Introduzione all'ottimizzazione discreta

Domenico Salvagnin

2018-10-01

1 Introduzione

Un problema di ottimizzazione P si puó formulare come:

$$\begin{aligned} \min(\textit{or} \max) f(x) \\ S \\ x \in D \end{aligned} \tag{1}$$

dove $f(x)$ è una funzione a valori reali nelle variabili x, D è il dominio di x e S è un insieme finito di vincoli. In generale, x è una tupla $(x_1, \dots, x_n)$ e D è un prodotto cartesiano $D_1 \times \cdots \times D_n$, e vale $x_j \in D_j$. Formalmente, un vincolo $c \in S$ è una funzione associata ad un sottoinsieme di variabili x_c il cui valore puó essere vero (in tal caso si parla di vincolo soddisfatto) o falso (vincolo violato).

Ogni $x \in D$ si dice *soluzione* di P. Una soluzione che soddisfi tutti i vincoli in S si dice *ammissibile*. L'insieme delle soluzioni ammissibili di un problema di ottimizzazione P si indica con $F(P)$.

Se il dominio D è discreto, si parla di *ottimizzazione discreta*. Se inoltre D è finito, cioé il numero di soluzioni è finito, si parla di *ottimizzazione combinatoria*.

Una soluzione ammissibile x^* si dice *ottima* se

$$f(x^*) \leq f(x) \quad \forall x \in F(P)$$

Un problema di ottimizzazione si dice *impossibile* (*infeasible*) se non ammette alcuna soluzione ammissibile, cioé $F(P) = \emptyset$. Un problema si dice *illimitato* (*unbounded*) se non esiste alcun limite inferiore a $f(x)$ per $x \in F(P)$. In questo corso assumeremo sempre che un problema di ottimizzazione sia impossibile, illimitato o abbia ottimo finito (ad esempio, sono esclusi problemi del tipo $\min e^{-x}, x \geq 0$).

Un problema di ottimizzazione si dice risolto quando ne viene trovata una soluzione ottima (e si dimostra che è tale) oppure quando si dimostra che il problema è impossibile o illimitato.

2 Paradigmi di ottimizzazione

In generale, un problema di ottimizzazione nella forma (1) è intrattabile, nel senso che non esistono algoritmi efficienti (o addirittura algoritmi) per la sua risoluzione.

È pertanto necessario considerare dei casi particolari di (1), in modo da poterne sfruttare la struttura. Ecco alcuni esempi:

2.1 Programmazione Lineare (LP)

Un problema di programmazione lineare consiste nella minimizzazione di una funzione lineare soggetta ad un numero finito di vincoli lineari. In generale si ha la forma:

$$\begin{aligned} &\min cx \\ &\quad a_i x \sim b_i \quad i = 1, \ldots, m \\ &\quad l_j \le x_j \le u_j \quad j = 1, \ldots, n \end{aligned} \tag{2}$$

dove $\sim \in \{\le, \ge, =\}$, $l_j \in \mathbb{R} \cup \{-\infty\}$ e $u_j \in \mathbb{R} \cup \{+\infty\}$. Pertanto i domini delle singole x_j sono intervalli di $\mathbb{R}$.

Si noti che a prima vista un problema di programmazione lineare *non* rientra nell'ottimizzazione discreta, in quanto i domini delle variabili sono continui. Vedremo in seguito che alcune proprietà strutturali dei problemi di programmazione lineare fanno sì che si possa comunque parlare di ottimizzazione discreta.

2.2 Programmazione Lineare Intera (MIP)

Un problema di programmazione lineare intera consiste nella minimizzazione di una funzione lineare soggetta ad un numero finito di vincoli lineari, con in piú il vincolo che alcune variabili devono assumere valori interi. In generale si ha:

$$\begin{aligned} &\min cx \\ &\quad a_i x \sim b_i \quad i = 1, \ldots, m \\ &\quad l_j \le x_j \le u_j \quad j = 1, \ldots, n = N \\ &\quad x_j \in \mathbb{Z} \quad \forall j \in J \subseteq N = \{1, \ldots, n\} \end{aligned} \tag{3}$$

Se $J = N$, si parla di programmazione lineare intera pura, altrimenti di programmazione lineare intera mista. Ovviamente se $J = \emptyset$ si ha un semplice problema di programmazione lineare.

I paradigmi LP e MIP restringono notevolmente il tipo di vincoli a disposizione per modellare un problema di ottimizzazione. Questo porta notevoli vantaggi per lo studio e l'implementazione degli algoritmi risolutivi, a costo di una piú faticosa (quando possibile) fase di modellazione. Vedremo comunque che, almeno per la classe MIP, questo non implica in ogni caso una eccessiva restrizione del tipo di problemi che si possono formulare.

2.3 Programmazione con Vincoli (CP)

Un problema di programmazione con vincoli è definito da una terna (X, D, C), dove X è un insieme di variabili con dominio *finito* D e C è un insieme finito di vincoli di natura arbitraria.

Il paradigma CP è volutamente molto piú libero nella definizione dei vincoli che si possono usare per modellare un problema, risultando spesso in una modellazione piú immediata e compatta. Ovviamente, se i vincoli usati non hanno una struttura che permetta lo sviluppo di algoritmi efficienti, la capacitá di risoluzione ne verrá penalizzata.

Si noti che, nella sua forma classica, un problema CP *non* ha una funzione obiettivo e lo scopo è quello di trovare "semplicemente" una soluzione ammissibile. Si parla in tal caso di problema di *ammissibilitá*.

2.4 Soddisfacilitá booleana (SAT)

Il problema della soddisfacibilitá booleana è un caso particolare di problema di ammissibilità in cui tutte le variabili sono booleane (cioé possono assumere solo i valori *True* o *False* e i vincoli sono un numero finito di clausole booleane. Ricordiamo che un *literal* è una variabile o la sua negazione e che una clausola (*clause*) è una disgiunzione di literals.

Come nel caso LP/MIP, in un problema SAT il linguaggio a disposizione per modellare un problema di ottimizzazione è molto ristretto. Nonostante questo, molti problemi di progettazione di circuiti logici (di notevole importanza pratica) possono essere espressi naturalmente con questo paradigma.

Anche un problema SAT è un problema di ammissibilitá.

3 Concetti generali dell'ottimizzazione

Nonostante la teoria sviluppata e gli algoritmi siano anche molto diversi per i quattro paradigmi considerati, alcuni concetti fondamentali sono comuni a tutti gli approcci all'ottimizzazione.

3.1 Ricerca

Dato un problema P, la ricerca consiste nel risolvere una sequenza di restrizioni $P_1, \dots, P_m$ di P. Ciascuna *restrizione* P_k è ottenuta da P aggiungendo degli ulteriori vincoli.

L'idea di base è che aggiungendo dei vincoli, e quindi restringendo lo spazio delle soluzioni ammissibili, il problema di ottimizzazione possa risultare piú semplice da risolvere.

Una ricerca si dice *esaustiva* se l'insieme delle restrizioni considerate "copre" tutto lo spazio delle soluzioni ammissibili di P, cioé se vale

$$\bigcup_i F(P_i) = F(P)$$

In caso di ricerca esaustiva è pertanto possibile risolvere P risolvendo tutte le restrizioni P_i e prendendo la soluzione migliore.

Si noti che data una restrizione, una qualsiasi sua soluzione ammissibile è anche una soluzione ammissibile per il problema di partenza, e pertanto costituisce un limite superiore (upper bound) valido per il valore della soluzione ottima di P.

La forma piú semplice di ricerca esaustiva prende il nome di *generate-and-test* e consiste nel generate esplicitamente tutte le soluzioni $x \in D$, verificare quali verifichino i vincoli del problema e prendere la migliore. Questo equivale a considerare le restrizioni di P in cui x è fissato ad un particolare valore $v \in D$.

Una forma piú evoluta di ricerca, che è alla base di quasi tutti gli algoritmi enumerativi usati in pratica, è la ricerca ad albero, in cui lo spazio di ricerca di P viene diviso ricorsivamente fino ad ottenere delle restrizioni (corrispondenti ai nodi foglia) sufficientemente facili da risolvere.

3.2 Inferenza

Un vincolo C è dedotto/derivato da un insieme di vincoli S se ogni $x \in D$ che soddisfa S soddisfa anche C.

Dal punto di vista della modellazione un tale vincolo è ridondante (per definizione non cambia l'insieme delle soluzioni ammissibili), ma potrebbe essere in ogni caso molto utile per l'efficienza degli algoritmi di ricerca, in quanto potrebbe "esplicitare" delle informazioni che sono valide globalmente, ma che non sono derivabili da nessun vincolo di S preso separatamente.

Si consideri ad esempio la seguente regione ammissibile

$$x_1 + x_2 \geq 1 \quad x_1 - x_2 \geq 0 \quad x_1, x_2 \in \{0, 1\}$$

È immediato verificare che il vincolo $x_1 = 1$ è valido per S, ma non è deducibile ne da $x_1 + x_2 \geq 1$ ne da $x_1 - x_2 \geq 0$ se presi separatamente.

3.3 Rilassamento

Un *rilassamento* di un problema di ottimizzazione P è un altro problema di ottimizzazione R ottenuto da P

1. eliminando alcuni vincoli
2. sostituendo la funzione obiettivo $f(x)$ con una sua approssimazione inferiore $g(x)$

Piú formalmente, R è un rilassamento di P se

1. $F(P) \subseteq F(R)$
2. $g(x) \leq f(x) \quad \forall x \in F(P)$

Il motivo per cui si ricorre ad un rilassamento R di P è essenzialmente lo stesso che per le restrizioni: si spera che R sia più facile da risolvere di P e che, seppur non equivalente, possa fornire delle informazioni utili per la risoluzione d P.

In particolare, dato un rilassamento è possibile derivare che:

1. se $F(R) = \emptyset$, allora P è impossibile.
2. se x^* è una soluzione ottima di R, $x^* \in F(P)$ e $g(x^*) = f(x^*)$, allora x^* è ottimo per P.
3. se x^* è una soluzione ottima di R, $g(x^*)$ è un limite inferiore (lower bound) valido per il valore ottimo di P.

Inoltre, un rilassamento puó essere usato in combinazione con uno schema di ricerca per velocizzarlo, in particolare considerando i rilassamenti R_k delle restrizioni P_k.

3.4 Dualitá

Un problema di ottimizzazione P è comunemente interpretato come un problema di ricerca, che consiste nel "trovare" la soluzione ottima x^*.

Alternativamente, lo stesso problema puó essere visto come un problema di inferenza, che consiste nel derivare dall'insieme dei vincoli di P il piú stretto lower bound possibile su $f(x)$. Il problema di trovare la dimostrazione (proof) che porta al lower bound piú stretto prende il nome di *problema duale*.

Tale punto di vista è un punto cardine della programmazione matematica e dell'ottimizzazione in generale.