International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

Improving AI Planning using Map Reduce

Mohamed Elkawkagy, Heba Elbeh

Abstract: ***Today, the Landmark concept is adapted from the classical planning to work in hierarchical task network planning. It was shown how it is used to extracts landmark literals from a given hierarchical planning domain and problem description and then use these literals to update the the planning domain by ruling out the irrelevant tasks and methods before the actual planning is performed. In this paper, we compine the landmark concept with the Map-reduce framework to increase the performance of the planning process. Our empirical evaluation shows that the combination between landmark and Map-Reduce framework dramatically improves performance of the planning process.***

Keywords **:** ***Classical Planning,HTN Planning,Search strateg,.***

I. INTRODUCTION

Artificial Intelligence planning is solving the various uses of technology in many applications like military operations, robot navigation, scheduling process, and human computer interaction. To get needed targets or particular missions, by courses of action generation , require high costs in spite of growing means to regularly debase the search trial and expand the operation orders is consequently a main attitude. Recently, a sum of approaches have been designed to carry out a preceding analysis of the artificial intelligence planning domain. The target is to withdraw and utilize information from the scope design and to classify problems so that they can debase the planning trial [1, 2]. In general, There are two different knid of Artificial intelligence planning: classical planning which uses strong search heuristics; and task network planning (HTN) which depends on the concept of tasks and methods [3, 4, 5]. Recently, some techniques have been presented use landmark to break down the planning domain and HTN planning problem description to a smaller subset and rule out the irrelevant parts from the HTN planning domain [6-9]. Using landmark is considered a new technique in hierarchical planning, but it is well defined concept in classical planning. Landmarks is defined as a technique that can discover intermediate facts of every solution plan of the specific problem.

Revised Manuscript Received on February 28, 2020.
* Correspondence Author
Mohamed Elkawkagy*, Computer Science Department, Community College, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.
Heba Elbeh, Computer Science Department, Community College, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.

The landmark concept was introduced by Porteous et al. [10] and further revised by [11, 12]. In general, Landmark facts and the relationships between them are exploited from a planning problem. After that the landmark are organized into sets as an intermediate subgoals to be acomplished [13] and then extract the disjunctive landmarks [14]. A disjunctive landmark is defined as a set of literals that should be achievied in the a solution plan. Many landmark approaches use landmark information to compute heuristic function for searching in classical and HTN planners [15-17].

In HTN planning, landmarks is defined by two different kind of tasks. Firstly, landmark primitive tasks which is a mandatory task(should be executed by any valid solution plan). Secondly, landmark abstract tasks which is a Local landmarks(should be refined by the decomposition method)[8,9].

Before introducing the proposed technique in Section 3, we will briefly review our underlying architecture (Section 2). Afterwards, Section 4 shows experimental results. The paper ends conclusion in (Section 5).

II. ARCHITECTURE

Several approaches have been revolved on using Cooperative Multi-agent planning (MAP) system to solve large planning problem. In this paper, we show how we inspire Map Reduce architecture [18] to improve the coordination of MAP in the context of hierarchical planning as shown in the following figure 1.

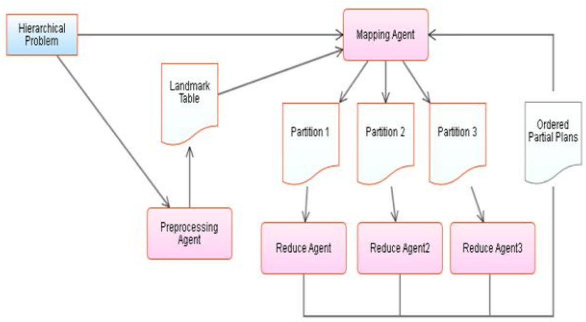

Figure 1: The proposed Arctitecture

We present Cooperative Multi-agent Partial order planning system, where agents cooperatively explore the plan search space to build refinement plans. The coordination and mapping among agents during search are guided with landmark heuristics. Mapping Agent apply a distributed greedy best-first search which enable constructing multi-agent search tree to coordinate the planning procedure with agents[19]. In hierarchical planning, landmarks are obligatory complex or primitive tasks, i.e. missions carried out through any solution plan.

For a primary task network that declares a contemorary planning problem, a pre-processing operation estimates the parallel landmarks. This is done through d systematically examining the means authorized to analyse the appropriate complex tasks.
Starting with the (landmark) tasks of the primary network, the operation traces the mould hierarchy until all complex tasks are no longer landmarks. According to primitive landmarks, an easily approached test is done; a downfall records that the way introduced the primitive landmark is no longer authorized. This information is transmitted, up the mould hierarchy and enrolls to detect all means that will never lead to a solution of the contemporary planning problem. Cutting back the growth of useless regions of the search space this way, a hierarchical planner carries out in an important way better than it does without drawing from the landmark information.
Preprocessing Agent build a Landmark Graph (LG). Landmarks are a mandatory goal or fluent that need to be satisfied in every plan solution. LG = {L, LO}, where L is the set of landmarks and LO is the necessary ordering between them. Conversely to other approaches, LG estimates the completeness of plan's refinement through stating out the unsatisfied single landmarks.
Therefore, it can be used as a heuristics function to estimate the goal distance with the number of the landmarks that still need to be satisfied from the state being evaluated onwards.
In the proposed technique the Mapping Agent holds information about the current node, ordered fragments node and landmark table.
It communicate and map iterating nodes with agents and it. It partitions the search space among different agents, where each reduce agent own a partition of the entire search space. Then, it merges the recieved refined plan from reduce agents to produce the final solution. On the other hand, Reduce Agents are identical agents that work in parallel to:

1. Reduce the given state space using abstraction of transition graph.
2. Refine a partial plan which are handled by their mapping agent.
3. Detect flaw using Least-committing-first (LCF).

III. THE PROPOSED PLAN SELECTION STRATEGIES

The process of choosing the way that is to be followed in the Induced search space which is supplied by removing options. With our architiecture, we investigate search strategies that allow agents to cooperatively explore search space. Plan selection strategy navigates the search space of changing plans examined to choose the line through the improved space.
We will use Parallel Greedy Best-First proposed by (Victoria María Sanz). It is based on HDA* (Hash Distributed A*) algorithm to enable execution on distributed memory machines. It also utilize other techniques to elminate the structure duplication such as Parallel Structured Duplicate Detection (PSDD).

3.1 Enforced Hill-Climbing (EHC)

EHC is based on the normally used hill-climbing algorithm for local search, but differs in that breadth-first search towards the global optimum is used to find a serial of actions leading to a relative follower if none is present in the current neighbourhood as shown in figure 2.

```
Procedure: EHCSearch
open_list = [initial_state];
best_heuristic = heuristic value of initial_state;
while open_list not empty do
   current_state = pop state from head of open_list;
   successors = the list of states visible from current_state;
   while successors is not empty do
      next_state = remove a state from successors;
      h = heuristic value of next_state;
      if next_state is a goal state then
         return next_state;
      end if
      if h better than best_heuristic then
         clear successors;
         clear open_list;
         best_heuristic = h;
      end if
      place next_state at back of open_list;
   end while
end while
```

Figure 2: Enforced Hill-Clumbing Search

3.2 An Abstraction Strategy

In this paper we will consider two different strategies:

1) The merging strategy which depends on the features of causal graph connectivity of the encoded task, and
2) The shrinking strategy which depends on the shortest distances in the abstract transition graphs.

The shrinking strategy is a special kind of a linear merging strategy. A linear strategy have a single non-atomic abstraction so-called the current abstraction. In first iteration, a new abstraction is created by combinig the current abstraction with is combined with an atomic projection. This process is repeated until all atomic projections have been considered.
Drager et al. use a non-linear merging strategy where each atomic abstraction is considered only once. On the other hand, a linear strategy is simpler and it achieves high performance. It is defined by the order in combination between atomic projections and the current abstraction. The order is determined according to the following rules:

1. Choose a variable from an arc in the causal graph to one of the previous added variables.
2. If there is no such variable, add a variable with defining a goal value..

3.3 Reduction strategy

3.3.1 Bisimulation

Determing the graph partition is an initial problem. In general, if two nodes share the features of basic structure they so-called bisimilar such as labeling and neighborhood topology. In data management, The crucial step is the process of reducing a graph, e.g., for indexing the graph for effcient query processing. Often, graphs of interest in the real world are massive such as social networks. For analytics on such graphs, it is becoming increasingly infeasible to rely on in-memory or even I/O- effcient solutions.

Hence, a trend in Big Data analytics is the use of distributed computing frameworks such as MapReduce. While there are both internal and external memory solutions for effciently computing bisimulation, there is, to our knowledge, no effective MapReduce-based solution for bisimulation.

Table1: Results for UM-Translog domain

Problem	MapReduce			EHC
	Search Time	Merge Time	Total Time	Total Time
1	0.000603	0.0104	0.011031	0.0058
2	0.000499	0.013	0.01346	0.022354
3	0.00107	0.449	0.450495	0.618466
4	0.00159	1.45	1.45103	2.00219
5	0.222151	16.4195	16.6417	23.8175
6	0.879119	0.997241	1.87636	2.12048
7	3.21756	6.06061	9.27817	20.3403
8	16.6123	7.4346	24.0469	36.374
9	3.91273	9.02737	12.9401	22.4723
10	19.12	11.745	30.865	550.874

IV. EVALUATION

We conducted our experiments on two well-known planning domain Satallite and UM-Translog. Satellite domain is a non-hierarchical which discusses the difficulties of managing stellar observation with satellite instruments. UM-Translog domain is a hierarchical for managing transportation and logistics. We choose a LAMA heuristics planning search algorithm as a baseline, which is Enforced-Hill-Climbing (EHC) with shrinking and merging strategy (see table 1 and 2). MapReduce show 85% average improving with regards to EHC. As we adopt a Co-MAP parallel best-first approach with landmarks to partition search space, we managed to obtain competitive performance for both hierarchical and non-hierarchical domains (see figure 3 and 4).

Table2: Results for satellite domain

Problem	MapReduce			EHC
	Search Time	Merge Time	Total Time	Total Time
1	12.97464	5.62136	18.596	123.723
2	12.99841	5.61949	18.6179	123.988
3	12.7473	5.4154	18.1627	124.491
4	13.06517	5.39433	18.4595	126.434
5	12.87239	5.48791	18.3603	127.539
6	12.87936	5.50994	18.3893	124.322
7	12.55959	5.49851	18.0581	125.2
8	12.51297	5.60033	18.1133	122.753
9	12.66758	5.74942	18.417	107.063
10	12.77379	5.60091	18.3747	114.064
11	12.45488	5.35612	17.811	116.538
12	12.47436	5.38564	17.86	124.32
13	12.93857	5.38023	18.3188	122.635
14	12.32952	5.13758	17.4671	121.441
15	12.2632	5.2678	17.531	123.649

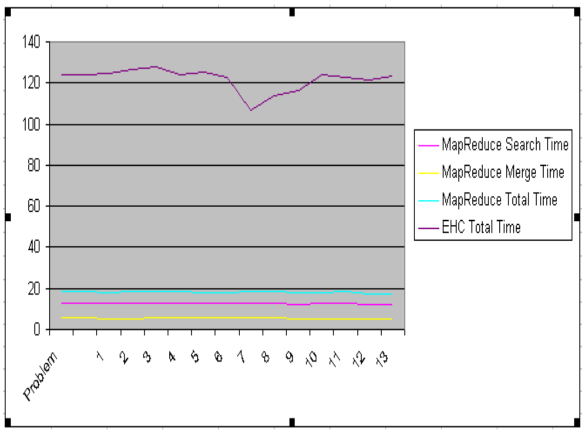

Figure 3: processing time of MS-Translog Domain

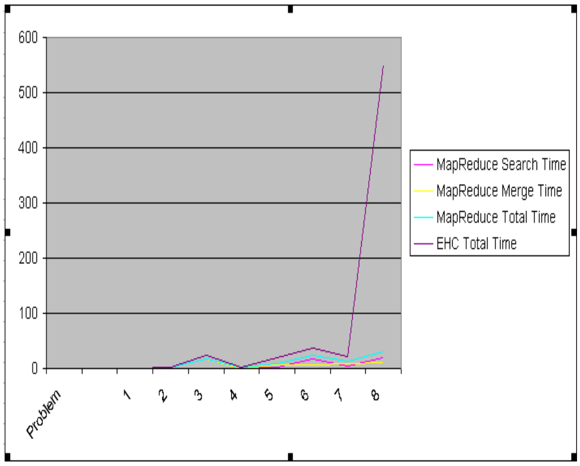

Figure 4: processing time of Satellite Domain

V. CONCLUSION

In this paper, we introduced cooperative Multi-agent Partial order planning system to build refinement plans. The coordination and mapping among agents during search are guided by landmark heuristics. Mapping Agent uses a distributed greedy best-first search to coordinate the planning procedure with agents. The evaluation is done over a number of HTN planning domains and problems that give reflect the practical relevance of our approach. The problems with a deep hierarchy of tasks works very well in the proposed technique. In general, our evaluation shows the impact effectiveness of the proposed technique.

REFERENCES

1. B. Schattenberg, J. Bidot, and S. Biundo, ‘On the construction and eval-uation of flexible plan-refinement strategies’,Proc. of German Confer-ence on Artificial Intelligence (KI), 367–381, (2007).
2. Blai Bonet and H´ector Geffner, ‘Planning as heuristic search’,Artificial Intelligence,129, 5–33, (2001).
3. Luis Castillo, Juan Fdez-Olivares, and Antonio Gonz´alez, ‘Onthe adequacy of hierarchical planning characteristics for real-world problem solving’, inProc. of the 6th European Confer-ence on Planning (ECP 2001), (2001).
4. Tara A. Estlin, Steve A. Chien, and Xuemei Wang, ‘An ar-gument for a hybrid HTN/operator-based approach to plan-ning’, inProc. of the 4th European Conference on Planning:Recent Advances in AI Planning, pp. 182–194, (1997).
5. Kutluhan Erol, James Hendler, and Dana S. Nau, ‘UMCP: Asound and complete procedure for hierarchical task-networkplanning’, inProc. of the 2nd International Conference onArtificial Intelligence Planning Systems (AIPS 1994), pp.249–254, (1994).

6. Mohamed Elkawkagy, Bernd Schattenberg, and Susanne Bi-undo, 'Landmarks in hierarchical planning', inProc. of the20th European Conference on Artificial Intelligence (ECAI2010), (2010).
7. Mohamed Elkawkagy, Pascal Bercher, Bernd Schattenberg, Susanne Biundo, 'Landmark-aware strategies for hierarchical planning', in proc. Of the 3rd Workshop on Heuristics for Domain-independent Planning (HDIP 2011).
8. Mohamed Elkawkagy, Pascal Bercher, Bernd Schattenberg, Susanne Biundo, ' Exploiting landmarks for hybrid planning', in proc. Of the 25th PuK Workshop Planen, Scheduling und Konfigurieren, Entwerfen (2010).
9. Mohamed Elkawkagy, ' Improving the performance of hybrid planning', International Journal of Artifitial Intelligence, volume 14, issue 2, PP 98-116.
10. Julie Porteous, Laura Sebastia, and J ̈org Hoffmann, 'On theextraction, ordering, and usage of landmarks in planning',inProc. of the 6th European Conference on Planning (ECP2001), eds., A. Cesta and D. Borrajo, pp. 37–48, (2001).
11. J ̈org Hoffmann, Julie Porteous, and Laura Sebastia, 'Orderedlandmarks in planning',Journal of Artificial Intelligence Re-search,22, 215–278, (2004).
12. Lin Zhu and Robert Givan, 'Landmark extraction via plan-ning graph propagation', inProc. of the ICAPS 2003 DoctoralConsortium, pp. 156–160, (2003).
13. Laura Sebastia, Eva Onaindia, and Eliseo Marzal, 'Decom-position of planning problems',AI Communications,19(1),49–81, (2006).
14. Julie Porteous and Stephen Cresswell, 'Extending landmarksanalysis to reason about resources and repetition', inIn Proc.of the 21st Workshop of the UK Planning and SchedulingSpecial Interest Group (PLANSIG 2002), pp. 45–54, (2002).
15. Malte Helmert and Carmen Domshlak, 'Landmarks, criticalpaths and abstractions: What's the difference anyway?', inProc. of the 19th International Conference on AutomatedPlanning and Scheduling (ICAPS 2009), pp. 162–169, (2009).
16. Erez Karpas and Carmel Domshlak, 'Cost-optimal planningwith landmarks', inProc. of the 21st International Joint Con-ference on Artificial Intelligence (IJCAI 2009), pp. 1728–1733, (2009).
17. Silvia Richter, Malte Helmert, and Matthias Westphal, 'Land-marks revisited', inProc. of the Twenty-Third AAAI Con-ference on Artificial Intelligence (AAAI 2008). AAAI Press,(2008).
18. Jens Dittrich, Jorge Arnulfo and Quiane Ruiz, 'Efficient Big Data Processing in Hadoop MapReduce', The 38th International Conference on Very Large Data Bases, August -2012, Istanbul, Turkey.
19. Mohamed Elkawkagy, Susanne Biundo, Hybrid multi-agent planning, in proc. Of the German Conference on Multiagent System Technologies, Publisher Springer, Berlin, Heidelberg, PP 16-28(2011).