

Survey on Heuristic Search Techniques to Solve Artificial Intelligence Problems

Ranjitha K S
MTech student
Computer Science & Engineering,
GSSSIETW, Mysuru

Arpitha T
MTech student,
Computer Science & Engineering,
GSSSIETW, Mysuru

Prakruthi S
MTech student,
Computer Science & Engineering,
GSSSIETW, Mysuru

Mouneshachari S
Associate Professor,
Dept of CSE,
GSSSIETW, Mysuru

Abstract: Artificial intelligence (AI) is an area of computer science that highlights the creation of machines that are intelligent, also they work and react like humans. Since AI problems are complex and cannot be solved with direct techniques we resort to heuristic search techniques. Heuristic search technique is any approach to problem solving, learning, or discovery which uses a practical methodology which is not guaranteed to be optimal or perfect, but it is sufficient for the immediate goals. This paper surveys some of the heuristic techniques which is used for searching an optimal solution among multiprocessor environment, followed by and method which enhances the search by doing a search in bidirection and also a method for task scheduling in multiprocessor environment. The paper also discusses about how heuristic is used to solve binary quadratic program and also how it is used in 3G (3rd Generation) Universal Mobile Telecommunication System (UMTS) network.

I. INTRODUCTION

Artificial intelligence can be defined as the study and design of intelligent agents, where the intelligent agent is a system that observes its environment and it takes action so that the chances of success can be maximized. Artificial intelligence is the intelligence which is shown by the machines or software's. Most of the problems that come under the horizon of artificial intelligence are too complicated; these problems cannot be solved using direct techniques that are available [12]. We need to solve these problems using suitable search method equipped with direct techniques whichever is available to guide the search. Heuristic is a search technique where the efficiency of search is improved, but by compromising with some claims of completeness [3]. The word heuristic has its origin from a greece word "HEURISKEN" which means find [13].The heuristic methods are applicable if the problem being solved is tough and cannot be solved using other ways and if the available method takes an infinite time or very long time to compute the solution [5]. Heuristic search techniques use domain specific knowledge to solve the problems, there are different heuristic techniques developed since many years. The rest of the paper discuss about some heuristic techniques which is used in the field of multiprocessor and some methods used in the field of networks.

II. SURVEY

a) Autonomous Parallel Heuristic Combinatorial Search

Heuristic search is nothing but the technique of searching the state space using the information provided by the evaluation function. Most of the heuristic search algorithms have concentrated on the problem of load balancing and reduction in memory use. The key feature that determines the performance of a system in artificial intelligent system is the efficiency of the search algorithm being used. In order to achieve performance gains, the

author in [2] proposes Autonomous Parallel Heuristic Combinatorial Search algorithm. Unlike the other parallel search algorithms the Autonomous Parallel Heuristic Combinatorial Search does not simply divide the search space among the processors which are available. The basic idea behind this Autonomous Parallel branch and bound Search algorithm is that, every available processor which is involved in searching an optimal path and also in altering its selection will be done according to the information sent from its peers. Here each processor works alone but they communicate with one another to pass on the information which audits the development of the search. This information is also used to redirect a processor's knowledge about those nodes which appears to be more likely to be an optimal path. Each processor broadcasts its selection of a candidate node asynchronously to the other processors. The communication here need not be synchronized as this information is only for reference. The key feature of this algorithm is that it avoids the interconnection network bottlenecks and also synchronization delays, because each processor can use the broadcast information from other processor if available, else it can proceed without it also. The algorithm is useful in multiprocessor environment.

b) Parallel Bidirectional Heuristic Search

The frequently used problem solving technique in artificial intelligence is searching. Searching is the process of finding an optimal path, when given an initial state which leads to goal state. Solving the search problems demands a lot of computational resources both in terms of memory usage and in terms of execution time. The author in [1] provides an approach called Parallel Bidirectional Heuristic Search (PBiHS) which is a combination of sequential A* search [4] and bidirectional search [9]. The PBiHS performs the search in both the directions and also the search in each direction will be performed in parallel [11]. To implement PBiHS in a multiprocessor environment, we need to

concentrate on two issues first processor assignment and then data structure allocation. Processor assignment is all about dividing the processor to implement parallel bidirectional search in a multiprocessor environment. Consider there are m processors; divide these m processors into two groups so that each group will have $m/2$ processors. One among these $m/2$ processors will be designated as master processor, which is responsible to controls all the other $m/2$ processors. Data structures associated with this approach is two lists OPEN and CLOSED. Each direction has these two lists. OPEN will have the nodes that are to be examined. CLOSED will have the nodes that are already examined. OPEN is kept on the master node so that the best $m/2$ nodes are selected for parallel expansion so that we can find an optimal path in optimal number of iterations. CLOSED will be kept on all nodes so that search within each direction and also between two search directions can be parallelized. The author has also implemented the eight puzzle and tower of Hanoi on 80-processor EM-4 multithreaded data flow multiprocessor whose result showed that PBiHS could solve tree depth of 20-40 of eight puzzle and 3-9 disks of tower of Hanoi in less than 2 seconds. Also PBiHS is highly scalable and gives over 40 fold speed up on 80-processor and also it gives an average of 10 times improvements over unidirectional search for the eight puzzle problem.

c) Task Scheduling Algorithms

The computing power on a single processor has got certain limitations, as the applications become more and more complicated and larger. Hence we started moving towards parallel processing technology on multiprocessor system. Efficient parallel processing means maximum performance of the multiprocessors in a multiprocessor environment. The key to efficient parallel processing is task scheduling technique that is a set of tasks from applications is assigned to many processors and these tasks must be executed in parallel in minimum processing time. In any task scheduling problem in order to get an optimal solution a full search is needed. But doing this full search in a practical time is not possible. The author in [8] says the DF|HIS (Depth First|Implicit Heuristic Search) [6] proposed by kasahara performs search based on heuristic and does not take into account communication overhead which is caused by assigning the two tasks which has preceding relation on two different processors.

The author in paper [8] adopts a search method in scheduling algorithm which takes into account the communication overhead. This paper proposes BSP(Branch at Same Priority Method) which uses Depth First Search on branch and bound method. If we have more than one task having same priority levels and also same number of immediate successors then they make multiple selection branches; which means, which tasks should be allocated to which processor. In such cases BSP does not search the full combination of all processor and task but makes partial search which is possible in practical time. The author proposes another algorithm called GP(Grouping Priority Levels) method. The main point of this method is how the range of priority level of tasks belong to same group is

determined. The maximum value of difference between the higher priority level and lowest priority level in same group is called the range. If this range is zero then GP method provides same solution as BSP method. If the range becomes wider that is if the number of combinations of tasks and processors that has to be searched becomes more, even the search time also becomes more. In such a case the GP does a full search. To get a better performance the GP method can also be parallelized.

d) Tabu Search

Several users communicate concurrently over the same channel using specific signature waveforms in direct sequence code division multiple access (DS-CDMA). Multiple access interference (MAI) will not occur if mutually orthogonal signature waveforms are selected for all users. But it is not possible in a mobile environment to ensure orthogonality between received signature waveforms. Hence MAI arises and it is a common problem encountered in mobile communication. In order to take maximum likelihood (ML) decision for multiuser detection (MUD) problem there is a need to solve binary constrained minimization problem which is also called as binary quadratic program (BQP). To find solution to the BQP problem many heuristic solutions are been proposed which includes local search, simulated annealing, genetic algorithms and tabu search (TS). Among which TS method is an effective and efficient heuristic when compared to other heuristic methods which is proposed by the author in [10]. TS use the memory structures of past events for guidance. The memory structure has a record of previously visited solutions, which allows TS to boost or transform the search to escape local optima. The key feature of this method is that it differentiates between short-term memory and long-term memory which is defined over the span of search history. TS use a tabu list of moves which is based on the recent past search. The tabu status will be examined or evaluated whenever a new solution is generated. The new solution will be accepted even if the solution generated is inferior if and only if the new solution generated is not tabu. Also a solution will be accepted if it satisfies the predefined criteria.

e) Local Search Heuristic

Over the last two decades wireless cellular network has spread across the globe rapidly. The 3G (3rd Generation) Universal Mobile Telecommunication System (UMTS) network is very popular in today's world. The design and planning of 3G network is very much challenging as it is more complicated and costlier even though it is very much flexible. The author in paper [7] proposes a good heuristic method called local search heuristic to find good feasible solution for planning problem in 3G UMTS. The algorithm starts with an initial solution and then iteratively starts exploring the neighborhood to find the better available solutions. The method first finds an initial solution to start with and note that this initial solution must meet all the constraints if not the program will terminate. If the initial state chosen to start meets all the constraints then the feasible solution is found and is stored as best cost. This

solution becomes the current solution. In the second step all neighbors are explored. The transformation operation that creates a new solution is called move. A move will be performed if and only if it gives feasible solution. If feasible solution is given its cost is calculated and it is stored as cost_temp. After all neighbors are evaluated the best cost is found and chosen. The main advantage of this method is the CPU time that is large problem size can be handled at the cost of solution quality.

III. CONCLUSION

Majority of the artificial intelligence problems cannot be solved using the direct techniques available hence we tend to use heuristic search techniques which may generate an excellent path but there are chances that it may overlook the optimal path in some cases. The paper discusses about some the heuristic techniques used in networks and task scheduling. Heuristic is not only limited to these fields, it is also used in the field of pattern sequencing, network design, search problems, metallurgic industry and also some programs use heuristics to find virus and malware. Heuristic search techniques can be enhanced so that they use the domain specific knowledge and applied to various other domains also.

REFERENCES:

- [1] Andrew Sohn Mitsuhsa Sato, Shuichi Sakaij-, Yuetsu Kodama, Yoshinori Yamaguchi "Parallel Bidirectional Heuristic Search on the EM-4 Multiprocessor".
- [2] Chao-Chun Wang and Leah H. Jamieson "Autonomous Parallel Heuristic Combinatorial Search".
- [3] Elaine Rich, Kevin Knight, Shivashanka B Nair: Artificial Intelligence, Tata McGraw Hill 3rd edition. 2013
- [4] Hart, P.E., Nilson, N.J., and Raphael, B., "A formal basis for the heuristic determination of minimum cost paths," IEEE Trans. on SMC 4(2), pp.100-107, 1974.
- [5] <http://citeseers.ist.psu.edu>
- [6] H. Kasahara and S. Narita, "Practical Multiprocessor scheduling Algorithms for Efficient Parallel Processing," IEEE Trans. Comput., Vol. C-33, No. 11, Nov. 1984.
- [7] Mohammad Reza Pasandideh and Marc St-Hilaire "A Local Search Heuristic for the Design Problem of 3G UMTS All-IP Release 4 Networks with Realistic Traffic".
- [8] Munenori Ka, Masanori Shimada "Task Scheduling Algorithms Based on Heuristic Search Taking Account of Communication Overhead".
- [9] Pohl, I., "Bidirectional Search," in B. Meltzer and D. Michie (eds.), Machine Intelligence 6, American Elsevier, NY, pp.219-236, 1971
- [10] Peng Hui Tan and Lars K. Rasmussen "Multiuser Detection in CDMA—A Comparison of Relaxations, Exact, and Heuristic Search Methods".
- [11] Sohn, A., "Parallel Bidirectional A* Search on a Symmetry Multiprocessor," in Proc. of Fijih IEEE Symposium Parallel and Distributed Processing, Dallas, TX, December 1993.
- [12] Stuart Russel, Peter Norvig: Artificial Intelligence A Modern Approach, Pearson 3rd edition 2013.
- [13] www.cs.cf.ac.uk