Heuristic for Planning with Action Costs Revisited by Emil Keyder and Héctor Geffner

Behrouz Tajoddin

Universität Basel

December 6, 2013

Overview

- Overview of HSP
- Set-additive heuristic
- Experimental results
- TSP problem

Planning Tasks in Strips formulation

- A language for expressing planning problems
- Defined as a quadruple $\langle F, I, O, G \rangle$
  - F is the set of atoms
  - I and G are the initial and goal conditions
  - O is a set of actions with precondition, add and delete lists
    - Each action has a cost of one
- A plan (π) is a sequence of actions
  - $\pi = a_1, ..., a_n$
  - $cost(\pi) = |\pi|$

Planning Tasks in Strips formulation (generalized)

- A language for expressing planning problems
- Defined as a quadruple $\langle F, I, O, G \rangle$
  - F is the set of atoms
  - I and G are the initial and goal conditions
  - O is a set of actions with precondition, add and delete lists
    - Each action has a different cost
- A plan (π) is a sequence of actions
  - $\pi = a_1, ..., a_n$
  - $cost(\pi) = \sum_{i=1}^{n} cost(a_i)$

Relaxed Planning Tasks

- Given a planning task $P = \langle F, I, O, G \rangle$
- The relaxation P' of P is defined as $P' = \langle F, I, O', G \rangle$
- $O' = \{(pre(o), add(o), \emptyset) | (pre(o), add(o), del(o)) \in O\}$
- Ignoring the delete list
- Relaxed plan
- Commonly used technique for heuristic estimation
- The heuristic is admissible
- The optimal relaxed plan is well-informed but intractable

HSP

- Search the state space using a heuristic
- Independance assumption between plans for individual atoms
- Estimates the cost of achieving a set of atoms as the sum of the costs of achieving individual atoms separately.

$$h_{add}(G, s) = \sum_{g \in G} h_{add}(g, s)$$

HSP (continued)

$$h_{add}(p,s) = \begin{cases} 0 & \text{if } p \in s, \\ h(a_p, s) & \text{otherwise} \end{cases}$$

$$a_p = argmin_{a \in O(p)} h(a,s)$$

$$h(a,s) = 1 + \sum_{q \in Prec(a)} h_{add}(q,s)$$

- $O(p)$ stands for actions that add the atom p
- a_p is the best support of atom p in s
- $h_{add}(p,s)$ is an estimate of the length of the solution for achieving the atom p from s

HSP with cost

$$h_{add}(p,s) = \begin{cases} 0 & \text{if } p \in s, \\ h(a_p, s) & \text{otherwise} \end{cases}$$

$$a_p = argmin_{a \in O(p)} h(a,s)$$

$$h(a,s) = cost + \sum_{q \in Prec(a)} h_{add}(q,s)$$

- $O(p)$ stands for actions that add the atom p
- a_p is the best support of atom p in s
- $h_{add}(p,s)$ is an estimate of the cost of achieving the atom p from s

Independance assumption in HSP

- Normally false as the delete-relaxation
- Reaching some sub-goals can make it easier to reach the goal
  - Because of the existence of positive interaction between atoms
- An example (from Hoffmann & Nebel 2001):

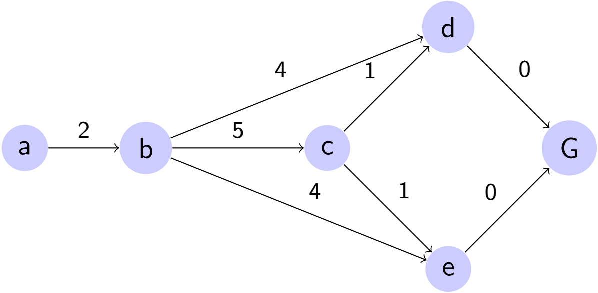
name		(pre,	add,	del)
$opG1$	$=$	$(\{P\},$	$(\{G_1\},$	$\emptyset)$
$opG2$	$=$	$(\{P\},$	$(\{G_2\},$	$\emptyset)$
opP	$=$	$(\emptyset,$	$(\{P\},$	$\emptyset)$

- The goal state is $\{G_1, G_2\}$
- The distance to the goal is estimated to 4 but solvable in 3 steps
- Additive heuristic is not admissible
- Overcounting of actions is the source of error

How to solve the problem?

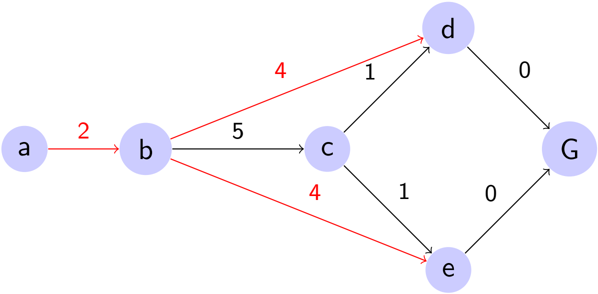
- The idea is to collect the best supporters recursively backward from the goal
- Keeping track of all the best supporters

How to solve the problem? (Example)



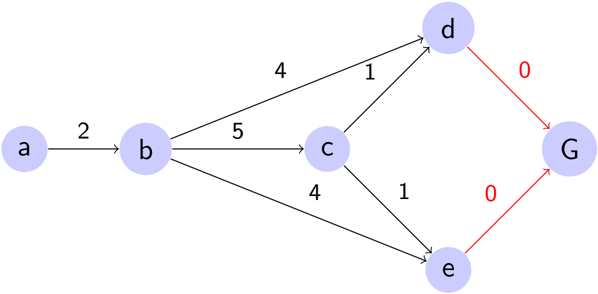
[1] Example is taken from a tutorial given by Emil Keydar at ICAPS 2009
http://icaps09.uom.gr/tutorials/tutorials.htm

How to solve the problem? (Example)



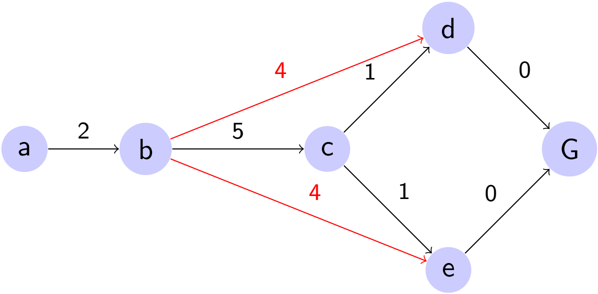
- Additive heuristic: $2 + 4 + 2 + 4 = 12$

How to solve the problem? (Example)



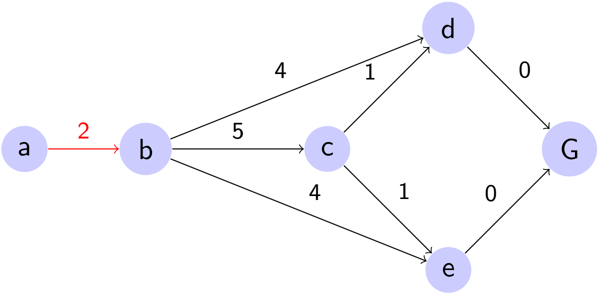
- New heuristic: $0 +$

How to solve the problem? (Example)



- New heuristic: $0 + 4 + 4 +$

How to solve the problem? (Example)



- New heuristic: $0 + 4 + 4 + 2 = 10$

Set-additive Heuristics

- Set-additive heuristic associates with every atom a relaxed plan $\pi_a(G; s)$
  - Is a set of actions
  - Replaced the heuristic value in HSP
- The heuristic defined as $h_a^s(G, s) = Cost(\pi_a(G; s))$
  - $Cost(\pi_a(p; s)) = \sum_{a' \in \pi_a(p;s)} cost(a')$
- How to define $\pi_a(p; s)$ in mathematic formulation?
- The same formulation as HSP
  - Instead of propagating the heuristic values the idea is to propagate the supports
  - Combine supports with a set-union instead of combining the cost with sum operator

Set-additive Heuristics (cnt'd)

- $Cost(\pi_a(p; s)) = \sum_{a' \in \pi_a(p;s)} cost(a')$
- $\pi_a(p; s) = \begin{cases} \{\} & \text{if } p \in s, \\ \pi_a(a_p; s) & \text{otherwise} \end{cases}$
- $a_p = argmin_{a \in O(p)} Cost(\pi_a(a; s))$
- $\pi_a(a; s) = \{a\} \cup \{\cup_{q \in Prec(a)} \pi_a(q; s)$
- The supports for joint preconditions are combined with the set-union operator
- So, no duplicate action! Actions count once

HSP

$$h_{add}(p,s) = \begin{cases} 0 & \text{if } p \in s, \\ h(a_p, s) & \text{otherwise} \end{cases}$$

$$a_p = argmin_{a \in O(p)} h(a,s)$$

$$h(a,s) = 1 + \sum_{q \in Prec(a)} h_{add}(q,s)$$

- $O(p)$ stands for actions that add the atom p
- a_p is the best support of atom p in s
- $h_{add}(p,s)$ is an estimate of the length of the solution for achieving the atom p from s

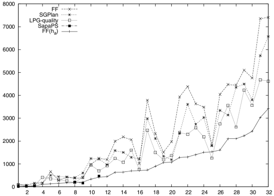
FF Planner

- An abbreviation for Fast Forward
- Same as HSP, search the state space using a heuristic
- Same as HSP, uses delete relaxtion planning task for heurisitc estimation
- Uses an *enforced* form of hill-climbing
- Helpful actions
  - Those applicable actions that add at least one goal at the lowest layer of the relaxed solution
  - $H(S) := \{o | pre(o) \subseteq S, add(o) \cap G_1 \neq \emptyset\}$

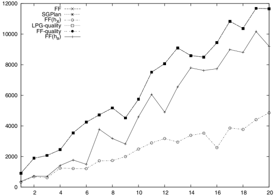
Experimental Results

- Experiments performed in 11 domains
  - 5 taken from IPC3 Competition
  - In the other 6, the length of the solution did not correlate with the cost
- Overall achieved better quality plans

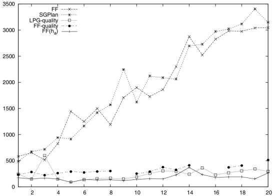
Experimental Results



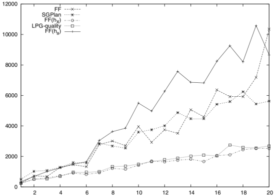
(a) Delivery



(b) Soft goal version of S. Rover domain



(c) Minimum Spanning Tree



(d) Hard goal version of S. Rover domain

The TSP heuristic

- There are n rocks $r_1, .., r_n$ to be picked up at locations $l_1, .., l_n$
- Having more general labels for each state
- A multivalued variable X $(at(l_1))$
- In each label $\pi(p, s)$ keeps two disjoint sets:
  - A set of actions that do not affect X
  - The set of X-*atoms* required as preconditions by the other actions
- $h_X(s) = Cost_X(\pi(G, s))$
- $Cost_X(\pi) = Cost(\pi \cap \bar{X}) + Cost_{TSP}(\pi \cap X)$

Experimental Results (Runtime)

- Not better than $FF(h_a)$
  - Computing h_a is costly
  - Longer plan may have lower costs which requires more search node and heuristic evaluation
- Also slower than additive heuristic

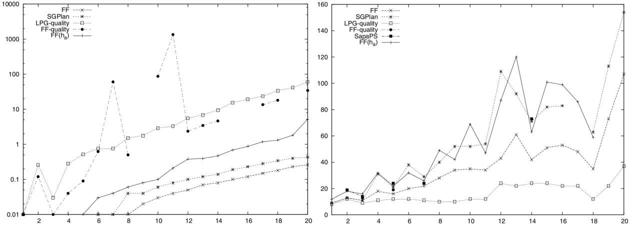


Figure: Left: Planning time, Right: Length of plans

Concolusion

- In h_s^a Labels are propagated rather than numbers
- h_s^a is generally better informed than h_{add}
- h_s^a cost-sensitive in compare to h_{FF}
- But more expensive to compute (Sum operator vs. Set-union operator)

Questions?