Examensarbete: fpt-algoritm för h^+
Preliminär beskrivning

Christer Bäckström

April 1, 2015

Examensarbetet handlar om att implementera en fpt-algoritm för att beräkna h^+-heuristiken och integrera denna i en färdig planerare, lämpligen Fast Downward (FD) [4].

1. Läs in en planeringsinstans $\mathbb{P} = \langle V, A, I, G \rangle$ i PDDL-kod.

2. Relaxera instansen, dvs. ta bort alla negativa preconditions och effekter i handlingarna. Kalla den relaxerade instansen $\mathbb{P}^+ = \langle V, A^+, I^+, G^+ \rangle$.

   Eftersom vi bara kan sätta variabler till sant, och inte till falskt så kan vi dessutom ta bort alla variabler som redan är sanna i I^+.

3. Beräkna den kausala grafen $\mathrm{CG}(\mathbb{P}^+)$ för $\mathbb{P}^+$. (Finns troligen färdigt i FD).

4. Beräkna en optimal tree decomposition $\langle N, T \rangle$ för $\mathrm{CG}(\mathbb{P}^+)$ med Bodlaenders algorithm [1]. Alt. kan även icke-optimala men snabbare algoritmer behöva testas.

   Varje nod N_i i N är en delmängd av variablerna i $\mathbb{P}^+$ och definierar ett delproblem $\mathbb{P}^+[N_i]$, dvs. $\mathbb{P}^+$ begränsat till variablerna i N_i.

   Beräkna $TG(\mathbb{P}^+)[N_i]$, dvs. projektionen av $\mathrm{TG}(\mathbb{P}^+)$ till N_i, för varje nod N_i.

5. För varje N_i, beräkna mängden av alla planer från $I[N_i]$ till $G[N_i]$, dvs. alla vägar från $I[N_i]$ till $G[N_i]$ i $TG(\mathbb{P}^+)[N_i]$.

   Denna mängd blir domänen $D(x_i)$ för den motsvarande CSP-variabeln x_i.

6. Trädet T definierar nu en CSP-instans med en variabel x_i för varje nod N_i i trädet. Eftersom grafen är ett träd så kan denna lösas i polynomisk tid [2] och kan lösas optimalt i fpt-tid [3].

7. Längden på den optimala planen blir nu det heuristiska värdet för $h^+(I)$.

Ovanstående är lite förenklat, eftersom det bara gäller för att beräkna h^+ för initialtillståndet I. I praktiken kommer planeraren att anropa vår algoritm för att beräkna $h^+(s)$ för många olika tillstånd s. Stegen 1–4 kanske kan göras helt eller delvis i början, en gång för alla; det finns nog avvägningar att göra här. Stegen 5–7 måste göras för varje sådant anrop.

Beräkningen av $TG(\mathbb{P}^+)[N_i]$ i steg 4 görs lämpligen genom att först konstruera DTGerna för varje variabel i N_i och sedan beräkna den synkrona produkten av dessa. För en enskild variabel v så gäller:

1. Om $G[v] = 1$ så ska vi bara ta med vägar i $DTG(v)$ som börjar i $v = 0$ och slutar i $v = 1$.

2. Annars är v inte definierad i G och då måste vi ta med alla både alla vägar från $v = 0$ till $v = 0$ och alla från $v = 0$ till $v = 1$.

References

[1] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[2] Rina Dechter and Judea Pearl. Tree clustering for constraint networks. *Artif. Intell.*, 38(3):353–366, 1989.

[3] Tommy Färnqvist. Constraint optimization problems and bounded tree-width revisited. In *Proc. 9th Int'l Conf. Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimzation Problems (CPAIOR 2012), Nantes, France.*, pages 163–179, 2012.

[4] Malte Helmert. The fast downward planning system. *J. Artif. Intell. Res.*, 26:191–246, 2006.