Constraint Systems

Tema D'Esame #1

Modellazione: "Si Parte"

Dopo un anno di duro lavoro il giovane ingegner Piero si appresta a partire per le ferie insieme alla sua dolce metà, con cui si è da poco accasato. La vita matrimoniale, si sa, è piena di cambiamenti e l'ingenuo Piero, che pure aveva previsto un certo aumento di bagagli, aveva clamorosamente sottostimato l'entità del fenomeno.

I due sono così davanti agli ultimi pacchetti, chiedendosi se riusciranno a farli entrare nell'ultima valigia (un gioiello high-tech, con due scomparti, ciascuno dei quali organizzabile in due strati).

Modellazione: "Si Parte"

Si sappia che:

1. Rimangono 2 pacchetti di vestiti, 1 di asciugamani ed 1 di scarpe.
2. La superficie di ogni scomparto (su ogni strato) è sufficiente a contenere 2 pacchetti stretti o 1 largo. I pacchetti di vestiti e quello di scarpe sono stretti, quello di asciugamani è largo.
3. Ogni pacchetto (stretto o largo) può essere messo nello strato inferiore o superiore. Quindi, ogni scomparto può contenere fino a 4 pacchetti stretti o 2 larghi.
4. Come eccezione, il pacchetto di scarpe è "alto" ed occupa la superficie di un pacchetto stretto su ambedue gli strati.

Modellazione: "Si Parte"

Si modelli il problema di impacchettamento dei bagagli mediante programmazione a vincoli e si specifichi se è risolvibile. Per modellare il problema, si faccia riferimento alla seguente libreria di vincoli

Suggerimento: si consiglia di costruire il modello incrementalmente, adattandolo e ristrutturandolo via via in modo da tenere conto delle caratteristiche del problema.

Soluzione

Nota:

- Il problema può essere risolto in molti modi!
- Questo è solo un esempio

La soluzione:

Se fosse possibile trascurare il punto 4, superficie e strati potrebbero essere considerati come due modi diversi di misurare un unico tipo di risorsa (spazio). Sotto questa ipotesi, sarebbe possibile modellare il problema introducendo una variabile per ogni pacchetto per indicare a quale scomparto è assegnato:

$$V1, V2, A, S \in \{0, 1\}$$

Soluzione

I vincoli sulle limitazioni di spazio sarebbero modellabili con un singolo un cumulative:

$$\textsc{cumulative}([V1, V2, A, S], [1, 1, 1, 1], [1, 1, 2, 2], 4)$$

Il problema è risolvibile, anche tenendo conto del punto 4, cosa che è possibile in diversi modi.

Un primo metodo consiste nell'osservare che un pacchetto alto può essere tranquillamente messo nello stesso scomparto con due pacchetti stretti (bassi), ma non con un pacchetto largo. Si può modellare questa restrizione aggiungendo il vincolo:

$$A \neq S$$

Soluzione

In alternativa, si può modellare la superficie di ciascuno scomparto come “tempo” in un cumulative. I domini diventano del tipo {0,1,2,3}, dove le posizioni 0/1 fanno riferimento al primo scomparto, mentre 2/3 al secondo scomparto. Un pacchetto stretto avrà un “durata” 1, un pacchetto largo avrà durata 2. Un pacchetto alto utilizzerà due unità di risorsa, mentre un pacchetto basso una sola. Il cumulative diventa:

$$\text{CUMULATIVE}([V1, V2, A, S], [1, 1, 2, 1], [1, 1, 1, 2], 2)$$

Si noti che la capacità è ora 2, anziché 4. Per prevenire che un pacchetto largo sia posizionato a cavallo di due scomparti, è necessario ridurre il dominio di A:

$$A \in \{0, 2\}$$

Esercizio: Propagazione di un CSP

Si consideri il seguente CSP:

$$\boxed{0} \quad x_0 + x_1 = 3$$
$$\boxed{1} \quad x_1 + x_2 = 3$$

con: $x_0 \in \{0..1\}, x_1 \in \{2..4\}, x_2 \in \{1..2\}$

Ipotesi:

- Nessun propagatore incrementale
- BC per +
- GAC per =

Esercizio: Propagazione di un CSP

Si consideri il seguente CSP:

$$\boxed{0} \quad x_0 + x_1 = 3$$
$$\boxed{1} \quad x_1 + x_2 = 3$$

con: $x_0 \in \{0..1\}, x_1 \in \{2..4\}, x_2 \in \{1..2\}$

- Si mostri la sequenza di attivazione dei vincoli, rispettivamente per l'algoritmo di propagazione AC1 ed AC3.
- Si mostri anche lo stato dei domini dopo ciascuna attivazione
- Si discutano eventuali differenze, mettendo in luce le caratteristiche dei due algoritmi

Soluzione

Sequenza di attivazione e domini per AC1:

1. Cst: $\boxed{0}$, $\{0..1\} + \{2..4\} = 3$
   - Domains: $x_0 \in \{0..1\}, x_1 \in \{2..3\}, x_2 \in \{1..2\}$
2. Cst: $\boxed{1}$, $\{2..3\} + \{1..2\} = 3$
   - Domains: $x_0 \in \{0..1\}, x_1 \in \{2\}, x_2 \in \{1\}$
3. Cst: $\boxed{0}$, $\{0..1\} + \{2\} = 3$
   - Domains: $x_0 \in \{1\}, x_1 \in \{2\}, x_2 \in \{1\}$
4. Cst: $\boxed{1}$, $\{2\} + \{1\} = 3$
   - Domains: $x_0 \in \{1\}, x_1 \in \{2\}, x_2 \in \{1\}$
5. Cst: $\boxed{0}$, $\{1\} + \{2\} = 3$
   - Domains: $x_0 \in \{1\}, x_1 \in \{2\}, x_2 \in \{1\}$
6. Cst: $\boxed{1}$, $\{2\} + \{1\} = 3$
   - Domains: $x_0 \in \{1\}, x_1 \in \{2\}, x_2 \in \{1\}$

Soluzione

Sequenza di attivazione e domini per AC3 (1/2):

Initial queue: $Q : [(c_0, *), (c_1, *)]$

1. Cst: $\boxed{0}$, $\{0..1\} + \{2..4\} = 3$
   - Domains: $x_0 \in \{0..1\}, x_1 \in \{2..3\}, x_2 \in \{1..2\}$
   - Queue: $Q : [(c_1, *), (c_0, *)]$
2. Cst: $\boxed{1}$,$\{2..3\} + \{1..2\} = 3$
   - Domains: $x_0 \in \{0..1\}, x_1 \in \{2\}, x_2 \in \{1\}$
   - Queue: $Q : [(c_0, *), (c_1, *)]$
3. Cst: $\boxed{0}$, $\{0..1\} + \{2\} = 3$
   - Domains: $x_0 \in \{1\}, x_1 \in \{2\}, x_2 \in \{1\}$
   - Queue: $Q : [(c_1, *), (c_0, *)]$

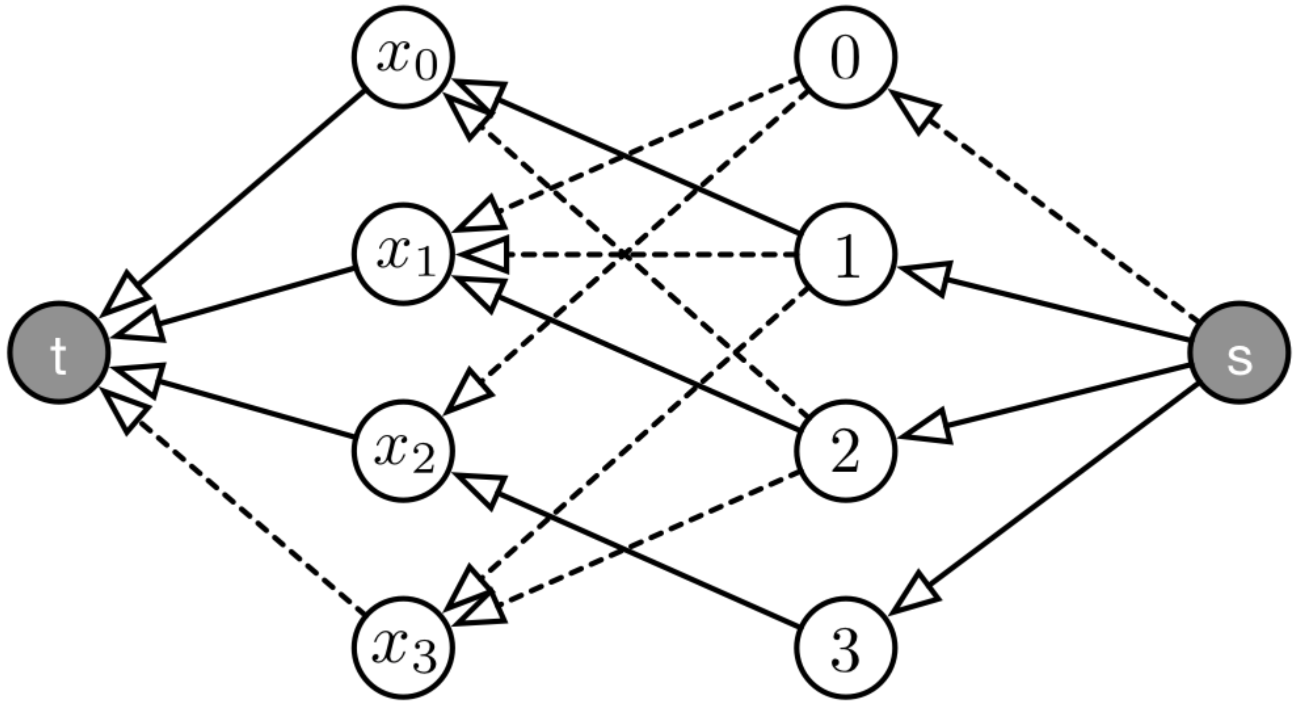
Soluzione

Sequenza di attivazione e domini per AC3 (2/2):

Domains from the previous step: $x_0 \in \{1\}, x_1 \in \{2\}, x_2 \in \{1\}$

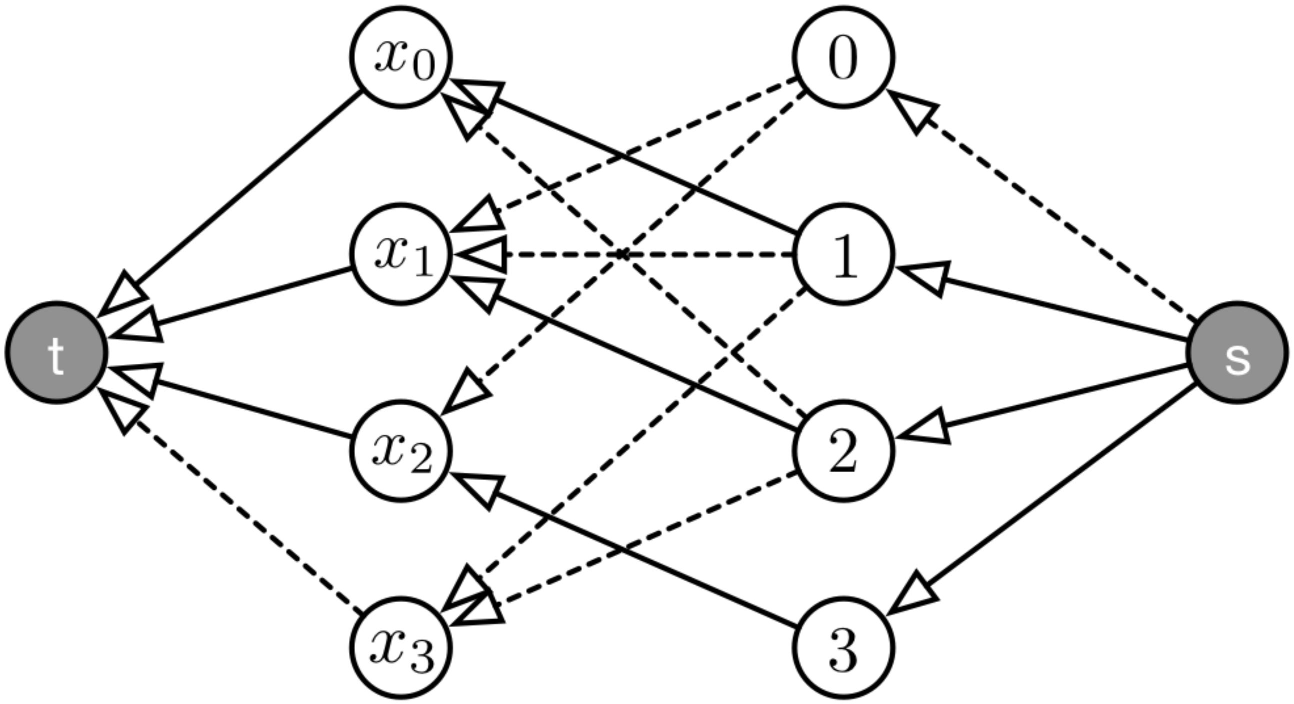
1. Cst: $\boxed{1}$, $\{2\} + \{1\} = 3$
   - Domains: $x_0 \in \{1\}, x_1 \in \{2\}, x_2 \in \{1\}$
   - Queue: $Q : [(c_1, *)]$
2. Cst: $\boxed{0}$, $\{1\} + \{2\} = 3$
   - Domains: $x_0 \in \{1\}, x_1 \in \{2\}, x_2 \in \{1\}$
   - Queue: $Q : []$

Esercizio: Filtering per ALLDIFF



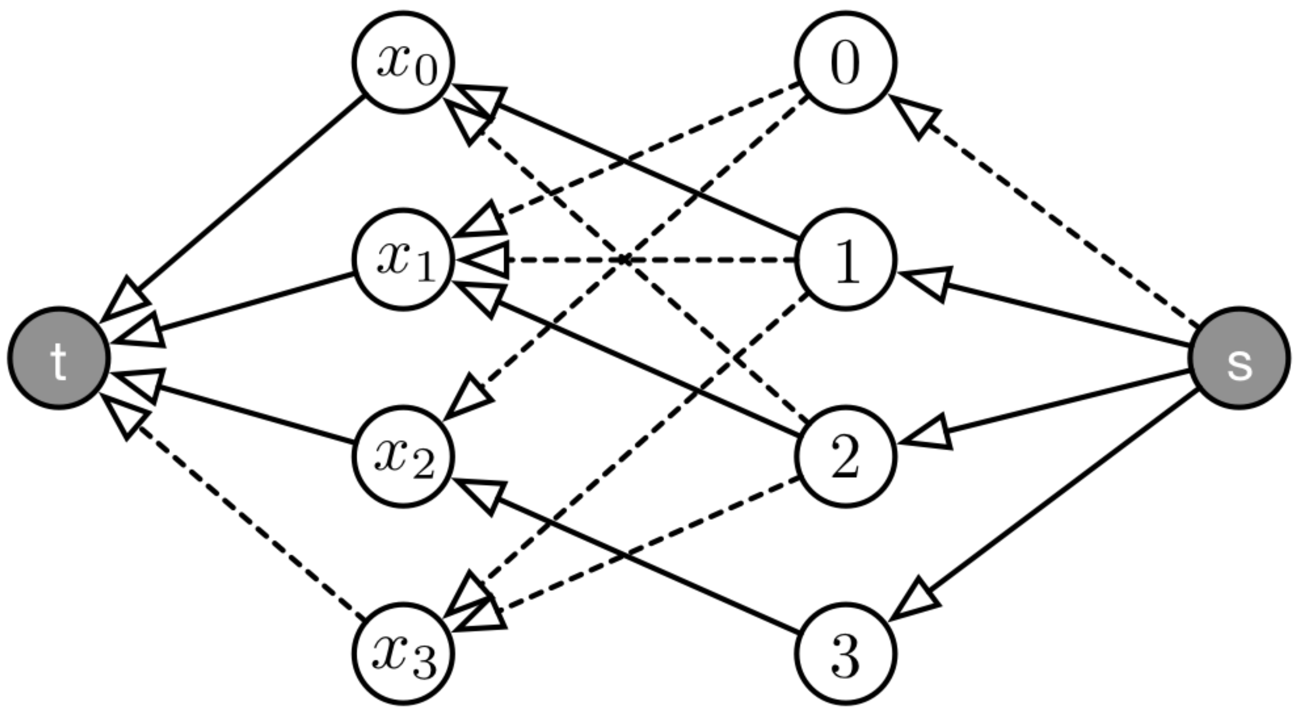
Sia dato questo grafo dei valori per il propagatore basato sul flusso del vincolo ALLDIFF. Gli archi non tratteggiati sono attraversati da una unità di flusso.

Esercizio: Filtering per ALLDIFF



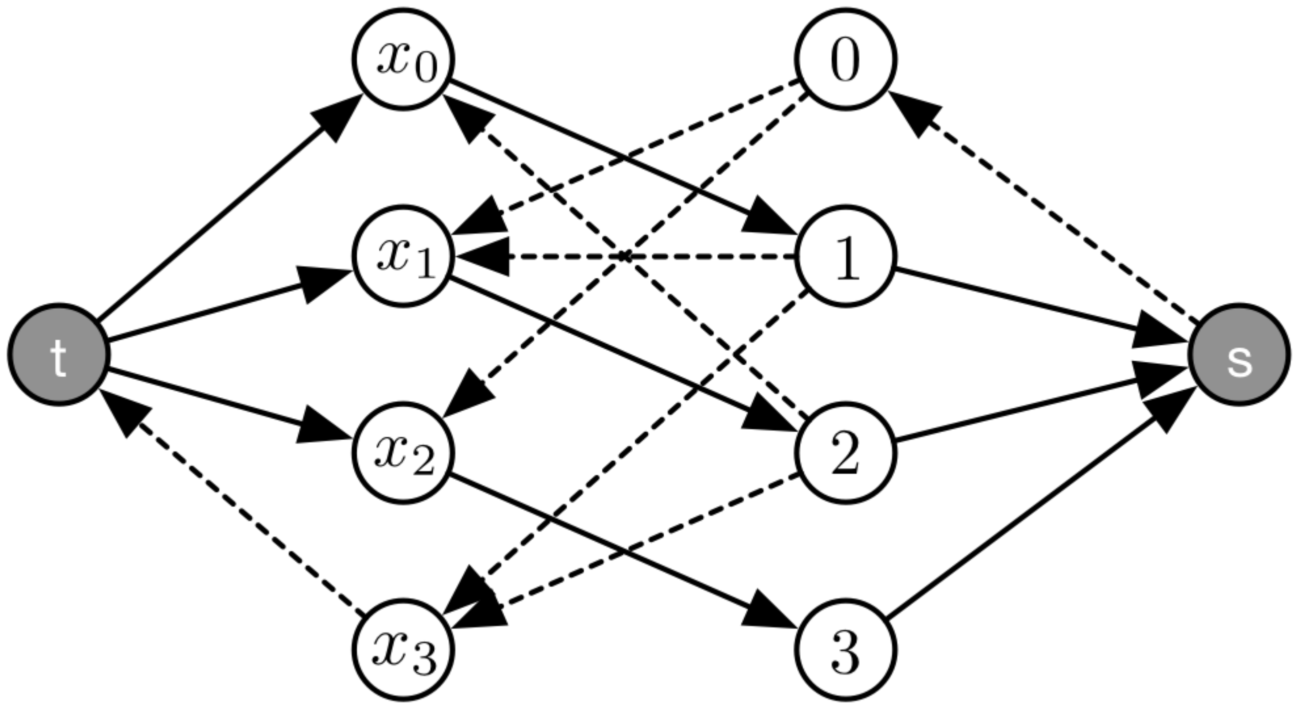
Si esegua il propagatore per determinare se il vincolo è ammissibile e per eliminare eventuali valori dai domini delle variabili

Esercizio: Filtering per ALLDIFF



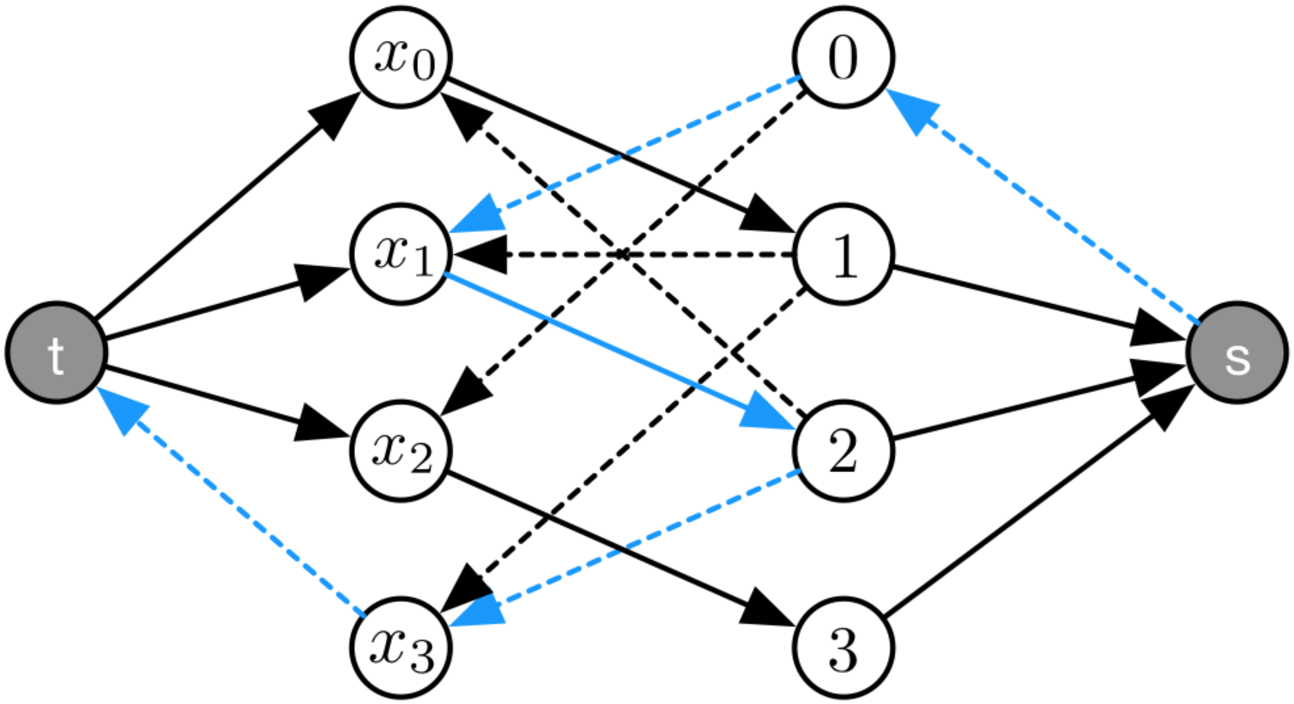
- Si costruisca un flusso massimo, operando sul grafo residuale
- Se il vincolo risulta ammissibile, si individuino i componenti fortemente connessi e si determinino gli archi da eliminare

Soluzione



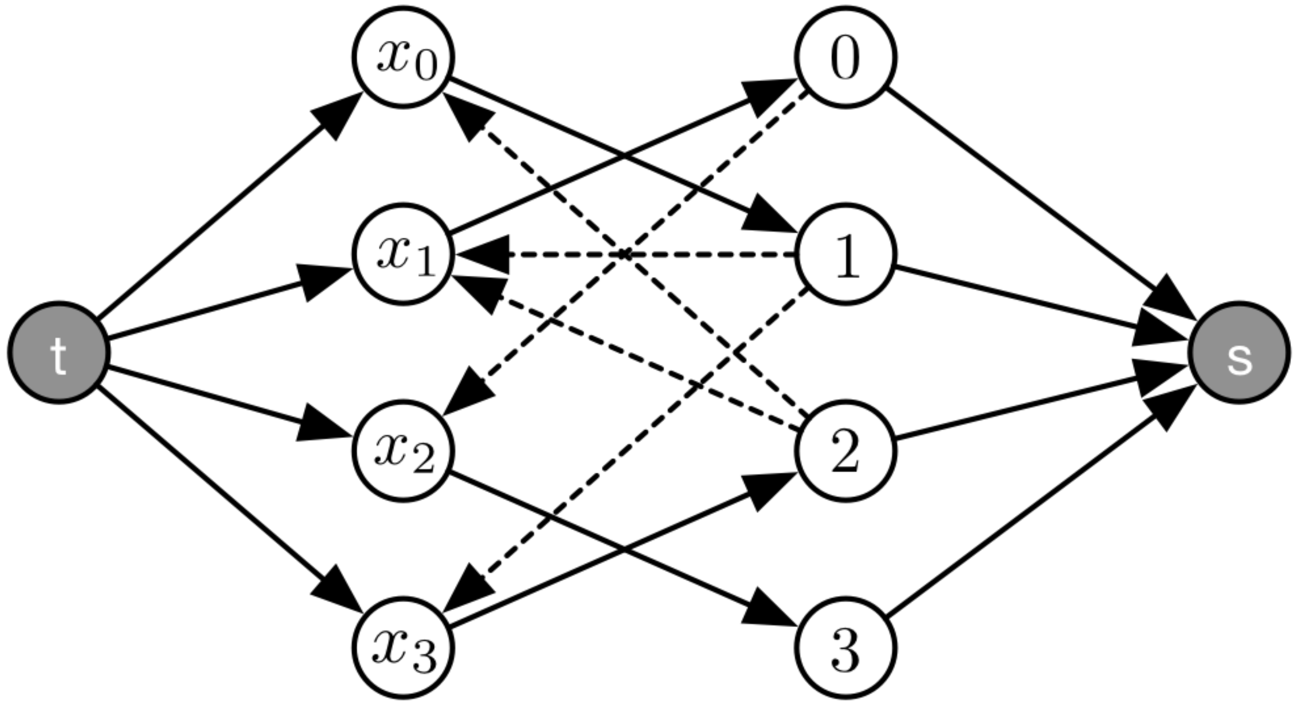
Per prima cosa, otteniamo il grafo residuale

Soluzione



Quindi individuiamo un percorso aumentante

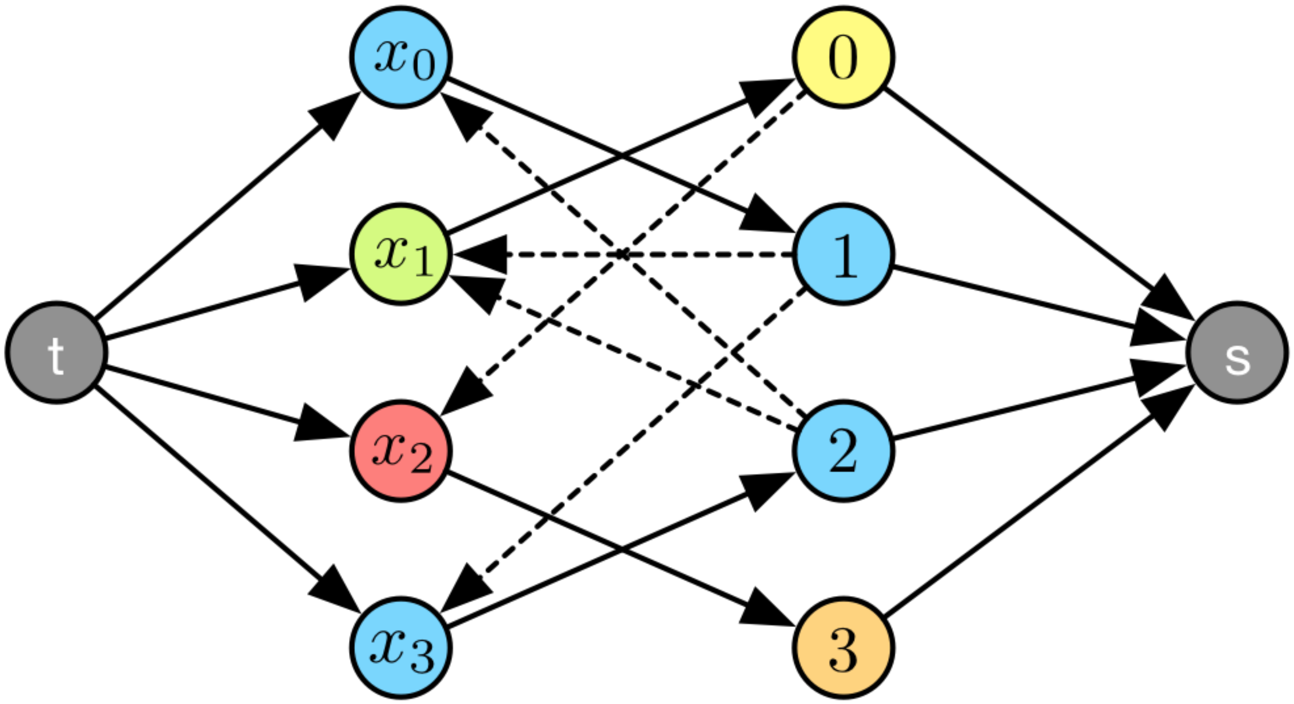
Soluzione



Aggiorniamo il grafo residuale

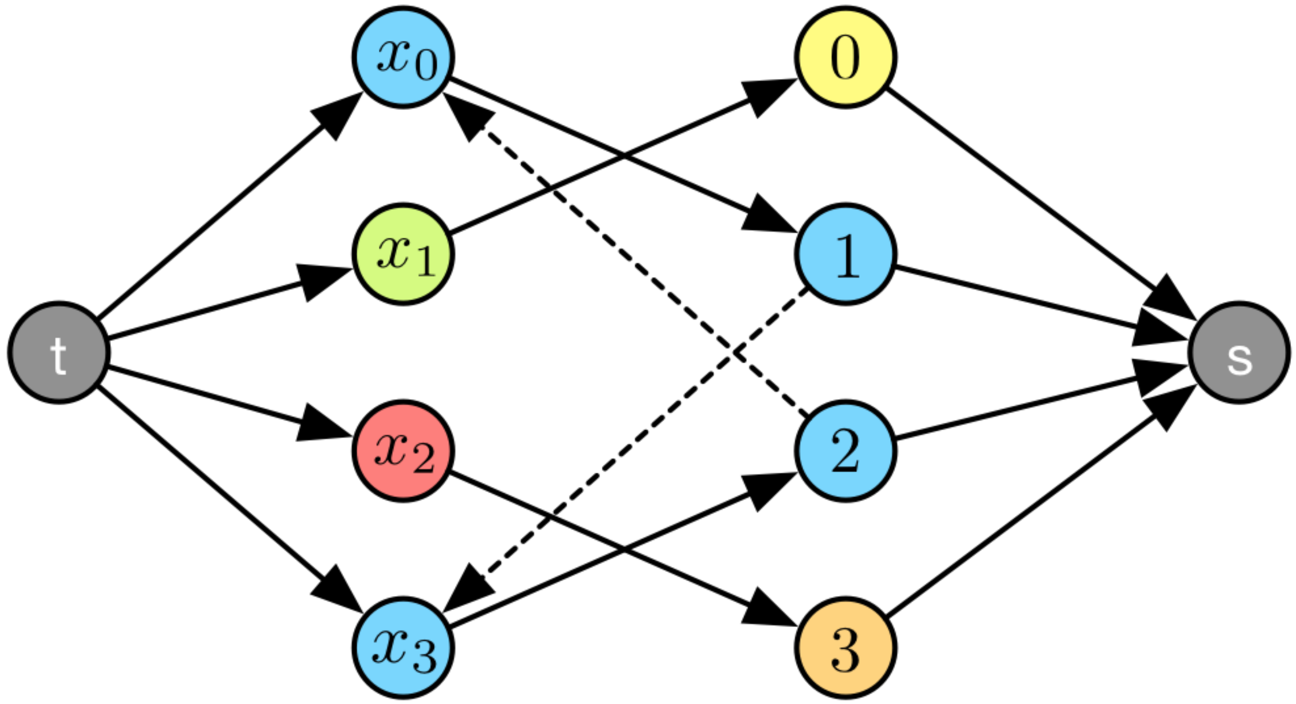
- Il flusso è massimo ed uguale al numero di variabili
- Quindi il vincolo è ammissibile

Soluzione



Individuiamo quindi i componenti fortemente connessi

Soluzione



Eliminiamo gli archi con flusso nullo che connettono SCC diversi

- I domini finali sono: $x_0 \in \{1, 2\}, x_1 \in \{0\}, x_2 \in \{3\}, x_3 \in \{1, 2\}$

Esercizio: Progettazione

Tre squadre ('0', '1 e '2') di calcio partecipano ad un mini-campionato. Ogni squadra deve scontrarsi con le altre in due partite (una "in casa" e l'altra "fuori casa"). Le 6 partite da giocare sono dunque:

$$(0, 1), (0, 2), (1, 0), (1, 2), (2, 0), (2, 1)$$

Occorre stabilire in quale sequenza effettuare le partite.

Il problema viene modellato utilizzando due serie di variabili principali:

- $x_k \in \{0..2\}, \forall k = 0..5$ per indicare la squadra di casa
- $y_k \in \{0..2\}, \forall k = 0..5$ per indicare la squadra ospite

Si definiscano vincoli (ed eventualmente variabili aggiuntive) per garantire che le variabili x e y rappresentino correttamente il problema. Si discutano brevemente le scelte fatte.

Soluzione

Una prima soluzione

- Per ogni partita, la squadra di casa e la squadra ospite devono essere diverse:

$$x_k \neq y_k, \quad \forall k = 0..5$$

- Ogni squadra gioca in casa esattamente due volte:

$$\textsc{gcc}(x, [0, 1, 2], [2, 2, 2], [2, 2, 2])$$

- Ogni squadra gioca come ospite esattamente due volte:

$$\textsc{gcc}(y, [0, 1, 2], [2, 2, 2], [2, 2, 2])$$

Soluzione

Una prima soluzione

Così facendo, può ancora accedere che una stessa coppia (i, j) compaia più volte nella sequenza.

È possibile eliminare il problema aggingendo i vincoli seguenti:

$$x_h = x_k \Rightarrow y_h \neq y_k, \quad \forall h, k = 0..5, h < k$$

Seguono due soluzioni più generali.

Soluzione

Una seconda soluzione

Una seconda soluzione consiste nel richiede che ognuna delle coppie (i, j) sia presente. È utile assegnare un indice numerico alle coppie:

$$\begin{array}{ccc} 0 \leftrightarrow (0,1) & 1 \leftrightarrow (0,2) & 2 \leftrightarrow (1,0) \\ 3 \leftrightarrow (1,2) & 4 \leftrightarrow (2,0) & 5 \leftrightarrow (2,1) \end{array}$$

Usiamo quindi la notazione h_i, g_i per riferirci rispettivamente alla squadra di casa ed alla squadra ospite per la coppia di indice i.

A questo punto basta utilizzare i vincoli:

$$\sum_{k=0}^{5} (x_k = h_i \wedge y_k = g_i) = 1 \quad \forall i = 0..5$$

Soluzione

Una terza soluzione

Più complessa, ma migliore in termini di propagazione

- Usiamo l'indice i per riferirci ad ognuna delle 6 coppie
- Introduciamo un vettore di variabili

$$z_k \in \{0..5\}, \forall k = 0..5$$

- $z_k = i$ sse la partita di indice i si svolge in posizione k
- A questo punto, tutte le partite devono svolgersi:

$$\text{ALLDIFF}(z)$$

Soluzione

- Per collegare le variabili x, y e z usiamo una serie di vincoli:

$$\text{TABLE}([x_k, y_k, z_k], T), \quad \forall k = 0..5$$

- Dove la tabella T contiene:

$\mathbf{x_k}$	$\mathbf{y_k}$	$\mathbf{z_k}$
0	1	0
0	2	1
1	0	2
1	2	3
2	0	4
2	1	5

Qualche domanda di teoria

L'esame conterrà una domanda di teoria. Alcuni esempi:

1) Che cosa è un vincolo globale in CP? Per quali ragioni i vincoli globali sono utili? Argomentare indicando alcuni vincoli come esempio.

2) Come si definisce un problema di soddisfacimento di vincoli? Come si definisce un vincolo?

3) Si definiscano la Generalized Arc Consistency e la Bound Consistency. Si discutano le due nozioni di consistenza, evidenziandone vantaggi e svantaggi