BioGecko Vol 12 Issue 02 2023

ISSN NO: 2230-5807

Parallel Machine and Task Scheduling Optimization using Advanced Mixed Integer Linear Programming Technique

Sudha Kumari
M. Tech. Scholar, Vivekananda Global University, Jaipur
sudha494@gmail.com

Dushyant Singh
Assistant Professor, Vivekananda Global University, Jaipur
dushyant.singh@vgu.ac.in

Abstract: Consumers can access internet services using cloud computing, a recent technology. Appropriate task scheduling is necessary for cloud computing to operate at a high efficiency. Traditional heuristic techniques are inefficient in this setting because job scheduling in cloud technology is thought of as an NP-complete issue. Due to this, the bulk of newly suggested work scheduling algorithms have concentrated on Mixed Integer Linear Programming (MILP) techniques. In this study, utilizing MILP, we offered a revolutionary approach. The suggested method outperforms in terms of make span, according to the simulation and statistical analysis's findings. Furthermore, Gantt charts have been used to describe the outcomes of task scheduling.

I. INTRODUCTION

Recent years have seen the emergence of cloud computing as one of the most important technologies and well-liked online services. In order to reduce operating and maintenance costs, a new framework that moves the physical site of computation and storage within the network has been made possible by the introduction of the cloud computing environment. A new resource and tool in high-performance computing is cloud computing. A centralized, internet-based service delivery technique for keeping data and applications is known as cloud computing. Customers may access a variety of apps over the internet without installing any software thanks to cloud computing. High-performance computing also makes use of cloud computing to combine storage, memory, processing, and bandwidth. [1] In addition, cloud computing is a method that scales and delivers information and communication technology dynamically through the internet. Additionally, cloud computing offers computation power to Internet users as a free service based on the demand-driven pay-as-you-go model. Contrarily, scheduling is seen as a decision-making procedure that is frequently employed in the majority of manufacturing and service-providing firms and is used to maximize efficiency. In actuality, scheduling is the distribution of resources across time to tasks for applications. [2-3] Scheduling is the difficult issue with cloud computing. Users in the cloud environment receive services from the cloud provider. The fact that the cloud provider provides services without identifying the location of host infrastructures and data centers is important. Additionally, due to business considerations, cloud computing must consider global customer tastes and expectations for service quality.

II. MATHEMATICAL MODELLING

A. Standard Mixed Integer Programming

The branch and cut method used by the MIP solver, which is implemented in this technique, is based on linear programming. By resolving the linear programming relaxations of a series of smaller subproblems, this divide-and-conquer strategy[4-5] seeks to resolve the main issue. To increase the effectiveness of the entire algorithm, the MIP solver additionally uses sophisticated techniques including pre-solving, creating cutting planes, and utilizing primal heuristics. The MIP solver offers a variety of control settings and solution approaches [6-7]. For the advanced techniques in particular, it enables, disables, or sets levels.

$min\ C^T\ x$

Subjected to

$$Ax\{\geq, =, \leq\}b$$

$$l \le x \le u$$
$$x_i \in Z \; \forall \; i \; \in S$$

Where x is a vector of organizational factors, A is a matrix of technical coefficients, c is a vector of optimization problem coefficients, b is a vector of constraints' right half (RHS), l is a vector of lower values on variables, u is a vector of outer boundaries on factors, and S is a nonempty subcategory of the set of indices 1,....,n.

B. Task Scheduling Optimization using MIP

Mixed Integer Programming (MIP) has been incorporated in the proposed work to represent the (data set including the time needed for machine k to perform responsibilities of a position j represented by $R_{j,k}$ and the sequential relying setup times of accessing task j after task I on machine k given by $Q_{i,j,k}$ is priori and calculated. [8-10] The model can reduce the maximum finish time (L_{max}). There are some symbolic definitions to explain the computational formula of this problem.

TABLE I: Symbolic definitions of various parameters of proposed problem

Symbol	Definition
Lj	The moment the machine's most recent work, j, finished.
Lmax	The makespan, or maximum completion time
Yijk	When job j is handled as soon as job I is handled by machine k
	It is 1; otherwise 0
Y0jk	It is 1 if machine k executes job j first; else, it is 0.
Yj0k	Task I is 1 if it is the final job to be handled on machine k; else, it is 0.
S	a significant enough positive number
Qijk + Rjk	Matrix of corrected machine processing times

Depending on the symbol definition in Table 1 above, this article establishes a mathematical model of (R_{jk} | Q_{ijk} | L_{max}) as shown below.

Objective function:

$$minimize \; T_{max} \quad (1)$$

$$s.t. \sum_{k}^{M} \sum_{\substack{i=0 \\ i \neq j}}^{N} y_{i,j,k} = 1, \forall j = 1, \ldots, N; \quad (2)$$

$$\sum_{j}^{N} y_{0,j,k} = 1, \; \forall \; k = 1, \ldots, M; \quad (3)$$

$$\sum_{\substack{i=0 \\ i \neq h}}^{N} y_{i,j,k} - \sum_{\substack{j=0 \\ j \neq h}}^{N} y_{h,j,k} = 0, \forall h = 1, \ldots, N; \quad (4)$$

$$L_j - \left[L_i + \sum_{k=1}^{M} y\left(Q_{i,j,k} + R_{j,k}\right) + S\left(\sum_{k=1}^{M} y_{i,j,k} - 1\right)\right] \ge 0, \forall i = 0, \ldots, N; (5)$$

$$Lj \; \le \; Lmax, \forall \; j \; = \; 1, \cdots, N; \quad (6)$$

$$L_0 \; = \; 0; \quad (7)$$

$$Lj \; \ge \; 0, \forall \; j \; = \; 1, \cdots, N; \quad (8)$$

$$y_{i,j,k} \; \in \; \{0, 1\}, \forall \; i \; = \; 0, \cdots, N; \; \forall \; j \; = \; 0, \cdots, N; \; \forall \; k \; = \; 1, \cdots, M \quad (9)$$

The objective function of the proposed approach is to minimize the maximum completion time, the constraint in formula ensures that the job will be executed only once, the constraint indicates that the number of jobs to be processed first by each machine is 1, the constraint in formula (4) ensures that each job is processed first or constructed into a tight post-processing job of other jobs, the formula (5) is used to calculate the completion time of each job and the formula (6) defines L_{max} as having to be greater than the completion time of any other job. The constraint (7) ensures that the completion of virtual job 0 is 0, he constraint (8) ensures that the completion time of the job is not negative, and the constraint (9) defines the value range of decision variable x [13-14]

Pseudo Code for proposed approach
Require: Identification of Minimize
Input: Processing Time, Task sequence, Task IDs and Virtual Machine IDs
Output: Minimum value of completion time L_{max}
Initialization: set bounds and iter = 0
Loop Initialize for L_{max} Objective function
$\sum L_{max}$
    Loop Iterations = No. of Tasks.
    end
Constraint Definition
Loop initialize for constraint
$\sum_{k}^{M} \sum_{\substack{i=0 \\ i \neq j}}^{N} y_{i,j,k} = 1, \forall j = 1, \ldots.., N;$
$\sum_{j}^{N} y_{0,j,k} = 1, \ \forall \ k = 1, \ldots, M;$
$\sum_{\substack{i=0 \\ i \neq h}}^{N} y_{i,j,k} - \sum_{\substack{j=0 \\ j \neq h}}^{N} y_{h,j,k} = 0, \forall h = 1, \ldots, N;$
$L_j - \left[L_i + \sum_{k=1}^{M} y\left(Q_{i,j,k} + R_{j,k}\right) + S\left(\sum_{k=1}^{M} y_{i,j,k} - 1\right)\right] \geq 0, \forall i = 0, \ldots, N;$(5)
$Lj \ \leq \ Lmax, \forall \ j \ = \ 1, \cdots, N;$
$L_0 \ = \ 0;$
$Lj \ \geq \ 0, \forall \ j \ = \ 1, \cdots, N;$
$y_{i,j,k} \ \in \ \{0,1\}, \forall \ i \ = \ 0, \cdots, N; \ \forall \ j \ = \ 0, \cdots, N; \ \forall \ k \ = \ 1, \cdots, M$
end
Solving problem using solver linear integer programming
Sorting L_{max} maximum values appending at top

In this work six test cases have been considered for validating the AMIP algorithm. The complete test cases are shown in the given table:

III. RESULTS

The above AMIP algorithm is implemented on the proposed task scheduling optimization problem on MATLAB and following results are obtained in terms of Completion Time (L_{max}). The given tables describe the various test cases, no. of virtual machines considered and n. of tasks assigned to the system.

TABLE II: Completion timings for different test cases

Test Case No.	**No. of Virtual Machines**	**No. of Tasks**	**L_{max}**
TC #1	3	4	23
TC #2	3	5	22
TC #3	3	3	17
TC #4	4	3	28
TC #5	4	5	33
TC #6	6	7	24

It has been observed that, for the different test cases with variations in both Virtual Machines (VMs) and Tasks (T) completion time varies. It can be clearly seen from table 2 that L_{max} for TC #1 was obtained as 34 mins which can also be validated from the gantt chart of Processing Time Vs Virtual Machine plot of respective test case. With an increment of 1 task with 3 virtual machines no significant changes were observed in the system.

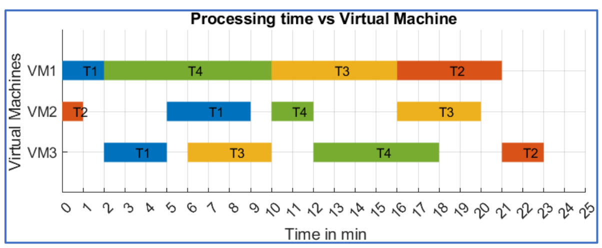

Figure 1: Gantt Chart for VM scheduling for test case 1

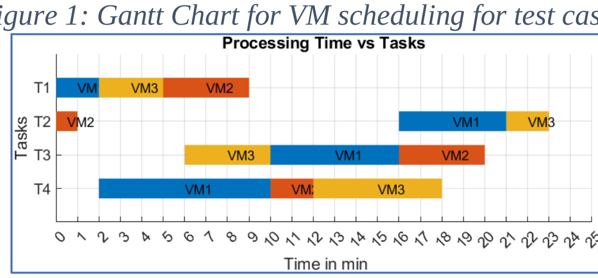

Figure 2: Gantt Chart for Task scheduling for test case 1

Similarly for the all other test cases plot are shown in figure 2 to figure 6. It has been evident that with the increase in virtual machines, the completion time increases to an extent not exceeding 28 mins but when the simultaneously tasks are also increased in the system the completion time increases.

BioGecko
Vol 12 Issue 02 2023
ISSN NO: 2230-5807

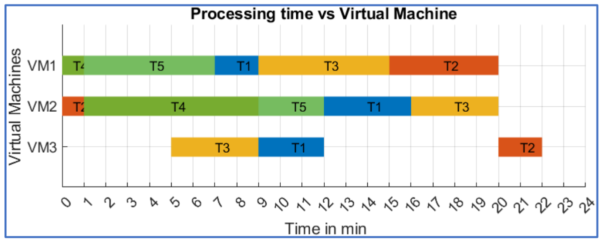

Figure 3: Gantt Chart for VM scheduling for test case 2

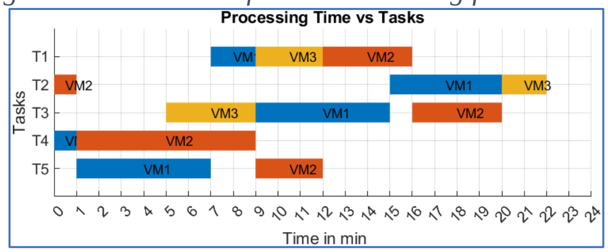

Figure 4: Gantt Chart for Task scheduling for test case 2

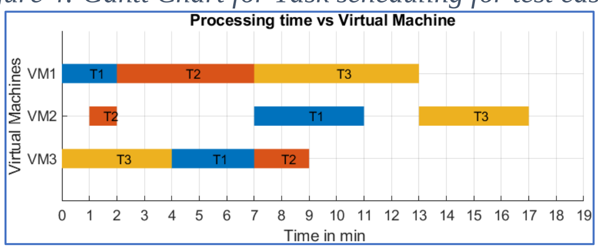

Figure 5: Gantt Chart for VM scheduling for test case 3

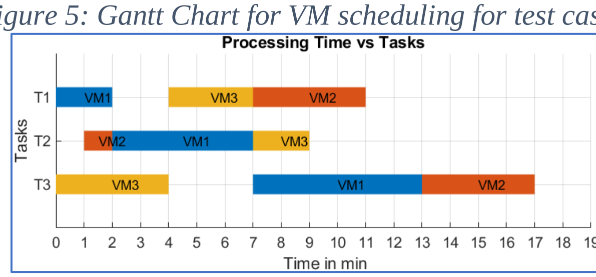

Figure 6: Gantt Chart for Task scheduling for test case 3

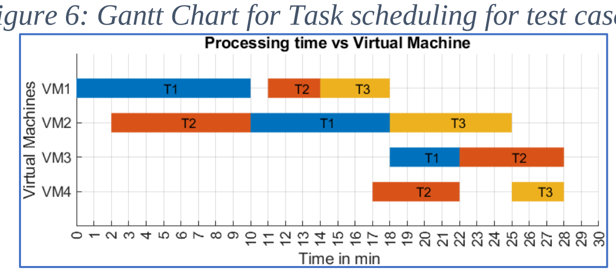

Figure 7: Gantt Chart for VM scheduling for test case 4

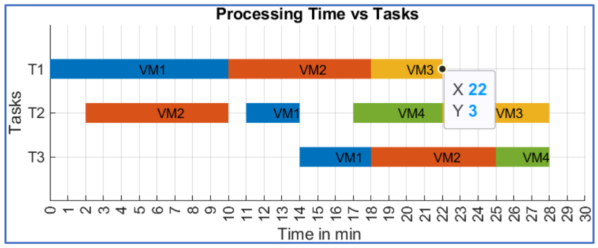

Figure 8: Gantt Chart for Task scheduling for test case 4

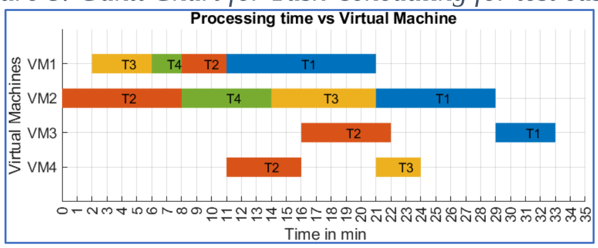

Figure 9: Gantt Chart for VM scheduling for test case 5

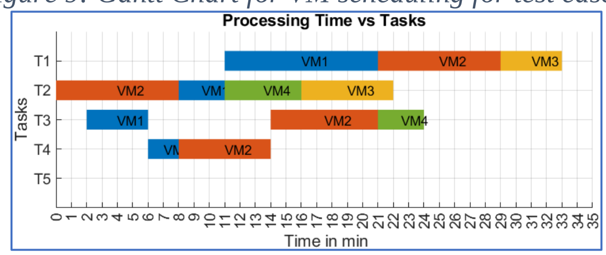

Figure 10: Gantt Chart for Task scheduling for test case 5

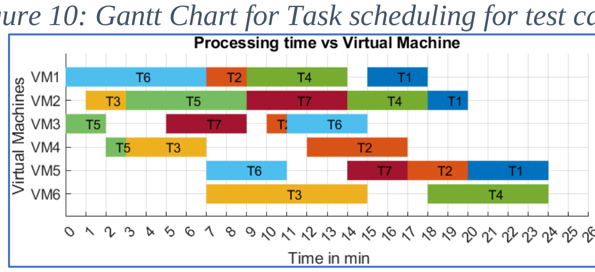

Figure 11: Gantt Chart for VM scheduling for test case 6

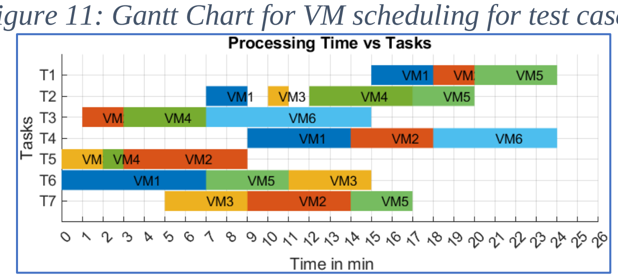

Figure 12: Gantt Chart for Task scheduling for test case 6

IV. CONCLUSION

The Task Scheduling and Virtual Machine Scheduling has been successfully implemented on optimization using mixed integer linear programming algorithm. It has been seen that without scheduling the tasks and considering them as per the process timing or by machine IDs or by tasks IDs the execution time increases, when the proposed algorithm has been implemented, the completion timing reduces to an extent which can be evident from gantt charts.

References

[1] M. A. Rodriguez and R. Buyya, “A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments,” *Concurrency and Computation: Practice and Experience*, vol. 29, no. 8, 2017, doi: 10.1002/CPE.4041.

[2] S. Mittal and A. Katal, “An Optimized Task Scheduling Algorithm in Cloud Computing,” *Proceedings - 6th International Advanced Computing Conference, IACC 2016*, pp. 197–202, 2016, doi: 10.1109/IACC.2016.45.

[3] J. Verma, S. Sobhanayak, S. Sharma, A. K. Turuk, and B. Sahoo, “Bacteria foraging based task scheduling algorithm in cloud computing environment,” *undefined*, vol. 2017-January, pp. 777–782, 2017, doi: 10.1109/CCAA.2017.8229901.

[4] J. Verma, S. Sobhanayak, S. Sharma, A. K. Turuk, and B. Sahoo, “Bacteria foraging based task scheduling algorithm in cloud computing environment,” *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017*, vol. 2017-January, pp. 777–782, 2017, doi: 10.1109/CCAA.2017.8229901.

[5] S. Mirjalili, “Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,” *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016, doi: 10.1007/S00521-015-1920-1/TABLES/12.

[6] M. S. Ajmal, Z. Iqbal, M. B. Umair, and M. S. Arif, “Flexible Genetic Algorithm Operators for Task Scheduling in Cloud Datacenters,” *2020 14th International Conference on Open Source Systems and Technologies, ICOSST 2020 - Proceedings*, 2020, doi: 10.1109/ICOSST51357.2020.9333057.

[7] A. Kamalinia and A. Ghaffari, “Hybrid Task Scheduling Method for Cloud Computing by Genetic and DE Algorithms,” *Wireless Personal Communications*, vol. 97, no. 4, pp. 6301–6323, 2017, doi: 10.1007/S11277-017-4839-2.

[8] A. Chraibi, S. Ben Alla, and A. Ezzati, “Makespan Optimisation in Cloudlet Scheduling with Improved DQN Algorithm in Cloud Computing,” *Scientific Programming*, vol. 2021, 2021, doi: 10.1155/2021/7216795.

[9] “Sci-Hub | [IEEE 2017 Fourth International Conference on Image Information Processing (ICIIP) - Shimla, India (2017.12.21-2017.12.23)] 2017 Fourth International Conference on Image Information Processing (ICIIP) - Monkey search algorithm for task scheduling in cloud IaaS | 10.1109/ICIIP.2017.8313789”.

[10] P. Mukherjee, P. K. Pattnaik, T. Swain, and A. Datta, “Task scheduling algorithm based on multi criteria decision making method for cloud computing environment: TSABMCDMCCE,” *Open Computer Science*, vol. 9, no. 1, pp. 279–291, 2019, doi: 10.1515/COMP-2019-0016/REFERENCES.

[11] X. Huang, C. Li, H. Chen, and D. An, “Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies,” *Cluster Computing*, vol. 23, no. 2, pp. 1137–1147, 2020, doi: 10.1007/S10586-019-02983-5.

[12] P. Mukherjee, P. K. Pattnaik, T. Swain, and A. Datta, “Task scheduling algorithm based on multi criteria decision making method for cloud computing environment: TSABMCDMCCE,” *Open Computer Science*, vol. 9, no. 1, pp. 279–291, 2019, doi: 10.1515/COMP-2019-0016/REFERENCES.

[13] A. Razaque, N. R. Vennapusa, N. Soni, G. S. Janapati, and khilesh R. Vangala, “Task scheduling in Cloud computing,” in *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Apr. 2016, pp. 1–5. doi: 10.1109/LISAT.2016.7494149.

[14] X. Huang, C. Li, H. Chen, and D. An, “Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies,” *Cluster Computing*, vol. 23, no. 2, pp. 1137–1147, 2020, doi: 10.1007/S10586-019-02983-5.

[15] Singh, Dushyant, and Baldev Singh. "Secure Chess-Based Data Exchange and User Validation." Journal of Cases on Information Technology (JCIT) 24.4 (2022): 1-10.

[16] Saxena, N., Kumar, R., Saxena, R., Mishra, S.K., Jain, R., Verma, S.K. (2023). Deep Learning Approach for Wind Power Forecasting. In: Dhar, S., Do, DT., Sur, S.N., Liu, H.CM. (eds) Advances in Communication, Devices and Networking. Lecture Notes in Electrical Engineering, vol 902. Springer, Singapore. https://doi.org/10.1007/978-981-19-2004-2_32
[17] Singh, D., & Singh, B. (2022). Secure Chess-Based Data Exchange and User Validation. Journal of Cases on Information Technology (JCIT), 24(4), 1-10.
[18] Singh, Dushyant, and Baldev Singh. "Secure Chess-Based Data Exchange and User Validation." Journal of Cases on Information Technology (JCIT) 24, no. 4 (2022): 1-10.
[19] Singh, D. and Singh, B., 2022. Secure Chess-Based Data Exchange and User Validation. Journal of Cases on Information Technology (JCIT), 24(4), pp.1-10.
[20] Singh D, Singh B. Secure Chess-Based Data Exchange and User Validation. Journal of Cases on Information Technology (JCIT). 2022 Oct 1;24(4):1-0.
[21] Singh, Dushyant. "A Review on Deep Learning Models." Integrated Emerging Methods of Artificial Intelligence & Cloud Computing (2022): 223-229.
[22] Singh, D. (2022). A Review on Deep Learning Models. Integrated Emerging Methods of Artificial Intelligence & Cloud Computing, 223-229.
[23] Singh, Dushyant. "A Review on Deep Learning Models." Integrated Emerging Methods of Artificial Intelligence & Cloud Computing (2022): 223-229.
[24] Singh, D., 2022. A Review on Deep Learning Models. Integrated Emerging Methods of Artificial Intelligence & Cloud Computing, pp.223-229.
[25] Singh D. A Review on Deep Learning Models. Integrated Emerging Methods of Artificial Intelligence & Cloud Computing. 2022:223-9.