AI PLANNING AND SEARCH KEY DEVELOPMENTS

Dinesh Bhat
bvdinesh@outlook.com

Table of Contents

• Goals

This paper provides the key development on the AI planning and search area. They are

1. Adaptive pattern Databse Heuristics
2. EU-FP7 PANDORA Persistent Autonomy for AUVs
3. Efficient Macroscopic Urban Traffic Models for Reducing Congestion

• Adaptive pattern Database Heuristics

The paper provides the detail outline of developing more accurate admissible heuristic functions. The goal is to provide more admissible search pattern without overestimating the cost of reaching the goal. The most important heuristic search algorithms include A*, iterative deepening-A*(ADA*) and depth-first branch-and-bound (DFBnB). All of these algorithms make use of heuristic evolution function h (n), which takes a state n and efficiency computes an estimate of the cost of an optimal solution from node n to a goal state. But this may not be cost effective and more accurate. To improve cost efficiency and accurate, addictive pattern is useful.

• Overview of the Paper

The research paper has been organized as follows

1. In section 2 represents previous work on designing more accurate heuristic functions, including pattern databases in general and statically-partitioned addictive pattern databases.
2. In section 3 represents the idea of dynamically-partitioned addictive pattern databases.
3. A section 4 represents the implementations and experiments on the tile puzzles.
4. A section 5 represents the two applications of addictive pattern databases including 4-peg Towers and Hanoi problem.
5. A section 6 represents finding a minimum vertex cover of a graph
6. A section 7 represents the general discussion on the methods
7. A section 8 suggests the further work and conclusions.

• Techniques introduced

The non-additive pattern database algorithm is applied to fifteen puzzles and Rubik's cube games.

In this games IDA* algorithm is used to search for an optimal solution to a particular problem instance. But one of the limitations of non-addictive database is that they don't scale upto larger problems. For example, since the Twenty-Four puzzle contains 25 different positions, a pattern database covering n tiles and the blank would require 25! /(25−n−1)! entries. A database of only six tiles and the blank would require over 2.4 billion entries. The non-additive pattern database algorithm takes the maximum of different pattern database values instead taking sum of their values, to get more accurate heuristic, without violating admissibility. To achieve this there are two algorithms

1. Statically-partitioned additive pattern databases
2. Dynamically-partitioned additive pattern databases

o Statically-partitioned additive pattern databases

The Statically-partitioned additive pattern databases are distinguished from dynamically-partitioned pattern databases. This algorithm same partition of the tiles is used for all states of the search. The key difference between additive databases and the non-additive databases is that the non-additive databases include all moves required to solve the pattern tiles, including moves of tiles not in the pattern group. In an additive pattern database, we only count moves of the tiles in the group.

But there are limitations in the Statically-partitioned additive pattern databases, this fail to capture fail to capture interactions between tiles in different groups of the partition. This issue is addressed using the Dynamically-partitioned additive pattern databases

o Dynamically-partitioned additive pattern databases

In dynamically-partitioned additive pattern databases is developed independently in the context of sliding-tile puzzles. This algorithm has different methods, they are

1. Computing the Pairwise Distances
2. Computing the Heuristic Value of a State
3. Triple and Higher-Order Distances

The overall this research document gives the different search algorithm to solve high end problems more accurately. For example

- Sliding-tile puzzles
- 4-Peg Towers of Hanoi Problem
- Vertex cover

• Result

The paper shows that, both static and dynamic additive pattern database heuristics improves the search more admissible. Apart from this algorithm improved in following areas

1. This is best known method for finding the optimal solutions to arbitrary instances these problems.
2. In this research studied similarities and the differences between the two versions of additive pattern databases
3. Studied dynamically partitioning the problem for each state of the search is most effective for large problems.
4. These algorithms avenues for future work on admissible heuristics for these problem domains.

• Reference

1. Journal of Artificial Intelligence Research 22 (2004) 279-318 by Ariel Felner, Richard E. Korf and Sarit Hanan,http://www.ise.bgu.ac.il/faculty/felner/research/jairpdb.pdf

EU-FP7 PANDORA Persistent Autonomy for AUVs

Per ref [3], PANDORA is persistent Autonomy, means going beyond what has been done before. PANDORA is creating a new class of AUVs that keep going under extreme uncertainty. AUVs that respond to system faults by doing what they can. AUVs that generate their own missions when idle. AUVs that act appropriately under unexpected environmental challenges per ref [2], such as an Offshore oil and gas fields

To perform the task there are following constraints

1. Energy Constraints
2. Inspection Outcomes
3. External Requests
4. Changing Environment Conditions

In this PANDORA, there are challenges to achieve inspection task within limited time and energy budget, so efficient paths are important. Path cost is determined not only by length, but by momentum at start and end and kinematics.

To achieve the solution, initially developed Random Coarse Roadmap Generation Algorithm from this coarse plan generated. Once coarse plan generated visibility points discovery and re-planning performed until reaches the final object.

Per Ref [1], To develop PANDORA, Robot Operating system (ROS) is used. ROS is a set of software libraries and tools used to build robotic systems. ROS has three parts: knowledge gathering, planning, and dispatch. The ROSPlan package provides a standard interface for PDDL planners in ROS. Per Ref [2], The planning system has following phase

1. Problem generation
2. Planner
3. Plan Parser
4. Plan Dispatch

The outcome was successful, with the AUV correcting all the valves. The ROSPLAN planning framework for ROS. Proposed this architecture as a standard solution to embedding a task planner in the PDDL family into a robotic system. This focused on providing a standardized solution for the integration, through ROS, of modern heuristic search planners with robotic systems.

Reference

1. https://www.aaai.org/ocs/index.php/ICAPS/ICAPS15/paper/view/10619/10379
2. http://kcl-planning.github.io/ROSPlan/tutorials/slides/AAAI17_PlanningAndRobotics.pdf
3. http://persistentautonomy.com/

• Efficient Macroscopic Urban Traffic Models for Reducing Congestion

The global growth in urbanization increases the demand for services including road transport infrastructure, presenting challenges in terms of mobility. In this scenario, optimizing the exploitation of urban road networks is a pivotal challenge. Existing urban traffic control approaches, based on complex mathematical models, can effectively deal with planned-ahead events, but are not able to cope with unexpected situations –such as roads blocked due to car accidents or weather-related events– because of their huge computational requirements. Therefore, such unexpected situations are mainly dealt with manually, or by exploiting pre-computed policies

The research presents a PDDL + formulation of urban traffic control, where continuous processes are used to model flows of cars, and show how planning can be used to efficiently reduce congestion of specified roads by controlling traffic light green phases.

In this implementation widely used system like SCOOT and SCATS, proposed after sensors were introduced, strategies are reactive to the given input from sensors and can control a set of connected intersections.

In this overcome from STRIPS based planning approaches to handle unexpected traffic conditions. To achieve this simplified model (S-Model) is used. They are

1. PDDL + Planning
2. PDDL + Formulation

One of the example of using above simplified model is “UPMurphi” planning system has been used for solving urban traffic control problems. The UPMurphi has been enhanced with higher end hardware to analyze real time sensors and generating the plans. The experiments are performed with two maps which has two intersections and 8 entry/exists points.

The experiments demonstrate the extent to which our approach is able to efficiently control traffic lights phases in order to cope with unexpected traffic conditions, and hence demonstrates its potential for increasing the resilience of network management.

The outcome of the experiment was successful, traffic green phases can be controlled by a planner for reducing network congestion, and designed a forward heuristic search in order to improve the performance of existing state-of-the-art planners. The performed experimental analysis shows that the proposed approach can effectively cope with unexpected traffic conditions.

Reference

1. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/download/11985/12079