LP-based Heuristics, Abstractions

h^{flow}, h^{pot}, Merge & Shrink

Michaela Urbanovská

PUI Tutorial
Week 7

Lecture check

- Any questions regarding the lecture?

Organization

Assignment #1-3

- No substitute date for 10.4.2023
- **12.4.2023** will be tutorial with consultation + computing more heuristic examples
- Deadline will be extended for everyone to **14.4.2023 23:59**

LP-based heuristics

Linear program

Linear program (LP) consists of:

- a finite set of real-valued variables **V**
- a finite set of linear **constraints over V**
- an **objective function** *(linear combination of V)*

Integer linear program (ILP) is the same thing with integer-valued variables.

LP-based heuristics

- LP - solution in **polynomial time**
- ILP - finding solution is **NP-complete**
- We can approximate ILP solution with corresponding LP
- Sounds familiar? **Relaxation**
- Flow heuristic - h^{flow}
- Potential heuristic - h^{pot}

Running example

FDR problem example

FDR planning task $P = \langle \mathbf{V}, O, s_{init}, s_{goal}, c \rangle$

- $\mathbf{V} = \{A, B, C\}$
- $D_A = \{D, E\}$; $D_B = \{F, G\}$; $D_C = \{H, J, K\}$
- $s_{init} = \{A = D, B = F, C = H\}$
- $s_{goal} = \{A = D, C = K\}$
- $O = \{o_1, o_2, o_3, o_4, o_5\}$

	pre	eff	c
o_1	$\{A = D, C = H\}$	$\{A = E, C = J\}$	2
o_2	$\{A = D\}$	$\{B = G\}$	1
o_3	$\{B = G, C = J\}$	$\{C = K\}$	1
o_4	$\{A = E\}$	$\{A = D\}$	2
o_5	$\{C = H\}$	$\{C = J\}$	5

h^{flow}

Producing and consuming

For every variable $V \in \mathbf{V}$ and every value $v \in D_V$ we define

- a set of operators **producing** $\langle V, v \rangle$:
  $prod(\langle V, v \rangle) = \{o | o \in O, V \in vars(eff(o)), eff(o)[V] = v\}$
- a set of operators **consuming** $\langle V, v \rangle$:
  $cons(\langle V, v \rangle) = \{o | o \in O, V \in vars(pre(o)) \cap vars(eff(o)), pre(o)[V] = v, pre(o)[V] \neq eff(o)[V]\}$

- FDR planning task $P = \langle \mathbf{V}, O, s_{init}, s_{goal}, c \rangle$
- real-valued variable x_o for each $o \in O$ - counts operators in plan

LP formulation

$$\text{minimize} \sum_{o \in O} c(o) x_o$$

$$\text{subject to } LB_{V,v} \leq \sum_{o \in prod(\langle V,v \rangle)} x_o - \sum_{o \in cons(\langle V,v \rangle)} x_o, \forall V \in \mathbf{V}, \forall v \in D_V$$

$$\text{where } LB_{V,v} = \begin{cases} 0 & \text{if } V \in vars(s_{goal}) \text{ and } s_{goal}[V] = v \text{ and } s[V] = v, \\ 1 & \text{if } V \in vars(s_{goal}) \text{ and } s_{goal}[V] = v \text{ and } s[V] \neq v, \\ -1 & \text{if } (V \notin vars(s_{goal}) \text{ or } s_{goal}[V] \neq v) \text{ and } s[V] = v, \\ 0 & \text{if } (V \notin vars(s_{goal}) \text{ or } s_{goal}[V] \neq v) \text{ and } s[V] \neq v, \end{cases}$$

h^{flow}

$$\mathrm{LB}_{V,v} = \begin{cases} 0 & \text{if } V \in \mathit{vars}(s_{goal}) \text{ and } s_{goal}[V] = v \text{ and } s[V] = v, \\ 1 & \text{if } V \in \mathit{vars}(s_{goal}) \text{ and } s_{goal}[V] = v \text{ and } s[V] \neq v, \\ -1 & \text{if } (V \notin \mathit{vars}(s_{goal}) \text{ or } s_{goal}[V] \neq v) \text{ and } s[V] = v, \\ 0 & \text{if } (V \notin \mathit{vars}(s_{goal}) \text{ or } s_{goal}[V] \neq v) \text{ and } s[V] \neq v, \end{cases}$$

- if $V = v$ in s then it cannot be consumed more times than produced to reach s_{goal}
- if $V = v$ is not true in s it has to be produced at least once to reach s_{goal}
- if $V = v$ is not set in s_{goal} but is set in s we don't know how many times it should be consumed or produced so we set the lower bound to -1 (can be consumed more then produced)
- if $V = v$ is not set in goal state but is not set in s we can produce it but also consume it so we set the lower bound to 0

LP formulation

$$\text{minimize} \sum_{o \in O} c(o) x_o$$

$$\text{subject to } LB_{V,v} \leq \sum_{o \in prod(\langle V,v \rangle)} x_o - \sum_{o \in cons(\langle V,v \rangle)} x_o, \forall V \in \mathbf{V}, \forall v \in D_V$$

$$\text{where } LB_{V,v} = \begin{cases} 0 & \text{if } V \in vars(s_{goal}) \text{ and } s_{goal}[V] = v \text{ and } s[V] = v, \\ 1 & \text{if } V \in vars(s_{goal}) \text{ and } s_{goal}[V] = v \text{ and } s[V] \neq v, \\ -1 & \text{if } (V \notin vars(s_{goal}) \text{ or } s_{goal}[V] \neq v) \text{ and } s[V] = v, \\ 0 & \text{if } (V \notin vars(s_{goal}) \text{ or } s_{goal}[V] \neq v) \text{ and } s[V] \neq v, \end{cases}$$

The **value of h^{flow} heuristic** for the state s is

$$\text{h}^{flow}(s) = \begin{cases} \lceil \sum_{o \in O} c(o) x_o \rceil & \text{if the solution is } \textbf{feasible} \\ \infty & \text{if the solution is } \textbf{not feasible} \end{cases}$$

Long story short

- Define variable x_o for each operator *(operator "counters")*
- Create *prod* and *cons* sets
- Write constraints with $LB_{V,v}$ constants on the left side
- Compute constants $LB_{V,v}$ based on the 4 rules
- Put it in a solver
- ...
- Profit!

h^{pot}

- FDR planning task $P = \langle \mathbf{V}, O, s_{init}, s_{goal}, c \rangle$
- real-valued variable $P_{V,v}$ for each variable $V \in \mathbf{V}$ and each value $v \in D_V$
  - **potential** corresponding to $\langle V, v \rangle$
- real-valued variable M_V for each variable $V \in \mathbf{V}$
  - **upper bound** on the potentials of variable V
  - used in situations where we don't know the value $\rightarrow$ prepare for the worst case
  - *example:* variable B in our problem P

h_{pot}

Goal-awareness constraint: $P_{A,D} + P_{C,K} \leq 0$ ...what about B?

- Add each case of B (possibly exponentially many)
  - $P_{A,D} + P_{B,F} + P_{C,K} \leq 0$
  - $P_{A,D} + P_{B,G} + P_{C,K} \leq 0$
- Use the M_B bound (linear)
  - $P_{A,D} + M_B + P_{C,K} \leq 0$
  - $P_{B,F} \leq M_B$
  - $P_{B,G} \leq M_B$

LP formulation

$$\text{maximize} \sum_{V \in \mathbf{V}} P_{V, s_{init}[V]}$$

$$\text{subject to } P_{V,v} \leq M_V, \forall V \in \mathbf{V}, \forall v \in D_V$$

$$\sum_{V \in \mathbf{V}} maxpot(V, s_{goal}) \leq 0$$

$$\sum_{V \in vars(eff(o))} (maxpot(V, pre(o)) - P_{V, eff(o)[V]}) \leq c(o), \forall o \in O$$

$$\text{where } maxpot(V, p) = \begin{cases} P_{V, p[V]} & \text{if } V \in vars(p), \\ M_V & otherwise. \end{cases}$$

The **value of h^{pot} heuristic** for the state s is

$$h^{pot}(s) = \begin{cases} \sum_{V \in \mathbf{V}} P_{V, s[V]} & \text{if the solution is } \textbf{feasible} \\ \infty & \text{if the solution is } \textbf{not feasible} \end{cases}$$

h^{pot}

Long story short

- Define potential $P_{V,v}$ for each variable and its possible value
- Define potential upper bound for each variable $V \in \mathbf{V}$
- When computing $h^{pot}(s)$ we want to maximize sum of potentials of $\langle V, v \rangle$ pairs in s
- define goal-awareness constraints
- define consistency constraints with respect to operator costs
- Solve $\rightarrow$ get the potentials

Abstraction heuristics

- Simplification of the problem
- Making the problem smaller by dropping state distinctions

Abstraction heuristics

Transition system $\mathcal{T} = \langle S, L, T, I, G \rangle$

- S - finite set of states
- L - finite set of labels
- $T \subseteq S \times L \times S$ - transition relation
- $I \subseteq S$ - set of initial states
- $G \subseteq S$ - set of goal states
- $c(l) \in \mathrm{R}_0^+, \forall l \in L$ - cost function for each label

Abstraction heuristics

Transition system for problem P

Transition system $\mathcal{T}(P)$ is defined for FDR problem $P = \langle V, O, s_{init}, s_{goal}, c \rangle$.
The mapping goes as followed:

- S is set of states over V
- $L = O$
- $T = \{(s, o, t) | res(o, s) = t\}$
- $I = \{s_{init}\}$
- $G = \{s | s \in S, s$ is consistent with $s_{goal}\}$

Abstraction heuristics

Abstraction definition

- Let's have two transition systems $\mathcal{T}^1 = \langle S^1, L, T^1, I^1, G^1 \rangle$ and $\mathcal{T}^2 = \langle S^2, L, T^2, I^2, G^2 \rangle$ with the same set of labels L.
- Let's have an **abstraction function** $\alpha : S^1 \mapsto S^2$ which maps S^1 to S^2.
- S^2 is an **abstraction** of S^1 if
  - $\forall s \in I^1$ holds that $\alpha(s) \in I^2$
  - $\forall s \in G^1$ holds that $\alpha(s) \in G^2$
  - $\forall (s, l, t) \in T^1$ holds that $(\alpha(s), l, \alpha(t)) \in T^2$

Abstraction heuristics

Abstraction heuristic

Let P denote an FDR planning task and $\mathcal{A}$ denote an **abstraction** of its transition system $\mathcal{T}(P)$.
Abstraction heuristic induced by $\mathcal{A}$ and α is the function

$$h^{\mathcal{A},\alpha} = h^*_{\mathcal{A}}(\alpha(s)), \forall s \in S$$

Abstraction heuristics

Abstraction heuristic

Let P denote an FDR planning task and $\mathcal{A}$ denote an **abstraction** of its transition system $\mathcal{T}(P)$.
Abstraction heuristic induced by $\mathcal{A}$ and α is the function

$$h^{\mathcal{A},\alpha} = h^*_{\mathcal{A}}(\alpha(s)), \forall s \in S$$

Synchronized product

Given two transition systems $\mathcal{T}^1 = \langle S^1, L, T^1, I^1, G^1 \rangle$ and $\mathcal{T}^2 = \langle S^2, L, T^2, I^2, G^2 \rangle$ with the same labels, their **synchronized product** $\mathcal{T}^1 \otimes \mathcal{T}^2 = \mathcal{T}$ is a transition system $\mathcal{T} = \langle S, L, T, I, G \rangle$, where

- $S = S^1 \times S^2$
- $T = \{((s_1, s_2), l, (t_1, t_2)) | (s_1, l, s_2) \in T^1, (s_2, l, t_2) \in T^2\}$
- $I = I^1 \times I^2$
- $G = G^1 \times G^2$

Merge & Shrink heuristic

- Different types of abstraction heuristics
- How to select α?
- In this tutorial: **merge & shrink**
- Consists of
  - merging = computing synchronized products of the abstractions
  - shrinking = abstracting the abstractions further
- There are many strategies...so we will just focus on the main thought behind it

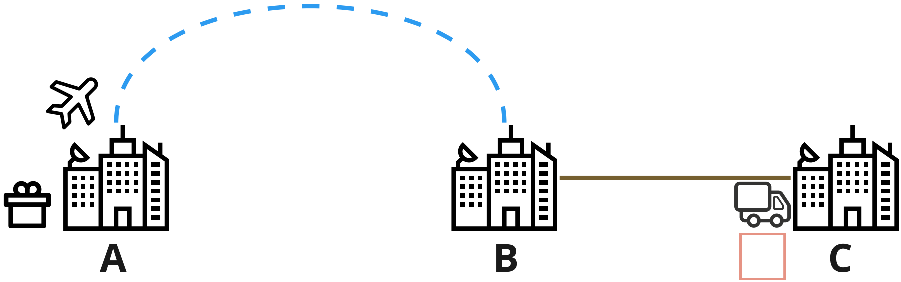
Merge & Shrink heuristic

Transition systems $\mathcal{T}^1$ and $\mathcal{T}^2$

- $L^1 = L^2 = \{a, b, c, d, e\}$
- $S^1 = \{A, B, C, D\}$
- $T^1 = \{(A, a, B), (B, b, C), (C, c, A), (A, d, A), (A, e, D)\}$
- $I^1 = \{A\}$
- $G^1 = \{A, C\}$
- $S^2 = \{X, Y, Z\}$
- $T^2 = \{(X, a, Y), (X, a, Z), (Y, b, Z), (Z, c, Y), (Z, d, Y), (Z, e, Z)\}$
- $I^2 = \{X\}$
- $G^2 = \{X\}$

Let's compute synchronized product!

Let's try an example we know well...



FDR representation

FDR problem $P = \langle V, O, s_I, s_G, c \rangle$

$V = \{a, t, p\}$

$D_a = \{A, B\}$ $D_t = \{B, C\}$ $D_p = \{A, B, C, a, t\}$

$s_I = \{a = A, t = C, p = A\}$

$s_G = \{p = C\}$

$O =$

	pre	eff	c
fAB	a=A	a=B	1
fBA	a=B	a=A	1
dBC	t=B	t=C	1
dCB	t=C	t=B	1
laA	a=A, p=A	p=a	1
laB	a=B, p=B	p=a	1
ltB	t=B, p=B	p=t	1
ltC	t=C, p=C	p=t	1
uaA	p=a, a=A	p=A	1
uaB	p=a, a=B	p=B	1
utB	p=t, t=B	p=B	1
utC	p=t, t=C	p=C	1

Atomic projections

- One possible representation is by **atomic projections**
- One transition system for one variable from $V = \{a, t, p\}$

$\mathcal{T}^{\mathbf{a}}$
$S^a = \{aA, aB\}$
$I^a = \{aA\}$
$G^a = \{aA, aB\}$

$\mathcal{T}^{\mathbf{t}}$
$S^t = \{tB, tC\}$
$I^t = \{tC\}$
$G^t = \{tB, tC\}$

$\mathcal{T}^{\mathbf{p}}$
$S^p =$
$\{pA, pB, pC, pa, pt\}$
$I^p = \{pA\}$
$G^p = \{pC\}$

Merge & Shrink

1. Create atomic projections (one per variable)
2. Merge two arbitrary transition systems (synchronized product)
3. Shrink the new transition graph (merge states together to create smaller abstraction)
4. Repeat 2 and 3 until you're left with one abstraction in which you can find the solution

Recap

- Know definition of h^{flow} and h^{pot} heuristics
- Know how to compute Merge & Shrink
  - How to create synchronized product
  - Atomic projections
  - Main principle
    - **merging** = creating synchronized products of two transition systems
    - **shrinking** = creating smaller abstraction

The End

Feedback form link