Speeding up search-based motion planning using Expansion delay heuristics

Jasmeet Kaur, Ishani Chatterjee, Maxim Likhachev

Robotics Institute, Carnegie Mellon University

1. Motivation and Contribution

- Local minima regions are where the heuristics are weakly correlated with the true cost-to-goal.
- Given a graph, identify these regions of local minima and compute a heuristic function to circumvent these.
- Inadmissible heuristics are computed by learning **Expansion delay** values.
- Two-step Algorithm : First, Expansion Delay values are learnt. Second, heuristic function is computed in an abstract space using learnt Expansion Delay values.

2. Abstract Space

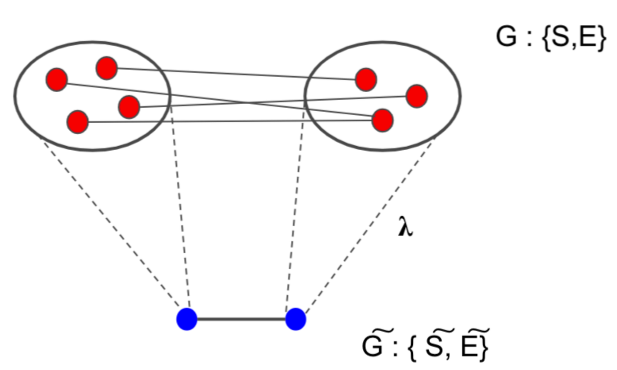

Figure 1: Abstract Space

Original Graph : G = $\{S, E\}$ and Abstract Space is also a graph denoted as $\tilde{G} = \{\tilde{S}, \tilde{E}\}$.

- $\lambda : S \to \tilde{S}$ is a many-to-one mapping representing the projection of each state in S to a state in abstract space $\tilde{S}$.
- For example :

  3D/2D : S = [x, y, θ] and $\tilde{S}$ = [x, y]

3. Expansion Delay

- Let e(s) be the total number of nodes that have been expanded before s is expanded during a search in graph G.
- **Expansion Delay** Δe = (s,s') where s is the parent node and s' is the successor is :

  Δe(s,s') = e(s') - e(s)
- Let A be the set of state-pairs (s,s') in G that project to (s,s') in $\tilde{G}$. For each edge $(\tilde{s}, \tilde{s}')$, $\Delta\tilde{e}(\tilde{s}, \tilde{s}')$ is the expected value of $\Delta e(s, s')$ over the set A :

  $\Delta\tilde{e}(\tilde{s}, \tilde{s}') = \mathbf{E}(\Delta e(s, s'))$

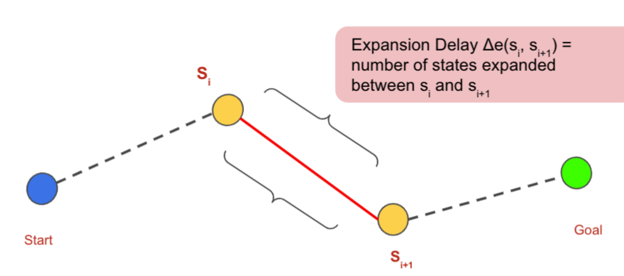

Figure 2: Expansion Delay between two states on a solution path

4. Heuristic Computation

- Compute a heuristic function that guides the search in graph G along paths that reduce the total Expansion Delay.
- If N($\tilde{s}$,$\tilde{s}'$) is the value of expansion delay predicted for an edge in abstract space $\tilde{G}$, the edge-costs are assigned as :

  $\tilde{c}(\tilde{s}_i, \tilde{s}_j) = N(\tilde{s}, \tilde{s}')$
- h($\tilde{s}$) is cost of path in $\tilde{G}$ using backward Dijkstra.

5. Experimental Setup

- Evaluated on a humanoid footstep domain.
- $\tilde{G}$ is a 2D 8-connected grid that corresponds to the position of the center of the robot in the map.
- λ projects the 2 feet-positions into the 2D grid by computing their mean.

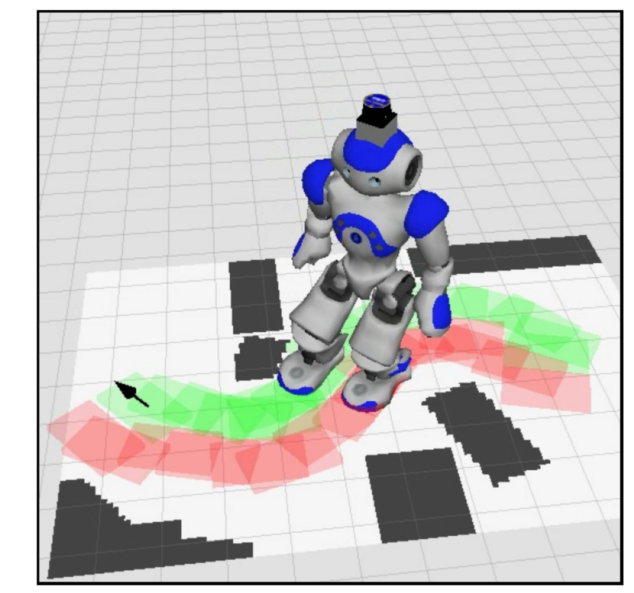

Figure 3: A humanoid footstep path in an environment

6. Results and Discussion

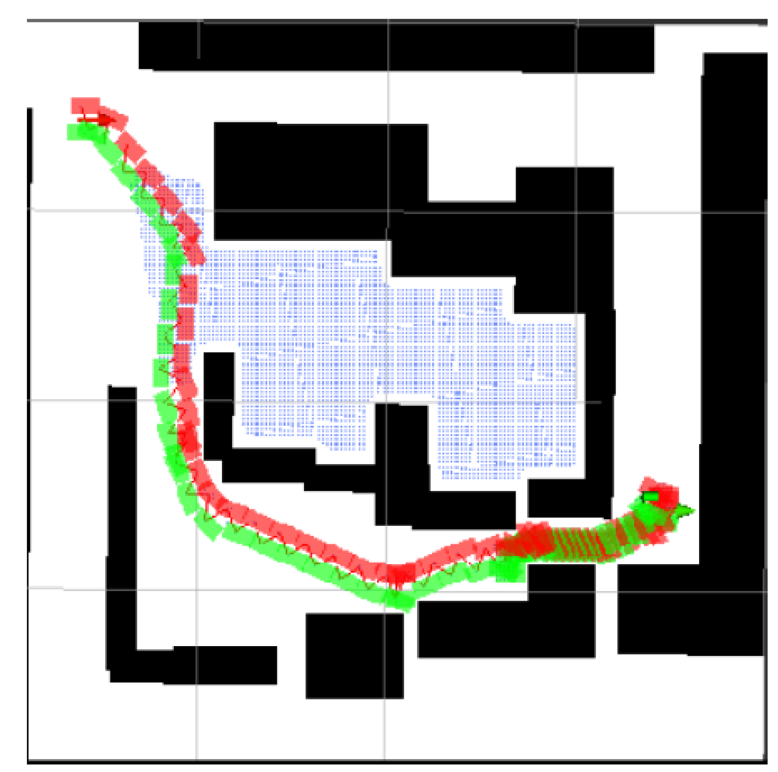

Figure 4: A path found using baseline heuristic h_b

Figure 5: A path found using heuristic computed using Expansion delay, h_{ed}

- Comparison with the baseline heuristic h_b which is the cost of the optimal path with regular euclidean 2D edge-costs.
- Solution size and quality are sub-optimal but there is significant reduction in number of expansions.

Heuristic	No. of Expansions	Planning Time(s)	Solution Cost
h_b	477777±75685	17.14±4.11	9226±754
h_{e-d}	183174±120187	5.09±3.47	18951±4393

Figure 6: Comparison of h_b and h_{ed}

7. Future Work

- Finding a generalization that transfers learning across different environments effectively.

8. References

- Vats, S.; Narayanan, V.; and Likhachev, M. 2017. Learning to Avoid Local Minima in Planning for Static EnvironmentsIn ICAPS, 572–576.