Lezione14

Table of contents

Problemi di Soddisfacimento dei Vincoli

I problemi di soddisfacimento dei vincoli come gli abbiamo visti fino a ora, sono un tanto informali: ci accorgiamo che alcune cose non sono complete. Per risolvere, formalizziamo parlando in senso matematico.

Un **constraint satisfaction problem** (CSP) è una terna $\langle V, D, C \rangle$ con

- un insieme non vuoto e finito di *variabili*, indicato con $V \neq 0$;
- un insieme non vuoto e finito di *domini*, indicato con $D \neq 0$ e di numero inferiore rispetto al numero di variabili ($|D| \leq n$);
- un insieme finito di *vincoli*, indicato con C (un CSP vuoto sarà consistente qualsiasi siano gli assegnamenti).

Ordinamento totale

Una funzione $\texttt{dom} : V \to D$ suriettiva, associa a ogni variabile un dominio.
Supponiamo che tutte le volte che andiamo a elencare le variabili, lo facciamo in un ordine predefinito; tutte le volte che elenchiamo le variabili in un certo ordine, seguiremo un certo *ordinamento totale*, permettendoci di riferirci a variabili in base alla loro posizione.

Domini

Il *dominio* del problema è il prodotto cartesiano dei domini di tutte le variabili, elencando le variabili secondo l'ordinamento totale scelto.

$$\texttt{dom}(\mathcal{P}) = \prod_{x \in V} \texttt{dom}(x)$$

Un dominio di x, y, z

Se abbiamo le variabili x, y, z che variano all'interno di 3 intervalli in $\mathbb{R}$, il dominio del problema sarà il prodotto cartesiano di queste, nell'ordine scelto da noi. Stiamo costruendo una scatola in questo caso, che identifica tutti i possibili assegnamenti delle variabili ai valori all'interno del dominio.

Vincoli

I vincoli per la maggior parte delle volte che li vedremo, saranno binari o ternari . Un vincolo è una copia che contiene sottoinsieme V_c dell'insieme delle variabili (variabili su cui lavora il vincolo) e un *sottoinsieme* Δ_c del dominio (coinvolto all'interno del vincolo).

$$\Delta_c \subseteq \prod_{x \in V_c} \mathtt{dom}(x)$$

> **Vincoli di x, y, z**
>
> Considerando un cubo costituito da 3 variabili in prodotto cartesiano, se prendiamo soltanto x, y per esempio, costruendo otteniamo un quadrato e questo è sottoinsieme. Non tutti i punti contenuti nel quadrato soddisfano il vincolo: il quadrato è l'insieme di tutti i punti possibili mentre il vincolo ci dice quali di questi soddisfano per davvero.

Assegnamenti

Δ_c ci dice quali sono gli *assegnamenti parziali*, perché lavorano su sottoinsiemi, *consistenti* (o ammissibili), perché sono quegli assegnamenti che soddisfano il vincolo.

- Se la cardinalità di $|V_c| = 1$ allora il vincolo c viene detto *unario* (eliminabili facilmente).
- Se $|V_c| = 2$ allora c viene detto *binario* (per algoritmi come ARC).
- In tutti gli altri casi, c si dice *globale* (hanno propagazioni particolari).

Fissato un vincolo V_c la notazione $\mathtt{vars}(c) = V_c$ ci dice quali sono le variabili del vincolo c: funzione che dato un vincolo ci ritorna le variabili associate. Il dominio di c è il prodotto cartesiano dei domini delle variabili coinvolte nel vincolo, seguendo l'ordine fissato

$$\mathtt{dom}(c) = \prod_{x \in V_c} \mathtt{dom}(x)$$

In possesso di un dominio di vincolo, dobbiamo capire come i domini dei vari vincoli interagiscono tra loro. Per farlo, si estende il dominio di un vincolo al dominio del problema: siccome una variabile non coinvolta non ha importanza, in un dominio dove solo alcune hanno significato, allora estendere a un valore qualsiasi queste variabili non fa differenza.

> **Una estensione di x, y, z**
>
> Sempre pensando a un cubo, prendiamo un sottoinsieme associato x, y: questo quadrato identifica tutte le coppie ammissibili delle 2 variabili ed estende aggiungendo tutte le z possibili. Avremo vettori di 3 componenti con x, y ammissibili dal vincolo e z una qualsiasi.

Chiamata immagine del vincolo $\mathtt{img} : C \to \Delta$ la funzione suriettiva totale che associa a ogni vincolo l'estensione a Δ dell'insieme degli assegnamenti (parziali) consistenti, parliamo per ogni vincolo $c \in C$ del suo insieme degli *assegnamenti totali consistenti* (o ammissibili) $\mathtt{img}(c)$.

Solo perché stiamo ragionando in senso di numeri, non vuole dire che solo numeri sono da considerarsi in assegnamenti. Se tutti i domini delle variabili di un vincolo sono finiti, allora è possibile descrivere il vincolo elencando tutti gli assegnamenti parziali consistenti in modo *estensionale* e il vincolo viene detto *tabellare*.

Le tabelle rappresentate efficientemente, permettono di ottenere velocemente dei risultati: a volte, la realizzazione di queste permette vantaggi rispetto a formulazioni.

> **Quali SSD su quali schede madre?**
>
> Realizzato un algoritmo di SBT, con funzione che verifica se un vincolo è soddisfatto o meno, possiamo realizzare una tabella di vincoli. Se ho un dominio che rappresenta i possibili tipi di SSD da installare su una scheda madre, e ho un vincolo che lega il tipo alla velocità del BUS, possiamo realizzare una tabella che mostra tutti i possibili SSD.

☰ **CSP con 3 variabili**

Supponiamo un CSP con 3 variabili x, y, z con dominio $I = [1..6]$. L'insieme delle variabili è $V = \{x, y, z\}$ e il dominio è $\Delta = I^3$ con ordinamento alfabetico. Il vincolo di CSP c è definito dalla proprietà: "x e z devono essere diversi ed entrambi pari".

Stiamo quindi dicendo che $c = \langle V_c, \Delta_c \rangle$ con $V_c = \{x, z\}$ e Δ_c esprimibile in forma estensionale come

$$\{(2,4),(2,6),(4,2),(4,6),(6,2),(6,4)\}$$

L'immagine del vincolo $\mathtt{img}(c)$ in forma estensionale di dimensione 6×6 come

$$\{(2,1,4),(2,2,4),\ldots,(6,6,4)\}$$

Soluzioni di CSP

Dato un CSP $\mathcal{P} = \langle V, D, C \rangle$, l'insieme delle soluzioni $\mathtt{sol}(\mathcal{P})$ del problema l'otteniamo facendo l'intersezione delle immagini di tutti i vincoli che sono presenti con la convenzione, e se non ce ne sono, la soluzione è il dominio stesso. $\mathcal{P}$ si dice *risolubile* se l'insieme ottenuto è non vuoto. Quindi: prendiamo ogni vincolo; per ognuno costruiamo il relativo dominio; per ogni dominio costruiamo l'immagine inserendo tutti i valori non coinvolti dai vincoli, ottenendo insieme di vettori tutti con stesso numero di elementi finito; facciamo l'intersezione.

$$\mathtt{sol}(\mathcal{P}) = \bigcap_{c \in C} \mathtt{img}(c)$$

Esiste almeno 1 vettore di n elementi, assegnamento completo e consistente, che è contenuto nelle immagini di tutti i vincoli che rispetta dunque tutti i vincoli. Questa rappresentazione ci permette di dire anche quando 2 CSP sono equivalenti: sfruttando il fatto che esiste un ordinamento delle variabili e che riusciamo a costruire l'insieme delle soluzioni, possiamo dire che CSP definiti su stesso insieme di variabili sono equivalenti, quando applicando lo stesso ordinamento i 2 insiemi di soluzioni ottenute, sono uguali.

Trasformazioni di CSP

Trasformazioni del CSP originario, possono mantenere l'equivalenza $\mathtt{sol}(\mathcal{P}_1) = \mathtt{sol}(\mathcal{P}_2)$, facendo semplificazioni che garantiscano la non modifica dell'insieme delle soluzioni (visto nella consistenza ad arco).

☰ **Example**

Consideriamo CSP $\mathcal{P}$ con 2 variabili x, y, con dominio per ciascuna $[-10..10]$. Un vincolo viene descritto tramite una proprietà $y = x^2$.

- Un $\mathcal{P}_1$ che restringa il set di soluzioni di y a $[0..10]$, è equivalente a $\mathcal{P}$, siccome qualsiasi numero al quadrato darà sempre e comunque un numero positivo.
- Un $\mathcal{P}_2$ che restringa il set di soluzioni di x a $[0..10]$, non è equivalente a $\mathcal{P}$, perché soluzioni di $\mathcal{P}$ non sono uqeuivalenti a soluzioni di $\mathcal{P}_1$.

$\mathcal{P}_2$ viene ottenuto tramite un'operazione detta di *rottura della simmetria* (symmetry breaking). È comunque un CSP che vale la pena di osservare, in quanto contiene un numero di elementi inferiori rispetto al dominio originale e ha le stesse soluzioni del CSP originali, quindi equivalente.

Un metodo generale di trasformazione di un problema in uno semplificato ed equivalente, prende il nome di propagazione dei vincoli e prevede di sfruttare le caratteristiche di ogni singolo vincolo, andando a eliminare valori dai domini su qui stiamo lavorando: se stiamo usando 1 vincolo soltanto per la propagazione, stiamo facendo una *propagazione locale*; se stiamo facendo una trasformazione usando un sottoinsieme dei vincoli, stiamo facendo un *filtro* (dei valori).

Un CSP dei quali vincoli unari sono stati propagati, si dice nodo-consistente, facendo valere la seguente proprietà

$$\forall v \in \mathtt{dom}(x), v \in \Delta_c$$

dal solito grafo dei vincoli tutti gli auto anelli vengono eliminati, riducendo così il dominio.

Un CSP dei quali vincoli binari sono stati propagati, si dice arco-consistente, se per qualsiasi valore del dominio della variabile x coinvolta nel vincolo, esiste un valore nel dominio di y, tale per cui la copia (v_x, v_y) sia parte del dominio del vincolo c

$$\forall v_x \in \mathtt{dom}(x), \exists v_y \in \mathtt{dom}(y) : (v_x, v_y) \in \Delta_c$$
$$\forall v_y \in \mathtt{dom}(y), \exists v_x \in \mathtt{dom}(x) : (v_x, v_y) \in \Delta_c$$

Solitamente, un buon equilibrio tra quella che è l'efficacia della propagazione (il togliere valori dai domini) e il tempo richiesto per farlo. Se troppo onerosa, farla per troppi assegnamenti richiede troppo tempo: AC-3 ha complessità polinomiale in caso pessimo $\mathcal{O}(ek^3)$, mentre per AC-4 è $\mathcal{O}(ek^2)$, con k il numero massimo di valori nel dominio ed e il numero di vincoli.

La proprietà di arco-consistenza può essere estesa a vincoli globali introducendo l'iperarco-consistenza. Una consistenza di percorso invece, sarebbe creata costruendo percorsi di più di due variabili collegate da vincoli binari.

Problemi con Vincoli Polinomiali

Consideriamo un CSP con un certo insieme di n variabili, ciascuna variabile all'interno di un dominio chiuso di reali $\mathtt{dom}(x_i) = [\underline{x}_i, \overline{x}_i]$. Un vincolo si dice polinomiale, se il suo Δ_c può essere identificato come il sottoinsieme del dominio del vincolo c, tali per cui esiste un polinomio p e un polinomio q nelle variabili in considerazione, per cui $p(\mathbf{x}) \odot q(\mathbf{x})$, scritto anche come

$$\Delta_c = \{\mathbf{x} \in \mathtt{dom}(c) : p(\mathbf{x}) \odot q(\mathbf{x})\}$$

dove $\odot$ rappresenta una relazione matematica tra $<, \leq, =, \neq, >, \geq$, mentre $p, q : \mathbb{R}^k \to \mathbb{R}$ sono due funzioni polinomiali con k cardinalità dell'insieme dei vincoli delle variabili.

In possesso di un vincolo polinomiale, in realtà ci bastano queste due descrizioni estensionali del vincolo.

$$\{\mathbf{x} \in \mathtt{dom}(c) : p(\mathbf{x}) \geq 0\} \qquad \{\mathbf{x} \in \mathtt{dom}(c) : p(\mathbf{x}) > 0\}$$

Non stiamo limitando nulla, perché note queste relazioni, riusciamo a ricostruire tutte le altre.

$$p(\mathbf{x}) = 0 \iff p(\mathbf{x}) \leq 0 \land p(\mathbf{x}) \geq 0$$
$$p(\mathbf{x}) \neq 0 \iff p^2(\mathbf{x}) > 0$$

Consistenza sugli intervalli

Per vincoli polinomiali, introduciamo un nuovo tipo di consistenza che prende il nome di consistenza sugli intervalli (bounds-consistent). Contrariamente a quello ipotizzato fino adesso, i valori nei domini sono tra di loro ordinati e le operazioni che andiamo a utilizzare per costruire il vincolo, lavorano sull'ordinamento.

I domini contengono valori numerici ed essendo tali sono intrinsecamente ordinati,(possiamo sempre dire che $1 \leq 2$). In aggiunta, abbiamo ipotizzato di partire da intervalli e quindi se facciamo un'operazione che mantiene l'equivalenza tra i problemi ottenendo altri intervalli ancora, allora stiamo semplificando il CSP lavorando localmente su ogni vincolo.

L'arco-consistenza non prevede che i valori nei domini siano ordinati, non prevede che le proprietà descriventi il vincolo lavorino sull'ordinamento. Basta soltanto capire se due valori sono uguali o diversi.

Nella consistenza a intervalli la questione è diversa: un CSP si dice consistente sugli intervalli se per ogni vincolo polinomiale c, vale che per ognuna delle sue variabili con dominio $[\underline{x}, \overline{x}]$, esiste almeno un assegnamento parziale del vincolo che abbia $\underline{x}$ per x e $\overline{x}$ per y, ammissibili. Andiamo a considerare un intervallo e diciamo:

- se i valori degli estremi dell'intervallo sono consistenti, allora non possiamo stringere l'intervallo perché altrimenti quella che andiamo a togliere è una coppia potenzialmente estendibile a soluzione unica;
- se i valori estremi non sono consistenti, allora possiamo provare a stringere l'intervallo perché non stiamo togliendo possibili soluzioni.

Il grande vantaggio è che così riusciamo a scartare numerosi valori sulla retta dei reali; più le porzioni sono lunghe e meno dovremo andare a lavorare per fare operazioni di assegnamento. Pensiamo ora di realizzare un algoritmo che sposti gli estremi d'intervallo fino a farli ad arrivare a 2 estremi che sono sicuramente bound-consistent: se l'intervallo scomparso, non esiste soluzione; se l'intervallo ha almeno un valore, allora soluzione c'è.

Dato $n \in \mathbb{N}$, $p : \mathbb{R}^n \to \mathbb{R}$ funzione polinomiale a coefficienti reali positivi, se $\underline{\mathbf{x}} \in \mathbb{R}^n_+$ e $\overline{\mathbf{x}} \in \mathbb{R}^n_+$, allora per ogni $\mathbf{v} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ vale

$$p(\underline{\mathbf{x}}) \leq p(\mathbf{v}) \leq p(\overline{\mathbf{x}})$$

Consideriamo un vincolo $p(\mathbf{x}) \geq 0$,
siccome p è una funzione polinomiale, abbiamo sempre un polinomio che la descrive, ipotizzando che questo sia scritto nel modo seguente

$$p(\mathbf{x}) = \sum_{i=1}^{n} a_i x_i + b$$

che sarebbe il polinomio lineare classico, dove un insieme di coefficienti viene moltiplicato e un termine noto sommato $b \geq 0$. A questo punto, spostiamo dalla parte destra, tutti i coefficienti che diventano effettivamente negativi: le x sono definite su $\mathbb{R}_+$, b è positivo, l'unica cosa che può essere negativa sono coefficienti; se il coefficiente è positivo allora non lo tocchiamo, se negativo $a_i < 0$, allora spostiamo il termine dalla parte opposta

$$\sum_{a_i > 0} a_i x_i + b \geq \sum_{a_i < 0} -a_i x_i$$

Fissiamo un coefficiente k e se $a_k > 0$ possiamo spezzare la somma in 2 parti: tutto quello che rimane tolto k, isolandolo, con

$$\sum_{i \neq k, a_i > 0} a_i x_i + a_k x_k + b \geq \sum_{a_i < 0} -a_i x_i$$

a_k con a un positivo $+$ qualcosa di positivo $+$ qualcosa di positivo, deve essere $\geq$ di un qualche cosa positivo. Se siamo interessati a capire che valore può assumere x_k, possiamo ricordarci che qualsiasi valore positivo assuma k, moltiplicato a positivo e sommato a positivo, è sicuramente maggiore dell'altro termine della proposizione (destra).

- u sarebbe quel qualche cosa di più grande, scritto come

$$\sum_{i \neq k, a_i > 0} a_i x_i \leq \sum_{i \neq k, a_i > 0} a_i \overline{x}_i = u$$

- l sarebbe quel qualche cosa di più piccolo, scritto come

$$l = \sum_{a_i < 0} -a_i \underline{x}_i \leq \sum_{a_i < 0} -a_i x_i$$

con proprietà che descrive il vincolo che impone

$$x_k \geq \frac{l - u - b}{a_k}$$

Per "sbucciare" la notazione formale magari poco chiara, vediamo un esempio per comprendere il significato dalla "notazione multi-indice".

CSP equivalente usando bound-consistency

Si consideri CSP con 3 variabili x, y, z, tutte con dominio reale $[1, 10]$, con la proprietà $2x + 3y - z \leq 1$. Per capire se siamo nel caso da noi interessato, poniamoci le domande:

- Le variabili sono positive? Sì, lo sono, quindi tutto il ragionamento di spezzare o meno il dominio delle variabili, non ci interessa.
- Il vincolo è uno polinomiale lineare? Sì, non ci sono prodotti tra variabili o potenze di variabili.
- Le variabili sono definite su intervallo? Sì, $[1, 10]$ è un intervallo.

Eseguiamo la procedura di bound-consistency, applicando la formula, cercando di spostare i due valori, almeno per una variabile.

Cominciamo da z, riscrivendo il vincolo originale in uno su cui è stata applicata la proprietà vista sopra (solo termini positivi, cambiando segno)

$$2x + 3y - z \leq 1 \quad \rightarrow \quad -2x - 3y + z \geq -1 \quad \rightarrow z + 1 \geq 2x + 3y$$

Siccome la x ha per ora un valore minimo $\underline{x} = 1$, e la y ha un valore minimo $\underline{y} = 1$ ($\texttt{dom}(x) = \texttt{dom}(y) = \texttt{dom}(z) = [1, 10]$), riscriviamo la disuguaglianza come

$$z + 1 \geq 2x + 3y \geq 5 \quad \rightarrow \quad z \geq 4$$

z è così stato ridefinito, trovandosi ora nel nuovo intervallo $[4, 10]$: abbiamo ridotto la dimensione del dominio.

Ragionando su x, sappiamo che $\underline{y} = 1$ e che $z + 1 \leq 11$ (siccome $\texttt{dom}(z) = [1, 10]$), riscriviamo la disuguaglianza come

$$11 \geq z + 1 \geq 2x + 3y \geq 2x + 3 \quad \rightarrow \quad x \leq 4$$

y ha un dominio ridotto di $[1, 4]$.

Ragionando su y,

$$11 \geq z + 1 \geq 2x + 3y \geq 2 + 3y \quad \rightarrow \quad y \leq 3$$

Il nuovo CSP equivalente a quello iniziale, ha i domini modificati

$$\texttt{dom}(x) = [1, 4] \quad \texttt{dom}(y) = [1, 3] \quad \texttt{dom}(z) = [4, 10]$$

CSP che non ammette soluzioni dopo bound-consistency

Si consideri CSP di 3 variabili x, y, z con dominio per ciascuna $[1, 10]$, con vincolo definito da $12x + 8y - z \leq 1$.

La proprietà che definisce il vincolo può essere riscritta come

$$12x + 8y - z \leq 1 \quad \rightarrow \quad -12x - 8y + z \geq -1 \quad \rightarrow \quad z + 1 \geq 12x + 8y$$

che considerando $\underline{x} = 1$ e $\underline{y} = 1$, risulta

$$z + 1 \geq 12x + 8y \geq 20$$

Ci viene quindi detto che $z + 1 \geq 20$ quando questo non è possibile, siccome $\overline{z} = 10$.

Abbiamo scoperto che il CSP non ammette soluzioni.

Il metodo per la riduzione dei domini, può essere usato anche in casi di non linearità, dove $\mathtt{dom}(x) = [\underline{x}, \overline{x}] \subseteq \mathbb{R}_+$ con $m \in \mathbb{N}_+$

$$\underline{x}^{m-1}x \leq x^m \leq \overline{x}^{m-1}x$$

i termini non lineari x^m possono essere ridotti a termini lineari.
In modo simile, se $\mathtt{dom}(y) = [\underline{y}, \overline{y}] \subseteq \mathbb{R}_+$ allora

$$\underline{x}y \leq xy \leq \overline{x}y \quad \wedge \quad \underline{y}x \leq xy \leq \overline{y}x$$

27/04/2023