

RILASSAMENTI: KNAPSACK, PACKING, MULTICOVERING

Irace Gianni

Indice

Sommario.....	3
Abstract.....	3
1. Introduzione.....	4
2. Tecniche di rilassamento.....	5
2.1 Definizione.....	5
Teorema 2.1.....	6
Teorema 2.2.....	6
2.2 Rilassamento lagrangiano e suo duale, rilassamento surrogato e suo duale.....	7
Teorema 2.3 (delle dominanze).....	8
Teorema 2.4 (condizioni di ottimo).....	9
2.3 Decomposizione lagrangiana e suo duale.....	10
Teorema 2.5 (delle dominanze).....	11
Teorema 2.6 (condizioni di ottimo).....	11
2.4 Duale debole (Weak dual).....	12
Teorema 2.7.....	12
2.5 Grafo delle dominanze.....	12
Grafo delle dominanze.....	13
3. Il problema del knapsack.....	15
3.1 Formulazione matematica.....	15
3.2 Rilassamento continuo, rilassamento lagrangiano e duale debole.....	15
Teorema 3.1.....	16
3.3 Un algoritmo greedy euristico per il duale debole del knapsack.....	16
Teorema 3.2.....	16
3.4 Decomposizione lagrangiana: una successione finita di greedy euristiche su duali deboli.....	17
Teorema 3.3.....	17
3.5 Un algoritmo greedy euristico per il knapsack.....	18
Teorema 3.4.....	19
4. Il problema del set packing.....	20
4.1 Formulazione matematica.....	20
4.2 Il grafo degli implicanti.....	20
4.3 Un algoritmo greedy euristico per il set packing.....	22
5. Il problema del multicovering.....	26
5.1 Formulazione matematica.....	26
5.2 Duale debole, rilassamento surrogato e rilassamento lagrangiano.....	27
Teorema 5.1.....	28
Corollario 5.1.....	28
5.3 Un greedy euristico per il duale debole del multicovering.....	29
Teorema 5.2.....	29
5.4 Decomposizione lagrangiana: una successione finita di greedy euristiche su duali deboli.....	30
Teorema 5.3.....	30
5.5 Un algoritmo greedy euristico per il multicovering.....	32
5.6 Un algoritmo branch & bound per il multicovering.....	35
6. Risultati sperimentali.....	36
Tabella dei risultati:.....	37
Bibliografia.....	38
Riferimento: Irace dott. Gianni (393-0679616).....	40

Sommario.

In questo lavoro dopo aver introdotto i rilassamenti, si effettua un confronto fra alcuni di essi rispetto ai problemi di programmazione lineare intera. Presentiamo tre algoritmi greedy euristiche per il "Knapsack Problem"=(KP), per il "Set Packing Problem"=(SPP) e per il "Multicovering Problem"=(MCP). Presentiamo infine un algoritmo branch & bound per il (MCP) basato su euristica primale, decomposizione lagrangiana, duale debole e rilassamento surrogato. Il "Set Covering Problem"=(SCP) è un caso particolare di (MCP). Si conclude con alcuni risultati sperimentali su problemi generati casualmente.

PAROLE CHIAVI: decomposizione lagrangiana, duale debole, rilassamento surrogato.

Abstract.

This paper starts with an illustration of relaxation techniques, the comparison between them is applied for the resolution of linear integer programming problem. We present a greedy heuristics algorithms for the "Knapsack Problem"=(KP), for the "Set Packing Problem"=(SPP) and for the "Multicovering Problem"=(MCP). After we present a branch & bound algorithm for the (MCP) based on primal heuristic, lagrangean decomposition, weak dual and surrogate relaxation. The "Set Covering Problem"=(SCP) is a particular case of (MCP). Following some computational results for randomly generated problems.

1. Introduzione.

Molti problemi di ottimizzazione combinatoria, ed in particolare la programmazione lineare intera, rientrano nella classe dei problemi np-hard. Esistono tecniche dette "Rilassamenti" che permettono di affrontare i problemi più agevolmente ma comunque risolti con algoritmi non polinomiali nella lunghezza dell'input.

Gli algoritmi branch & bound utilizzano i rilassamenti per effettuare dei "tagli" sull'albero di ricerca, e, più questi si avvicinano all'ottimo meno nodi dell'albero vengono esaminati. E' logico, quindi, andare alla ricerca di rilassamenti che siano più efficaci e dominanti rispetto ad altri.

Il lavoro fatto in questo articolo produce un algoritmo greedy euristico per il problema del "knapsack" in $O(n^2)$, che dà un risultato migliore del suo rilassamento continuo la cui complessità è $O(n \log n)$, con una occupazione della memoria equivalente.

Successivamente viene dato un greedy euristico per il "set packing" grazie ad una nuova definizione: il grafo degli implicanti. Tale greedy verrà utilizzato per fornire, infine, un greedy per il "multicovering problem" basata sulla stessa idea di quella del knapsack.

La base da cui si parte è il rilassamento "decomposizione lagrangiana" sul quale applicheremo un trucco: la clonazione dei vincoli.

2. Tecniche di rilassamento.

2.1 Definizione.

Un problema di ottimizzazione può essere espresso in forma matematica:

$$(P) = (F, c; \min) = \min \{c(\mathbf{x}) / \mathbf{x} \in F\}$$

dove F è chiamata regione ammissibile mentre c è una funzione definita come $c : F \rightarrow \mathbb{R}$ ed è chiamata funzione obiettivo.

Quando a P assegniamo dei valori ai termini noti diremo che P è una istanza appartenente alla classe delle istanze Π .

Sia $(P') = (F', c'; \min) = \min \{c'(\mathbf{x}) / \mathbf{x} \in F'\}$ una istanza della classe Π'

e $\sigma : \Pi \rightarrow \Pi'$ una applicazione con $\sigma(P) = P'$

la coppia $[\Pi', \sigma]$ è uno schema di rilassamento per Π

se $\forall P \in \Pi$ e $P' = \sigma(P)$ si ha:

- 1) $F \subseteq F'$
- 2) $c'(\mathbf{x}) \leq c(\mathbf{x}) \quad \forall \mathbf{x} \in F$

e diremo che P' è un rilassamento di P .

notazione:

Se P è un problema di ottimizzazione allora $z(P)$ indica l'ottimo di P .

Teorema 2.1

Se P' è un rilassamento di P allora

$$z(P') = \min\{c'(x) / x \in F'\} \leq \min\{c(x) / x \in F\} = z(P) .$$

Teorema 2.2

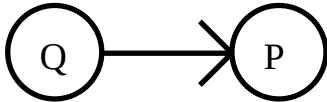
Se P' è un rilassamento di P allora

- x^* ottima per P'
- $x^* \in F \quad \Rightarrow \quad x^*$ è ottima per P .
- $c(x^*) = c'(x^*)$

Principali tecniche di rilassamento:

- eliminazione di vincoli
- rilassamento continuo
- rilassamento lagrangiano e duale lagrangiano
- rilassamento surrogato e duale surrogato
- decomposizione lagrangiana e duale decomposizione lagrangiana
- duale debole (weak dual)
- convessificazione funzione obiettivo
- linearizzazione
- decomposizione di benders

notazione: (per problemi di minimo)

$$z(Q) \leq z(P) \quad \Leftrightarrow \quad P \text{ domina } Q \quad \Leftrightarrow \quad \text{Diagramma}$$


2.2 Rilassamento lagrangiano e suo duale, rilassamento surrogato e suo duale.

Ipotesi sul problema di ottimizzazione: $(P) = (F, c; \min)$

$F = \{ \mathbf{x} \in S : \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \}$ regione ammissibile

$c : F \rightarrow \mathbb{R}$ funzione obiettivo

$\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ applicazione che definisce i vincoli

$S \subseteq \mathbb{R}^n$ spazio delle variabili

$$(P) \equiv \begin{cases} \min c(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{x} \in S \end{cases}$$

Rilassamento lagrangiano di (P): $\forall \boldsymbol{\lambda} \in \mathbb{R}_+^m$

equivale a premoltiplicare i vincoli per un vettore $\boldsymbol{\lambda}$ non negativo e sommare il risultato alla funzione obiettivo

$$\text{RLP}(\boldsymbol{\lambda}) \equiv \begin{cases} \min c(\mathbf{x}) + \boldsymbol{\lambda} \mathbf{g}(\mathbf{x}) \\ \mathbf{x} \in S \end{cases}$$

Duale lagrangiano di (P):

equivale a calcolare il massimo lagrangiano al variare di $\boldsymbol{\lambda} \in \mathbb{R}_+^m$

$$(\text{DLP}) \equiv \max_{\boldsymbol{\lambda} \in \mathbb{R}_+^m} \{ z(\text{RLP}(\boldsymbol{\lambda})) \}$$

Rilassamento surrogato di (P): $\forall \boldsymbol{\lambda} \in \mathbb{R}_+^m$

equivale a premoltiplicare i vincoli per un vettore $\boldsymbol{\lambda}$ non negativo in modo da ottenere uno ed un solo vincolo

$$\text{RSP}(\boldsymbol{\lambda}) \equiv \begin{cases} \min c(\mathbf{x}) \\ \boldsymbol{\lambda} \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{x} \in S \end{cases}$$

Duale surrogato di (P):

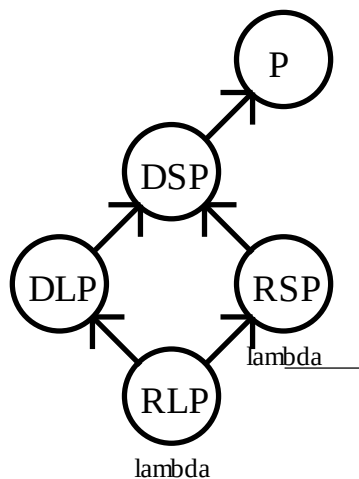
equivale a calcolare il massimo rilassamento surrogato al variare di $\boldsymbol{\lambda} \in \mathbb{R}_+^m$

$$(\text{DSP}) \equiv \max_{\boldsymbol{\lambda} \in \mathbb{R}_+^m} \{ z(\text{RSP}(\boldsymbol{\lambda})) \}$$

Teorema 2.3 (delle dominanze)

$$1) \quad z(\text{RLP}(\lambda)) \leq z(\text{RSP}(\lambda)) \leq z(P) \quad \forall \lambda \in \mathcal{R}_+^m$$

$$2) \quad z(\text{DLP}) \leq z(\text{DSP}) \leq z(P)$$



il teorema 2.3 afferma che:

Il rilassamento lagrangiano, fissato λ ,
è dominato dal rilassamento surrogato.

Il duale lagrangiano è dominato dal duale
surrogato il quale è dominato da (P).

Dimostrazione

Cominciamo con il dimostrare il punto 1) del Teorema 2.3

$$\begin{array}{ccc} \left\{ \min_{\mathbf{x} \in S} c(\mathbf{x}) + \lambda \mathbf{g}(\mathbf{x}) \right\} & \stackrel{\text{(III)}}{\leq} & \left\{ \min_{\mathbf{x} \in S} c(\mathbf{x}) + \lambda \mathbf{g}(\mathbf{x}) \right\} \\ & & \lambda \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ & & \mathbf{x} \in S \\ \parallel & & \parallel \\ z(\text{RLP}(\lambda)) & \leq & z(\text{RSP}(\lambda)) \end{array} \stackrel{\text{(II)}}{\leq} \left\{ \min_{\mathbf{x} \in S} c(\mathbf{x}) \right\} \stackrel{\text{(I)}}{\leq} \left\{ \min_{\mathbf{x} \in S} c(\mathbf{x}) \right\} \stackrel{\text{(I)}}{\leq} \left\{ \min_{\mathbf{x} \in S} c(\mathbf{x}) \right\}$$

la disuguaglianza (I) vale per il rilassamento surrogato, la disuguaglianza (II) vale perchè si somma alla funzione obiettivo una quantità negativa, la disuguaglianza (III) vale per il rilassamento eliminazione di vincoli;

passando al $\sup$ su $\lambda \in \mathcal{R}_+^m$ si dimostra anche il punto 2).

Teorema 2.4 (condizioni di ottimo)

Sia $\lambda \in \mathfrak{R}_+^m$

(1) $\mathbf{x}^*$ ottimo per (RLP)

(2) $\mathbf{g}(\mathbf{x}^*) \leq \mathbf{0}$

(3) $\lambda \mathbf{g}(\mathbf{x}^*) = \mathbf{0}$

(4) $\mathbf{x}^*$ ottimo per (RSP)

valgono le seguenti proposizioni:

- (1),(2),(3) - $\Rightarrow$ $\mathbf{x}^*$ ottimo per (P)
- (1),(2) - - $\Rightarrow$ $\mathbf{x}^*$ soluzione approssimata per (P) con un
"errore assoluto" $EA = | \lambda \mathbf{g}(\mathbf{x}^*) |$
- (1), - (3) - $\Rightarrow$ (4)
- - (2), - (4) $\Rightarrow$ $\mathbf{x}^*$ ottimo per (P).

2.3 Decomposizione lagrangiana e suo duale.

Ipotesi sul problema di ottimizzazione: $(P)=(F,c;\min)$

$F=\{ \mathbf{x} \in S : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{h}(\mathbf{x}) \leq \mathbf{0} \}$ regione ammissibile

$c : F \rightarrow \mathbb{R}, c(\mathbf{x})=\mathbf{c}\mathbf{x}$ lineare funzione obiettivo

$\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ applicazione che definisce i vincoli

$\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^k$ applicazione che definisce i vincoli

$S \subseteq \mathbb{R}^n$ spazio delle variabili

La decomposizione lagrangiana si ottiene aggiungendo dei vincoli a (P) , rilassando tali vincoli con lagrange ed effettuando un cambio variabili:

$$\begin{aligned}
 (P) = \left\{ \begin{array}{l} \min \mathbf{c}\mathbf{x} \\ \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{h}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{x} \in S \end{array} \right\} &= \left\{ \begin{array}{l} \min \mathbf{c}\mathbf{x}^{(1)} \\ \mathbf{g}(\mathbf{x}^{(1)}) \leq \mathbf{0} \\ \mathbf{x}^{(1)} = \mathbf{x}^{(2)} \\ \mathbf{h}(\mathbf{x}^{(2)}) \leq \mathbf{0} \\ \mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in S \end{array} \right\} \geq \left\{ \begin{array}{l} \min \mathbf{c}\mathbf{x}^{(1)} + \mu(\mathbf{x}^{(1)} - \mathbf{x}^{(2)}) \\ \mathbf{g}(\mathbf{x}^{(1)}) \leq \mathbf{0} \\ \mathbf{h}(\mathbf{x}^{(2)}) \leq \mathbf{0} \\ \mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in S \end{array} \right\} = \\
 &= \left\{ \begin{array}{l} \min \mathbf{c}^{(1)}\mathbf{x} \\ \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{x} \in S \end{array} \right\} + \left\{ \begin{array}{l} \min \mathbf{c}^{(2)}\mathbf{x} \\ \mathbf{h}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{x} \in S \end{array} \right\} \text{avendo posto } \begin{array}{l} \mathbf{c}^{(1)} = \mathbf{c} + \mu \\ \mathbf{c}^{(2)} = -\mu \end{array}
 \end{aligned}$$

notiamo che applicare il rilassamento lagrangiano ai vincoli di uguaglianza significa che il vettore delle variabili $\mu \in \mathbb{R}^n$ non è vincolato ad avere tutti i termini positivi.

Decomposizione lagrangiana di (P) : $\left[\forall \mathbf{c}^{(1)} + \mathbf{c}^{(2)} = \mathbf{c} \right]$

$$\text{DCLP}(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}) = \begin{cases} \min \mathbf{c}^{(1)} \mathbf{x} \\ \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{x} \in S \end{cases} + \begin{cases} \min \mathbf{c}^{(2)} \mathbf{x} \\ \mathbf{h}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{x} \in S \end{cases} = P^{(g)}(\mathbf{c}^{(1)}) + P^{(h)}(\mathbf{c}^{(2)})$$

la determinazione della partizione dei costi è essenziale per trovare un buon rilassamento e in effetti il suo duale va alla ricerca della partizione ottima.

Notiamo anche il perchè si è ipotizzato due tipi diversi di vincoli, magari un tipo difficile e l'altro facile, in modo da poter spezzare in due il problema.

Duale decomposizione lagrangiana di(P) :

$$(\text{DDCLP}) = \underset{\mathbf{c}^{(1)} + \mathbf{c}^{(2)} = \mathbf{c}}{\text{MAX}} \left\{ z(\text{DCLP}(\mathbf{c}^{(1)}, \mathbf{c}^{(2)})) \right\}$$

Teorema 2.5 (delle dominanze)

$$z(\text{DCLP}(\mathbf{c}^{(1)}, \mathbf{c}^{(2)})) \leq z(\text{DDCLP}) \leq z(P)$$



Teorema 2.6 (condizioni di ottimo)

Sia $\mathbf{c}^{(1)} + \mathbf{c}^{(2)} = \mathbf{c}$

(1) $\mathbf{x}^{(1)}$ ottimo per $P^{(g)}(\mathbf{c}^{(1)})$

(2) $\mathbf{x}^{(2)}$ ottimo per $P^{(h)}(\mathbf{c}^{(2)})$

(3) $\mathbf{x}^{(1)} = \mathbf{x}^{(2)}$

(1),(2),(3) $\Rightarrow \mathbf{x}^* = \mathbf{x}^{(1)} = \mathbf{x}^{(2)}$ ottimo per (P).

2.4 Duale debole (Weak dual).

$(P) = (F, c; \min) = \min \{c(\mathbf{x}) / \mathbf{x} \in F\}$ problema di minimo

$(D) = (Q, b; \max) = \max \{b(\mathbf{y}) / \mathbf{y} \in Q\}$ problema di massimo

(D) è un Duale debole di (P) se esiste un problema (RP) di minimo ed un problema (RD) di massimo tale che:

- (RP) è un rilassamento di (P) oppure è (P) stesso
- (RD) è un rilassamento di (D) oppure è (D) stesso
- $z(RD) = z(RP)$

per cui vale valgono le seguenti dominanze:



Teorema 2.7

$\forall \mathbf{y}$ ammissibile per $(D) \Rightarrow b(\mathbf{y}) \leq z(P)$.

Dimostrazione

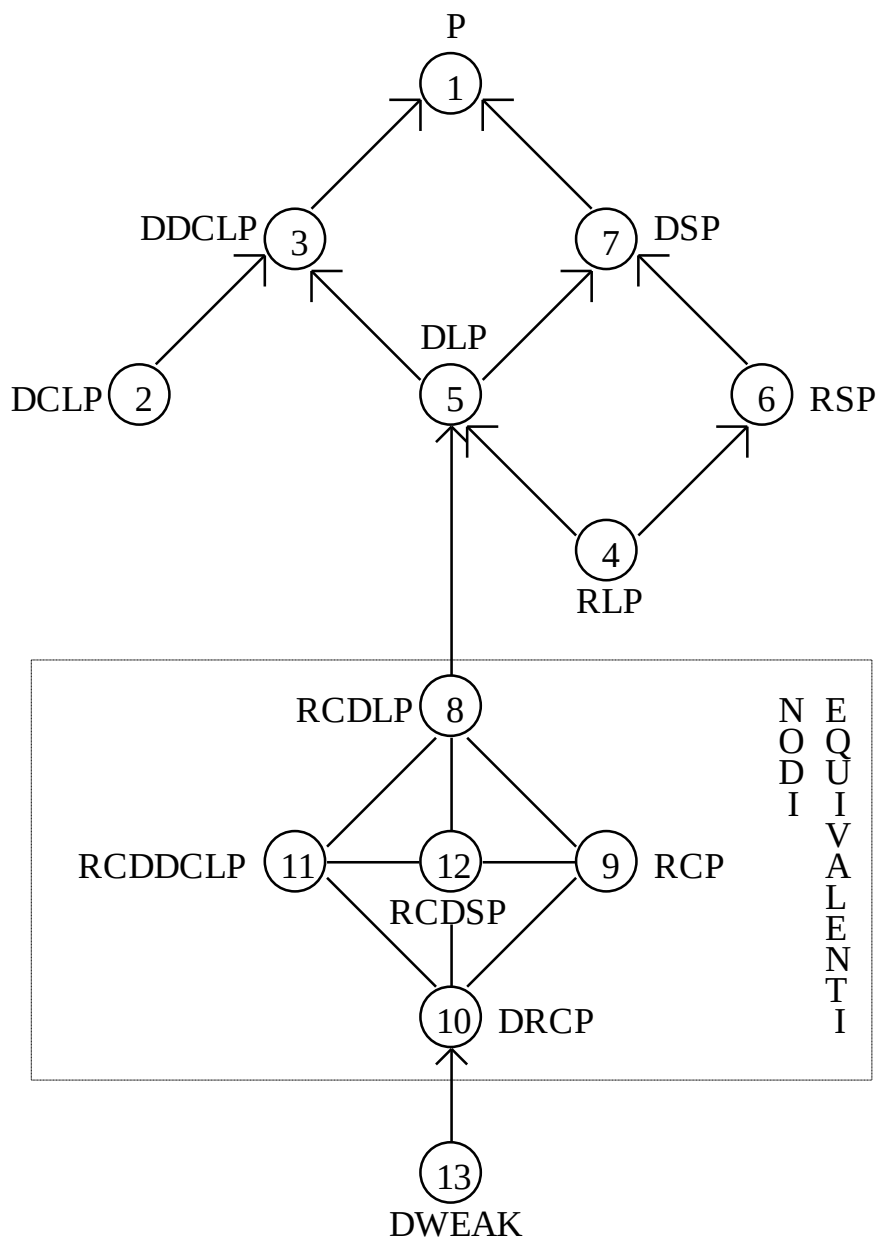
se $\mathbf{y}$ ammissibile per (D) allora $b(\mathbf{y}) \leq z(D) \leq z(RD) = z(RP) \leq z(P)$.

In generale se (P) è un problema di minimo, allora un suo rilassamento è anch'esso un problema di minimo, e se vogliamo usarlo come valutazione inferiore di (P) allora siamo vincolati a trovare l'ottimo di tale rilassamento senza commettere errori, invece il duale debole è un problema di massimo per cui ogni soluzione ammissibile del duale debole fornisce una valutazione inferiore di (P) .

In questo modo non siamo più vincolati a trovare l'ottimo del rilassamento e possiamo ad esempio utilizzare dei greedy.

2.5 Grafo delle dominanze.

Il grafo delle dominanze mette in risalto i problemi che dominano e i problemi che sono dominati, cioè le relazioni che esistono fra il problema e i suoi rilassamenti.



Grafo delle dominanze.

1 :	(P)	problema P
2 :	(DCLP)	decomposizione lagrangiana di P
3 :	(DDCLP)	duale decomposizione lagrangiana di P
4 :	(RLP)	rilassamento lagrangiano di P
5 :	(DLP)	duale lagrangiano di P
6 :	(RSP)	rilassamento surrogato di P
7 :	(DSP)	duale surrogato di P
8 :	(RCDLP)	rilassamento continuo-duale lagrangiano di P
9 :	(RCP)	rilassamento continuo di P
10 :	(DRCP)	duale-rilassamento continuo di P
11 :	(RCDDCLP)	rilassamento continuo-duale decomposizione lagrangiana di P
12 :	(RCDSP)	rilassamento continuo-duale surrogato di P
13 :	(DWEAK)	duale debole di P

si osserva che:

- i rilassamenti continui si ottengono rilassando i vincoli di interezza sulle variabili in oggetto.
- i duali dei rilassamenti continui sono i duali reali nel senso proprio della programmazione lineare a variabili reali.

Ad esempio risolvere il rilassamento continuo del duale lagrangiano(nodo-8) oppure il rilassamento continuo del duale surrogato(nodo-12) oppure il rilassamento continuo del duale decomposizione lagrangiano(nodo-11) è lo stesso che risolvere il rilassamento continuo di P (nodo-9) oppure il duale del rilassamento continuo di P (nodo-10). Si può inoltre osservare che i duali deboli di P sono dominati da tutti.

Dal *grafo delle dominanze* si trae la conclusione che:

fra tutti i rilassamenti di (P) il duale surrogato di P (nodo-7) ed il duale decomposizione lagrangiana di P (nodo-3) sono i "dominanti" nel senso che i loro ottimi si avvicinano di più degli altri all'ottimo del problema (P) .

3. Il problema del knapsack.

3.1 Formulazione matematica.

Il problema del knapsack è il seguente: ci sono n oggetti ognuno dei quali ha un costo ed un peso, si vuole scegliere una combinazione di questi oggetti in modo tale che il peso totale sia maggiore o uguale di un certo valore b e il costo totale sia minimo.

$$(KP) = \begin{cases} \min \mathbf{cx} & (1) \quad \mathbf{a}, \mathbf{c} \in \mathfrak{R}_+^n, b \in \mathfrak{R}_+ \\ \mathbf{ax} \geq b & (2) \quad \mathbf{a}_j \leq b, \quad j=1,2,\dots,n \\ \mathbf{x} \in \{0,1\}^n & (3) \quad \sum_j \mathbf{a}_j \geq b \end{cases}$$

l'ipotesi (2) non è restrittivo infatti modificando $\mathbf{a}_j \leftarrow \min\{\mathbf{a}_j, b\}$, $j=1,\dots,n$ si ottiene un problema equivalente a (KP).

3.2 Rilassamento continuo, rilassamento lagrangiano e duale debole.

rilassamento continuo di (KP):

si ottiene rilassando i vincoli di interezza $\mathbf{x} \in \{0,1\}^n \rightarrow \mathbf{x} \in [0,1]^n$

$$(RCKP) = \begin{cases} \min \mathbf{cx} \\ \mathbf{ax} \geq b \\ \mathbf{x} \in [0,1]^n \end{cases}$$

rilassamento lagrangiano di (KP): $\forall \lambda \in \mathfrak{R}_+$

si ottiene moltiplicando λ per $\mathbf{g}(\mathbf{x}) = (b - \mathbf{ax}) \leq 0$ e sommandolo a $\mathbf{cx}$

$$(RLKP)(\lambda) = \begin{cases} \min \mathbf{cx} + \lambda (b - \mathbf{ax}) \\ \mathbf{x} \in \{0,1\}^n \end{cases} = \begin{cases} \min \lambda b + (\mathbf{c} - \lambda \mathbf{a})\mathbf{x} \\ \mathbf{x} \in \{0,1\}^n \end{cases}$$

duale debole di (KP):

questo duale debole si ottiene dal rilassamento continuo di (KP) in cui vengono eliminati i vincoli $\mathbf{x} \leq \mathbf{e}$, dove $\mathbf{e} = [1, \dots, 1]$, e calcolando il duale

$$(DWKP) = \begin{cases} \max \mathbf{yb} \\ \mathbf{ya} \leq \mathbf{c} \\ \mathbf{y} \in \mathfrak{R}_+ \end{cases} = \begin{cases} \max \mathbf{yb} \\ \mathbf{ya}_j \leq \mathbf{c}_j, \quad j=1,2,\dots,n \\ \mathbf{y} \in \mathfrak{R}_+ \end{cases}$$

Teorema 3.1

$\forall y$ ammissibile per (DWKP) $\Rightarrow z(\text{RLKP}(y)) = yb$.

Dimostrazione

$z(\text{RLKP}(y)) = yb + \min\{ (c-ya)x : x \in \{0,1\}^n \} = yb$, poichè $(c-ya) \geq 0$, e $0 \in \{0,1\}^n$, per cui l'ottimo si ottiene per $x=0$.

Il Teorema 3.1 afferma che presa una qualsiasi soluzione ammissibile y per il duale debole del knapsack, se diamo y in "input" al rilassamento lagrangiano del knapsack allora l'ottimo del lagrangiano è uguale al valore della funzione obiettivo del duale debole yb .

3.3 Un algoritmo greedy euristico per il duale debole del knapsack.

Il seguente *greedy euristico* restituisce:

- la soluzione ottima y^* per il duale debole del knapsack (DWKP);
- un lower bound z^* per il knapsack (KP);

procedure greedy_dwkp (a,b,c,y^*,z^*); { complessità: $O(n)$ }

begin

| $y^* := c_k/a_k = \min \{ c_j/a_j : j = 1,2,...,n \};$

| $z^* := y^* \cdot b$

end.

Teorema 3.2

L'algoritmo greedy_dwkp dà sempre la soluzione ottima: $y^*b = z(\text{DWKP})$.

Dimostrazione

Per assurdo se esistesse un $\bar{y}$ ammissibile per (DWKP) con $\bar{y}b > y^*b$ allora risulterebbe $\bar{y} > y^* = c_k/a_k$ e quindi $\bar{y} a_k > y^* a_k = c_k$ il che contraddice l'ipotesi di ammissibilità $\bar{y} a_k \leq c_k$.

3.4 Decomposizione lagrangiana: una successione finita di greedy euristiche su duali deboli.

Applichiamo il rilassamento *decomposizione lagrangiana* al problema del knapsack con un trucco: viene fatta una copia dei vincoli

$$KP(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \left\{ \begin{array}{l} \min \mathbf{c}\mathbf{x} \\ \mathbf{a}\mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \in \{0,1\}^n \end{array} \right\} = \left\{ \begin{array}{l} \min \mathbf{c}\mathbf{x} \\ \mathbf{a}\mathbf{x} \geq \mathbf{b} \\ \mathbf{a}\mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \in \{0,1\}^n \end{array} \right\} \geq \left\{ \begin{array}{l} \min \mathbf{c}^{(1)}\mathbf{x} \\ \mathbf{a}\mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \in \{0,1\}^n \end{array} \right\} + \left\{ \begin{array}{l} \min (\mathbf{c} - \mathbf{c}^{(1)})\mathbf{x} \\ \mathbf{a}\mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \in \{0,1\}^n \end{array} \right\}$$

$$KP(\mathbf{a}, \mathbf{b}, \mathbf{c}) \geq KP(\mathbf{a}, \mathbf{b}, \mathbf{c}^{(1)}) + KP(\mathbf{a}, \mathbf{b}, \mathbf{c} - \mathbf{c}^{(1)})$$

La decomposizione lagrangiana necessita di una partizione dei costi, noi useremo come decompositore dei costi il duale debole $DWKP(\mathbf{a}, \mathbf{b}, \mathbf{c})$.

Teorema 3.3

Sia $\mathbf{y}^{(1)}$ la soluzione ottima di $DWKP(\mathbf{a}, \mathbf{b}, \mathbf{c})$ data dal greedy_dwkp,

sia $\mathbf{c}^{(1)} = \mathbf{y}^{(1)} \cdot \mathbf{a}$ vale che:

- 1) $\Rightarrow \mathbf{c}^{(1)} \leq \mathbf{c}$
- 2) $\Rightarrow \mathbf{y}^{(1)}$ è soluzione ottima di $DWKP(\mathbf{a}, \mathbf{b}, \mathbf{c}^{(1)})$
- 3) $\Rightarrow KP(\mathbf{a}, \mathbf{b}, \mathbf{c}) \geq \mathbf{y}^{(1)} \cdot \mathbf{b} + KP(\mathbf{a}, \mathbf{b}, \mathbf{c} - \mathbf{c}^{(1)})$

Dimostrazione

- 1): $\mathbf{c}^{(1)} = \mathbf{y}^{(1)} \cdot \mathbf{a} \leq \mathbf{c}$ poichè $\mathbf{y}^{(1)}$ è ammissibile per $DWKP(\mathbf{a}, \mathbf{b}, \mathbf{c})$.
- 2): $z(DWKP(\mathbf{a}, \mathbf{b}, \mathbf{c}^{(1)})) = \max\{ \mathbf{y}\mathbf{b} : \mathbf{y}\mathbf{a} \leq \mathbf{y}^{(1)}\mathbf{a}, \mathbf{y} \in \mathcal{R}_+ \} = \mathbf{y}^{(1)}\mathbf{b}$.
- 3): segue dalla decomposizione lagrangiana e dal punto 2).

Il Teorema 3.3 suggerisce di iterare il procedimento ponendo $\mathbf{c} \leftarrow \mathbf{c} - \mathbf{c}^{(1)}$, previa modifica del problema in quanto possono sorgere dei costi $\mathbf{c}_j$ nulli.

Il knapsack $KP(\mathbf{a}, b, \mathbf{c})$ viene così modificato ad ogni iterazione:

- $\forall \mathbf{c}_k = 0$ essendo il problema di minimo si pone $\mathbf{x}_k = 1$, l'incremento di costo è nullo, in compenso diminuisce $b \leftarrow b - \mathbf{a}_k$.
- $\forall j: \mathbf{c}_j > 0 \quad \mathbf{a}_j \leftarrow \min\{\mathbf{a}_j, b\}$.

Il procedimento di iterazione termina quando $b=0$ oppure $\mathbf{c}=\mathbf{0}$.

Se la terminazione avviene per $\mathbf{c}=\mathbf{0}$ quando ancora risulta $b > 0$ allora vuol dire che il problema non ammette soluzioni.

Osservazione

Il numero delle iterazioni sopra descritte sono al più "n" infatti ad ogni iterazione almeno un elemento di $\mathbf{c}$ va a zero per come viene risolto il duale debole e quindi dopo al più "n" iterazioni il vettore $\mathbf{c}$ diventa nullo.

Riassumendo, se applichiamo la decomposizione lagrangiana al problema (KP) con k copie di $\mathbf{a}\mathbf{x} \geq b$ si ha:

$$KP(\mathbf{a}, b, \mathbf{c}) \geq \sum_{i=1}^k KP(\mathbf{a}, b, \mathbf{c}^{(i)}) \geq \sum_{i=1}^k DWKP(\mathbf{a}, b, \mathbf{c}^{(i)})$$

tale che $\sum_{i=1}^k \mathbf{c}^{(i)} = \mathbf{c}$, $\mathbf{c}^{(i)}$, k da determinare

previa modifica di $\mathbf{a}$, b ad ogni passo.

3.5 Un algoritmo greedy euristico per il knapsack.

Il seguente greedy euristico restituisce un:

- un peso $= y^*$ del vincolo $\mathbf{a}\mathbf{x} \geq b$
- un lower_bound $= z^*$ per (KP)
- una soluzione ammissibile $= \mathbf{x}^*$ per (KP).

```

procedure greedy_kp (a,b,c,x*,y*,z*);      { complessità:  $O(n^2)$  }
begin
|   x* := { j : cj = 0 };      y* := 0;      z* := 0;
|
|   repeat
|   |   for each j : (cj > 0) do aj ← min{aj,b};
|   |   greedy_dwkp(a,b,c,y,z);
|   |   y* := y* + y;      z* := z* + z;      c := c - ya;
|   |   for each j : (cj = 0) and (j ∉ x*) and (b > 0) do
|   |   |   begin
|   |   |   |   x* := x* ∪ {j};      b := b - aj
|   |   |   end
|   until (b = 0) or (c = 0);
|   if (b > 0) then "problema non ammissibile"
end.

```

Teorema 3.4

Sia z^* un output di $\text{greedy_kp}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{x}^*, y^*, z^*) \Rightarrow z^* \geq z(\text{RCKP})$

dove $z(\text{RCKP})$ è il rilassamento continuo del knapsack problem.

Dimostrazione

Il rilassamento continuo fornisce un ottimo in base ai "costi unitari crescenti $\mathbf{c}_j/\mathbf{a}_j$ ". Il greedy_kp ad ogni iterazione, modifica il problema mediante $\mathbf{a}_j \leftarrow \min\{\mathbf{a}_j, \mathbf{b}\}$, calcola il minimo costo unitario crescente y , modifica il problema mediante $\mathbf{c} \leftarrow \mathbf{c} - y\mathbf{a}$, $\mathbf{b} \leftarrow \mathbf{b} - \mathbf{a}_j$, tali modifiche permettono di calcolare gli effettivi costi unitari crescenti al passo successivo. Quindi fornisce un lower_bound per il (KP) superiore a quello fornito da (RCKP).

4. Il problema del set packing.

4.1 Formulazione matematica.

Il problema del set packing è formulato nel modo seguente

$$(SPP) = \begin{cases} \max \mathbf{b}\mathbf{y} & (1) \quad \mathbf{A} \in \mathbf{M}(n, m, \{0,1\}) \\ \mathbf{A}\mathbf{y} \leq \mathbf{e} & (2) \quad \mathbf{b} \in \mathbf{Z}_+^m \\ \mathbf{y} \in \{0,1\}^m & (3) \quad \mathbf{e} = (1,1,\dots,1) \in \mathbf{Z}^n \end{cases}$$

dove $\mathbf{M}(n,m,\{0,1\})$ è la classe delle matrici ad n righe, m colonne ed a elementi nulli o unitari.

Questo problema è di difficile risoluzione, e ci sono stati risultati positivi come ad esempio:

- se la matrice $\mathbf{A}$ è *totalmente unimodulare* allora il rilassamento continuo di (SPP) dà la soluzione ottima intera [3],[13],[14],[15];
- se tutte le sottomatrici quadrate non singolari di $\mathbf{A}$ sono *triangolari* allora esiste un algoritmo polinomiale che risolve (SPP) [4].

Nel seguito daremo una definizione nuova grazie alla quale saremo in grado di dare un algoritmo greedy euristico per (SPP).

4.2 Il grafo degli implicanti.

Sia $\mathbf{A} \in \mathbf{M}(n,m,\{0,1\})$

sia $[\mathbf{A}\mathbf{y} \leq \mathbf{e}]$ un sistema di n disequazioni ad m incognite con $\mathbf{y} \in \{0,1\}^m$

si definisce *grafo degli implicanti* di $[\mathbf{A}\mathbf{y} \leq \mathbf{e}]$ il grafo $\mathbf{G}=(\mathbf{N},\mathbf{E})$ dove:

- $\mathbf{N}=\{1,2,\dots,m\}$ =insieme dei nodi associati a $y_1, y_2, \dots, y_m$
- $\mathbf{E}=\{(i,j) : i,j \in \mathbf{N}, [\mathbf{A}\mathbf{y} \leq \mathbf{e}] \Rightarrow [y_i=1 \Rightarrow y_j=0]\}$ = insieme degli archi

in altri termini il grafo $\mathbf{G}$ contiene l'arco (i,j) se la formula $[y_i=1 \Rightarrow y_j=0]$ può essere *implicata* dai vincoli $[\mathbf{A}\mathbf{y} \leq \mathbf{e}]$.

Vediamo un *esempio*:

$$\mathbf{A}\mathbf{y} \leq \mathbf{e} \equiv \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{cases} y_1 + y_3 + y_4 \leq 1 \\ y_2 + y_3 \leq 1 \\ y_1 + y_2 + y_3 \leq 1 \\ y_2 + y_4 \leq 1 \end{cases}$$

$$y_1 + y_3 + y_4 \leq 1 \Rightarrow \begin{cases} y_1 = 1 \Rightarrow y_3 = 0 \\ y_1 = 1 \Rightarrow y_4 = 0 \\ y_3 = 1 \Rightarrow y_4 = 0 \end{cases} \quad y_1 + y_2 + y_3 \leq 1 \Rightarrow \begin{cases} y_1 = 1 \Rightarrow y_2 = 0 \\ y_1 = 1 \Rightarrow y_3 = 0 \\ y_2 = 1 \Rightarrow y_3 = 0 \end{cases}$$

$$y_2 + y_3 \leq 1 \Rightarrow \{y_2 = 1 \Rightarrow y_3 = 0\} \quad y_2 + y_4 \leq 1 \Rightarrow \{y_2 = 1 \Rightarrow y_4 = 0\}$$

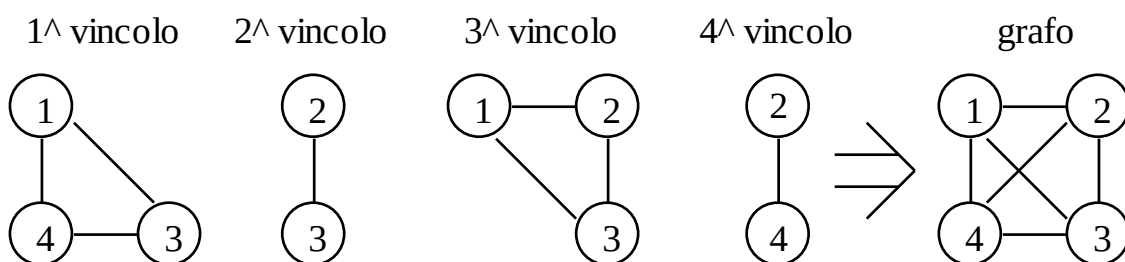
$$1^{\wedge} \text{ vincolo} \Rightarrow \mathbf{N}(1) = \{1,3,4\} \quad \mathbf{E}(1) = \{ (1,3), (1,4), (3,4) \}$$

$$2^{\wedge} \text{ vincolo} \Rightarrow \mathbf{N}(2) = \{2,3\} \quad \mathbf{E}(2) = \{ (2,3) \}$$

$$3^{\wedge} \text{ vincolo} \Rightarrow \mathbf{N}(3) = \{1,2,3\} \quad \mathbf{E}(3) = \{ (1,2), (1,3), (2,3) \}$$

$$4^{\wedge} \text{ vincolo} \Rightarrow \mathbf{N}(4) = \{2,4\} \quad \mathbf{E}(4) = \{ (2,4) \}$$

$$\text{grafo} \Rightarrow \mathbf{N} = \{1,2,3,4\} \quad \mathbf{E} = \{ (1,2), (1,3), (1,4), (2,3), (2,4), (3,4) \}$$



COSTRUZIONE DEL GRAFO DEGLI IMPLICANTI:

al j -esimo vincolo di $[\mathbf{A}\mathbf{y} \leq \mathbf{e}]$ del tipo " $\mathbf{y}_{j_1} + \mathbf{y}_{j_2} + \dots + \mathbf{y}_{j_k} \leq 1$ " corrisponde un grafo

$\mathbf{G}(j) = (\mathbf{N}(j), \mathbf{E}(j))$ completo di archi dove:

- $\mathbf{N}(j) = \{j_1, j_2, \dots, j_k\}$
- $\mathbf{E}(j) = \{ (u,v) : u,v \in \mathbf{N}(j), u < v \}, \quad |\mathbf{E}(j)| = k(k-1)/2$

$$\bullet \quad \mathbf{G} = \bigcup_j \mathbf{G}(j) \quad \mathbf{N} = \bigcup_j \mathbf{N}(j) \quad \mathbf{E} = \bigcup_j \mathbf{E}(j) .$$

Al nodo $i \in N$ viene associato una **tripla** : $T_i = (FS(i), b_i, p_i)$

- $FS(i) = \{ j \in N : (i,j) \in E \} = \text{forward_star_}i$
- $b_i = \text{guadagno}$ derivante dal porre $y_i = 1$
- $p_i = \sum_{j \in FS(i)} b_j = \text{perdita}$ derivante dal porre $y_i = 1$

LEMMA Se $(i,j) \in E$ allora $[Ay \leq e] \Rightarrow [y_i + y_j \leq 1]$.

Il lemma ci dice che il problema (SPP) si può trasformare in un altro problema, in cui ci sono $|E| \leq n(n-1)/2$ vincoli a due variabili, chiamato "*Vertex packing generalizzato*" (VPG) definito come:

$$(VPG) = \begin{cases} \max & by \\ y_i + y_j \leq 1, & \forall (i, j) \in E. \\ y \in \{0,1\}^m \end{cases}$$

Vemuganti [12] dà un algoritmo di enumerazione parziale al problema "*Vertex packing*" (VP), il quale è il problema (VPG) con $b=e$, basato su rilassamento continuo e determinante della matrice di base associata. Comunque il problema (VPG) resta difficile da risolvere.

4.3 Un algoritmo greedy euristico per il set packing.

Sappiamo che ad ogni nodo "i" è associato un guadagno b_i e una perdita p_i , derivante dal porre $y_i = 1$.

Adesso possiamo *selezionare il nodo* che ha:

- 1) il *massimo* guadagno e a parità quello che ha la *minima* perdita
- 2) la *minima* perdita e a parità quello che ha il *massimo* guadagno
- 3) il *massimo* rapporto b_i / p_i oppure il *minimo* rapporto p_i / b_i

Regola di selezione del greedy : per esempio usiamo 2) $O(m)$

$BEST(N) = u \in U, \quad U = \{j \in V : b_j \text{ è massimo}\} \quad V = \{i \in N : p_i \text{ è minima}\}$

Regola di ammissibilità del greedy : $O(n)$

$$y_i = 1 \quad \Rightarrow \quad \forall j \in FS(i) \quad y_j = 0$$

Il seguente greedy euristico restituisce:

- una soluzione ammissibile = $\mathbf{y}^*$ per il set packing (SPP)
- il valore $z^* = \mathbf{b}\mathbf{y}^*$ della funzione obiettivo di (SPP)

Input:

- $\mathbf{G} = (\mathbf{N}, \mathbf{E})$ grafo di $[\mathbf{A}\mathbf{y} \leq \mathbf{e}]$;
- $\mathbf{T} = (\mathbf{FS}, \mathbf{b}, \mathbf{p})$ triple di $\mathbf{N}$.

procedure greedy_spp ($\mathbf{N}, \mathbf{T}, \mathbf{y}^*, z^*$);

begin

```
|    $\mathbf{y}^* := \mathbf{0}; \quad z^* := 0;$ 
|   repeat
|   |    $\mathbf{u} := \text{BEST}(\mathbf{N}); \quad \mathbf{y}^*_{\mathbf{u}} := 1; \quad z^* := z^* + \mathbf{b}_{\mathbf{u}};$ 
|   |   for each  $\mathbf{j} \in (\mathbf{FS}[\mathbf{u}] \cap \mathbf{N})$  do
|   |   |   begin
|   |   |   |    $\mathbf{y}^*_{\mathbf{j}} := 0;$ 
|   |   |   |   for each  $\mathbf{i} \in (\mathbf{FS}[\mathbf{j}] \cap [\mathbf{N} \setminus \mathbf{FS}[\mathbf{u}] \setminus \{\mathbf{u}\}])$  do  $\mathbf{p}_{\mathbf{i}} := \mathbf{p}_{\mathbf{i}} - \mathbf{b}_{\mathbf{j}};$ 
|   |   |   end;
|   |    $\mathbf{N} := \mathbf{N} \setminus \mathbf{FS}[\mathbf{u}] \setminus \{\mathbf{u}\}$ 
|   until  $\mathbf{N} = \emptyset$ 
```

end.

$\text{BEST}(\mathbf{N}) = \mathbf{u} \in \mathbf{U}, \quad \mathbf{U} = \{\mathbf{j} \in \mathbf{V} : \mathbf{b}_{\mathbf{j}} \text{ è massimo}\} \quad \mathbf{V} = \{\mathbf{i} \in \mathbf{N} : \mathbf{p}_{\mathbf{i}} \text{ è minima}\}$

Complessità di calcolo:

- costruzione del grafo: $O(n \cdot m^2 \cdot d^2)$, $0 \leq d \leq 1$
 - greedy_spp: $O(m^2)$ $d = \text{densità (\%)} \text{ di } \mathbf{A}$
-

esempio:

$$\max y_1 + 2y_2 + 4y_3 + y_4 + 2y_5 + y_6$$

$$1: \quad y_1 + y_3 + y_6 \leq 1$$

$$2: \quad y_2 + y_3 + y_4 + y_6 \leq 1$$

$$3: \quad y_3 + y_5 \leq 1$$

$$4: \quad y_2 + y_4 \leq 1$$

$$5: \quad y_5 + y_6 \leq 1$$

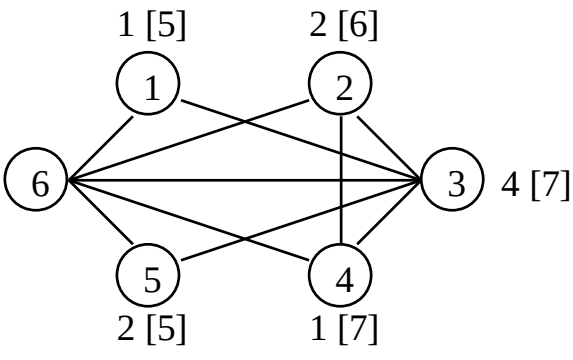
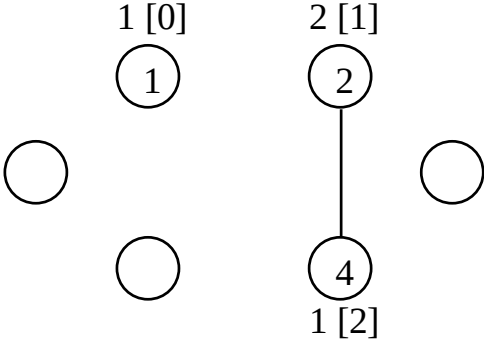
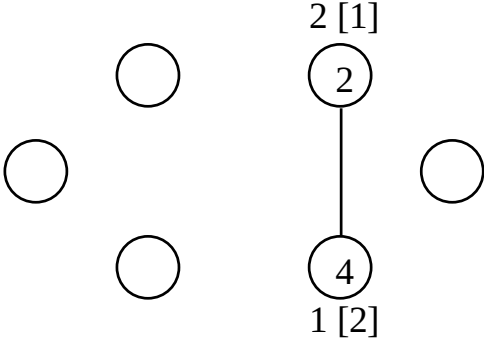
$$N(1)=\{1,3,6\} \quad E(1)=\{(1,3),(1,6),(3,6)\}$$

$$N(2)=\{2,3,4,6\} \quad E(2)=\{(2,3),(2,4),(2,6),(3,4),(3,6),(4,6)\}$$

$$N(3)=\{3,5\} \quad E(3)=\{(3,5)\}$$

$$N(4)=\{2,4\} \quad E(4)=\{(2,4)\}$$

$$N(5)=\{5,6\} \quad E(5)=\{(5,6)\}$$

	$\begin{matrix} \textcircled{i} & b_i[p_i] \\ N = \{1, 2, 3, 4, 5, 6\} \\ \text{BEST}(N) = 5 \\ y_5 = 1 \implies y_3 = y_6 = 0 \end{matrix}$
	$\begin{matrix} N = \{1, 2, 4\} \\ \text{BEST}(N) = 1 \\ y_1 = 1 \implies \emptyset \end{matrix}$
	$\begin{matrix} N = \{2, 4\} \\ \text{BEST}(N) = 2 \\ y_2 = 1 \implies y_4 = 0 \end{matrix}$
grafo vuoto	$\begin{matrix} y = (1, 1, 0, 0, 1, 0) \\ z = yb = 5 \end{matrix}$

Osserviamo che se il BEST(N) fosse implementato con la tecnica 1) "del massimo guadagno e a parità quello a minima perdita" allora il migliore sarebbe il nodo 3, quindi la rispettiva variabile settata ad 1 implicherebbe tutte le altre variabili a 0 come si deduce dal grafo iniziale, e l'ottimo sarebbe $y = (0, 0, 1, 0, 0, 0)$ $z = yb = 4$.

5. Il problema del multicovering.

5.1 Formulazione matematica.

Il problema multicovering viene formulato come:

$$(\text{MCP}) = \begin{cases} \min \mathbf{c}\mathbf{x} & (1) \quad \mathbf{A} \in \mathbf{M}(m, n, \{0,1\}) \\ \mathbf{A}\mathbf{x} \geq \mathbf{b} & (2) \quad \mathbf{b} \in \mathbf{Z}_+^m \\ \mathbf{x} \in \{0,1\}^n & (3) \quad \mathbf{c} \in \mathbf{Z}_+^n \end{cases}$$

dove $\mathbf{M}(m,n,\{0,1\})$ è la classe delle matrici con m righe ed n colonne ad elementi 0,1.

Il problema multicovering è la generalizzazione del set covering (SCP), il quale si ottiene dal (MCP) ponendo $\mathbf{b}=\mathbf{e}=(1,1,\dots,1)$, ed è stato largamente studiato.

Etcheberry [5] dà un algoritmo di branch & bound per (MCP) basato su rilassamento continuo e tecniche di ottimizzazione di subgradienti.

Beasley [1] dà un algoritmo di branch & bound per (MCP) basato su test di riduzione, greedy euristico sul duale, e duale lagrangiano che riceve come punto iniziale la soluzione data dal greedy euristico.

Hall [11] dà un algoritmo di branch & bound per (MCP) basato su una euristica primale e un greedy euristico sul duale.

Gonsalvez [6] dà un algoritmo greedy euristico per (MCP) basato su 10 criteri diversi di selezione della variabile.

Hall [10] dà un algoritmo di branch & bound per (MCP) basato su una euristica primale (scelta fra i 10 criteri dati da Gonsalvez [6]), da questa ricava una soluzione ammissibile duale usata:

- per generare un piano di tagli ed eliminare la soluzione primale,
- come punto iniziale del duale lagrangiano.

Noi introdurremo in questo articolo degli elementi nuovi e daremo un algoritmo di branch & bound per (MCP) nel paragrafo 5.6, basato su una euristica primale e come rilassamenti:

- 1) un greedy euristico sul duale decomposizione lagrangiana;
- 2) il rilassamento surrogato con i pesi $\mathbf{y}^*$ dato dal greedy del punto 1);
- 3) il rilassamento surrogato con i pesi $\mathbf{y}^*=\mathbf{e}$.

5.2 Duale debole, rilassamento surrogato e rilassamento lagrangiano.

Notazione: $\mathbf{yA}=\mathbf{A}^t \mathbf{y}$

Il duale debole di (MCP):

il rilassamento continuo del duale debole non è altro che il duale del rilassamento continuo di (MCP) in cui vengono eliminati i vincoli $\mathbf{x} \leq \mathbf{e}$, dove $\mathbf{e}=[1,...,1]$, e calcolando il duale

$$(\text{DWMCP}) = \begin{cases} \max \mathbf{yb} \\ \mathbf{yA} \leq \mathbf{c} \\ \mathbf{y} \in \mathbf{Z}_+^m \end{cases}$$

osserviamo che se i costi sono unitari, $\mathbf{c}=\mathbf{e}$, allora il duale debole diventa un problema di set packing (DWMCP)=(SPP), infatti $\mathbf{yA} \leq \mathbf{e} \Rightarrow \mathbf{y} \in \{0,1\}^m$, dato che $\mathbf{A}$ è una matrice di 0-1.

Il rilassamento surrogato di (MCP): $\forall \lambda \in \mathcal{R}_+^m$

si ottiene premoltiplicando i vincoli per i pesi e sommando tutti i vincoli

$$(\text{RSMCP}) = \begin{cases} \min \mathbf{cx} \\ [\lambda \mathbf{A}] \mathbf{x} \geq [\lambda \mathbf{b}] \\ \mathbf{x} \in \{0,1\}^n \end{cases}$$

osserviamo che il rilassamento surrogato è un problema di knapsack, inoltre se i costi sono unitari $\mathbf{c}=\mathbf{e}$, allora il suo rilassamento continuo dà sempre la soluzione ottima.

Il rilassamento lagrangiano di (MCP): $\forall \lambda \in \mathcal{R}_+^m$

si ottiene premoltiplicando il vettore λ per $\mathbf{b}-\mathbf{Ax}$ e sommando il tutto alla funzione obiettivo

$$(\text{RLMCP}) = \begin{cases} \min \mathbf{cx} + \lambda (\mathbf{b} - \mathbf{Ax}) \\ \mathbf{x} \in \{0,1\}^n \end{cases} = \begin{cases} \min \lambda \mathbf{b} + (\mathbf{c} - \lambda \mathbf{A}) \mathbf{x} \\ \mathbf{x} \in \{0,1\}^n \end{cases}$$

osserviamo che se $\mathbf{c}-\lambda \mathbf{A} \geq 0$ allora l'ottimo del rilassamento lagrangiano è proprio $\lambda \mathbf{b}$ e si ha per $\mathbf{x}=\mathbf{0}$.

Teorema 5.1

$\forall \mathbf{y}$ ammissibile per (DWMCP)

$$1) \Rightarrow z(\text{RLMCP}(\mathbf{y})) = \mathbf{yb}$$

$$2) \Rightarrow z(\text{RSMCP}(\mathbf{y})) \geq \mathbf{yb}$$

Dimostrazione

$$1) \quad z(\text{RLMCP}(\mathbf{y})) = \mathbf{yb} + \min\{(\mathbf{c} - \mathbf{yA})\mathbf{x} : \mathbf{x} \in \{0,1\}^n\} = \mathbf{yb},$$

poichè $(\mathbf{c} - \mathbf{yA}) \geq \mathbf{0}$, e $\mathbf{0} \in \{0,1\}^n$, per cui l'ottimo si ottiene per $\mathbf{x} = \mathbf{0}$.

2) il rilassamento surrogato "domina" il rilassamento lagrangiano.

Il Teorema 5.1 afferma che ogni soluzione ammissibile per il duale debole di (MCP) passato come "input":

- 1) al rilassamento lagrangiano, questo dà un ottimo uguale al valore della funzione obiettivo del duale debole.
- 2) al rilassamento surrogato, questo dà un ottimo maggiore o uguale del valore della funzione obiettivo del duale debole.

Corollario 5.1

$\forall \mathbf{y}$ ammissibile per (DWMCP) se $\mathbf{c} = \mathbf{yA} \Rightarrow$

$\Rightarrow z(\text{RSMCP}(\mathbf{y})) = \mathbf{yb}$ se risolviamo $\text{RSMCP}(\mathbf{y})$ mediante il greedy_kp dato nel capitolo del knapsack, oppure mediante il suo rilassamento continuo.

Dimostrazione

se risolviamo $\text{RSMCP}(\mathbf{y})$ con:

- il greedy_kp otteniamo la soluzione

$$\begin{cases} \min \mathbf{cx} \\ [\mathbf{yA}]\mathbf{x} \geq [\mathbf{yb}] \\ \mathbf{x} \in \{0,1\}^n \end{cases} = \begin{cases} \min \mathbf{cx} \\ \mathbf{cx} \geq [\mathbf{yb}] \\ \mathbf{x} \in \{0,1\}^n \end{cases} \geq \begin{cases} \max \mu[\mathbf{yb}] \\ \mu\mathbf{c} \leq \mathbf{c} \\ \mu \in \mathbb{R}_+ \end{cases} = \mathbf{yb} \quad (\mu^* = 1)$$

- con il suo rilassamento continuo

$$\begin{cases} \min \mathbf{cx} \\ [\mathbf{yA}]\mathbf{x} \geq [\mathbf{yb}] \\ \mathbf{x} \in [0,1]^n \end{cases} \geq \begin{cases} \min \mathbf{cx} \\ \mathbf{cx} \geq [\mathbf{yb}] \\ \mathbf{x} \in [0,1]^n \end{cases} = \mathbf{yb}$$

5.3 Un greedy euristico per il duale debole del multicovering.

$$\text{Consideriamo il duale debole : DWMCP}(\mathbf{A}, \mathbf{b}, \mathbf{c}) = \begin{cases} \max \mathbf{y}\mathbf{b} \\ \mathbf{y}\mathbf{A} \leq \mathbf{c} \\ \mathbf{y} \in \mathbf{Z}_+^m \end{cases}$$

Teorema 5.2

$\forall \mathbf{y}$ ammissibile per $\text{DWMCP}(\mathbf{A}, \mathbf{b}, \mathbf{e}) = \text{set packing (SPP)}$, $t = \min\{ \mathbf{c}_j : \mathbf{c}_j > 0, j=1,2,\dots,n \}$

1) $\Rightarrow \mathbf{t}\mathbf{y}$ ammissibile per $\text{DWMCP}(\mathbf{A}, \mathbf{b}, \mathbf{t}\mathbf{e})$

2) $\Rightarrow \mathbf{t}\mathbf{y}$ ammissibile per $\text{DWMCP}(\mathbf{A}, \mathbf{b}, \mathbf{c})$.

Dimostrazione

per ipotesi è $\mathbf{y}\mathbf{A} \leq \mathbf{e} \Rightarrow (\mathbf{t}\mathbf{y})\mathbf{A} \leq \mathbf{t}\mathbf{e} \leq \mathbf{c}$ per come è stato scelto t .

Il seguente greedy euristico restituisce:

- una soluzione ammissibile $\mathbf{y}^*$ per (DWMCP)
- un lower_bound z^* per (MCP)

procedure greedy_dwmcp ($\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{y}^*, z^*$);

begin

```
| BREAK-GRAFO; {  $\mathbf{G}=(\mathbf{N}, \mathbf{E})$  di  $[\mathbf{y}\mathbf{A} \leq \mathbf{e}]$  ;  $\mathbf{T}=(\mathbf{FS}, \mathbf{b}, \mathbf{p})$  triple di  $\mathbf{N}$ 
| costruzione del grafo degli impicanti }
| greedy_spp (  $\mathbf{N}, \mathbf{T}, \mathbf{y}, z$  );
|  $t := \min\{ \mathbf{c}_j : \mathbf{c}_j > 0, j=1,2,\dots,n \}$ ;
|  $\mathbf{y}^* := \mathbf{t}\mathbf{y}$ ;
|  $z^* := \mathbf{t}\mathbf{y}\mathbf{b}$ 
```

end.

Complessità di calcolo:

- costruzione del grafo: $O(n \cdot m^2 \cdot d^2)$, $0 \leq d \leq 1$
- greedy_spp: $O(m^2)$ $d = \text{densità (\%)} \text{ di } \mathbf{A}$
- greedy_dwmcp $O(m^2 [1+n \cdot d^2])$

5.4 Decomposizione lagrangiana: una successione finita di greedy euristiche su duali deboli.

Applichiamo il rilassamento *decomposizione lagrangiana* al problema del multicovering con un trucco: viene fatta una copia dei vincoli

$$\text{MCP}(\mathbf{A}, \mathbf{b}, \mathbf{c}) = \left\{ \begin{array}{l} \min \mathbf{c}\mathbf{x} \\ \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \in \{0,1\}^n \end{array} \right\} \equiv \left\{ \begin{array}{l} \min \mathbf{c}\mathbf{x} \\ \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \in \{0,1\}^n \end{array} \right\} \geq \left\{ \begin{array}{l} \min \mathbf{c}^{(1)}\mathbf{x} \\ \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \in \{0,1\}^n \end{array} \right\} + \left\{ \begin{array}{l} \min (\mathbf{c} - \mathbf{c}^{(1)})\mathbf{x} \\ \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \in \{0,1\}^n \end{array} \right\}$$

$$\text{MCP}(\mathbf{A}, \mathbf{b}, \mathbf{c}) \geq \text{MCP}(\mathbf{A}, \mathbf{b}, \mathbf{c}^{(1)}) + \text{MCP}(\mathbf{A}, \mathbf{b}, \mathbf{c} - \mathbf{c}^{(1)})$$

La decomposizione lagrangiana necessita di una partizione dei costi, noi useremo come decompositore dei costi il duale debole $\text{DWMCP}(\mathbf{A}, \mathbf{b}, \mathbf{c})$.

Teorema 5.3

Sia $\mathbf{y}^{(1)}$ la soluzione di $\text{DWMCP}(\mathbf{A}, \mathbf{b}, \mathbf{c})$ data dal greedy_dwmcp,

sia $\mathbf{c}^{(1)} = \mathbf{y}^{(1)} \mathbf{A}$ vale che:

- 1) $\Rightarrow \mathbf{c}^{(1)} \leq \mathbf{c}$
- 2) $\Rightarrow \mathbf{y}^{(1)}$ è ammissibile per $\text{DWMCP}(\mathbf{A}, \mathbf{b}, \mathbf{c}^{(1)})$
- 3) $\Rightarrow \text{MCP}(\mathbf{A}, \mathbf{b}, \mathbf{c}) \geq \mathbf{y}^{(1)} \mathbf{b} + \text{KP}(\mathbf{A}, \mathbf{b}, \mathbf{c} - \mathbf{c}^{(1)})$

Dimostrazione

- 1): $\mathbf{c}^{(1)} = \mathbf{y}^{(1)} \cdot \mathbf{A} \leq \mathbf{c}$ poichè $\mathbf{y}^{(1)}$ è ammissibile per $\text{DWMCP}(\mathbf{A}, \mathbf{b}, \mathbf{c})$.
- 2): $\mathbf{y}^{(1)}$ è ammissibile per $\text{DWMCP}(\mathbf{A}, \mathbf{b}, \mathbf{c}^{(1)})$ per come è costruito $\mathbf{c}^{(1)}$ e in particolare vale come disuguaglianza $\mathbf{y}^{(1)} \mathbf{A} \leq \mathbf{c}^{(1)}$.
- 3): segue dalla decomposizione lagrangiana e dal punto 2).

Il Teorema 5.3 suggerisce di iterare il procedimento ponendo $\mathbf{c} \leftarrow \mathbf{c} - \mathbf{c}^{(1)}$, previa modifica del problema in quanto possono sorgere dei costi $\mathbf{c}_j$ nulli.

Il multicovering MCP($\mathbf{A}, \mathbf{b}, \mathbf{c}$) viene così modificato ad ogni iterazione:

- $\forall \mathbf{c}_k = 0$ essendo il problema di minimo si pone $\mathbf{x}_k = 1$, l'incremento di costo è nullo, in compenso diminuisce $\mathbf{b} \leftarrow \mathbf{b} - \mathbf{A}^k$ (dove $\mathbf{A}^k$ è la k -esima colonna di $\mathbf{A}$), e cambia $\mathbf{A} \leftarrow \mathbf{A} \setminus \mathbf{A}^k$, $\mathbf{c} \leftarrow \mathbf{c} \setminus \mathbf{c}_k$.
- $\forall \mathbf{b}_i = 0$ $\mathbf{A} \leftarrow \mathbf{A} \setminus \mathbf{A}_i$ dove $\mathbf{A}_i$ è la i -esima riga di $\mathbf{A}$.

Il procedimento si itera fino a quando $\mathbf{b} = \mathbf{0}$ oppure $\mathbf{c} = \mathbf{0}$.

Riassumendo, se applichiamo la decomposizione lagrangiana al problema (MCP) con k copie di $\mathbf{Ax} \geq \mathbf{b}$ si ha:

$$\text{MCP}(\mathbf{A}, \mathbf{b}, \mathbf{c}) \geq \sum_{i=1}^k \text{MCP}(\mathbf{A}, \mathbf{b}, \mathbf{c}^{(i)}) \geq \sum_{i=1}^k \text{DWMCP}(\mathbf{A}, \mathbf{b}, \mathbf{c}^{(i)})$$

tale che $\sum_{i=1}^k \mathbf{c}^{(i)} = \mathbf{c}$, $\mathbf{c}^{(i)}$, k da determinare

previa modifica ad ogni passo di $\mathbf{A}, \mathbf{b}$.

Osservazione

Il numero delle iterazioni sopra descritte è limitato superiormente da

$$M = \sum_i \mathbf{b}_i + \sum_j \mathbf{c}_j + \text{opt}, \quad \text{opt} = \text{ottimo di (MCP)}.$$

Infatti ad ogni iterazione almeno una delle seguenti tre asserzioni è vera:

- 1) $\mathbf{b} \geq \mathbf{0}$ decresce almeno di una unità intera di almeno un componente.
- 2) $\mathbf{c} \geq \mathbf{0}$ decresce almeno di una unità intera di almeno un componente.
- 3) la funzione obiettivo $\mathbf{y}^{(1)} \mathbf{b}$ cresce almeno di una unità intera e non può superare l'ottimo di (MCP) perchè si tratta di un rilassamento.

5.5 Un algoritmo greedy euristico per il multicovering.

Il seguente greedy euristico restituisce:

- i pesi $= y^*$ dei vincoli $Ax \geq b$ (input per il rilassamento surrogato)
- un lower_bound $= z^*$ per (MCP)
- una soluzione ammissibile $= x^*$ per (MCP).

procedure greedy_mcp (A, b, c, x^*, y^*, z^*);

begin

| $x^* := \{ j : c_j = 0 \}; \quad y^* := 0; \quad z^* := 0; \quad I := \{ i : b_i > 0 \};$

| **repeat**

| | greedy_dwmcp(A, b, c, y, z);

| | $y^* := y^* + y; \quad z^* := z^* + z; \quad c := c - yA;$

| | **for each** $j : (c_j = 0) \text{ and } (j \notin x^*) \text{ and } (I \neq \emptyset) \text{ do}$

| | **begin**

| | | $x^* := x^* \cup \{j\}; \quad b := b - A^j; \quad A := A \setminus A^j; \quad c := c \setminus c_j;$

| | | **for each** $i : (b_i = 0) \text{ and } (i \in I) \text{ do}$

| | | **begin**

| | | | $I := I \setminus \{i\}; \quad A := A \setminus A_i$

| | | **end**

| | **end**

| **until** $(b = 0) \text{ or } (c = 0);$

| if $(b > 0)$ then "problema non ammissibile"

end.

Complessità di calcolo:

- greedy_dwmcp: $O(m^2 [1 + n \cdot d^2])$ $0 \leq d \leq 1$
- greedy_mcp: $O(M \cdot m^2 [1 + n \cdot d^2])$ $d = \text{densità (\%)} \text{ di } A$
 $M = \min \{ \sum_i b_i, \sum_j c_j, \text{opt} \},$ $\text{opt} = \text{ottimo di (MCP)}$

esempio: greedy_mcp

$$\begin{array}{c|ccccc}
 \mathbf{A} & x_1 & x_2 & x_3 & x_4 & x_5 \\
 \hline
 y_1 & 1 & 0 & 1 & 0 & 1 \\
 y_2 & 1 & 0 & 0 & 0 & 1 \\
 y_3 & 0 & 1 & 1 & 0 & 0 \\
 y_4 & 1 & 0 & 0 & 1 & 0
 \end{array}
 \begin{array}{c}
 \mathbf{b} \\
 || \\
 \hline
 1 \\
 1 \\
 1 \\
 1
 \end{array}
 \begin{array}{l}
 \text{Ottimo di (P)} \\
 \mathbf{x}^* = (1,1,0,0,0) \\
 \mathbf{c}\mathbf{x}^* = 13
 \end{array}$$

$$\mathbf{c} = \begin{array}{|c|c|c|c|c|}
 \hline
 9 & 4 & 7 & 2 & 8 \\
 \hline
 \end{array}$$

Modifiche al grafo degli implicanti di $[\mathbf{yA} \leq \mathbf{e}]$

$1[3] \quad 1[2]$ $1[2] \quad 1[1]$	$1[2] \quad 1[2]$ $1[2]$	$1[1]$ $1[1]$	 $1[0]$	grafo vuoto
1,2: $\mathbf{yA} \leq \mathbf{e}$	3,4: $\mathbf{yA}^2 = 0$	5: $\mathbf{yA}^3 = 0$	6: $\mathbf{yA}^5 = 0$	7: $\mathbf{yA}^1 = 0$

Evoluzione del greedy-mcp.

N° = Numero di iterazione di greedy_mcp, ad ogni iterazione :

- $\mathbf{c} = \mathbf{c} - \mathbf{c}^{(i)}$, costi ancora da partizionare
- $\mathbf{b} = \mathbf{b} \setminus \mathbf{A}^k$, quando qualche nuovo $\mathbf{c}_k = 0$
- Grafo $[\mathbf{yA} \leq \mathbf{e}]$, modifiche al grafo quando qualche nuovo $\mathbf{c}_k = 0$,
- $\mathbf{y}^*$ = Soluzione data da greedy_dwmcp
- $\mathbf{y}^* \mathbf{b}$ = Valore f. obiettivo dato da greedy_dwmcp
- z^* = Valore corrente f. obiettivo dato da greedy_mcp
- $\mathbf{c}^{(i)} = \mathbf{y}^* \mathbf{A}$, i -esima partizione dei costi

N°	$\mathbf{c}$	$\mathbf{b}$	Grafo $[\mathbf{yA} \leq \mathbf{e}]$	$\mathbf{y}^*$	$\mathbf{y}^* \mathbf{b}$	z^*	$\mathbf{c}^{(i)}$
1	(9,4,7,2,8)	(1,1,1,1)	. . .	(0,2,2,0)	4	4	(2,2,2,0,2)
2	(7,2,5,2,6)	(1,1,1,1)	. . .	(0,2,2,0)	4	8	(2,2,2,0,2)
3	(5,0,3,2,4)	(1,1,0,1)	$\mathbf{c}_2=0 \Rightarrow \mathbf{yA}^2=\mathbf{0}$	(2,0,0,0)	2	10	(2,0,2,0,2)
4	(3,0,1,2,2)	(1,1,0,1)	. . .	(1,0,0,0)	1	11	(1,0,1,0,1)
5	(2,0,0,2,1)	(0,1,0,1)	$\mathbf{c}_3=0 \Rightarrow \mathbf{yA}^3=\mathbf{0}$	(0,1,0,0)	1	12	(1,0,0,0,1)
6	(1,0,0,2,0)	(0,0,0,1)	$\mathbf{c}_5=0 \Rightarrow \mathbf{yA}^5=\mathbf{0}$	(0,0,0,1)	1	13	(1,0,0,1,0)
7	(0,0,0,1,0)	(0,0,0,0)	$\mathbf{c}_1=0 \Rightarrow \mathbf{yA}^1=\mathbf{0}$	(0,0,0,0)	0	13	(0,0,0,1,0)
Σ				(3,5,4,1)	13		(9,4,7,2,8)

Tabella di evoluzione di greedy_mcp

La colonna "Grafo $[\mathbf{yA} \leq \mathbf{e}]$ " indica ad ogni iterazione il grafo su cui viene calcolato il greedy_dwmcp che si può vedere nella tabella delle modifiche a tale grafo.

5.6 Un algoritmo branch & bound per il multicovering.

L'algoritmo branch & bound ha queste caratteristiche:

visita: *depth_first* sull'albero binario, branch: *bipartito*

rilassamenti:

1) *greedy_mcp(A,b,c,x*,y*,z*)*: un greedy sulla decomposizione lagrangiana di (MCP). Esso dà in output:

- un lower_bound z^* per (MCP)
- una soluzione ammissibile x^* per (MCP) da cui si ricava un upper_bound cx^* per (MCP)
- dei pesi y^* da utilizzare come "input" per il rilassamento surrogato.

2) *greedy_kp([y*A],[y*b],c,x,y,z)*: un greedy sulla decomposizione lagrangiana-del rilassamento surrogato di (MCP) con i pesi= y^* . Esso dà in output:

- un lower_bound z per (MCP) migliore del rilassamento continuo-del rilassamento surrogato di (MCP) con i pesi= y^* .
- una soluzione ammissibile x ed un peso y per il rilassamento surrogato di (MCP) con i pesi= y^* che in questo caso non usiamo.

3) *greedy_kp([eA],[eb],c,x,y,z)*: un greedy sulla decomposizione lagrangiana-del rilassamento surrogato di (MCP) con i pesi= e . Esso dà in output:

- un lower_bound z per (MCP) migliore del rilassamento continuo-del rilassamento surrogato di (MCP) con i pesi= e .
- una soluzione ammissibile x ed un peso y per il rilassamento surrogato di (MCP) con i pesi= e che in questo caso non usiamo.

selezione:

la selezione della variabile viene fatta scegliendo la variabile a minor costo unitario del knapsack risultante dal rilassamento surrogato di (MCP) con i pesi= y^* dato dal greedy del punto 1), e precisamente:

$$select = x_k \quad \text{se} \quad \frac{c_k}{[y^*A]_k} = \min_j \left\{ \frac{c_j}{[y^*A]_j} \right\}$$

6. Risultati sperimentali.

Diamo di seguito alcuni risultati sperimentali dell'algoritmo branch & bound per (MCP) su istanze di problemi generati casualmente al variare dei vincoli e delle variabili:

n=numero delle variabili

m=numero dei vincoli

d=densità di **A** espresso in percentuale %

LB1=lower_bound dato da *greedy_mcp*(**A**,**b**,**c**,**x***,**y***,**z***)

LB2=lower_bound dato da *greedy_kp*([**y*****A**],[**y*****b**],**c**,**x**,**y**,**z**)

LB3=lower_bound dato da *greedy_kp*([**eA**],[**eb**],**c**,**x**,**y**,**z**)

NODI=numero del nodo relativo all'ultima soluzione trovata

l'algoritmo viene stoppato a 100 sec. oppure è terminato prima di 100 sec.

la macchina su cui le prove sono state fatte è *Dec_Vax*.

Tabella dei risultati:

n	m	d	LB1	LB2	LB3	SOL	NODI
50	50	36 %	321	327	95.6	345	307
50	50	30 %	349	392	127.8	487	428
50	50	19 %	1237	1285	695.6	1561	2070
50	50	11 %	831	831	306	918	45
100	100	38 %	2422	2422	449.3	2512	429
100	100	20 %	1983	2129	474.1	2373	48
100	100	6 %	3577	3645	807.2	3833	158
100	100	2.5 %	8006	8006	2225.7	8093	16
100	200	7 %	926	943	262.6	1151	101
100	500	2 %	2971	2987	898	3018	184
100	700	2 %	4057	4451	1781	4736	112
100	1000	2 %	3531	3864	1666.5	4092	203
200	100	18 %	610	714	229.1	911	49
500	100	6 %	1622	1698	269.4	1925	54
700	100	2 %	6939	6939	1320.1	6942	82
1000	100	2 %	4096	4096	1062.7	4696	92
100	500	2 %	4540	4540	1228.6	4548	156
500	100	3.9 %	2038	2038	341.7	2112	763
300	500	3.8 %	2681	2735	954.1	2963	185
500	300	1.8 %	3524	3652	724.8	3876	130
400	700	1 %	4806	4978	1324.3	5330	188
700	400	1 %	6622	6702	1386.3	6805	186
500	500	1 %	4477	4894	1274.4	5031	168
500	1000	0.5 %	9783	9783	4920	9951	246
1000	500	0.5 %	12145	12145	6223	12145	120

Ringraziamenti

Un particolare ringraziamento al prof. **G. Gallo** per essersi costantemente interessato allo sviluppo e alle variazioni del contenuto di questo articolo.

Bibliografia.

- [1] BEASLEY J. E. An algorithm for the set covering problem.
European Journal or Operational Research, 31, [85-93], 1987.
- [2] CARPANATO G., DI PILLO G. Metodi e algoritmi per
l'ottimizzazione, *Gris Collana Pitagora*, 1982.
- [3] CARRARESI P., GALLO G., PALLOTTINO S. Appunti di
programmazione matematica, *Seu Servizio Editoriale Universitario
di Pisa*, 1993.
- [4] CRAMA Y., HAMMERS P. L., IBARAKI T. Packing, covering
and partition problems with strongly unimodular constraint
matrices, *Mathematics of Operations Research*, vol. 15, N°=2,
[258-267], 1990.
- [5] ETCHEBERRY J. The set covering problem: a new implicit
enumeration algorithm, *Operation Research*, vol. 25, N°=5,
[760-772], 1987.
- [6] GONSALVEZ D. J., HALL N. G., RHEE W. T., SIFERD S. P.
Heuristic solution and confidence intervals for the multicovering
problem, *European Journal or Operational Research*, 31,
[94-101], 1987.
- [7] GUIGNARD M., KIM S. Lagrangean decomposition: a model
yielding stronger lagrangean bound, *Mathematical Programming*,
39, [215-228], 1987.

- [8] GUIGNARD M. KIM S. Lagrangean decomposition for integer programming: theory and application, *RAIRO Operation Research*, 21, [307-323], 1987.
- [9] GUIGNARD M. General aggregation schemes in lagrangean decomposition: theory and potential applications, *Departement of Decision Science, the Wharton School, University of Pensylvania, Philadelphia, PA 19104-6366, USA*, 1991.
- [10] HALL H. G., HOCHBAUM D. S. The multicovering problem *European Journal of Operational Research*, 62, [323-339], 1992.
- [11] HALL H. G., HOCHBAUM D. S. A fast approssimation algorithm for the multicovering problem, *Discrete Applied Mathematics*, 15, [35-40], 1986.
- [12] HOUCH D. J., VEMUGANTI R. R. An algorithm for the vertex paching problem, *Operations Research*, vol. 25, N°=5, [773-788], 1977.
- [13] NEMHAUSER G., WOLSEY L. Integer and combinatorial optimization, *Wiley_Interscience Pubblication, John Wiley & Sons*, 1988.
- [14] PARKER G., RARDIN R. L. Discrete optimization, *Accademic Press, Inc.*, 1988.
- [15] RARDIN R. L., KARWAN M. H. Surrogate dual multiplier search procedure in integer programming, *Operation Research*, vol. 32, N°=1, 1984.
- [16] SALKIN H. M., MATHUR K. Foundation of integer programming, *New York, North-Holland*, 1989.

Riferimento: Irace dott. Gianni (393-0679616)

e-mail: cons.iracegianni@siassb.eu
gianni.irace@hotmail.it
gianniirace@virgilio.it
gianni.irace@irasoft.it