queue theory. In 2010, Omaraa presented CPGA and TDGA[8] algorithms by making a few modifications in standard genetic algorithm to enhance performance. These modifications consist of adding two greedy algorithms to generic algorithm to enable it to reduce task start time and as a consequence, reduce makespan. Khanli in 2010 presented RFOH [9] for task scheduling with fault tolerance in computational grid. Maleki [10] in 2010 suggested an algorithm for minimizing makespan. Sadi et al [11] presented and algorithm for scheduling independent subtasks in computational grid. S.Suryadevera et al [12] presented an ant colony optimization algorithm for achieving load balancing in computational grid environment. In most methods above, only one factor of quality of service has been considered and neither of the above methods considered establishing proportional balance between criteria using multiple criteria decision making.

4. PROPOSED METHOD

In proposed method, at first, reliability for stochastic scheduling based on algorithm presented in [6] is computed.

Then other scheduling for the same task has been done and again, reliability for each of the above schedules are computed. After studying n combination of resources for a specific task, load balancing for each case is computed and finally, using multiple criteria decision making and specifically TOSIS, alternatives are rated and the most preferable alternative in which reliability and load balancing are in a win-win level are chosen.

You can view the colored petri model in figure 1. First, reliability based on suggested model [6] for all permutation of resources for allocation to a subtask of a specific task is computed. Afterwards, service load balancing for different cases of resource allocation is computed. For achieving maximum load balance, productivity of resources must be the same. For computing load balance value, sum of productivity deviation is computed. So resource productivity must be computed first. Productivity is computed as follows. First, largest execution time is computed:

$$E = \text{MAX}\{T_{ij}, \tau_{ij}\} \quad (1)$$

For computing e, data transfer time from RMS (τ_{ij}) and resource processing time (T_{ij}) are used. In this case, E is computed. If productivity for i-th resource in execution of of j-th subtask is represented with μ_i the following is yielded:

$$\mu_i = \frac{T_{ij}, \tau_{ij}}{e} \quad (2)$$

For computing the productivity value of resource, the same transaction for computing E value above is used. In other words, E is computed first and then, based on above formula, productivity of each resource is computed. Attention must be paid to the point that productivity value based on suggested model for added resources (resources that have a higher execution time compared to selected resource) is zero. For computing deviation, average of total productivity is computed as follows:

$$\mu = \frac{\sum_{i=1}^m \mu_i}{m} \quad (3)$$

For doing this, all values must be added. This is done using V transaction. Then, the following formula is used for computing load balancing:

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (\mu_i - \mu)^2 \quad (4)$$

Finally, for computing system load balance, T_{load-balancing} is used. In other words, all productivity values entered into T_{load-balancing} and then T_{load-balancing} will compute the final system load balance so that if $\sigma^2 = 0$, maximum load balance is gained. After computing reliability and load balance of each of permutations of resources for subtask, multiple criteria decision making is used for selecting the most preferable alternative in which system will be reached to a proportional balance in aspects of reliability and load balance [13].

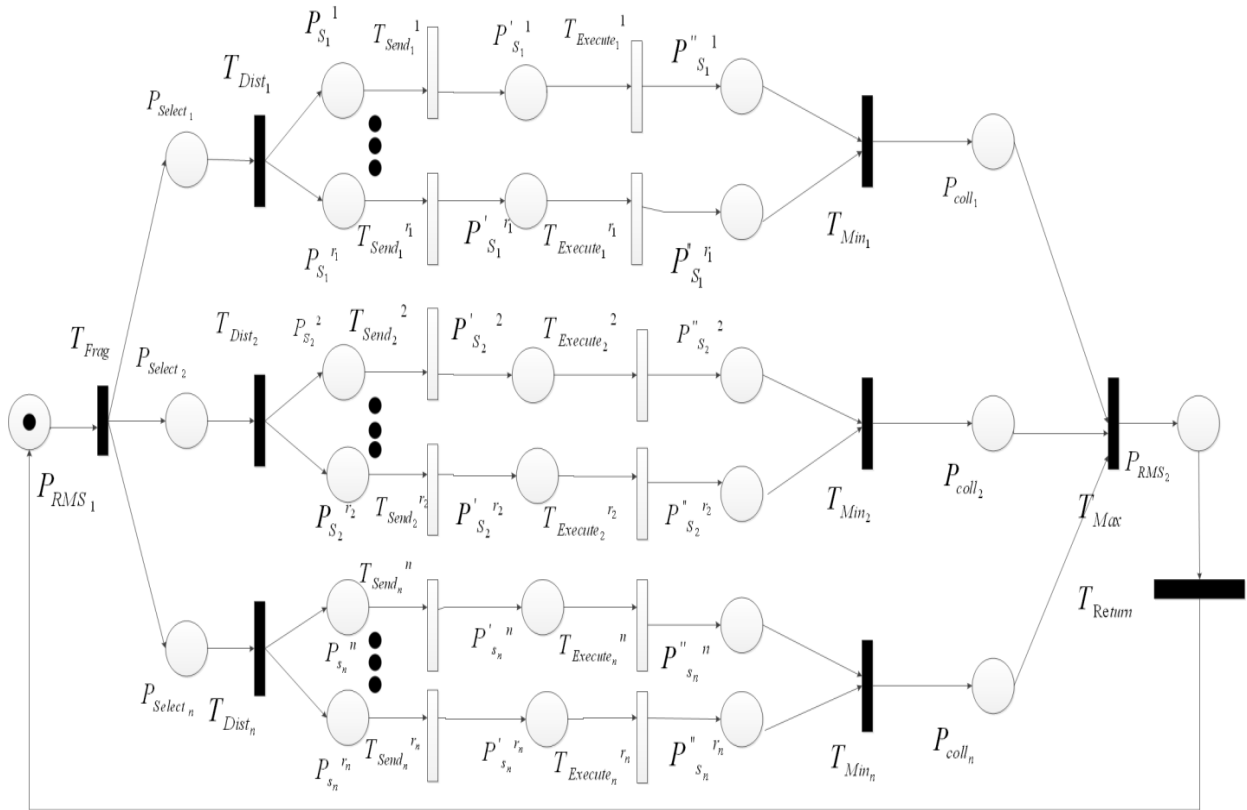


Fig 1. Colored petri net mode

4.1 TOPSIS Algorithm Steps

First, using entropy technique, weights vector (W) for attributes is computed. In this method, selection matrix is computed initially as P_{ij} :

$$P_{ij} = \frac{r_{ij}}{\sum_{i=1}^m r_{ij}}; \quad \forall i, j \quad (5)$$

for E_j from set P_{ij} the following is computed for each attribute:

$$E_j = -k \sum_{i=1}^m [p_{ij} \cdot \ln p_{ij}] \quad \forall j, \quad k = \frac{1}{\ln m} \quad (6)$$

deviation (d_j) for j -th attribute is:

$$d_j = 1 - E_j; \quad \forall j \quad (7)$$

and finally, weights (W_j) for existing attributes is computed:

$$w_j = \frac{d_j}{\sum_{j=1}^n d_j}; \quad \forall j \quad (8)$$

existing decision matrix (matrix D) is converted to a weighted normalization matrix N_D using the following relation:

$$n_{ij} = \frac{r_{ij}}{\sqrt{\sum_{i=1}^m r_{ij}^2}} \quad (9)$$

creating weighted normalization matrix is defined by assuming vector W as input for algorithm, that is:

$$W = \{w_1, w_2, \dots, w_n\} \approx$$

Assumed from DM

$$V = N_D \cdot W_{n \times n} = \begin{bmatrix} V_{11} & \dots & V_{1j} & \dots & V_{1n} \\ \vdots & & \vdots & & \vdots \\ V_{m1} & \dots & V_{mj} & \dots & V_{mn} \end{bmatrix} \quad \text{Weighted normalization matrix} \quad (10)$$

and determining ideal solution and negative ideal solution for ideal alternative (A^+) and negative ideal alternative (A^-):

$$A^+ = \{(\max V_{ij} | j \in J), (\min V_{ij} | j \in J') | i = 1, 2, \dots, m\} = \{V_1^+, V_2^+, \dots, V_j^+, \dots, V_n^+\} \quad (11)$$

$$A^- = \{(\min V_{ij} | j \in J), (\max V_{ij} | j \in J') | i = 1, 2, \dots, m\} = \{V_1^-, V_2^-, \dots, V_j^-, \dots, V_n^-\} \quad (12)$$

$d_{i+} = J$'s related to benefit

assume

$J' = \{j = 1, 2, \dots, n | J$'s related to cost}

distance between i -th alternative with ideal alternatives are computed using Euclidean method as follows:

i -th alternative distance from ideal:

$$d_{i+} = \sqrt{\sum_{j=1}^n (V_{ij} - V_j^+)^2}; i = 1, 2, \dots, m \quad (13)$$

i -th alternative distance from negative ideal:

$$d_{i-} = \sqrt{\sum_{j=1}^n (V_{ij} - V_j^-)^2}; i = 1, 2, \dots, m \quad (14)$$

Computation of proportional proximity of A_i to ideal solution is defined as follows:

$$cl_{i+} = \frac{d_{i-}}{(d_{i+} + d_{i-})} ; 0 \leq cl_{i+} \leq 1 ; i = 1, 2, \dots, m \quad (15)$$

And finally, sorting alternatives in descending order of cl_{i+} can rate existing alternatives of the current problem in the order of preference.

5. CASE STUDY

In this section, an example of a sample grid network is given. Modeling and reliability of the example network is evaluated using colored petri net and applying CPN tools. As you see in figure 1, 4 resources are assumed in grid environment that each resource has its own bandwidth, processing speed, failure probability and communication links. Processing speed and bandwidth for communication links of resources are shown in table 2. It is assumed that RMS divided input subtask into two subtasks and submit them to resources stochastically. Data value that must be transferred between RMS and subtask, as well as computational complexity of each subtask can be seen in table 3. Table 4 and 5 accordingly show no-failure probability in resources and communication links.

Table2. Process speed and bandwidth for each resource

	R ₁	R ₂	R ₃	R ₄
Process speed (MIPS)	4	6	3	8
bandwidth(Mbps)	30	55	4	5

Table3. Subtasks Specialty

	S ₁	S ₂
Computational complexity(MI)	60	100
Required Data(Mb)	300	80

Table4. Probability of no failure in resources towards the process (p)

	R ₁	R ₂	R ₃	R ₄
Subtask1(S ₁)	0/95	0/64	0/92	0/60
Subtask2(S ₂)	0/94	0/63	0/87	0/67

Table5. Probability of no fault in communication links during transfer

	R ₁	R ₂	R ₃	R ₄
Subtask1(S ₁)	0/98	0/56	0/88	0/58
Subtask2(S ₂)	0/91	0/60	0/93	0/61

Subtask submitted by RMS is divided into S₁ and S₂ and based on the method specified in [6], subtask are submitted stochastically to resources for execution. After execution, total system reliability and task execution time is computed. For finding the best reliability and execution time, all permutations of resource allocation to subtasks are considered and reliability and execution time for each case is computed. In the current example, there are 6 different permutations of resource allocation that are shown in table 6.

Table 6. Combinations of allocating resources to subtasks

	Allocation of resources to subtasks
Alternative1	Subtask S ₁ , selects R ₁ , R ₂ and Subtask S ₂ , selects R ₃ and R ₄ resources
Alternative 2	Subtask S ₁ , selects R ₁ , R ₃ and Subtask S ₂ , selects R ₂ and R ₄ resources
Alternative 3	Subtask S ₁ , selects R ₁ , R ₄ and Subtask S ₂ , selects R ₂ and R ₃ resources
Alternative 4	Subtask S ₁ , selects R ₂ , R ₃ and Subtask S ₂ , selects R ₁ and R ₄ resources
Alternative 5	Subtask S ₁ , selects R ₂ , R ₄ and Subtask S ₂ , selects R ₁ and R ₃ resources
Alternative 6	Subtask S ₁ , selects R ₃ , R ₄ and Subtask S ₂ , selects R ₁ and R ₂ resources

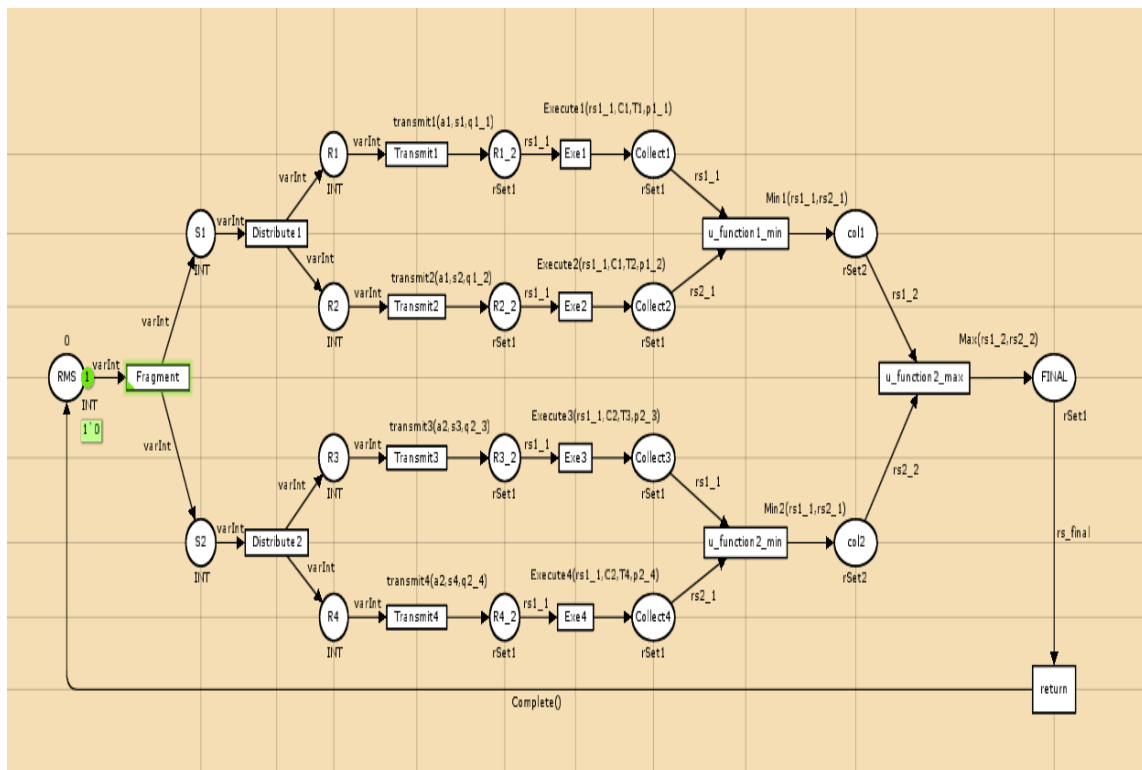


Fig 2: Simulation of Colored petri net model in CPN Tools for grid example environment

Table 7. Execution time, reliability, variance and standard deviation of resources workload in various combinations of allocating resources to subtasks

	Execution time	Reliability	Workload variance	Workload standard deviation
Alternative 1	28	0/391	0/0108	0/104
Alternative 2	25	0/588	0/107	0327
Alternative 3	25	0/820	0/1	0/316
Alternative 4	27	0/751	0/084	0/291
Alternative 5	27	0/497	0/0713	0/267
Alternative 6	67	0/316	0/008	0/090

Table 8. Decision making matrix (D matrix)

$A_i \backslash X_j$	Reliability	Load balancing
Alternative 1 (A_1)	391	104
Alternative 2 (A_2)	588	327
Alternative 3 (A_3)	820	316
Alternative 4 (A_4)	751	291
Alternative 5 (A_5)	497	267
Alternative 6 (A_6)	316	90

Table 9. w_j matrix

	x_1	x_2
w_j	0/37	0/63

Based on proportional attribute importance above, it can be said that proportional importance of load balancing attribute is more than reliability.

And in final step, alternatives are sorted descending in the order of preference. Table 10 shows a comparison between sorting alternatives based on reliability criteria, load balance and finally TOPSIS.

Table 10. Sorting alternatives based on TOPSIS algorithm, reliability load balancing criteria

	TOPSIS	Reliability	Load balancing
Option 1	A_2	A_3	A_6
Option 2	A_3	A_4	A_1
Option 3	A_4	A_2	A_5
Option 4	A_5	A_5	A_4
Option 5	A_1	A_1	A_3
Option 6	A_6	A_6	A_2

A comparison between reliability in normal resource allocation method with TOPSIS method is done in figure 3. It can be seen that alternatives 2, 3 and 4 are the best cases of resource allocation based on TOPSIS. Also, a comparison between load balancing in normal resource allocation with TOPSIS method is done in figure 4 and it can be seen that alternatives 3 and 4 are the best cases of resource allocation. Finally in figure 5, a comparison is made between harmonic mean of reliability and load balancing with TOPSIS and it can be seen that based on harmonic mean of two criteria, alternatives 1,2,3 and 4 are the best cases of resource allocation but based on figure 2, load balancing of alternatives 1 and 2 are very low that results to rejection of alternatives 1 and 2. Although alternative 4 can be a selected alternative, it is not the best option due to low reliability. It can be seen that alternative 3 is in a win-win state based on reliability and load balancing so it can have the best resource allocation.

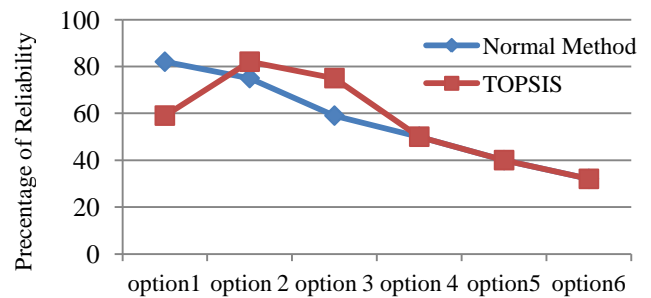


Figure 3. Reliability in normal resource allocation vs TOPSIS method

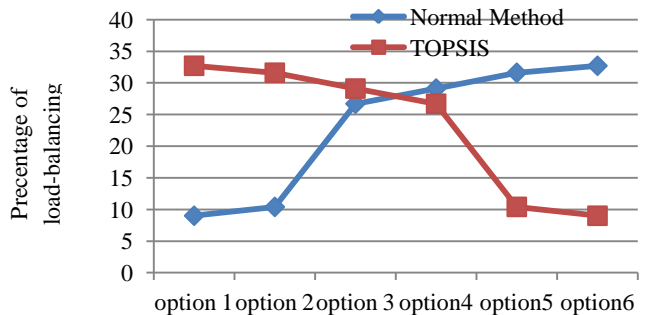


Figure 4. Load balancing in normal resource allocation vs. TOPSIS method

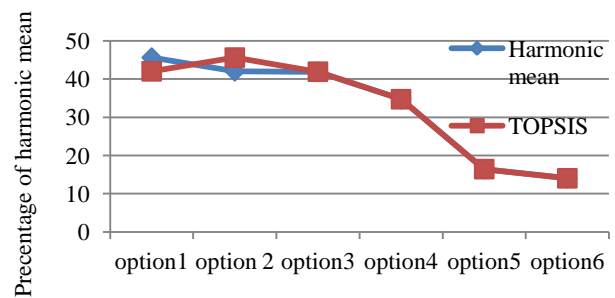


Figure 5. Harmonic mean of Reliability and load balancing in normal resource allocation vs TOPSIS method

6. CONCLUSION

Task scheduling is one of most essential and fundamental topics in grid environments. Task scheduling must be done in a way that maximum quality of service can be attained. In this paper, task execution with considering reliability, execution time and load balance parameters were modeled and the result model was simulated using CPN Tools. Finally, it is attempted to use multiple criteria decision making methods to find a balance between these parameters and achieve a win-win state of resource allocation. Some future works that can be done in this area, are as follows:

1. Extending the proposed model to other topologies of the grid environment such as tree or hierarchical topologies that are more similar to real grid environments.
2. Considering other parameters of quality of service such as cost.
3. Using fuzzy based methods.

7. REFERENCES

- [1] Foster, I., Kesselman, C., Nick, J. and Tuecke, S., 2002, *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*, Computer, Vol. 35, No. 6, pp. 1-4
- [2] W. Hoschek, Peer-to-peer grid databases for web service discovery, in: F. Berman, A.J.G. Hey, G.C. Fox (Eds.), *Grid Computing: Making the Global Infrastructure a Reality*, Wiley, 2003.
- [3] Afzal, A., McGough, A.S., Darlington, J., "Capacity planning and scheduling in Grid computing environment", *Journal of Future Generation Computer Systems* 24 , pp. 404-414, 2008.
- [4] K. Jensen, L.M. Kristensen, L. Wells, "Coloured Petri Nets and CPN Tools for modeling and validation of concurrent systems", *International Journal on Software Tools for Technology Transfer (STTT)* Vol.9., May. 2007, pp. 213-254.
- [5] S. Parsa and R. Entezari-Maleki, "Subtask Scheduling Algorithm in Grid Environment Considering QoS Measures," *Proceedings of 14th Annual National CSI Computer Conference (CSICC'09)*, Amirkabir University of Technology, Tehran, Iran, March 10-11, 2009.
- [6] M. Abdollahi Azgomi and R. Entezari Maleki, "Task Scheduling Modeling and Reliability Evaluation of Grid Services Using Coloured Petri Nets," *Future Generation Computer Systems (FGCS)*, Elsevier, ISSN: 0167-739X, Vol. 26, No. 8, 2010, pp. 1141-1150.
- [7] Li L., Fang Chun Y.; "Modeling and Performance Analysis of a Priority-based Scheduling Scheme in Service Grid", *Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06)*, IEEE 2006.
- [8] Omaraa, F. A., Arafa, M.M., 2010, Genetic algorithms for task scheduling problem, *Journal Parallel Distributed Computing*, Elsevier, Vol. 70, No. 1, pp. 13-22.
- [9] Khanli, L., Etminan Far, M., Ghaffari, A., 2010, Reliable Job Scheduler using RFOH in Grid Computing, *Journal of Emerging Trends in Computing and Information Sciences*, Vol. 1, No. 1, pp. 43- 47.
- [10] Entezari-Maleki, R. and Movaghar, A., 2010, A Genetic-Based Scheduling Algorithm to Minimize the Makespan of the Grid Applications, *Communications in Computer and Information Science*, Springer, Vol. 121, pp. 22-31.
- [11] H. Sadi, J. Habibi, H. Mohammadi, "Design a Scheduler for Computational Grid Using Genetic Algorithm", *13th Annual Conference of Computer Society of Iran*, 2007.
- [12] S. Suryadevera, J. Chourasia, S. Rathore, A. Jhummarwala, "Load Balancing in Computational Grids Using Ant Colony Optimization Algorithm", *International Journal of Computer & Communication Technology (IJCCCT)*, Vol.3, 2012, pp. 20-23.
- [13] M.J. Asgharpour, "Multiple Criteria Decision Making" University of Tehran Press, 10th edition, 2011.