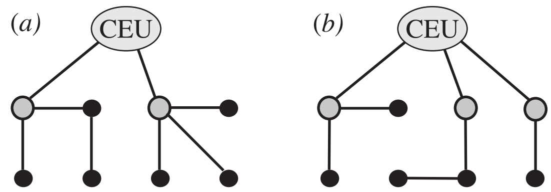
Capitolo 4

Ottimizzazione Combinatoria

4.1 Introduzione

L'*Ottimizzazione Combinatoria* (*OC*) studia i problemi di ottimizzazione in cui l'insieme ammissibile è definito in termini di strutture combinatorie, tra le quali svolgono sicuramente un ruolo di rilievo i grafi. La caratteristica fondamentale di tali problemi è quindi quella di avere insiemi ammissibili *discreti*, a differenza ad esempio della *PL* in cui l'insieme ammissibile è *continuo*. Ciò comporta che le metodologie necessarie per affrontare problemi di *OC* sono spesso diverse da quelle utilizzate per risolvere problemi nel continuo.
Nei Capitoli 1 e 3 abbiamo già incontrato molti problemi di *OC*; in particolare, nel Capitolo 3 abbiamo descritto alcune importanti classi di problemi di *OC* che ammettono algoritmi risolutivi di complessità polinomiale. Moltissimi problemi di *OC* sono invece "difficili" ($\mathcal{NP}$-ardui, si veda l'Appendice A), ed è di questi problemi che ci occuperemo in questo capitolo e nei successivi. In effetti, i problemi di *OC* per i quali esistono algoritmi efficienti hanno caratteristiche molto specifiche: capita molto spesso che varianti apparentemente minori di un problema di *OC* "facile" siano invece difficili.

Esempio 4.1: Un problema di progetto di rete



Si consideri il problema della Banca Gatto & Volpe definito in 1.2.2.2. Si supponga adesso che la banca intenda aggiornare il sistema informativo, sostituendo la gestione attraverso terminali a caratteri che operano su un mainframe centralizzato con un più moderno sistema client-server in cui le filiali possono effettuare interrogazioni sul database centrale ed elaborare i risultati localmente. Chiaramente, questo tipo di cambiamento aumenta sensibilmente la quantità di informazione che viene inviata sulla rete. Data una soluzione, cioè un albero, tutto il traffico inviato al CEU da tutte le filiali appartenenti ad un certo sottoalbero deve passare per l'unico collegamento tra il nodo "radice", che rappresenta il CEU, e la filiale che funge da radice del sottoalbero. Quindi, sarebbero proprio questi collegamenti "critici" ad essere saturati per primi qualora la rete non fosse dimensionata opportunamente. Inoltre, in caso di un guasto ad una di queste linee tutte le filiali rappresentate da nodi nel sottoalbero corrispondente verrebbero disconnesse dal CEU. Ad ogni nodo i diverso dalla radice possiamo associare quindi un peso b_i corrispondente alla massima banda utilizzata dalla corrispondente filiale: per fare in modo che tutte le filiali abbiano sempre sufficiente banda per comunicare con il CEU, dobbiamo richiedere che nella soluzione del problema la somma dei pesi dei nodi in ciascun sottoalbero della radice sia al più Q, dove Q è la capacità dei collegamenti. Esempi di un albero di copertura non ammissibile ad ammissibile per $Q = 3$ sono dati nella figura qui sopra, ove tutti i nodi hanno peso unitario ed i nodi evidenziati sono quelli collegati al CEU dai collegamenti "critici". Il corrispondente problema di *OC*, noto come *Constrained MST* (CMST) è una variante apparentemente minore di (MST), ma è un problema $\mathcal{NP}$-arduo (tranne per valori particolari di Q) mentre, come abbiamo visto, (MST) è polinomiale. Infatti, nella pratica istanze di (MST) su grafi con 10000 e più nodi non presentano alcuna difficoltà, mentre istanze di (CMST) con soli 200 nodi sono a tutt'oggi per lo più insolubili.

Si può affermare che la grande maggioranza dei problemi di Ottimizzazione Combinatoria che si incontrano nella realtà sono difficili. Per questo, la conoscenza dei problemi di *OC* più rilevanti, e dei relativi algoritmi risolutivi, è una parte importante delle competenze specifiche di un esperto di Ricerca Operativa. In effetti, il fatto che non esistano algoritmi generali in grado di risolvere problemi di *OC* di grande dimensione, mentre esistono metodologie generali che possono essere applicate caso per caso per la costruzione di algoritmi ad-hoc per un certo problema, giustifica in buona parte la necessità di formare specialisti con competenze di ottimizzazione. Inoltre, tipicamente gli algoritmi per problemi di *OC* "difficili" fanno ricorso ad algoritmi per problemi più "facili", quali la *PL* o i problemi di flusso su reti: questo giustifica in buona parte l'interesse per la soluzione efficiente di problemi "facili", anche se la maggior parte dei modelli provenienti da applicazioni reali corrispondono a problemi "difficili".

In questo capitolo introdurremo e discuteremo in modo generale alcune delle principali proprietà dei problemi di *OC* che hanno rilevanza per lo sviluppo di approcci algoritmici al problema. I Capitoli 5, 6 e 7 descriveranno invece alcune classi di algoritmi per problemi di *OC*.

4.2 Programmazione Lineare Intera (Mista)

I problemi di *Programmazione Lineare Intera* (*PLI*) si differenziano da quelli di *PL* unicamente per il fatto che tutte le variabili possono assumere solamente valori interi; questo *vincolo di integralità* ha però un enorme impatto. Innanzi tutto, come abbiamo visto nel Capitolo 1, "l'espressività" del modello aumenta in maniera consistente: le variabili intere possono essere utilizzate per modellare condizioni logiche (decisioni "tutto o niente") e situazioni in cui le decisioni si prendono tra un numero finito di possibili alternative. Si può affermare che la grande maggioranza dei modelli utilizzati in pratica sono di *PLI*, in quanto nella maggior parte delle applicazioni reali esistono condizioni logiche ed è necessario compiere scelte discrete. Come abbiamo visto nel Capitolo 1, molti problemi di *OC* possono essere formulati come problemi di *PLI*. In effetti, nella pratica si tende a considerare sostanzialmente coincidenti le due classi dei problemi. Questo è dovuto a due ragioni concomitanti:

- da una parte, i problemi di *OC* vengono normalmente formulati come problemi di *PLI*, e buona parte degli approcci risolutivi per i problemi di *OC* si basa su tali formulazioni;
- d'altra parte, quasi tutte le tecniche efficienti per la soluzione di problemi di *PLI* si fondano sull'individuazione e sullo sfruttamento di strutture combinatorie specifiche all'interno del modello *PLI*, ossia di (sotto)problemi di *OC* che corrispondono al modello di *PLI* o a sue parti.

Si consideri ad esempio il problema (MST): per tale problema abbiamo fornito sia una formulazione in termini di *OC*, nella quale l'insieme ammissibile è definito come l'insieme di tutti gli alberi di copertura di un grafo dato, sia una formulazione in termini di *PLI* in cui l'insieme ammissibile è definito come l'insieme di tutti i vettori in $\{0,1\}^m$ ($m = |A|$) che rispettano un certo insieme di vincoli lineari. Tali vincoli assicurano che tutte le soluzioni ammissibili del problema possano essere interpretate come vettori di incidenza di sottografi connessi del grafo originario, e la funzione obiettivo assicura che tra tutti questi venga selezionato il vettore di incidenza di un albero di copertura di costo minimo.

Esistono quindi forti relazioni tra i problemi di *OC* e quelli di *PLI*; le due classi non sono però completamente coincidenti. Da una parte la *PLI* fornisce un potente linguaggio per esprimere in modo uniforme problemi di *OC* definiti su strutture molto diverse, ed anche problemi che può essere molto difficile ricondurre a problemi di *OC* ben definiti: la *PLI* è in qualche senso "più espressiva" dell'*OC*. D'altra parte, possono esistere molte formulazioni di *PLI* diverse dello stesso problema di *OC*, e la formulazione come problema di *OC* è spesso "più informativa" di quelle come *PLI*, nel senso che può essere più facile derivare proprietà utili per la soluzione del problema lavorando direttamente sulla sua struttura combinatoria piuttosto che sulle sue formulazioni come *PLI*.

Visto che ai problemi di *PLI* si possono applicare le stesse trasformazioni che abbiamo visto nel Capitolo 2 per i problemi di *PL*, possiamo pensare che tutti i problemi di *PLI* siano espressi in forme standard analoghe a quelle già viste, ad esempio

$$(PLI) \qquad \max\{\, cx \;:\; Ax \leq b \;,\; x \in \mathbb{Z}^n \,\} \;.$$

Si parla inoltre di problemi di *Programmazione Lineare Mista* (*PLM*) quando solamente alcune delle variabili sono vincolate ad essere intere: tali problemi hanno quindi la forma

$$(PLM) \qquad \max\{\, c'x' + c''x'' \;:\; A'x' + A''x'' \leq b \;,\; x' \in \mathbb{Z}^n \,\} \;.$$

Quasi tutti gli approcci per la *PLI* che descriveremo possono essere generalizzati alla *PLM*, spesso solamente al costo di complicazioni nella descrizione. Per semplicità ci riferiremo quindi sempre a problemi di *PLI*.

4.2.1 Il rilassamento continuo

Il motivo principale per cui è ampiamente diffusa la pratica di formulare e risolvere problemi di *OC* come problemi di *PLI* risiede nel fatto che per questi ultimi si possono utilizzare i potenti risultati teorici e le efficienti metodologie algoritmiche relative ai problemi di *PL*. Descriviamo adesso brevemente le principali relazioni tra la *PLI* e la *PL*; questo argomento sarà poi ripreso in maggiore dettaglio nel Capitolo 6. L'insieme ammissibile del problema (PLI),

$$\mathcal{F} = \{ \, x \in \mathbb{Z}^n \; : \; Ax \leq b \, \} \;\; ,$$

ha fondamentalmente le caratteristiche di quello mostrato in Figura 4.1(a): i vincoli lineari $Ax \leq b$ definiscono un poliedro convesso

$$\bar{\mathcal{F}} = \; \{ \, x \in \mathbb{R}^n \; : \; Ax \leq b \, \} \quad ,$$

e l'insieme ammissibile è formato dall'intersezione tra la "griglia" dei punti a coordinate intere e $\bar{\mathcal{F}}$, ossia da tutti i punti a coordinate intere che appartengono al poliedro (punti bianchi). Essendo formato da punti isolati, $\mathcal{F}$ non è convesso, il che spiega sostanzialmente la difficoltà del problema.

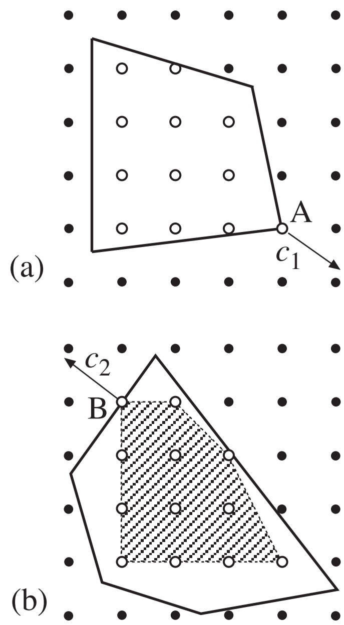


Figura 4.1: *PL* e *PLI*

Non entreremo qui in una spiegazione dettagliata del perchè la non convessità renda il problema difficile, limitandoci ad illustrare i principi generali con un esempio; ulteriori commenti su questo punto saranno poi fatti nel seguito. Consideriamo per questo la minimizzazione di una funzione non convessa su $\mathbb{R}^n$: questo compito è in generale difficile quanto minimizzare (o massimizzare) una funzione lineare su un insieme non convesso. Infatti, è possibile riformulare (quasi) tutti i problemi di *PLI* come problemi di minimizzazione di funzioni non convesse; si noti che abbiamo visto esempi della trasformazione inversa, cioè da problemi di minimizzazione non convessa a problemi di *PLI*, nel Capitolo 1.

Esercizio 4.1 *Si dimostri formalmente l'equivalenza tra la* PLI *e la minimizzazione di una funzione nonconvessa (suggerimento: si determini un* singolo vincolo nonlineare *equivalente alla richiesta* $x \in \{0,1\}$, *si estenda l'idea a* $x \in \{0,1\}^n$ *e poi a* $x \in \mathbb{Z}^n$, *ed infine si rimpiazzi il vincolo con un'opportuna modifica della funzione obiettivo, sotto opportune ipotesi se necessario).*

In Figura 4.2 sono mostrate un esempio di funzione convessa (a) e uno di funzione non convessa (b) in una sola variabile: con x^* sono indicati i *minimi globali* delle funzioni, ossia i punti che si cercano quiando si minimizzano le funzioni. Con x_1 ed x_2 sono indicati due *minimi locali* della funzione nonconvessa che non sono minimi globali; invece, tutti i minimi locali di qualsiasi funzione convessa sono anche minimi globali. In generale, determinare un minimo locale di una funzione con opportune proprietà di regolarità, ad esempio differenziabile con continuità, è "facile": l'informazione al primo ordine sulla funzione indica "da che parte andare". Nel caso non convesso, però, una volta determinato un minimo locale non si ha nessuna indicazione sull'esistenza di altri minimi locali (migliori) e sulla loro posizione. Questi concetti saranno comunque ripresi nel Capitolo 5.

Dato che $\mathcal{F}$ è contenuto in $\bar{\mathcal{F}}$, quest'ultimo fornisce "un'approssimazione" di $\mathcal{F}$ che può essere sfruttata algoritmicamente. Si consideri infatti il *rilassamento continuo* di (PLI)

$$(RC) \qquad \max \{ \, cx \; : \; Ax \leq b \, \} \;\; ,$$

cioè il problema di *PL* corrispondente al rilassamento dei vincoli di integralità $x \in \mathbb{Z}^n$. Questo problema può essere efficientemente risolto, e permette di derivare informazione sul problema originario. Ad

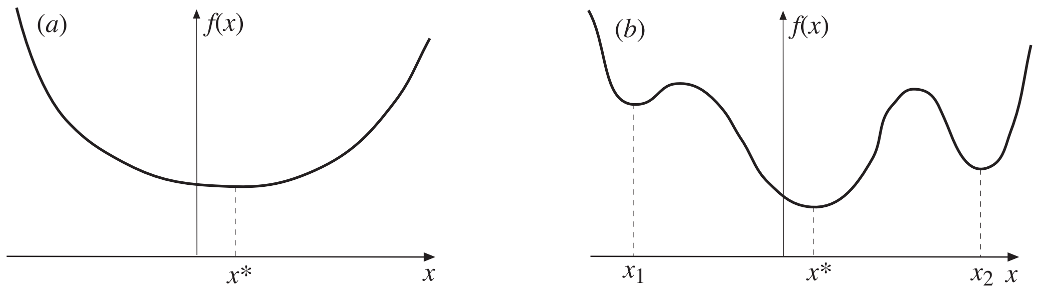


Figura 4.2: Funzioni convesse e nonconvesse

esempio, il valore ottimo della sua funzione obiettivo, $z(RC)$, fornisce una valutazione superiore del valore ottimo della funzione obiettivo di (PLI), ossia $z(RC) \geq z(PLI)$; l'utilità di questa relazione sta nel fatto che $z(RC)$, al contrario di $z(PLI)$, è efficientemente calcolabile in quanto (RC) è facile. Inoltre, è immediato verificare che vale il seguente risultato:

Lemma 4.1 *Sia x^* una soluzione ottima di (RC): se $x^* \in \mathbb{Z}^n$, ossia x^* è ammissibile per (PLI), allora x^* è ottima per (PLI).*

Un caso in cui si verificano le condizioni del Lemma 4.1 è mostrato in Figura 4.1(a), se la funzione obiettivo è c_1: è immediato verificare geometricamente che il punto "A", la soluzione ottima del rilassamento continuo, è intera ed è anche la soluzione ottima del problema di *PLI*. Il rilassamento continuo fornisce quindi un modo per tentare di calcolare una soluzione ammissibile per (PLI), ed in ogni caso fornisce una valutazione superiore sul valore di $z(PLI)$.

Esempio 4.2: Valutazioni superiori ed inferiori
Si consideri ad esempio il problema della Pintel: abbiamo visto che il suo rilassamento continuo ha soluzione ottima $(4, 1)$ (in unità di 100000 processori). Se consideriamo il vincolo di integralità sui wafers tale soluzione non è ammissibile, ma ci fornisce comunque una stima per eccesso del massimo ricavo disponibile, pari a 220 milioni di dollari. Si consideri adesso la soluzione ammissibile $(w_P, w_C) = (2666, 334)$ corrispondente a $(x_P, x_C) = (3.995, 1.002)$: tale soluzione permette un ricavo di 219.99 milioni di dollari. Possiamo quindi affermare che la soluzione ammissibile fornisce, alla peggio, un ricavo inferiore di 10000$ al massimo possibile, ossia più del 99.995% del massimo ricavo possibile. Per la maggior parte degli scopi pratici, determinare una soluzione di questo tipo può essere considerato equivalente ad aver risolto il problema. Si noti che l'aver determinato la soluzione non è di per sé sufficiente: quello che permette di "esserne soddisfatti" è l'essere in grado di valutarne la "qualità", il che è reso possibile dalla valutazione superiore fornita dal rilassamento continuo.

4.2.2 Formulazioni di PL equivalenti per la PLI

Un'importante osservazione è che *lo stesso insieme* ammissibile per un problema di *PLI* può essere specificato attraverso poliedri "sostanzialmente diversi"; ciò è mostrato in Figura 4.1(b), dove il poliedro mostrato (in tratto continuo) definisce lo stesso insieme di soluzioni ammissibili di quello in Figura 4.1(a) pur essendo "completamente diverso". Quindi esistono *formulazioni diverse dello stesso problema di PLI*. Queste formulazioni sono equivalenti per quanto riguarda il problema di *PLI*, ma non per quanto riguarda i rilassamenti continui. Risulta infatti intuitivamente chiaro come la valutazione superiore su $z(PLI)$ fornita da $z(RC)$ sarà tanto migliore quanto più il poliedro $\bar{\mathcal{F}}$ è "aderente" all'insieme ammissibile $\mathcal{F}$ di (PLI). In effetti questa nozione dipende anche dalla funzione obiettivo: ad esempio, come abbiamo visto il poliedro di Figura 4.1(a) è "buono" se la funzione obiettivo è c_1, mentre se la funzione obiettivo fosse c_2 allora sarebbe il poliedro di Figura 4.1(b) ad essere "buono", in quanto la soluzione ottima del corrispondente rilassamento continuo è il punto "B" che è anche la soluzione ottima del problema intero. Viceversa, nessuno dei due poliedri è "buono" per l'altra funzione obiettivo: è immediato verificare che le soluzioni ottime del rilassamento continuo non sono intere, e che vale $z(RC) > z(PLI)$.
Tra tutti i rilassamenti continui, ne esiste peró uno "buono" per *qualsiasi* funzione obiettivo; esso è "completamente aderente" a $\mathcal{F}$. Infatti, è possibile dimostrare che se tutti gli elementi della matrice

A e del vettore b sono razionali (interi)[1], allora *l'inviluppo convesso* di $\mathcal{F}$

$$\tilde{\mathcal{F}} = Conv(\mathcal{F})$$

è un poliedro, cioè può essere rappresentato da un sistema finito di disequazioni $\tilde{A}x \leq \tilde{b}$; in Figura 4.1(b), il poliedro tratteggiato è l'inviluppo convesso dei punti ammissibili. Quanto appena detto sembrerebbe contraddire l'affermazione secondo cui i problemi di *PLI* sono difficili: infatti, il problema

$$(\tilde{RC}) \qquad \max\{\ cx\ :\ \tilde{A}x \leq \tilde{b}\ \}\quad .$$

$(\tilde{RC})$ è chiaramente un rilassamento di (PLI), in quanto $\mathcal{F} \subset \tilde{\mathcal{F}}$. È però possibile dimostrare che $\tilde{\mathcal{F}}$ gode della seguente *proprietà di integralità*:

Definizione 4.1 *Un poliedro $\mathcal{P}$ (non vuoto) gode della* proprietà di integralità *se vale una delle due seguenti definizioni equivalenti:*

- *tutti i vertici di $\mathcal{P}$ hanno coordinate intere;*
- *il problema* $\max\ \{\ cx\ :\ x \in \mathcal{P}\ \}$ *ammette una soluzione ottima intera per* qualsiasi *scelta del vettore $c \in \mathbb{R}^n$ per cui non è superiormente illimitato.*

Esercizio 4.2 *Dimostrare l'equivalenza delle due condizioni nella definizione precedente (suggerimento: un'implicazione è ovvia, per l'altra si usi il fatto che ad ogni vertice del poliedro è associata almeno una base B di cui il vertice è la soluzione di base associata e si usi la matrice di base A_B per costruire un vettore c tale per cui quel vertice è ottimo).*

Come sappiamo dalla teoria della *PL*, esiste una soluzione ottima x^* di $(\tilde{RC})$ che giace su un vertice di $\tilde{\mathcal{F}}$: siccome tutti i vertici di $\tilde{\mathcal{F}}$ hanno coordinate intere, x^* è anche una soluzione ottima di (PLI). Di conseguenza, è possibile risolvere (PLI) al costo della soluzione del problema di *PL* $(\tilde{RC})$.

Abbiamo visto (nel Teorema 3.11) che la formulazione "naturale" del problema di flusso di costo minimo, e quindi di tutti i problemi del Capitolo 3 che ad esso possono essere ricondotti, gode della proprietà di integralità (purchè le capacità degli archi e i deficit dei nodi siano interi). Come vedremo nel paragrafo 5.1.3, questo vale anche per il problema dell'MST: si può quindi affermare che questa proprietà, ossia il fatto di trovarsi "al confine" tra l'ottimizzazione discreta e quella continua, è ciò che rende "facili" i problemi del Capitolo 3. Per il generico problema di *PLI*, però, la rappresentazione di $\tilde{\mathcal{F}}$, ossia l'insieme dei vincoli $\tilde{A}x \leq \tilde{b}$ che lo definiscono, non è nota: tutto quello di cui si dispone è la sua rappresentazione "approssimata" data dai vincoli $Ax \leq b$. In generale, è possibile dimostrare che costruire la rappresentazione di $\tilde{\mathcal{F}}$ per un qualunque problema di (PLI) è "difficile"; è interessante formalizzare questa nozione, in quanto per farlo dobbiamo introdurre alcuni concetti che risulteranno utili in seguito.

4.2.3 Diseguaglianze valide

Una diseguaglianza $dx \leq \delta$ si dice *valida* per $\tilde{\mathcal{F}}$ se è soddisfatta da ciascun punto $x \in \tilde{\mathcal{F}}$; equivalentemente, si può dire che $dx \leq \delta$ è valida per (PLI) se è soddisfatta da tutte le soluzioni (intere) del problema. Si noti che tutte le disequazioni $\tilde{A}_i x \leq \tilde{b}_i$ che definiscono $\tilde{\mathcal{F}}$ sono valide; $\tilde{A}$ e $\tilde{b}$ formano una rappresentazione minimale di $\tilde{\mathcal{F}}$ se la rimozione di una qualsiasi di queste disequazioni definisce un poliedro che contiene strettamente $\tilde{\mathcal{F}}$. In generale, non è necessario disporre di tutta la rappresentazione di $\tilde{\mathcal{F}}$ per risolvere $(\tilde{RC})$: è sufficiente essere in grado di risolvere il seguente *problema di separazione*

Dato $\bar{x} \in \mathbb{R}^n$, esiste una diseguaglianza valida $dx \leq \delta$ per $\tilde{\mathcal{F}}$
che non è soddisfatta da $\bar{x}$, ossia tale che $d\bar{x} > \delta$?

[1]Questi sono comunque i numeri che i calcolatori digitali trattano efficientemente; mediante sistemi di calcolo simbolico è possibile trattare anche numeri irrazionali ma con un'efficienza enormemente minore, il che ne limita fortemente l'uso in applicazioni come quelle qui descritte.

Chiaramente, il problema di separazione permette di determinare se un dato punto $\bar{x}$ appartiene o no a $\tilde{\mathcal{F}}$; per il caso in cui $\bar{x} \notin \tilde{\mathcal{F}}$ viene richiesta una "dimostrazione" della non appartenenza sotto forma di una diseguaglianza che separa $\bar{x}$ da $\tilde{\mathcal{F}}$. Una tale diseguaglianza viene anche detta un *taglio* per $\bar{x}$. Per meglio comprendere l'utilità del problema di separazione, si assuma di avere una rappresentazione "approssimata" $A^{(0)}x \leq b^{(0)}$ di $\tilde{\mathcal{F}}$, ad esempio quella fornita dal sistema $Ax \leq b$, e di risolvere il corrispondente rilassamento continuo. Sia $x^{(0)}$ la soluzione ottima del rilassamento, e si risolva il corrispondente problema di separazione: se $x^{(0)} \in \tilde{\mathcal{F}}$ allora è una soluzione ottima per $(\tilde{RC})$, e quindi $cx^{(0)} = z(PLI)$, altrimenti viene determinato un taglio $dx \leq \delta$ per $x^{(0)}$. Si consideri quindi il sistema $A^{(1)}x \leq b^{(1)}$ in cui

$$A^{(1)} = \begin{bmatrix} A^{(0)} \\ d \end{bmatrix} \qquad \text{e} \qquad b^{(1)} = \begin{bmatrix} b^{(0)} \\ \delta \end{bmatrix} \;;$$

chiaramente, $A^{(1)}$ e $b^{(1)}$ definiscono una nuova rappresentazione di $\tilde{\mathcal{F}}$ "meno approssimata" della precedente: infatti, almeno il punto $x^{(0)}$ non ne fa più parte. Risolvendo il nuovo rilassamento di (PLI) corrispondente a questo nuovo sistema si otterrà una nuova soluzione $x^{(1)} \neq x^{(0)}$, e si potrà iterare il procedimento fino a determinare una soluzione ottima di $(\tilde{RC})$.

Disponendo di un algoritmo efficiente per risolvere il problema di separazione rispetto a $\tilde{\mathcal{F}}$, si disporrebbe quindi di un algoritmo per risolvere $(\tilde{RC})$, e quindi (PLI). In effetti, è possibile dimostrare che, con tecniche in parte analoghe a quella sopra accennata (ma che utilizzano un diverso modo per risolvere i problemi di *PL*, detto *metodo degli ellissoidi*), è possibile risolvere in tempo polinomiale qualunque problema di *PL* su un poliedro $\mathcal{P}$, anche se definito da un numero esponenziale di disequazioni, purché si disponga di un *separatore polinomiale* per $\mathcal{P}$, ossia si sia in grado di risolvere il corrispondente problema di separazione in tempo polinomiale. La conseguenza di questo risultato è che, dato un problema $\mathcal{NP}$-arduo, il problema di separazione associato all'inviluppo convesso delle soluzioni intere di qualunque sua formulazione come *PLI* è anch'esso $\mathcal{NP}$-arduo.
Può essere utile discutere un esempio in qualche modo inverso, in cui un problema la cui formulazione *PLI* "esatta" richiede un numero esponenziale di vincoli risulta ciò nonostante facilmente risolubile. Il problema è (MST): si può infatti dimostrare che la formulazione presentata nel paragrafo 1.2.2.2, che usa le disuguaglianze (1.6) (dette *cutset inequalities*), individua esattamente l'inviluppo convesso delle soluzioni intere del problema (si veda il paragrafo 5.1.3). Il fatto che il problema ammetta algoritmi risolutivi polinomiali corrisponde al fatto che esista un separatore polinomiale per le disuguaglianze (1.6). Si consideri infatti una soluzione x^* (possibilmente frazionaria): vogliamo verificare se esiste $S \subseteq V$ a cui corrisponde un vincolo è violato, ossia tale che risulti

$$\textstyle\sum_{i \in S, j \notin S} x^*_{ij} < 1 \;\; .$$

Per determinarlo, si può determinare il sottoinsieme S a cui corrisponde il *minimo* di quei valori; per questo basta notare che il valore corrisponde alla *capacità del taglio* $(S, V \setminus S)$ in cui x^*_{ij} è vista come la capacità del lato $\{i, j\}$. Infatti, supponiamo di determinare il taglio (V', V'') di capacità minima rispetto a quelle cpacità: se la capacità del taglio è minore di 1 allora abbiamo individuato una specifica disuguaglianza violata, mentre se è maggiore od uguale ad 1 allora non esiste nessuna disuguaglianza violata. Il problema del taglio di capacità minima può essere risolto in tempo polinomiale (si veda il paragrafo 3.3), e quindi esiste un separatore polinomiale per le diseguaglianze; infatti (MST) ammette algoritmi polinomiali (si veda il paragrafo B.4). Si noti che in pratica gli algoritmi combinatori per (MST) risultano enormemente più efficienti rispetto alla soluzione di una sequenza di problemi di *PL* e di separazione; questo esempio serve principalmente a sottolineare la fondamentale relazione tra i due concetti.

Esercizio 4.3 *I problemi di taglio di capacità minima per i quali abbiamo discusso algoritmi risolutivi efficienti sono su grafi orientati e con una precisa scelta dei nodi s e t; si discuta come adattarli alla costruzione di un separatore per (MST).*

Per riassumere, l'esistenza di una formulazione di *PL* equivalente a ciascun problema di *PLI* non lo rende, per ciò stesso, facile: la formulazione esiste, ma non abbiamo nessun modo efficiente per generarla, e neanche per generarne una parte sufficiente a caratterizzare la soluzione ottima (si noti che basterebbe una base ottima). Questo però suggerisce alcuni interessanti approcci per la *PLI*, detti *metodi poliedrali*, che possono risutare molto utili per la soluzione di ampie classi di problemi. Questi metodi verranno descritti pì in dettaglio nel seguito, ma la loro idea fondamentale può essere descritta facilmente: anche se il problema di separazione per $\tilde{\mathcal{F}}$ è "difficile", accade sovente che molte delle disequazioni importanti che lo caratterizzano (faccette) possano essere determinate efficientemente. In altri termini, è possibile risolvere un rilassamento del problema originale "più debole" di $(\tilde{RC})$, ma che comunque comprende molta più "informazione" sul problema originario rispetto ad una formulazione *PLI* "naturale", quali quelle che abbiamo visto nel Capitolo 1. Questo rilassamento fornisce quindi valutazioni superiori su $z(PLI)$ più accurate di quelle fornite da una formulazione più semplice, e questo può rendere possibile risolvere (PLI) in modo più efficiente.

4.3 Dimostrazioni di ottimalità

In generale, il processo di soluzione di un qualsiasi problema di ottimizzazione, ad esempio

$$(P) \qquad \max \{\, c(x) \;:\; x \in X \,\} \quad ,$$

può essere considerato come composto di due parti distinte:

- produrre una soluzione ottima x^*;
- produrre una valutazione superiore su $z(P)$ che *dimostri* l'ottimalità di x^*, ossia un valore $\bar{z}$ per il quale sia garantito che $z(P) \leq \bar{z}$ ma per il quale risulti anche $c(x^*) = \bar{z}$, in modo tale che

$$\bar{z} = c(x^*) \leq z(P) \leq \bar{z} \ .$$

In molti algoritmi visti nei capitoli precedenti le valutazioni superiori (o inferiori) erano esplicitamente descritte. In altri le dimostrazioni di ottimalità non facevano uso esplicito di valutazioni superiori (o inferiori), ma tali valutazioni potrebbero essere costruite e mostrate.

Per valutazioni superiori disponibili in modo esplicito, si considerino i problemi dell'albero dei cammini minimi, del Flusso Massimo e di *PL*. Nel primo caso, è facile dimostrare, usando il Teorema 3.3 e la definizione della funzione obiettivo, che una valutazione inferiore sul costo dell'albero ottimo è data dalla somma delle etichette associate ai nodi per qualsiasi vettore di etichette che rispetta le condizioni di Bellman: quando l'algoritmo termina, la valutazione inferiore ha costo pari a quello dell'albero individuato. Nel secondo caso, una valutazione superiore sul valore del massimo flusso è fornita dalla capacità di un qualsiasi taglio (Teorema 3.5): a terminazione, l'algoritmo ha costruito un taglio con capacità esattamente pari al valore del flusso determinato, che risulta quandi massimo. Nel terzo caso, una valutazione superiore al valore della funzione obiettivo di qualsiasi soluzione primale ammissibile è data dal valore della funzione obiettivo di qualsiasi soluzione duale ammissibile (Teorema 2.9): a terminazione, gli algoritmi hanno costruito una coppia di soluzioni ammissibili per il primale ed il duale con lo stesso valore di funzione obiettivo, e quindi hanno una dimostrazione esplicita di ottimalità per la soluzione primale (e per quella duale). Nel caso di (MCF), valutazioni sul valore ottimo della funzione obiettivo possono essere ricavate utilizzando argomenti duali, anche se non ne abbiamo fatto esplicitamente uso; per (MST) questo è mostrato nel paragrafo 5.1.3. Ad una più attenta ispezione, tutte queste valutazioni superiori (o inferiori) si rivelano derivare dalla teoria della dualità per la Programmazione Lineare, che è quindi uno dei metodi più potenti e generali per dimostrare l'ottimalità di un problema. Nel Capitolo 6 vedremo comunque una tecnica che la generalizza.

Per i problemi $\mathcal{NP}$-ardui di *OC*, non si hanno in generale tecniche per determinare efficacemente valutazioni esatte del valore della funzione obiettivo del problema. Un diverso modo di vedere la cosa è il seguente: per quasi tutti i problemi di *OC*, la difficoltà del problema consiste fondamentalmente nella determinazione del *valore ottimo* della funzione obiettivo. Infatti, moltissimi problemi di *OC* godono di una proprietà chiamata *auto-riducibilità*, dalla quale segue che se esistesse un algoritmo efficiente per determinare il valore ottimo della funzione obiettivo per il problema, allora esisterebbe un algoritmo efficiente per determinare una soluzione ottima del problema.

Senza formalizzare completamente questi concetti, ne illustriamo l'uso con un esempio. Si consideri ancora il (CMST), e si supponga di avere un algoritmo $\mathcal{A}$ in grado di calcolare, data una qualunque istanza I di (CMST), il valore ottimo della funzione obiettivo $z(I)$, senza però fornire una corrispondente soluzione ottima. Vediamo come sia possibile utilizzare $\mathcal{A}$ per costruire una soluzione ottima del prolema.

Per prima cosa, si utilizza $\mathcal{A}$ per calcolare $z^* = z(I)$. A questo punto, si seleziona un qualunque lato tra due nodi i e j (entrambe diversi dalla radice) e si verifica, utilizzando l'algoritmo $\mathcal{A}$, se $\{i, j\}$ fa oppure no parte di una soluzione ottima del problema. Per questo, è sufficiente construire una nuova istanza I' in cui il lato $\{i, j\}$ è cancellato, e calcolare $z(I')$ utilizzando $\mathcal{A}$. Se $z(I') = z^*$, allora esiste effettivamente una soluzione ottima di I in cui $\{i, j\}$ non è presente: è quindi possibile cancellare definitivamente $\{i, j\}$. Se invece $z(I') > z^*$, allora $\{i, j\}$ fa parte di qualsiasi soluzione ottima di I. È facile costruire una nuova istanza I'' di (CMST) corrispondente al fissare $\{i, j\}$ come parte della soluzione ottima: per questo, è sufficiente "accorpare" i e j in un nuovo nodo h, con $b_h = b_i + b_j$, fissando il costo dei lati $\{h, k\}$ per $k \neq i$ e $k \neq j$ al minimo tra c_{ik} e c_{jk}. È immediato verificare che ogni soluzione ottima di (CMST) su questa nuova istanza corrisponde ad una soluzione ottima per l'istanza originale se il nodo h viene nuovamente "espanso" nella coppia di nodi i e j collegati dal lato $\{i, j\}$. In entrambo i casi abbiamo ricondotto il problema ad uno "più piccolo", con un lato oppure un nodo in meno: iterando il procedimento è possibile costruire una soluzione per l'istanza originale I con al più $O(n^2)$ chiamate all'algoritmo $\mathcal{A}$ (si noti che il problema in cui esistono solo lati uscenti dalla radice è banale). Di conseguenza, se $\mathcal{A}$ fosse polinomiale sarebbe possibile costruire una soluzione ottima di (CMST) in tempo polinomiale.

Questa osservazione giustifica anche dal punto di vista teorico l'interesse per tecniche in grado di determinare valutazioni superiori (inferiori nel caso di un problema di minimo) sul valore ottimo della funzione obiettivo di un dato problema di *OC*. Per usare la terminologia della teoria della complessità computazionale, possiamo considerare il valore ottimo della funzione obiettivo di un problema di *OC* come un *certificato* di ottimalità. I problemi di *OC* "facili" sono quelli per i quali sono disponibili tecniche efficienti per costruire un tale certificato, mentre quelli "difficili" sono quelli per cui non sono note tecniche in grado di svolgere efficientemente questo compito.

Le tecniche per determinare valutazioni superiori sul valore ottimo della funzione obiettivo di un dato problema di *OC* sono comunque molto importanti anche nella pratica. In effetti, per molte classi di problemi di *OC* esiste *in pratica* una consistente differenza tra *produrre* una soluzione ε-ottima e *certificare* la ε-ottimalità di una soluzione data. Ad esempio, in molti algoritmi enumerativi per problemi "difficli" (si veda il Capitolo 7) capita sovente che l'algoritmo determini la soluzione ottima in tempo relativamente breve, ma sia poi ancora necessario un grandissimo sforzo computazionale per *dimostrare* che tale soluzione è effettivamente ottima. In altri termini, la difficoltà del problema risiede non tanto nel costruire una soluzione ottima, quando nel verificarne l'ottimalità, ossia nel determinare il valore ottimo della funzione obiettivo. Alle tecniche utili a determinare questo valore, o una sua approssimazione accurata, è dedicata una parte rilevante della ricerca attuale volta a sviluppare algoritmi "efficienti" per problemi di *OC*.

Riferimenti Bibliografici

F. Maffioli "**Elementi di Programmazione Matematica**", *Casa Editrice Ambrosiana*, 2000.

A. Sassano "**Modelli e Algoritmi della Ricerca Operativa**", *FrancoAngeli*, 1999.

L. Wolsey "**Integer Programming**", *Wiley-Interscience*, 1998.