

disjunctive action landmarks

landmark = something that holds for every plan

disjunctive action landmarks: every plan contains either action c or action d
heuristics based on them for the delete-relaxed problem

[note] The heuristics described in this page is based on disjunctive action landmarks, a particular form of landmarks. It is an heuristics for delete-relaxed problems.

delete-relaxed problem

assume no negative precondition
remove delete effects

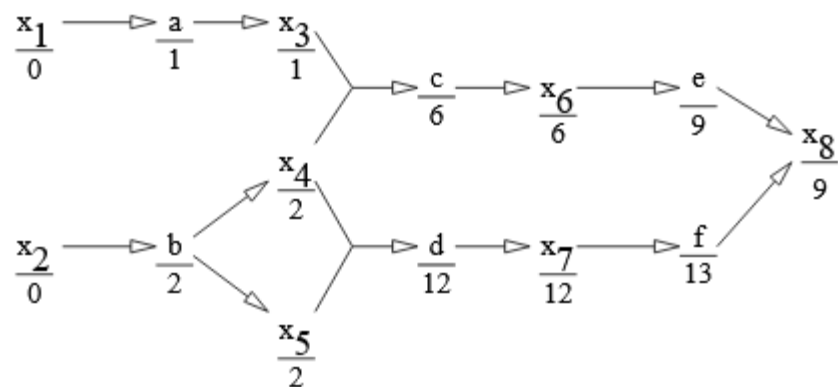
build the graph of variables and actions

maximal heuristics h^{\max} admissible but not precise

obtain a better admissible heuristics

[note] This is one use of one particular type of landmarks, the disjunctive action landmarks. Other uses of landmarks are briefly described at the end of this page.

maximal heuristics, in retrospect



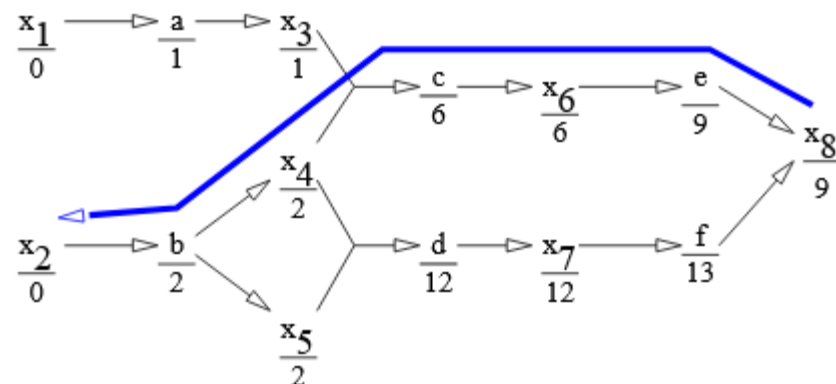
graph of variables in a delete-relaxed problem
what does the maximal heuristics do?

e and f both make x_8 true
alternative actions: choose the cheapest

x_3 and x_4 both needed to execute c
preconditions: choose only the one of maximal cost



what the maximal heuristics overlooks



in retrospect: one path from the goal to the initial state

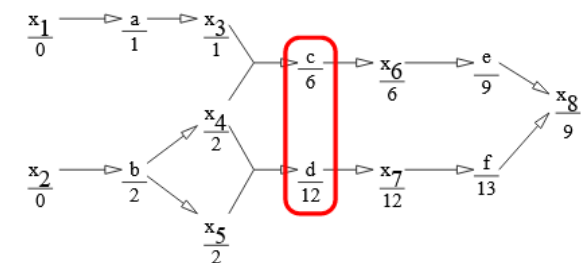
disregard the other paths \Rightarrow underestimate the costs

solution: **keep into account actions in different paths**

done by:

- critical path (next page)
turn the line into a bounded-width tree
- disjunctive action landmarks (this page)
sets of actions in different paths

disjunctive action landmarks



graph of variables of a delete-relaxed problem

actions c and d deleted: no way to reach the goal

either c or d is in every plan

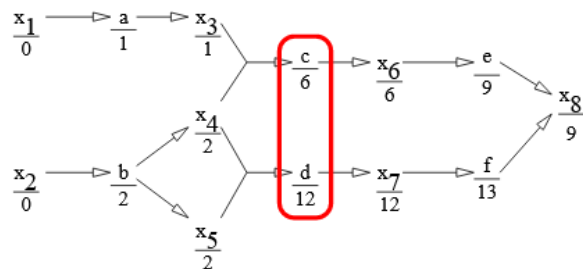
the set $\{c, d\}$ is a disjunctive action landmark

keeps into account alternative roads to the goal

[note] A disjunctive action landmark is defined as a set of actions such that every plan contains at least one of these actions. In other words, no plan exists that contains no action in the landmark. In the figure, either c or d is in every plan, since cutting (deleting) both actions makes the goal unreachable. Therefore, $\{c, d\}$ is a disjunctive action landmark.

As evident from the figure, a disjunctive action landmark contains actions from alternative paths to the goal.

from landmark to heuristics, first attempt



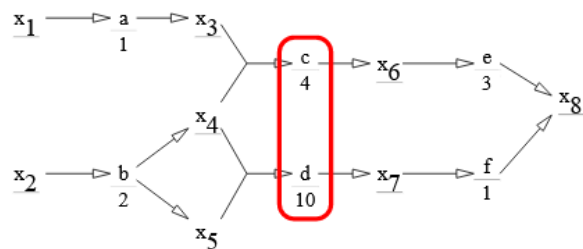
landmark $\{c, d\}$: every plan contains either c or d

heuristics: minimal cost of action in a landmark

in this case: minimal is 6

cost of goal with maximal heuristics h^{\max} was 9, better

from landmark to heuristics, second attempt



use only action costs

do not incorporate cost of making preconditions true

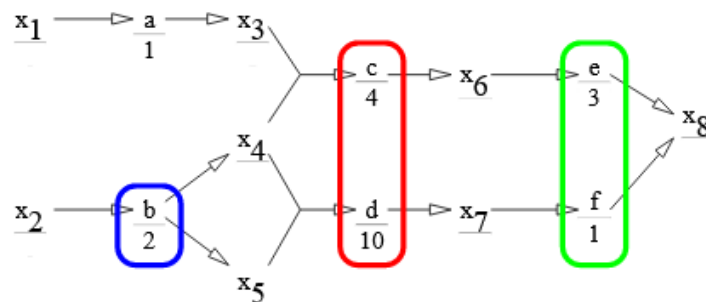
cost of landmark $\{c, d\}$: lowest cost of action

why: every plan contains *either* c or d

heuristics: 4

even worse than before?

from multiple landmarks to heuristics



for each landmark, each plan contains at least an action

cost is at least $2+4+1 = 8$

still lower than maximal heuristics, 9

but...

remember cost partitioning?

make some copies of the problem

distribute cost of each action among the copies

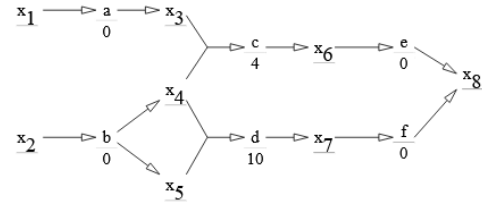
sum of cost of each action over all copies = original cost of action

heuristics for each copy

heuristics for original problem is sum of heuristics

[note] Cost partitioning is described in the page about the [properties of heuristics](#).

an example copy



put all cost of c and d here
is zero in the other copies

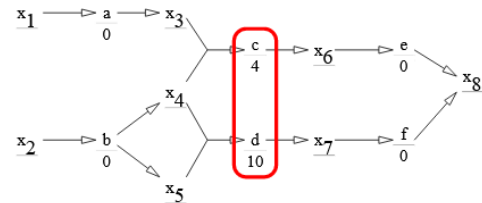
all other actions have cost 0 here
their cost can be put in other copies

select a landmark

obvious choice: prefer larger heuristics = landmark of maximal cost

[note] Since the landmark heuristics are admissible, the aim is to obtain a value that is as large as possible. Therefore, it is better to choose a landmark of large cost.

a landmark for a copy



$\{c, d\}$ is still a landmark

costs do not affect landmarks: without c and d , no way to reach the goal

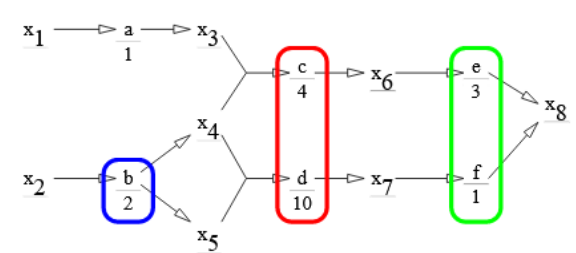
cost is of this landmark is 4

the copies and their landmarks

landmark = deleting all its actions, goal unreachable
costs not involved in the definition of landmarks

copies only differ on the costs
they have exactly the same landmarks

from landmarks to heuristics, cost partitioning



three copies
landmarks are the same

- first copy: b whole cost, other actions zero
select landmark {b}, cost 2
- second copy: c and d whole cost, other actions zero
(shown in previous slides)
select landmark {c, d}, cost 4
- third copy: e and f whole cost, other actions zero
landmark {e, f}, cost 1

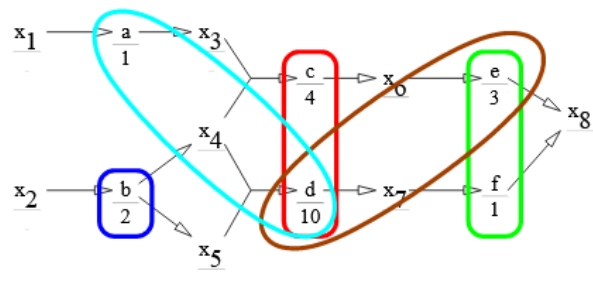
cost of each actions distributed among the copies \Rightarrow **additivity**
heuristics is: $2+4+1 = 7$

not enough...

[note] The landmark are the same as explained in the previous slide.

Since the cost of actions has been distributed among the copies, so that the sum of the cost for each action is equal to the cost of the original action, summing the admissible heuristics for each copy results in an admissible heuristics for the original problem.

from landmarks to heuristics, overlapping landmarks



five copies

landmarks are the same

- first copy: whole cost of b
landmark {b}, cost 2
- second copy: whole cost of a, 1 of d
landmark {a, d}, cost 1
- third copy: whole cost of c, 4 of d
landmark {c, d}, cost 4
- fourth copy: cost 1 of e, whole of f
landmark {e, f}, cost 1
- fifth copy: cost 5 of d, 2 of e
landmark {d, e}, cost 2

cost of each actions distributed among the copies \Rightarrow **additivity**

heuristics: $2+1+4+1+2 = 10$

better than max heuristics (9)

all this for...?

disjunctive action landmarks heuristics improves over maximal heuristics

still admissible, but keeps into account multiple alternative paths

only effective with additivity (cost partitioning)

[note] The landmarks shown in the previous slides are disjunctive action landmarks.

using disjunctive action landmark

previous slides had magic:

- landmarks come out of the blue
- distribution of the cost of actions as well

algorithm needed to automate the process:

find landmarks, find distribution of cost

LM-cut

principles:

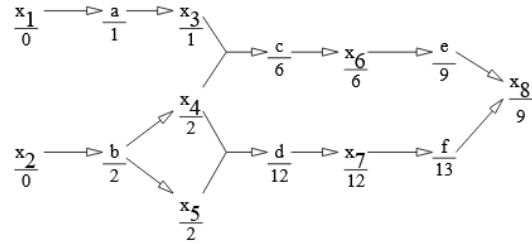
- an action needs all its preconditions to be true
including the most costly one
- do cost partitioning on the fly
find a landmark \Rightarrow create a copy for it

implementation:

1. temporarily restrict each action to have only its most expensive precondition
2. find a cut of cost $m > 0$ to make the goal unachievable
 \Rightarrow this is a landmark
3. decrease cost of all actions of the landmark by m
4. repeat until goal reachable with cost 0

what does step 3 do?

LM-cut: step 1



compute maximal heuristics

choose a preconditions of maximal cost for each action:

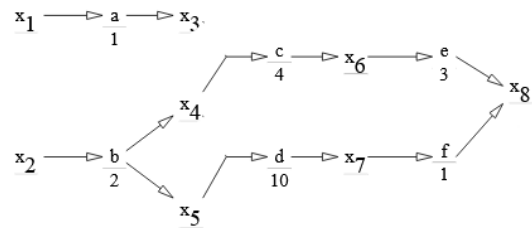
x_4 for c

x_5 for d (also x_4 was possible, tie)

temporarily erase the other arrows to c and d

[note] Before starting the algorithm, actions with no preconditions are turned into actions with a single precondition, a new variable that is true in the initial state and never made false.

LM-cut: steps 2 and 3



cost of actions without preconditions

find a non-zero cut in the graph

for example, $\{e, f\}$

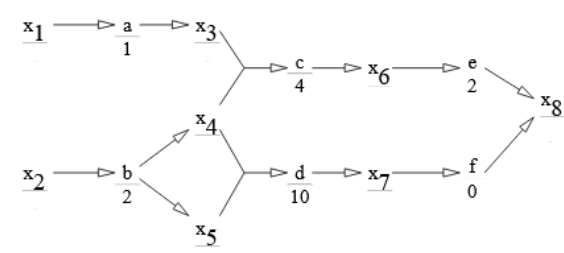
cost is $\min(3, 1) = 1$

first landmark: $\{e, f\}$

decrease cost of e and f by 1

[note] The cut needs not to be one of maximal cost, nor even to be irredundant. All that is required is that it has no zero-cost action. It can be found for example proceeding back from the goal, skipping actions of cost zero.

LM-cut: step 4



now e and f have cost 2 and 0

back to the graph with all preconditions

recompute maximal heuristics, etc.

[note] Computing cuts back from the goal, f is skipped because of its zero cost, and the the next cut is {d, e}. Its cost 2 is subtracted from the cost of d and e.

At this step, recomputing the maximal heuristics results in the same costs as before. However, since the cost of actions keeps decreasing, the maximal heuristics at some point may produce a different result. This affects the selection of the precondition to keep for each action.

LM-cut: why decreasing the cost?

landmark (cut) found: {e, f}
cost is minimal cost of actions: 1

decrease cost of e and f by 1
= split the problem in two

- the first has cost 1 for e and f and zero for the others
- the second has the remaining costs:
2 for e, 0 for f and the original cost for all other actions

heuristics based on landmark {e, f} on the first copy
continue the search on the second copy

the landmark {e, f} has now cost zero in the second copy
⇒ it is not found again (only cut of non-zero cost found)

[note] Point 2 of algorithm is "find a cut of non-zero cost". Since the landmark {e, f} now has zero cost in the second copy of the problem, where the algorithm continues, it is not found again.

other things about landmarks

- find a better set of disjunctive action landmarks
- better way to combine disjunctive action landmarks
 - approximate hitting sets (more precise, but more expensive to calculate)
- landmarks for non-relaxed problems
- fact landmarks: *variables* that need to be made true, instead of actions
- formula landmarks, instead of just disjunctions
- other heuristics: number of currently unachieved landmarks
- landmark ordering, planning as achieving the landmarks in order