Subgoaling in Real-Time Heuristic Search

Vadim Bulitko

http://ircl.cs.ualberta.ca
Acknowledgements

- Ramon Lawrence (UBCo)
- Yngvi Bjornsson (RU)
- Valeriy Bulitko (AU)
Outline

- Heuristic search
- learning in heuristic search
- How to avoid on-line learning
  - D LRTA*
  - kNN LRTA*
- Work in progress
Algorithm

- A* computes an entire path before taking its first action
- does not scale well with problem size
- Various techniques/tricks are used to mitigate
  - weighted A*
  - PRA*
- We want *true* real-time planning
  - time per move is constant-bounded
    - constant is independent of the number of states
These small, suicidal creatures are filled with explosive chemicals and corrosive acids.

- Conduct limited lookahead around the agent
- Update heuristic function
- Take one action

A special case of real-time dynamic programming (RTDP)

- Similar to simple RL algorithms (e.g., Q-learning)
- Planning time per action independent of number of states
- Poor paths
The Problem

- Heuristic is inaccurate
- misleads the agent
- fixing the heuristic takes a long time

LRTA*. Korf, 1990
Solutions

- Learn heuristic function more efficiently: LRTS, etc.

- Learn heuristic in a smaller abstract space: PR LRTS

- Learn less by starting with a better initial heuristic
  - Pre-compute a better heuristic
  - Choose closer goals: D LRTA*
    - Heuristic is better closer to a goal
For every *abstract* state pair
- compute an optimal path
- store *first gateway* state on this path as subgoal

- Complex algorithm
- Reasonable precomputation time
- Memory can still be excessive
  - square of abstract states
For $N$ random state pairs:

- compute an optimal path
- compress the path into a series of subgoals

Simple(r) algorithm

Reasonable precomputation time

Reasonable memory $O(N)$
Open Questions

- kNN LRTA* parameters
- number of records
  - how many to guarantee certain suboptimality?
- placement of records
  - random/non-random? How?
- selection of the best record on-line
  - optimal selectors?
Work in Progress
Open Questions

- kNN LRTA* parameters

- number of records
  - how many to guarantee certain suboptimality?

- placement of records
  - random/non-random? How?

- selection of the best record on-line
  - optimal selectors?
Record Selection

![Graph showing the relationship between database size (%) on the x-axis and 1st HCable record suboptimality (%) on the y-axis. The graph includes lines for different database sizes: 29583, 45513, 67183, and 94578. Each line represents a different database size, with the 29583 line being the highest and the 94578 line being the lowest. The x-axis ranges from 1 to 11, and the y-axis ranges from 8 to 20.]

Vadim Bulitko
November 23, 2011
10% test problems x 40 folds, total time 65279.3s
What we know (empirically)

- With random records, the number of records needed to guarantee a solution suboptimality grows linearly with the total number of problems even with a perfect selector.
Open Questions

- kNN LRTA* parameters
- number of records
- how many to guarantee certain suboptimality?

- placement of records
- random/non-random? How?

- selection of the best record on-line
- optimal selectors?
65792 test problems on game map, total time 2705.3s

- random best 10%
- random first 10%
- heurError best 10%
- heurError first 10%

Random database works better
Record Placement

random 10%

heurError 10%
What we know (empirically)

- With random records, the number of records needed to guarantee a solution suboptimality grows linearly with the total number of problems even with a perfect selector.

- Random records are better than hardest records.
Open Questions

- kNN LRTA* parameters
- number of records
  - how many to guarantee certain suboptimality?
- placement of records
  - random/non-random? How?
- selection of the best record on-line
  - optimal selectors?
Record Selection

Record 1: dist from problem 1, record cost 11.0, solution cost 13.8
Spearman correlation between the record distance and the resulting solution quality

Database size (%)
What we know (empirically)

- With random records, the number of records needed to guarantee a solution suboptimality grows linearly with the total number of problems even with a perfect selector.
- Distance between a record and a problem has no correlation with the resulting suboptimality.
- Random records are better than hardest records.
Theory

"I think you should be more explicit here in step two."
The Big Picture
Probability of Depression

![Graph showing the relationship between effective obstacle density and probability of depression]
Suboptimality w.r.t. r

Trials 5, density 45%, fraction of test problems 25.0%, total time 2624.2s

Empirical
Theoretical
Map side

Expected $h^*$ of hill-climbable problems

45.0% density, 1% problems, 1000 trials, 11226.6s

Empirical

$(\alpha(1-\delta))^x$
Mean $h^*$

5 trials, 2316.5s

Vadim Bulitko
November 23, 2011
1% test problems x 100 folds, 25 x 25 from maps/wc3maps/divideandconquer.map, total time 36010.0s

- Best
- 1st HC
- Random
- Random Theory
- HB
Limitations

- Random records only
  - off-line
  - on-line
- 2D gridworlds, 4-connected
- Sampled constants
  - mean h*
  - suboptimality of LRTA*
- HCability