The Grid-Based Path-Planning Competition

Nathan R. Sturtevant

■ *While there have been many papers published on path planning in grids, there has not been significant work on comparing existing approaches, and it is difficult to evaluate new work in comparison to existing work. After creating a public repository of grid-based path-planning problems I created the Grid-Based Path-Planning Competition (GPPC) to facilitate these comparisons. This article describes the motivation and design of the competition, as well as plans for the future of the competition.*

A large number of papers have been written that deal with path planning in grids. These papers contain a wide variety of techniques and operate under a wide variety of constraints. All of them offer significant improvement over and beyond a basic A* search. But there has been no unified study comparing techniques and measuring the trade-offs implicit in the approaches. While this comparison could be performed by a single individual implementing existing approaches, the overhead of implementing all existing techniques is significant, and the final payoff is small relative to the amount of work performed. Furthermore, there is no guarantee that all implementations will be properly optimized, or that edge cases are handled correctly.

This leads to two significant issues. First, it is difficult to identify which existing approaches are best for a given task, and second, it is difficult to evaluate new approaches and assess their strengths and weaknesses. Related to this is the problem of metrics. There are many metrics that can be optimized, such as planning time, plan quality, preprocessing

 ISSN 0738-4602

time, and others. When different papers optimize different metrics, it can be difficult to compare approaches directly.

All of this might be surprising, given that grids are one of the easiest formats to get started with, and people have been planning on grids for decades (Thrun and Bücken 1996). Given these observations, some effort has been devoted to address these issues and improve the quality of new research. The work in this area has had two thrusts. The first thrust has been in collecting an archive of grid maps (Sturtevant 2012). The existence of an accepted repository of maps and testing problems makes it easier for authors to compare their work using a common test set of problems. These problems come from a variety of human-designed and artificially designed sources. The second thrust is the organization of the Grid-Based Path-Planning Competition (GPPC), which has been running since 2012. The focus of this article is on the GPPC — how the event is organized and future plans for the event.

Competition Organization

The Grid-Based Path-Planning Competition has been running for two years on an annual basis, with events in 2012 and 2013, and a planned event in December, 2014. I considered multiple different venues for hosting the competition, and it has been thus far affiliated with the Symposium on Combinatorial Search (SoCS). In 2012 SoCS encouraged participation in the competition as part of its call for papers, and entrants into the competition were able to write short abstracts that were published at SoCS. This option wasn't offered in 2013, although the competition results were still presented at SoCS in 2013.

Entries in the competition are required to open up their source code to the community. As such, the source code for each year's entries have been placed in a Google code repository.

The competition currently has a single organizer, with informal feedback from a number of researchers that have performed research in the area. This is primarily a function of the size of the event and the computational resources required to run it, which are relatively small compared to some competitions. In the future it is possible that a formal advisory board will be formed as is needed with more participants and tracks.

Competition Description

The competition is designed around a C++ interface. The interface is relatively simple, requiring four functions to be implemented by the competitor, one of which just requests the name of the competition entry. The other three calls ask the entry to (1) preprocess the map, saving the results, (2) prepare for search, potentially loading preprocessed data, and (3) compute a path between two points. This last call is designed to allow incremental path computation. If the entire path is not returned by an entry, the call will be repeated until the full path is returned. The competition is run by first asking all entries to preprocess all maps, and then later asking them to load the data and answer our queries.

The map and scenario files, which are used to specify maps and problems, are loaded by a master program and passed to a particular entry through the API. The file formats have been used for many years as part of the HOG2 framework.[1] Actions on these maps are restricted to moving to the eight neighboring cells, and the underlying cost is the straight-line distance (1 or $\sqrt{2}$).

The competition itself is designed to allow as many comparisons as possible, and so the following statistics are collected:

The total time required for finding a complete path. This is one of the most common measurements used for evaluating a path-planning system.

The time required to find the first 20 steps of a path. This metric is important in real-time systems, where an agent can start moving as soon as a sufficiently long initial path has been found.

The maximum time for returning any set of actions. This is another metric important for real-time systems, when the amount of time for each planning step is limited.

The memory required for preprocessing. Memory can almost always be used to speed things up; however, mobile, embedded, or other memory-constrained applications have hard limits on how much offline or online storage must be used. (Online storage is that which is dynamically allocated for search, while offline storage is precomputed and stored or shipped with an application.)

The time required for preprocessing. Preprocessing can usually be performed in an offline phase, amortizing its cost over many problem instances, but in some applications the cost of preprocessing matters.

The memory required at run time. This is related to the previous metrics, just measuring online memory instead of offline memory.

The quality of the paths found. Path quality is an important metric, but can be misleading. It is measured in two ways. The first is comparing the sum of the length of all paths found to the sum of the optimal path lengths. This measurement gives more weight to long paths. The second is by comparing the average ratio of path lengths found. This measurement gives equal weight to all paths.

In past competitions the time and disk space used during preprocessing have been limited, although this restriction is now being lifted.

Competitors and Results

A total of 12 teams have submitted entries into the competition over the last two years, and three of these entries have entered multiple variants that optimize metrics slightly differently. One difficulty in the first year was that many competitors had minor errors that caused the programs not to handle all problems in the test set. This improved in the second year, although there were fewer entries. There was a drop in entries in 2013 because several researchers were encouraged to submit older research projects in 2012, which were then not updated in 2013. There were also several participants who did not finish entries in time for the 2013 deadline and were forced to withdraw. I highlight a few approaches here, which are notable.

Ken Anderson submitted two tree-based approaches in 2012 and 2013. These approaches are extremely fast — the 2013 entry was able to solve 1.74 million problems in 50.9 seconds (29 microsecnds per path). The drawback is that the approach is highly suboptimal. When looking at the length of all paths summed together, the approach was approximately 20 percent worse than optimal. But, when looking at the average suboptimality on a per path basis it was more than two times optimal on average. In 2012 Álvaro Parra, Álvaro Torralba, and Carlos Linares López submitted an entry that had very good real-time performance. It was one of the few entries that took advantage of incremental computations, allowing it to return the first 20 moves in approximately 8 microseconds on average. The best optimal approach was submitted by Tansel Uras, Carlos Hernández, and Sven Koenig in both 2012 and 2013 (Uras, Koenig, and Hernández 2013). It isn't faster than the compressed path database (CPD) approach of Adi Botea (Botea 2011), but it can handle the larger problems, which CPD cannot.

Lessons

I have learned several things from running the competition thus far. First is the importance of publicity. Although we have advertised broadly within our own research community, we have not reached broader communities that are doing related work. We have found very recent publications from researchers in other fields making state of art claims that need to be validated against data in the competition, yet they have not been. We hope that more publicity will help encourage reviewers to ask for comparisons, as they are now relatively easy to make, especially since all entries are available.

Second is the difficulty of persuading people to enter their work. I have talked to a number of people who have work they could submit to the competition, yet they have not. This is often because the graduate student doing the research is not willing to do the work required to submit an entry. I suspect there is also a psychological influence, in that there can be an aversion to finding out that your own work doesn't compare as well as you thought it would.

To my knowledge, the approaches submitted to the competition have yet to be widely used by researchers or practitioners in other fields. Again, this will be helped by more publicity, which is a motivation in writing this article.

Future Competitions

Given these results, the competition will be moving forward as follows. First, I recognize that the existing track represents a problem definition that is much simpler than many real-world problems. Despite this, the metrics collected allow for the evaluation of a richer set of algorithms than those that have been entered into the competition. For example, there has been a large amount of work on real-time agent-centered search (Koenig 2011), none of which has appeared in the competition. So I will be individually encouraging researchers to submit a broader class of algorithms to the basic competition.

To make this easier, the submission process is being opened up, thus allowing submissions to be entered at any time during the year when they are ready. These results will be collected and officially announced at the competition each year, but researchers will be able to get results from running their program on our server at almost any time during the year.

The second step is to introduce a new competition track. This track will introduce two changes to the maps. The first change is that the maps will be weighted. The cost of moving through each type of terrain will be weighted by a constant. Weights are commonly used in real-world planning to model obstacle uncertainty or tactical considerations that should be taken into account while planning. The second change is that the maps themselves will no longer be static. After a variable number of pathfinding requests, an entry will be provided with a set of changes to the map. The entry will be able to perform updates to its representation, and then pathfinding requests will resume. This will once again allow entries in the competition to address many more problem domains.

Conclusions

I have described the grid-based path-planning competition (GPPC), a new competition which seeks to facilitate comparison between approaches that have traditionally lacked sufficient comparisons. The hope is that this competition will improve the quality of research performed within this field by providing standard performance benchmarks and encouraging authors working on grid-based path planning to make their approaches publicly available.

In domains with standardized problem formats and competitions for evaluating programs on those problems, it is easy for researchers outside of the field to come and use the results of that research as part of their own work. The hope is that the GPPC and its associated benchmark problems will do something similar for grid-based pathfinding, providing accurate evaluation of existing approaches, as well as standardized implementations that can be easily adapted for use elsewhere. I encourage everyone interested in this area to look at the competition results and consider submitting programs to the competition in the future.

Note

1. See code.google.com/p/hog2.

References

Botea, A. 2011. Ultra-Fast Optimal Pathfinding Without Runtime Search. In *Proceedings of the Seventh Artificial Intelligence and Interactive Digital Entertainment Conference,* 122–127. Palo Alto, CA: The AAAI Press.

Koenig, S. 2011. Agent-Centered Search. *AI Magazine* 22(4): 109–132.

Sturtevant, N. 2009. Benchmarks for Grid-Based Pathfinding. *Transactions on Computational Intelligence and AI in Games* 4(2):144–148. dx.doi.org/10.1109/TCIAIG.2012.2197681

Thrun, S., and Bücken, A. 1996. Integrating Grid-Based and Topological Maps for Mobile Robot Navigation. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 944–950. Menlo Park, CA: AAAI Press / The MIT Press.

Uras, T.; Koenig, S.; and Hernández, C. 2013. Subgoal Graphs for Optimal Pathfinding in Eight-Neighbor Grids. In *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling,* 224–232. Palo Alto, CA: AAAI Press.

Nathan Sturtevant is an assistant professor in the Computer Science Department at the University of Denver. His scientific research focuses on artificial intelligence and search in single-agent and multiagent settings. This research includes work on parallel large-scale search, work on real-time environments, such as movement planning in video games, and work in competitive environments, such as traditional games. He has implemented his research in the commercial game Dragon Age: Origins, and works with industry on a variety of research problems. Sturtevant received his Ph.D. from the University of California, Los Angeles in 2003.