IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-2, Issue-7 E-ISSN: 2347-2693

Optimization Techniques for Task Scheduling in Multiprocessor System - A Review

Asha Rani[1*] and Sukhvir Singh Boora[2]

[1]*Computer Science Department, N.C. College of Engineering, India*
[2]*Computer Science Department, N.C. College of Engineering, India*

Asha22mar@gmail.com, boora_s@gmail.com

www.ijcaonline.org

Received: 3 July 2014 Revised: 10 July 2014 Accepted: 26 July 2014 Published: 31 July 2014

Abstract— The multiprocessor scheduling can be defined as scheduling a task graph in a way such that performance criteria is optimized . Optimization is a mechanism of finding minimum and maximum values from a given set of values. This is done by taking some particular parameters and calculating results according to it. In this paper we describe various optimization technique such as heuristic scheduling, Local search algorithm and Global search algorithm. These algorithm are characterized further as simulated annealing , Tabu search and genetic algorithm comes under local search algorithm , and branch and algorithm comes under global search algorithm . All these optimization techniques plays important role in reducing scheduling length efficiently in order to increase the efficiency of multiprocessor system.

***Keywords*:** Optimization Techniques, Simulated Annealing, Tabu Search, Genetic Algorithm, Branch And Bound Algorithm.

I. INTRODUCTION

The multiprocessor scheduling can be defined as scheduling a task graph in a way such that performance criteria are optimized. Optimization is a mechanism of finding minimum and maximum values from a given set of values. This is done by taking some particular parameters and calculating results according to it. Typically, some attempt is made to optimize the schedule [2] according to some quality metric.

- Heuristic algorithm
- Local search algorithm
- Global search algorithm

These all algorithm are used to minimize the execution time of tasks in multiprocessor environment.

II. OPTIMIZATION TECHNIQUES

There are various optimization techniques that are used for scheduling that are as follows:

- Heuristic algorithm.
- Local search algorithm.
- Global search algorithm.

A. *heuristic algorithm*

Heuristic scheduling [4] algorithms typically attempt to achieve reasonably good, feasible schedules by assigning tasks to resources according to an order based upon some criticality measure. For example, tasks may be ordered according to the ratio of their availability duration (due date less release date) to their processing time; or resources may be ordered according to their total load (which changes as the algorithm schedules tasks). Typically, heuristic algorithms do not revise task assignments even if the schedule turns out to be poor.

B. *local search algorithm*

A local search algorithm computes on a single candidate schedule that is already complete – in the sense that it contains reservations for all of the tasks that are to be scheduled – but that may be infeasible or sub-optima [3]. A local search algorithm typically performs a transformation on the candidate schedule to produce a better neighboring schedule, and iterates this process until a satisfactory schedule is produced or a local optimum is reached. An example of a transformation is shifting a reservation forward in time to correct a due-date violation. It is not guaranteed that the final schedule [1] is globally optimal. Local search algorithms are also called iterative repair algorithms.

The primary considerations in designing a local search algorithm are:

- Generation of an initial seed schedule;
- Generation of the neighbors of a candidate solution;
- Selection of the best neighbor;
- Avoiding being trapped in unsatisfactory local optima.

Local search algorithm are further divided into various parts such as:

- Simulated annealing
- Taboo search
- Genetic algorithm

1) *simulated annealing*

In standard local search algorithms, the algorithm only progresses from some candidate schedule to a better neighbouring schedule. As a result, the algorithm may become stuck in a local optima and fail to find global optima. Simulated annealing is a variant of local search that tries to escape from local minima by occasionally progressing to neighbors that are worse than the current candidate schedule. At each iteration of the search, each worse neighbor of the current candidate schedule has a (low) probability of being accepted as the next candidate schedule. The probabilities are assigned according to some function of:

- The qualities of each neighbor relative to the quality of the current candidate;
- The length of time the algorithm has been running.

The longer the algorithm runs, the less likely the worse neighbors are to be accepted. Eventually, probability of progress to a worse neighbor becomes negligible and the search settles into a local optimum.

2) tabu search

In tabu search, the search algorithm may progress from a current candidate solution to any neighboring solution – better or worse – provided only that the transformation to the neighbor is not contained in the current-list. The-list is a fixed length list of schedule transformations that are currently forbidden; when a transformation is applied to the current candidate schedule, that transformation is added to the head of the list and the tail of the list is removed. The intention is to try to prevent cycling in the search process.

3) genetic algorithm

Genetic algorithms operate on finite-sized populations of candidate schedules. At each iteration of the algorithm, relatively poor schedules are removed from the population and are replaced with new candidate schedules generated by:
(1) Applying mutations to individual schedules in the population;
(2) Applying cross-over operations to pairs of schedules in the population.

Genetic algorithms [4] [5] as powerful and broadly applicable stochastic search and optimization techniques, are the most widely known types of evolutionary computation [6] [5] methods today. The father of the original Genetic Algorithm was John Holland [7] who invented it in the early 1970's. In general, a genetic algorithm has five basic components as follows:

- An encoding method that is a genetic representation [5] (genotype) of solutions to the program.
- A way to create an initial population of individuals [Davis, 1991].
- An evaluation function, rating solutions in terms of their fitness, and a selection mechanism.
- The genetic operators (crossover and mutation) that alter the genetic composition of offspring during reproduction.
- Values for the parameters of genetic algorithm

C. *global search algorithms*

Global search algorithms find feasible or globally optimal schedules by searching through schedule spaces. Typically, a search tree is constructed in which each node represents, say, a task that is to be scheduled and each branch from that node represents a choice of to which resource the task is assigned. Each partial schedule constructed in this way is required to be feasible. If a node is reached for which no choice of task and resource produces a feasible schedule, then back-tacking occurs.
For finding a feasible schedule, the algorithm terminates when it has constructed a feasible schedule and no tasks remain to be scheduled. For finding an optimal schedule, each such complete, feasible schedule is compared using some quality metric and the best one is returned; this is a globally optimal schedule.
The primary considerations in designing a practical global search algorithm are:

1) reducing the size of the search space.
2) Selecting a good order for considering the siblings branches at each level (since this may affect the ability to reduce the search space).

1). *branch and bound algorithm*

Global search algorithms may also use the schedule quality metric to prune off search branches. For example, assume that a feasible schedule is required that maximizes some metric. It may be possible to compute an upper bound on the metric for any given branch in the search tree, such that no feasible schedule contained in that branch has a metric that exceeds the upper bound computed for that branch. If a branches upper bound does not exceed the metric of some schedule that has already been constructed, then the branch cannot contain an optimal schedule and the branch can be pruned. If a heuristic scheduler is available that produces reasonable schedule, then it can be used to seed the branch & bound algorithm.
If an approximately optimal solution is sufficient, a more liberal pruning policy can reduce the search space still further: a branch is pruned off if its upper bound does not exceed, by some defined tolerance, the highest value of the metric found so far. For example, if the tolerance is set to 10%, then the schedule returned by the algorithm is guaranteed to be within approximately 9% of optimal.

III. CONCLUSION

In this paper we have described various optimization techniques their advantages and disadvantages and focuses more in explaining the practical use of these techniques. We believe that, through working out with these algorithm one could easily grasp the idea of how and where we can used these algorithm. The main aim of using technique is to complete the tasks in minimum time and without violating the time constraint. In this paper we also describes the advantages of one techniques over the other and also which one is better in which situation such as local search algorithm traps in local minima in order to get global optimum solution we can use Global search algorithm.

REFERENCES

[1] G. Syswerda and J. Palmucci, "The application of genetic algorithms to resource scheduling", Proceedings of the Fourth International Conference on Genetic Algorithms and Their Applications, pages 502-508, San Mateo, CA, July 1991.

[2] G. A. Cleveland and S. F. Smith, "Using genetic algorithms to schedule flow shop releases", Proceedings of the Third International Conference on Genetic Algorithms and Their Applications, pages 160-169, San Mateo, CA, June 1989.

[3] J. H. Holland, "Adaptation in Natural and Artificial Systems", The University of Michigan Press, Ann Arbor, MI, 1975.

[4] Cottet, F., Delacroix, J, Kaiser, C., Mammeri, Z., "Scheduling in Real-time Systems", John Wiley & Sons Ltd, England, 2002.

[5] Goldberg, David E, "Genetic Algorithms in Search, Optimization and Machine Learning", Kluwer Academic Publishers, Boston, 1989.

[6] Mitchell, Melanie, "An Introduction to Genetic Algorithms", MIT Press, Cambridge, MA. 1996.

[7] L.M.Schmitt, "Fundamental Study Theory of Genetic Algorithms", International Journal of Modelling and Simulation Theoretical Computer Science. 2001.

[8] C. V. Ramamoorthy, "Optimal scheduling strategies in a multiprocessor system", IEEE Trans. Computers, vol. C-2I.,Feb. 1972.

[9] I. H. Kasahara and S. Narita, "Practical multiprocessing scheduling algorithms for efficient parallel processing", IEEE Transactions on Computers, 1998.

[10] E. S. H. Hou, R. Hong, and N. Ansari, "Efficient multiprocessor scheduling based on genetic algorithms", IEEE 1990.

[11] Muhhamad K. Dhodhi, Imtiaz Ahmad, Ishfaq ahmad, "A multiprocessor scheduling scheme using Problem-space genetic algorithms", IEEE 1995.

[12] Michael Bohler, Frank Moore, Yi Pan, "Multiprocessor Scheduling Using Genetic Algorithms", Twelfth International FLAIRS Conference, 1999.

[13] Yi-Wen Zhongiz, Jian-Gang Yang, "A Genetic algorithm for tasks scheduling in parallel multiprocessor systems", Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2-5 November 2003

[14] Michael Rinehart, Vida Kianzad, and Shuvra S. Bhattacharyya, "A Modular Genetic Algorithm for Scheduling Task Graphs", 2003.

[15] Andrew J. Page and Thomas J. Naughton, "Framework for task scheduling in heterogeneous distributed computing using genetic algorithms", pp. 1-14, 2005.

[16] Andrew J. Page and Thomas J. Naughton, "Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing", 2005.

[17] Faezeh Montazeri, Mehdi Salmani-Jelodar, S. Najmeh Fakhraie and S. Mehdi Fakhraie, "Evolutionary Multiprocessor Task Scheduling", Proceedings of the International Symposium on Parallel Computing in Electrical Engineering (PARELEC'06), IEEE 2006.

[18] M. Salmani Jelodar, S. N. Fakhraie, F. Montazeri, S. M. Fakhraie, M. Nili Ahmadabadi, "A Representation for Genetic-Algorithm-Based Multiprocessor Task Scheduling", Congress on Evolutionary Computation, Vancouver, BC, Canada, IEEE July 16-21, 2006.

[19] M. Nikravan and M. H. Kashani, "A Genetic algorithm for process scheduling in distributed operating system considering load balancing ", Proceedings 21st European Conference on Modelling and Simulation, 2007.

[20] Keshav Dahal, Alamgir Hossain, Benzy Varghese, "Scheduling in Multiprocessor System Using Genetic Algorithms", 7th Computer Information Systems and Industrial Management Applications, IEEE 2008.

[21] Amir Masoud Rahamani, Mohamad Ali Vahedi "A novel Task Scheduling in Multiprocessor Systems with Genetic Algorithm by using Elitism stepping method", 2008.

[22] Yajun Li, Yuhang Yang, Maode Ma, Rongbo Zhu, "A Problem-Specific Genetic Algorithm for Multiprocessor Real-time Task Scheduling", The 3rd Intetnational Conference on Innovative Computing Information and Control (ICICIC'08), IEEE 2008.

[23] Peyman Almasi Nejad, Ahmad Farahi, Davood Karim Zadegan Moghadam, Reza Asgary Moghadam, "An Intelligent Method for Multi Processor Scheduling using Genetic Algorithms", International Conference on MultiMedia and Information Technology, IEEE 2008.

[24] Ali Pedram, "A method for scheduling multi processing systems with genetic algorithm", International Journal of Engineering and Technology Vol. 1, No. 2, June, 2009.

[25] Intisar A.Majied Al-Said, Nedhal Al-Saiyd, Firas Turki Attia, "Multiprocessor scheduling based on genetic algorithms", 2009.

[26] Sachi Gupta, Gaurav Agarwal, "Task Scheduling in Multiprocessor System Using Genetic Algorithm", Second International Conference on Machine Learning and Computing, IEEE 2010.

[27] Adel Manaa and Chengbin Chu, "Scheduling multiprocessor tasks to minimize the makespan on two dedicated processors", European Journal of Industrial Engineering, pp. 265 – 279, Volume 4, Number 3 / 2010.

[28] Amir Masoud Rahmani and Mojtaba Rezvani, "A Novel Genetic Algorithm for Static Task Scheduling in Distributed Systems", 2009.

[29] Carnegie-Mellon, "Genetic Algorithms and Their Applications", Proc. of the First Int. Conference, July 24-26, 1985.

[30] Dr. Franz Rathlauf, "Representations for Genetic and Evolutionary Algorithms", 2nd edition, @ Springer. 2006.

[31] S. Beaty, "Genetic algorithms and instruction scheduling", Proceedings of the 24th Microprogramming Workshop (MICRO-24), Albuquerque, NM, November 1991.
[32] John J. Grefenstette, "Genetic Algorithms and Their Applications", Proc. 2nd Int. Conf, July 28-31, 1987, MIT, Cambridge,1987.
[33] Davis, "Handbook of Genetic Algorithms", Van Nostrand Reinhold, 1991.
[34] E. Hou, R. Hong, and N. Ansari, "Multiprocessor scheduling based on genetic algorithms", Dept of ECE, New Jersey Institute of Technology, Technical Report, Aug. 1990.
[35] Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer, 1996.
[36] Goldberg D., "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wessley publishing company Inc., 1989.
[37] Allen, F. & Karjalainen, "Using Genetic Algorithms to Find Technical Trading Rules", Journal of Financial Economics, 1999.
[38] Forrest, Stephanie. "Genetic algorithms: principles of natural selection applied to computation", Science, vol.261, 1993.
[39] Kirkpatrick, S., C.D. Gelatt and M.P. Vecchi, "Optimization by simulated annealing", Science, vol.220, p.671-678, 1983.
[40] Tang, K.S., K.F. Man, S. Kwong and Q. He. "Genetic algorithms and their applications", IEEE Signal Processing Magazine, vol.13, 2004.
[41] Michael Bohler, Frank Moore, Yi Pan. "Improved Multiprocessor Task Scheduling Using Genetic Algorithms", Twelfth International FLAIRS Conference, 1999.
[42] S.N.Sivanandam, S.N.Deepa, "Introduction to Genetic Algorithms", Springer-Verlag Berlin Heidelberg, 2008.
[43] Forrest, Stephanie. "Genetic algorithms: principles of natural selection applied to computation" Science, vol.261. 1993.
[44] G. Syswerda, "Uniform crossover in genetic algorithms", Proceedings of the Third International Conference on Genetic Algorithms and Their Applications, pages 2-9, San Mateo, CA, June 1989.
[45] G. G. Robertson, "Population size in classifier systems", Proceedings of the Fifth International Workshop on Machine Learning, pages 142-152, San Mateo, CA, June 1988.
[46] M. F. Bramlette, "Initialization, mutation and selection methods in genetic algorithms for function optimization", Proceedings of the Fourth International Conference on Genetic Algorithms and Their Applications, pages 100-107, San Mateo, CA, July 1991.
[47] Whitley, D., "A genetic algorithm tutorial", Statistics and Computing", 1994.
[48] Vose, Michael D, "The Simple Genetic Algorithm: Foundations and Theory", MIT Press, Cambridge, MA, 1999.
[49] J. J. Grefenstette and J. E. Baker, "How genetic algorithms work: A critical look at implicit parallelism", Proceedings of the Third International Conference on Genetic Algorithms and Their Applications, pages 20-27, San Mateo, CA, June 1989.
[50] K. A. De Jong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", Ph.D. dissertation, University of Michigan, 1975.

AUTHORS PROFILE

Name: Asha Rani
D/O: Suresh chand
State: Haryana
Has Done BTech in computer Science and Engineering from Jind Institute of Engineering and Technology (JIET) Jind, Haryana-India.
The Author is presently working on her thesis in the fourth semester of her Mtech in operating system in department of Computer Science, N.C. College of Engineering (NCCE) Israna Panipat-India.

Dr Sukhvir Singh is the Assosiative Professor and Head, Department of Computer Engineering . He has obtained his Ph.D from M.D University Rohtak in Computer Science and Engineering. He has more than 17 years of teaching and industry experience . He has published more than 30 papers in national and international journals and proceedings of conferences.

