

# **Business Analytics – 2022/23**

## **Decomposition methods in optimization**

Prof. Paolo Brandimarte

Dip. di Scienze Matematiche – Politecnico di Torino

e-mail: [paolo.brandimarte@polito.it](mailto:paolo.brandimarte@polito.it)

URL: [staff.polito.it/paolo.brandimarte](http://staff.polito.it/paolo.brandimarte)

This version: June 15, 2023

**NOTE:** For internal teaching use within the Masters' Program in Mathematical Engineering. Do not post or distribute.

## References

---

- A.J. Conejo , E. Castillo , R. Mínguez, R. García-Bertrand. *Decomposition Techniques in Mathematical Programming*. Springer, 2006.
- P. Kall, S.W. Wallace. *Stochastic Programming*. Wiley, 1994. The book pdf can be downloaded from <http://stoprog.org/index.html?resources.html>
- J.L. Higle, S. Sen. *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*. Kluwer, 1996.

## Motivation

↳ Lo devi implementare manualmente, non lo si compra

Decomposition methods play a prominent role in optimization, as they allow:

- To take advantage of favorable structure: → Sfruttare la struttura del modello
  - network sub problems; → Sono risolvibili (polinomiale) con algoritmi specifici e particolarmente efficienti
  - from NP-hard to polynomial. → Risoluvi in problema approssimata il problema NPH con problemi P
- To parallelize the solution of large-scale problems. → Decomposizione in parallelo  
↳ Caso particolare
- To tackle large-scale, scenario-based stochastic optimization models.
- To tackle hard combinatorial problems by dealing with a sequence of simpler subproblems, possibly mixing different solution strategies. → Decomposizione sequenziale  
↳ Affrontare
- To avoid modeling issues with very difficult constraints. → Algoritmi di assemblazione di colonne
  - ↳ A volte scrivere i vincoli risulta essere molto difficile
  - Es. Decidere gli equipaggi di aerei/treni

Consider the optimization problem

E' decomponibile sottrando  
l'ottimo all'interno della  
somma.

$$\text{opt} \quad \sum_{j \in [n]} f_j(x_j)$$

s.t.  $x_j \in S_j$ . → Regioni di ammissibilità indipendenti

$\rightarrow$  Insieme degli interi da 1 a n, cioè  $[n] = \{1, \dots, n\}$

Clearly, we may decompose the problem into decoupled subproblems  $\text{opt}_{x_j \in S_j} f_j(x_j)$ , as subvectors  $x_j$  are subject to independent constraints.

In a linear programming model,

$$\begin{aligned} & \min \quad c^T x \\ & \text{s.t.} \quad Ax = b \\ & \quad x \geq 0, \end{aligned}$$

this would correspond to a block-diagonal matrix structure (missing submatrices are all zeros):

$$A = \begin{bmatrix} D_1 & & & & \\ & D_2 & & & \\ & & D_3 & & \\ & & & \ddots & \\ & & & & D_n \end{bmatrix}$$

Usually, we are not that lucky, and there will be a complicating factor.

For instance, we may deal with a large-scale LP featuring a block-angular structure in the technological matrix: *Vincoli o Variabili che rompono?*

$$\boxed{A} = \begin{bmatrix} C_1 & C_2 & C_3 & \cdots & C_n \\ D_1 & & & & \\ & D_2 & & & \\ & & D_3 & & \\ & & & \ddots & \\ & & & & D_n \end{bmatrix}, \quad \text{or} \quad \boxed{A} = \begin{bmatrix} C_1 & D_1 & & & \\ C_2 & & D_2 & & \\ C_3 & & & D_3 & \\ \vdots & & & & \ddots \\ C_n & & & & D_n \end{bmatrix}.$$

In the first case, the complicating factor is a set of interaction constraints coupling subproblems and preventing the decomposition by the block diagonal structure. We may decompose the problem if we relax the interaction constraints in some way (e.g., Lagrangian dual decomposition).  $\rightarrow$  *Decomposizione lagrangiana tramite la teoria della duality*

In the second case, we have interaction variables preventing the decomposition. We may decompose the problem if we fix the interaction variables (e.g., L-shaped decomposition, which is Benders decomposition for stochastic programming).

$\hookrightarrow$  *Decomposizione di Benders*

There is a fair number of inter-related decomposition methods. Some are exact, at least in principle, some lead to approximate algorithms:

- Lagrangian relaxation and Lagrangian decomposition.  
*E' un modello basato sul buon senso*
- Dual heuristics.
- Dantzig–Wolfe decomposition.
- Column generation.
- Hierarchical decomposition.  
*Costruire heuristiche su modelli matematici*
- Matheuristics (not to be confused with metaheuristics).  
*Principio generale, non usano la matematica*
- Benders decomposition for MILPs with special structure.
- L-shaped decomposition in two-stage stochastic programming with recourse.
- Progressive hedging for multistage stochastic programming with recourse.
- Dynamic programming (time-based decomposition). → *Decomposizione basata sul tempo*  
*Da problemi multi radio, a problemi a radio singola*

# Part 1

**Dual decomposition and progressive hedging for stochastic programming**

• Applichiamo concetti con vincoli di interazione che «corrispondono»

Gia' visto nel newsletter multiplosettore

• Condizioni KKT

• Dualità forte vs debole

## Dual decomposition

To see how duality can help in a simple setting, let us consider a problem like

Prendiamo la lagrangiana  
e moltiplichiamo il vincolo per  
il moltiplicatore di Lagrange  $\mu > 0$   
che lo consideriamo come un numero

$$\begin{aligned} \max \quad & \sum_{i=1}^n f_i(x_i) \rightarrow \text{Non} \\ \text{s.t.} \quad & \sum_{i=1}^n g_i(x_i) \leq b, \\ & x_i \in S_i, \quad i = 1, \dots, n \end{aligned}$$

posso mettere il massimo dentro  
Se  $\mu = 0$  e tutto funziona vuol dire che il  
vincolo e' superfluo e allora e' eliminabile  
Sono vincoli di interazione

Questo vincolo non ci permette di  
risolvere facilmente il problema  
Budget di risorsa finito

(3)

Let us interpret the decision variables  $x_i, i = 1, \dots, n$ , as activities yielding a profit  $f_i(x_i)$  and consuming a resource amount  $g_i(x_i)$ . The objective function (1) is total profit, and (2) is a budget constraint on the resource.

Note that the objective function is measured in monetary terms, whereas  $b$  is measured in resource units. → accoppia le decisioni

If we could get rid of the budget constraint, the problem could be decomposed.

Recap-

$$\begin{aligned} \min_{\mathbf{x}} f(\mathbf{x}) & \quad \mathcal{L}(\mathbf{x}, \mu) = f(\mathbf{x}) + \mu^T g(\mathbf{x}) \quad \text{funzione lagrangiana} \\ g(\mathbf{x}) \leq 0 \Rightarrow \min_{\mathbf{x} \in S} W(\mu) & = \min_{\mathbf{x} \in S} \mathcal{L}(\mathbf{x}, \mu) \quad \text{funzione duale} \rightarrow \text{otteniamo il lower bound sul problema originale} \\ \hookrightarrow \text{Vogliamo quindi} & \max_{\mu \geq 0} W(\mu) \end{aligned}$$

### Dual decomposition

Let us dualize the budget constraint by introducing the multiplier  $\mu \geq 0$  and writing the Lagrangian function:

$$\mathcal{L}(\mathbf{x}, \mu) = \sum_{i=1}^n f_i(\mathbf{x}_i) + \mu \left( b - \sum_{i=1}^n g_i(\mathbf{x}_i) \right) = \sum_{i=1}^n [f_i(\mathbf{x}_i) - \mu g_i(\mathbf{x}_i)] + \mu b.$$

Tengo conto del fatto che sto analizzando un problema di massimi mittutazione

Avendo dualizzato il vincolo di integrazione,  $\mu$  e' fissato

Abbiamo decomposto rispetto alle  $\mathbf{x}_i$ .

Note that here we must adjust the problem to account for the optimization sense.

The Lagrangian function should be maximized with respect to the primal variable, resulting in a set of problems: → Per un  $\mu$  fissato

$$\max_{\mathbf{x}_i \in S_i} [f_i(\mathbf{x}_i) - \mu g_i(\mathbf{x}_i)] \equiv \pi_i(\mathbf{x}_i)$$

"Sconsiglio" la risorsa, e' un costo

Così disaccoppiamo il problema  
→ Troviamo un lower bound sull'ottimo

- Each subproblem requires maximizing profit contribution minus resource cost.
- The multiplier  $\mu$  is a shadow price, measured in unit of money per unit amount of resource.
- In this case, we should minimize the dual function with respect to  $\mu$ .

non c' e' detto che  
sia sempre differentiabile

la funzione duale e' concava: esiste  
sempre il subgradiente  
Coordinando i problemi ho bisogno di una variabile,  $x$ ,  
per andare a modificare e andare oltre  $\mu$

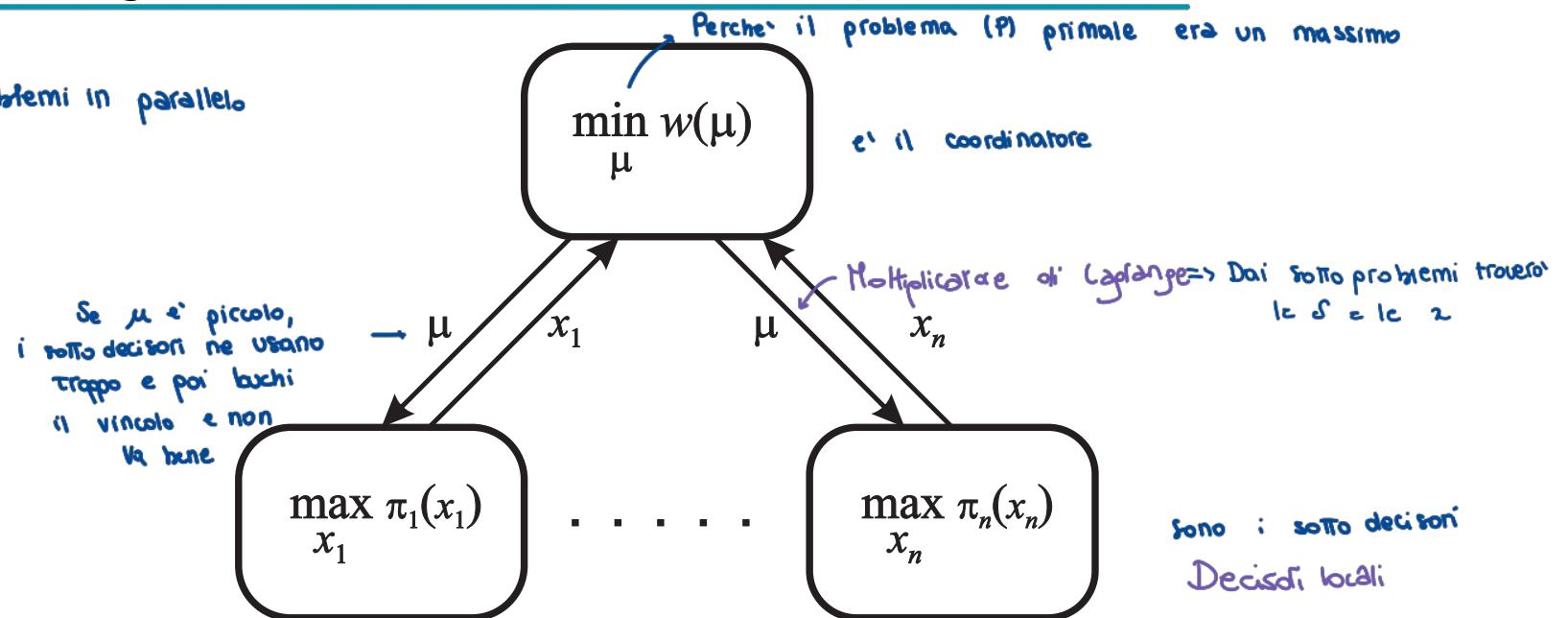
- Given relaxed solutions  $x_i^*$ , we get a subgradient of the dual function:

$$\sum_{i=1}^n g_i(x_i^*) - b$$

- Se  $\epsilon > 0$ , per cui il consumo della risorsa  
e' andato sopra il budget  $\rightarrow$  devo aumentare  $\mu$
- Se  $< 0 \rightarrow$  devo diminuire  $\mu$

- This is positive when the budget is exceeded, in which case we should increase the resource price. The price must be reduced when the budget is not exceeded.
- The resulting demand-offer scheme can be depicted as follows.

Ho decomposto in problemi in parallelo



Deterministico

RiconSIDERIAMO il problema di **Lot Sizing** con una sola risorsa e senza costi di set-up

$$\min \sum_i \sum_t (h_i I_{it} + f_i S_{it})$$

E' decomponibile rispetto ai prodotti? No, c'e' il vincolo di interazione.

$$I_{it} = I_{i,t-1} + z_{it} - d_{it} \quad V_{it} \rightarrow \text{NON} \text{ dare sommare su } i$$

$$\sum_i R_i z_{it} \leq B$$

$V_C \rightarrow$  Vincolo che scoppia i prodotti, per rendere più verosimile che anche il tempo di setup

$$\sum_{i=1}^n (R_i z_{it} + f_i S_{it}) \leq R \quad \text{il nuovo vincolo}$$

$$I_{it}, z_{it} \geq 0$$

$S_{it} \in \{0,1\} \rightarrow$  Usiamo l'euroistica duale per trovare le  $s_i$ , dopo un po' di iterazioni, e poi mettere  $\rightarrow$  il branch&cut filassa nel

$$z_{it} \leq M s_{it}$$

continuo questa variabile nel problema originale e ottenere la soluzione

Parto da un «problema camion» e lo approssimo con una serie di problemi più piccoli.

Metiamo i moltiplicatori  $\mu_C$  associati ai vari periodi per il vincolo di interazione,  $\mu_C \geq 0$   $\rightarrow$  Problema NP-H

perche' e' un problema di min

$$\hookrightarrow \sum_i \sum_t (h_i I_{it} + f_i S_{it}) + \sum_t \mu_C (\sum_i R_i z_{it} - B)$$

Che definiscono realmente le soluzioni  
Fissando le  $z_i$ , le  $s_i$  e il magazzino vamo a rimettere  
conto di produzione!

Posso decomporre rispetto ai prodotti commutando le somme:

$$\sum_i \left( \sum_t (h_i I_{it} + f_i S_{it} + \mu_C R_i z_{it}) \right) - B \sum_t \mu_C$$

Ma le variabili binarie sono ancora presenti! Il problema a capacità su un solo prodotto ha complessità polinomiale

Da NPH a tanti problemi P! Ma vale solo la dualità debole (e' non connesso) non e' detto che si trovi la soluzione ottima

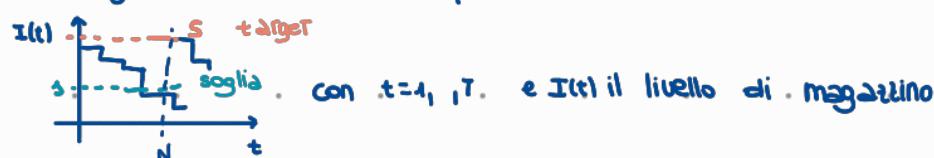
Si possono ottenere lower bound  $\rightarrow$  cosi da certificare la qualità della soluzione.

euroistica: quindi il moltiplicatore più giusto da cui possiamo fissare le  $s_i$  e ottenere un problema lineare  
duale

## Esercizio di simulazione con politica S- periodic review

NF

Problema: gestire le scorte a tempo discreto



In questa politica, ogni  $N$  (tempo) bisogna ordinare.  $\rightarrow$  con quale criterio?

1. Osservo il livello di magazzino con una certa periodicità

2. Se  $I(t) > s$ . Allora non ordino

Se  $I(t) < s$  allora ordino per arrivare  $\Rightarrow I(t)=s$   
cioè  $S - I(t)$  quindi riporto il magazzino al target

Le assunzioni implicite sono:  
1) Quello che ordino amico subito, quindi il tempo di fornitura (lead time) è zero

2) Ma se il magazzino va a zero prima del riordino? Assunzione di loss sales, cioè ho perso la vendita

↳ Se non considero questa politica devo vedere gli ordini in arretro

3) Assumiamo una distribuzione di probabilità della domanda  $\rightarrow$  es. Processo di Poisson (composto = non omogeneo)

4) La soglia  $f_S$  è che non si ordinerà mai più di  $S - I(t)$  e lo fai perché c'è un costo fisso che non vuoi pagare costantemente

## Problema del commesso viaggiatore simmetrico

E' NP-H

NF

La matrice delle distanze sono simmetriche

grafo non orientato

Vogliamo trovare un ciclo Hamiltoniano che visita tutti i vertici (nodi) partendo da uno e li ritornare

Il problema e' formalizzabile aggiungendo le s e ricordando che e' tutto simmetrico

(es. lunghezza)

$\min \sum_{e \in E} c_e z_e \Rightarrow$  Voglio trovare una collezione di lati a costo minimo

→ il cliente e' visitato 1 volta

s.t.  $\sum_{e \in E(v)} z_e = 2, \forall v \in V \Rightarrow$  Arrivi da un lato e parti da un altro. → ogni vertice ha un lato di entrata e di uscita  
 $e \in E(v) \rightarrow$  tutti i nodi che raggiungono con i vertici che hanno

Vincolo che elimina i sub-tour

$\sum_{e \in E(S)} z_e \leq |S|-1 \quad \forall S \subset V: 3 \leq |S| \leq |V|-1 \Rightarrow$  Non voglio creare cicli (sub-tour) perché non otterrei cicli Hamiltoniani  
 e neanche grafi comessi ⇒ Numero esponenziale di vincoli!

$z_e \in \{0,1\}$

Per tutti i sub-set meno quello completo

Sottoinsieme diretto di vertici

Es. Non vogliamo la soluzione



Così ho 3 vertici = 3 lati! Non va bene ⇒ 3 vertici corrispondono

3! più  $= |S|-1$

$= |S|$

Vale per tutti i sub-set di nodi meno del tour completo

E' necessario rilassare i vincoli e poi dinamicamente aggiungere quelli che non vengono soddisfatti

Non ha senso doppizzarli ⇒ ottieniamo troppi moltiplicatori e non risolveremmo nulla

## Spanning Tree KRUSKAL'S ALGORITHM

NF

Risolve il minimo Spanning Tree, cioè grafo connesso, di complessità polinomiale  $\Rightarrow$  prende sempre i lati più corti se questi non creano un ciclo

$\hookrightarrow$  Ha questi vincoli impliciti sono quelli del problema

$$\sum_{e \in E(S)} z_e \leq |S|-1 \quad \forall S \subset V: 3 \leq |S| \leq |V|-1$$

Ha allora il problema mi sorgono sugli altri vincoli

$$\sum_{e \in \delta(v)} z_e = 2, \quad \forall v \in V$$

$\Rightarrow$  Rilassiamo allora questi vincoli scrivendo  $\sum_{e \in E} z_e = n$

Dualizziamo questi vincoli tranne per un vertice.

Così l'algoritmo A-TRDE forza i 2 lati incidenti solo per il 1° nodo

Il rilassamento lagrangiano prende il problema NP-H e lo porta in un problema di Spanning Tree con moltiplicatori di Lagrange (positivi o negativi) che mi portano a risolvere il problema  $\Rightarrow$  il connesso viaggiatore è stato svoltato così

$\hookrightarrow$  OTteniamo soluzioni approssimate = c'è il prezzo che paghi per passare da complessità NP-H a polinomiale

## Dual decomposition

- Dual decomposition may converge poorly in practice, but it might be a good approach for some specially structured large-scale problems.
- Sometimes, we are satisfied by a suitably good solution. If we may recover a good primal feasible solution from dual decomposition, we obtain dual heuristic algorithm.  
*Deve essere ammissibile*
- Lagrangian methods can be integrated with penalty function methods, resulting in augmented Lagrangian schemes based, e.g., on the minimization of

$$f(x) + \sum_{i \in I} \lambda_i h_i(x) + \sigma \sum_{i \in I} h_i^2(x)$$

*lagrange + funzione di penalità*      *Coefficiente di penalità*      *→ Schema lagrangiano Aumentato*

for an equality-constrained problem.

Consideriamo

$$\begin{aligned} \min \quad & f(z) \\ \text{h}(z) = 0 \quad & i \in I \end{aligned}$$

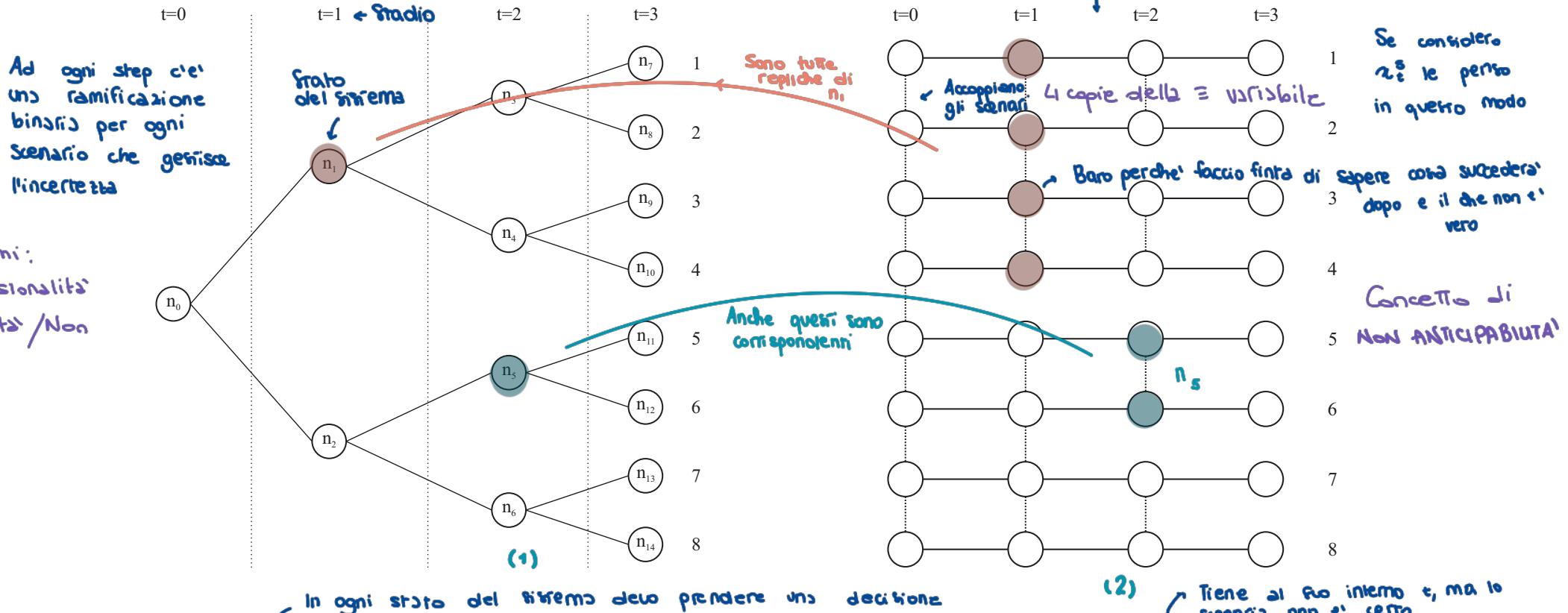
*Strategia per migliorare la convergenza*      *E' la funzione di penalità*  
*L'algoritmo ha tutte le ragioni di portare h il più possibile vicino allo 0*

## Progressive hedging for multistage stochastic programming

↳ Modo per decomporre problemi di prog. stocastica multistadio, uno per ogni scenario

Questa idea assume che tu abbia 12 sfere di cristallo

Uncertainty in multistage stochastic programming is represented by scenario trees:



- (1) • In a **compact model formulation**, we define decision variables  $x^n$  associated with node  $n$ .
  - ↳ In ogni stato del sistema devo prendere una decisione
- (2) • In a **split-variable model formulation**, we define decision variables  $x_t^s$ , associated with scenario  $s$  at time  $t$ .
  - ↳ Non è compatto
  - ↳ Ma questo mi porta a poter dividere in scenari
  - ↳ Tiene al suo interno  $t$ , ma lo scenario non è certo
  - ↳ Potrebbero essere variabili nel continuo
  - ↳ Devo aggiungere vincoli (per mantenere uguali delle variabili)
  - ↳ In questa formulazione ho più variabili
  - Nello scenario  $s$  al tempo  $t$  compio una certa azione

Voglio dei vincoli esplicativi per evitare di rendere certe delle assunzioni che faccio sul futuro e sui possibili sviluppi

In the second case, we need to enforce non-anticipativity constraints explicitly: decision variables corresponding to different scenarios at the same time  $t$  must take the same value, if the two scenarios are indistinguishable at time  $t$

In the case depicted in the figures, let us consider node  $n_1$ . Scenarios  $s = 1, 2, 3, 4$  pass through this node and are indistinguishable at time  $t = 1$ . Hence, we must have:

$$x_{i1}^1 = x_{i1}^2 = x_{i1}^3 = x_{i1}^4, \quad i = 1, \dots, I.$$

Le decisioni su un nodo radice per diversi scenari devono essere uguali  
Questo concetto è la Filtrazione

By the same token, at time  $t = 2$  we have constraints like

$$x_{i2}^5 = x_{i2}^6, \quad i = 1, \dots, I.$$

Definisco un sottoinsieme di scenari indistinguibili al tempo  $t$   
Cioè al tempo  $t$ , per esempio,  $s_1$  e  $s_2$  sono indistinguibili

More generally, it is customary to denote by  $\{s\}_t$  the set of scenarios which are not distinguishable from  $s$  up to time  $t$ . For instance:

$$\begin{aligned} \{1\}_0 &= \{1, 2, 3, 4, 5, 6, 7, 8\} && \rightarrow \text{e' come se } 1, \dots, 8 \text{ fossero 8 copie dello stesso nodo} \\ \{2\}_1 &= \{1, 2, 3, 4\} && \rightarrow \text{Al tempo 1 rispetto allo scenario 2 non distinguo i nodi 1, 2, 3, 4} \\ \{5\}_2 &= \{5, 6\}. \end{aligned}$$

Then, the non-anticipativity constraints may be written as

$$x_t^s = x_t^{s'} \quad \forall t, s, s' \in \{s\}_t.$$

Vincoli che accoppiano gli scenari  
Vincoli di anticipatività

This is not the only way of expressing the non-anticipativity requirement, and the best approach depends on the chosen solution algorithm.

Nel problema nel continuo lineare multistadio, non serve questa tecnica, se il problema non lineare e' utile  
 Proattivo  
 Utile per avere una buona eunistica duale

**Progressive hedging** is a strategy that can be applied to multistage, possibly non-linear stochastic programming problems.

Let us consider a dynamic system with state vector  $z_t$  and state equation  $z_{t+1} = G_t(z_t, x_t, \xi_{t+1})$ ,  $t = 0, 1, \dots, T$ , where  $z_0$  is given,  $x_t$  is the control variable, and  $\xi_{t+1}$  is a random variable realized after the decision at time  $t$ .

Se la dinamica  
varia nel tempo  
 $G_t$   
altrimenti si  
considera  $G$

Let  $s = (\xi_1^s, \xi_2^s, \dots, \xi_{T+1}^s)$  be a scenario, with probability  $\pi^s$ , and let us consider the individual scenario problem:

(\*) In un'altra tipologia di problemi +  
potrebbe essere proattivo

Costo/Guadagno  
traiettoria

Vogliamo ottimizzare dal tempo 0 al tempo  $T$

$$\min \sum_{t=0}^T \gamma^t f_t(z_t, x_t, \xi_{t+1}^s) + \gamma^{T+1} Q(z_{T+1})$$

s.t.  $z_{t+1} = G_t(z_t, x_t, \xi_{t+1}^s), \quad t = 0, 1, \dots, T$

$$L_t(z_t) \leq x_t \leq U_t(z_t), \quad t = 0, 1, \dots, T$$

Come si valuta lo stato finale?

Ultimo istante in cui prendo una decisione  
Ortante temporale dato ( $s$ )

Ultima realizzazione che posso guardare  
Costo allo Stato Terminale

Cosa faccio ( $z_t$ ), in quale stato ( $z_t$ )

Quello rappresenta lo stato terminale

Guardando

adesso, inizio a prendere la sequenza di decisione

Vincoli sui controlli delle equazioni dinamiche

Vincoli sulle decisioni

$x_t$  := Variabile di controllo

$z_t$  := Variabile di stato

with terminal state cost  $Q(\cdot)$  and discount factor  $\gamma \in (0, 1]$ .

Let  $(x_0^s, x_1^s, \dots, x_T^s)$  be the optimal solution for the scenario problem  $s$ .

Can we just aggregate the individual solutions and choose  $\underline{x}_t = \sum_{s \in S} \pi^s x_t^s$ ? unfortunately, there is no reason to believe that this would be the optimal solution (or even a feasible one, for that matter).

A good policy should take advantage of information, but it cannot be anticipative. We can formulate the multistage problem within a split-variable modeling framework, introducing nonanticipativity constraints in a suitable form.

Let  $\{s\}_t$  be the set of scenarios that, up to time instant  $t$ , cannot be distinguished from scenario  $s$ , and let  $\mathbb{P}(\{s\}_t)$  be the sum of their probabilities.

The stochastic optimization problem can be stated as:

Mi piacerebbe dividere e decomporre rispetto agli scenari.

$$\begin{aligned}
 \min \quad & \sum_{s \in \mathcal{S}} \pi^s \left( \sum_{t=0}^T \gamma^t f_t(\mathbf{z}_t^s, \mathbf{x}_t^s, \boldsymbol{\xi}_{t+1}^s) + \gamma^{T+1} Q(\mathbf{z}_{T+1}^s) \right) \xrightarrow{\text{Aggiro gli scenari per ottenere il valore atteso}} \\
 \text{s.t.} \quad & \mathbf{z}_{t+1}^s = G_t(\mathbf{z}_t^s, \mathbf{x}_t^s, \boldsymbol{\xi}_{t+1}^s), \quad \forall s \in \mathcal{S}, t = 0, 1, \dots, T \\
 & \mathbf{L}_t(\mathbf{z}_t^s) \leq \mathbf{x}_t^s \leq \mathbf{U}_t(\mathbf{z}_t^s), \quad \forall s \in \mathcal{S}, t = 0, 1, \dots, T \\
 & \mathbf{x}_t^s = \sum_{s' \in \{s\}_t} \frac{\pi^{s'} \mathbf{x}_t^{s'}}{P(\{s\}_t)}, \quad \forall s \in \mathcal{S}, t = 0, 1, \dots, T \xrightarrow{\substack{\text{Probabilità} \\ \text{Variabili decisionali}}} \xrightarrow{\substack{\text{Per evitare l'anticipatività}}} \xrightarrow{\text{Vincoli di non anticipatività}}
 \end{aligned} \tag{4}$$

↳ Valore atteso condizionale, si sommano gli scenari indistinguibili = e' una proiezione nello spazio delle politiche non anticipative

Constraints (4) enforce nonanticipativity: each decision in  $\{s\}_t$  must be the same at time  $t$ . Here we use a single conditional expectation per scenario and time period, rather than pairwise equalities between decisions  $\{s\}_t$ . Let us introduce Lagrange multipliers  $w_t^s$  for nonanticipativity constraints.

↳ Set di scenari che al tempo  $t$  non sono distinguibili

Then, dualization of (4) yields:

$$\min \sum_{s \in \mathcal{S}} \pi^s \left\{ \sum_{t=0}^T \gamma^t [f_t(\mathbf{z}_t^s, \mathbf{x}_t^s, \boldsymbol{\xi}_{t+1}^s) + w_t^s (\mathbf{x}_t^s - \underline{\mathbf{x}}(\{s\}_t))] + \gamma^{T+1} Q(\mathbf{z}_{T+1}^s) \right\}$$

Vengono incrementati o decrementati in base a  $\mathbf{z}_t^s - \underline{\mathbf{x}}(\{s\}_t) = 0$

Definisco un moltiplicatore di Lagrange per ogni vincolo di non anticipatività

↳ Dualizzato il vincolo che legava le variabili

where the conditional expectation

$$\underline{\mathbf{x}}(\{s\}_t) \equiv \sum_{s' \in \{s\}_t} \frac{\pi^{s'} \mathbf{x}_t^{s'}}{P(\{s\}_t)}$$

→ Valore atteso condizionale per il set di scenari indistinguibili

can be interpreted as a projection on the set of nonanticipative policies.

- Per problemi II non si giungerà a convergenza perché non vale la dualità forte (non convessa)
- In generale, in pratica non converge questo metodo anche se in teoria dovrebbe convergere

Il trucco per stabilizzare l'algoritmo e' ancorarsi all'iterazione precedente tra la soluzione dei problemi duali - primali precedenti  
Si risolvono i problemi per scenario.

This problem can be rearranged and decomposed by scenarios. In order to improve convergence, we may add a quadratic penalty term to yield an augmented Lagrangian method.

Unfortunately, the penalty term would destroy the decomposition structure.

However, we may use the scenario-aggregated solution  $\underline{x}(\{s\}_t)$  from the previous iteration, which yields (neglecting constant terms), the individual scenario problem:

$$\min \sum_{t=0}^T \gamma^t \left\{ f_t(z_t^s, x_t^s, \xi_{t+1}^s) + w_t^s x_t^s + \frac{1}{2} \rho [x_t^s - \underline{x}(\{s\}_t)]^2 \right\} + \gamma^{T+1} Q(z_{T+1}^s)$$

Prendiamo questo all'iterata precedente  
Somma dei vincoli che ho rilassato, vorrei che fosse una funzione di penalità su un vincolo di uguaglianza  
'e una costante  
Si verificano accoppiamenti di scenari

where the quadratic term may be interpreted as a regularization term. Non vogliamo riscapillare per gli scenari  
Per migliorare la convergenza aggiungo penalità

The choice of  $\rho$  and of the multiplier adjustment scheme are quite critical for convergence.

Nevertheless, progressive hedging is useful in order to take advantage of problem structure, and it has been used to devise heuristics for hard mixed-integer, multistage stochastic programs.

Tipico modo di usare questi tipi di algoritmi  
Quando si stabilizzano le variabili binarie, le si pone nel problema come costanti e si risolve poi il problema nel continuo

Consideriamo la variabile di stato  $z_t$  che è la popolazione di pesci

Mentre la variabile  $t$  ( $0 \leq t \leq 1$ ) rappresenta quante aringhe uccidi, è la variabile di controllo

Quante aringhe al tempo  $t$ ?  $z_{t+1} = z_t - \bar{z}_t^2 + p z_t$

quante ne hai pescate capacità dell'ambiente di sopportare le aringhe, cioè quanti pesci possono esistere nel sistema

quanto la popolazione aumenta o decrece (nascita e morti)

Si noti che  $z_t$  non è del tutto osservabile ma noi la consideriamo osservabile

Notiamo che  $p$  sarebbe più significativo un  $p$  che varia nel tempo

La funzione obiettivo risulta quindi  $\max \sum_{t=0}^T r^t z_t$  fattore di sconto che tendenzialmente fa convergere in maniera migliore il problema

Domanda. Come si fa a valutare lo stato finale  $z_{T_1}$ ? Bisogna mettere un vincolo o una funzione obiettivo per evitare di «finire» tutto prima del tempo  $T_1$ .

Cosa possiamo fare per troncare bene l'orizzonte? Devo aggiungere un petto per non sterminare le aringhe, sul lungo periodo assumo  $\bar{p}$  una costante

Si ipotizzi un valore  $z_{T_1}$  in modo tale che sia costante da lì in avanti, cioè la  $z_t$  è costante

Aiuterà  $z_t = \bar{p} \left(1 - \frac{z_{T_1}}{K}\right)$ , cioè esattamente di quanto cresce la popolazione. Così si ottiene una soluzione periodica  $t > T_1$

E' necessario introdurre la funzione  $a(z_{T_1}) = \sum_{t=T_1}^{\infty} r^{t-T_1} z_t z_t = \bar{p} z_{T_1} \left(1 - \frac{z_{T_1}}{K}\right) \frac{1}{1-r}$

serie geometrica

Si è poi applicato l'algoritmo appena descritto per risolvere questo problema

Questo problema ha la struttura  $A = \begin{bmatrix} c_1 & c_2 & \dots & c_n \\ d_1 & d_2 & \dots & d_n \end{bmatrix}$

## Part 2

Dividere tra le variabili continue  
e quelle intere

Questa idea è utilizzabile anche  
a problemi multiraggi

### Benders decomposition and L-shaped decomposition for stochastic programming

Consideriamo vincoli del tipo  $Az + By = b$  con  $z$  intera e  $y$  continuo  
come progettare la rete  
come fai viaggiare il flusso

Studiamo ora problemi con la struttura

$$A = \begin{bmatrix} c_1 & D_1 \\ c_2 & \ddots \\ \vdots & \\ c_n & D_n \end{bmatrix}$$

Variabili al 1° livello che accoppiano gli scenari  
Variabili al 2° livello

Riformulazione epigrafica. Si riscrive il problema di ottimizzazione sfruttando l'epigrafo della funzione

$$\min_{\mathbf{x} \in S} f(\mathbf{x}) \Rightarrow \min_{\mathbf{x} \in S} g^* \text{ - coincide con il subgradiente}$$

$\Omega^* f(\mathbf{x}) \leftarrow$  iper piano di supporto che sono funzioni affini

$\Rightarrow$  "problema approssimato diventa"

$\min_{\mathbf{x} \in S} u'$   
 $\Omega^* \alpha_k + \gamma_k^T \mathbf{x} \leq u' \quad \forall k \in \{1, \dots, m\}$

## Background: Kelley's cutting planes

Supponiamo i vincoli sulla regione di ammissibilità che sono lineari

Funzione convessa, ma non lineare

Consider the convex problem  $\min_{\mathbf{x} \in S} f(\mathbf{x})$ , and suppose that, for a given point  $\mathbf{x}^k$ , we are not only able to compute the function value  $f(\mathbf{x}^k) = \alpha_k$ , but also a subgradient  $\gamma_k$ , which does exist if the function is convex on the set  $S$ .

Nessuno assicura che la funzione sia derivabile

In other words, we are able to find an affine function such that

Definizione di iper piano di supporto  $\left\{ \begin{array}{l} f(\mathbf{x}^k) = \alpha_k + \gamma_k^T \mathbf{x}^k \\ f(\mathbf{x}) \geq \alpha_k + \gamma_k^T \mathbf{x} \end{array} \right.$  → Riesco ad approssimare con una funzione affine (5)

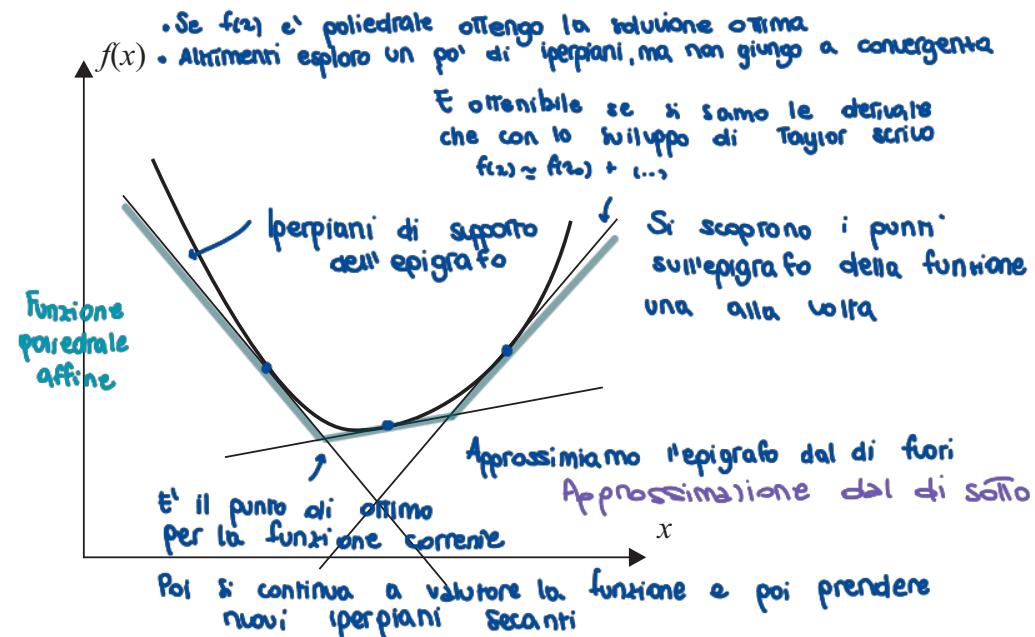
$$\forall \mathbf{x} \in S. \quad (6)$$

The availability of such a support hyperplane suggests the possibility of approximating  $f$  from below, by the upper envelope of support hyperplanes.

The Kelley's cutting plane algorithm exploits this idea by building and improving a lower bounding function until some convergence criterion is met.

If  $S$  is polyhedral, we have to solve a sequence of LPs.

Se la funzione non è differenziabile ⇒ Sub gradiente



## Kelley's cutting planes

*Soluzione iniziale ammmissibile*

**Step 0.** Let  $x^1 \in S$  be an initial feasible solution; initialize the iteration counter  $k \leftarrow 0$ , the upper bound  $u_0 = f(x^1)$ , the lower bound  $l_0 = -\infty$ , and the lower bounding function  $\beta_0(x) = -\infty$ .

**Step 1.** Increment the iteration counter  $k \leftarrow k + 1$ . Find a subgradient of  $f$  at  $x^k$ , such that equation (5) and condition (6) hold.

**Step 2.** Update the upper bound

$$u_k = \min\{u_{k-1}, f(x^k)\}$$

and the lower bounding function

$$\beta_k(x) = \max\{\beta_{k-1}(x), \alpha_k + \gamma_k^T x\}. \rightarrow \text{Funzione che dà il lower bound}$$

*Cioè si aggiunge un altro iperpiano una volta valutata la differenza tra lower e upper bound*

**Step 3.** Solve the problem  $l_k = \min_{x \in S} \beta_k(x)$ , and let  $x^{k+1}$  be the optimal solution.

**Step 4.** If  $u_k - l_k < \epsilon$ , stop:  $x^{k+1}$  is a satisfactory approximation of the optimal solution; otherwise, go to step 1. *→ quindi calcola un altro iperpiano*

*L'algoritmo si ferma quando il gap è piccolo  
tra lower e upper bound della funzione  
iperpiani funzioni*

*(\*) Se fisso un valore della variabile al primo  
stadio, non è poi sempre vero che c'è una  
soluzione ammmissibile al secondo stadio*

Si puo' dimostrare anche senza porti in un caso «facile» come questo, cioe' con la regola del duale deterministica  
L'idea principale e' passare al duale  $\rightarrow$  la funzione che voglio approssimare con i tagli e' la funzione di ricorso

## Convexity properties of stochastic programs

Let us consider the two-stage SLP with recourse

$$\begin{aligned} \min \quad & c^T x + \mathbb{E}_{\xi} [Q(x, \xi)] \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

↗ Costo deterministico al primo stadio  
 ↗ Variabile decisionale al primo stadio  
 ↘ Costo atteso  
 ↗  $c^T$  definita attraverso un problema di minimo

$\Rightarrow$  Puo' essere un integrale nel continuo o discretizzato per esempio.  
al secondo stadio

Il problema si dice a ricorso completo  
se al secondo stadio non si verifica mai la non ammissibilita'.  
Si verifica se sono presenti vincoli deboli/rilassati.

where

$$Q(x, \xi) \equiv \min_y \{ q(\xi)^T y \mid W(\xi)y = h(\xi) - T(\xi)x, y \geq 0 \}.$$

We speak of fixed recourse when  $W$  deterministic, and of random recourse in general.

Let us consider the recourse function  $Q(x) \equiv \mathbb{E}[Q(x, \xi)]$  in the case of deterministic recourse cost:

$$Q(x, \xi) \equiv \min_y \{ q^T y \mid Wy = h(\xi) - T(\xi)x, y \geq 0 \}$$

↗ Le supponiamo deterministiche - esempio costi e macchinari deterministici  
 ↗ Consideriamo questo come il problema primale  
 ↗ Le assumiamo non deterministiche - esempio incertezza sulla domanda

The effective domain of  $Q(x)$  consists of vectors  $x$  that are feasible for the first-stage constraints and such that  $Q(x) < +\infty$ , i.e., the second stage problem is (almost surely) feasible.

Note: we assume that, barring second-stage infeasibility, the above expectation is always defined, i.e., random variables are not heavy-tailed.

→ Non consideriamo code pesanti perché vogliamo che il valore atteso sia definito

Visto che l'incertezza e' solo qui, allora la regione di ammissibilita' del problema duale e' deterministica

By LP duality, we have  $Q(x, \xi) = \max_{\pi} \{ [h(\xi) - T(\xi)x]^T \pi \mid W^T \pi \leq q \}$ .  $\rightarrow$  Consideriamo la versione duale del problema.

Let us denote the feasible set of the dual by  $\Pi = \{ \pi \mid W^T \pi \leq q \}$ . Note that, since  $q$  is deterministic, that this set is nonrandom.  $\hookrightarrow$  e' un poliedro deterministico e non dipende da  $\xi$

Let us consider  $x^1, x^2 \in \mathcal{D}$ ,  $\lambda \in [0, 1]$ , and  $x_\lambda = \lambda x^1 + (1 - \lambda)x^2$ .

Voglio dimostrare che  $Q(x_\lambda)$  e' una funzione convessa, non implica differentiabile

Funzione di ricorso valutata nel punto  $x_\lambda$

$$Q(x_\lambda) = \int_{\Xi} Q(x_\lambda, \xi) dP$$

Sostituisco la definizione di  $Q$  come problema di ottimizzazione al secondo studio nella versione duale

$$= \int_{\Xi} \max_{\pi \in \Pi} \{ [h(\xi) - T(\xi)x_\lambda]^T \pi \} dP$$

$\hookrightarrow$  Supporto della distribuzione delle variabili casuali  $\xi$   
Un problema e' che in generale non si conosce questa misura di probabilita'

$$= \int_{\Xi} \max_{\pi \in \Pi} \left\{ \lambda [h(\xi) - T(\xi)x^1]^T \pi + (1 - \lambda) [h(\xi) - T(\xi)x^2]^T \pi \right\} dP$$

Notiamo che  $x_1, x_2$  non hanno impatti sulla regione di ammissibilita'

Si spettano le due massimizzazioni  
Da cui, avendo piu' liberta', il massimo del problema unito e' minore del massimo per i due problemi separati

$$\leq \lambda \int_{\Xi} \max_{\pi \in \Pi} \{ [h(\xi) - T(\xi)x^1]^T \pi \} dP$$

$$+ (1 - \lambda) \int_{\Xi} \max_{\pi \in \Pi} \{ [h(\xi) - T(\xi)x^2]^T \pi \} dP$$

$$= \lambda Q(x^1) + (1 - \lambda) Q(x^2) \quad \rightarrow \text{Definizione di funzione convessa}$$

## Convexity of the recourse function

---

The previous proof is rather simple, because the dual feasible region is nonrandom. Actually, convexity of the recourse function can be shown in a more general setting.

To cut a long story short (and cutting a few technical corners along the way...):

- The recourse function is typically differentiable for continuous probability distributions. → Quando le variabili casuali hanno una distribuzione nel continuo siamo in questa situazione
- The recourse function is polyhedral for discrete probability distributions.

In both cases, we may rely on Kelley's cutting planes to solve the problem.

The bonus, given problem structure, is that we may find cutting planes by a decomposition strategy.

## L-shaped decomposition

Consider the deterministic equivalent:

$$\begin{aligned}
 \min \quad & \mathbf{c}^T \mathbf{x} + \sum_{s \in S} p^s \mathbf{q}_s^T \mathbf{y}^s \\
 \text{s.t.} \quad & \mathbf{A} \mathbf{x} = \mathbf{b} \\
 & \mathbf{W} \mathbf{y}_s + \mathbf{T}_s \mathbf{x} = \mathbf{h}_s \quad \forall s \in S \\
 & \mathbf{x}, \mathbf{y}_s \geq \mathbf{0}
 \end{aligned}$$

Consideriamo diversi scenari; quindi abbiamo ottenuto un problema deterministico  
 Molti scenari differenti per avere tante opzioni da prendere  
 in considerazione

The technological matrix of the overall problem has a dual block angular structure:

$$\left[ \begin{array}{ccccc}
 \mathbf{A} & 0 & 0 & \cdots & 0 \\
 \mathbf{T}_1 & \mathbf{W} & 0 & \cdots & 0 \\
 \mathbf{T}_2 & 0 & \mathbf{W} & \cdots & 0 \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 \mathbf{T}_S & 0 & 0 & \cdots & \mathbf{W}
 \end{array} \right]$$

Primo stadio in cui fissiamo e risolvendo il problema master  
 Problemi al secondo stadio che possiamo suddividere

For a given first-stage, we get independent second-stage problems (dual structure of the one used in Lagrangian decomposition).

Let us rewrite the two-stage LP as:

Problema master rilassato  
e' pilota }  
non scrivo gli oo vincoli nel caso in cui la funzione non fosse poliedrale

$$\begin{aligned} \min \quad & c^T x + \theta \\ \text{s.t.} \quad & Ax = b \\ & \theta \geq Q(x) \\ & x \geq 0. \end{aligned}$$

NB Non c' è sempre detto che al 2° stadio sia sempre tutto ammissibile

Riscrittura del modello, non approssimazione

$\rightarrow Q(x) \rightarrow \infty$  se non ci sono più soluzioni ammissibili (7)

Di questa funzione sappiamo solo dire che è convessa.

↳ Se risolvo un rilassamento ott

We form a relaxed master problem by relaxing the constraint (7), which is approximated by cutting planes:

Rilassiamo il problema tramite dei tagli che dinamicamente vado ad aggiungere andando ad approssimare sempre meglio la funzione di ricorso

- Optimality cuts of form  $\theta \geq \alpha^T x + \beta$ .

→ Dalla funzione riescono a ricavare gli iperplani di supporto

- Feasibility cuts of form  $0 \geq \alpha^T x + \beta$ .

→ I tagli gli ottengo risolvendo i problemi al secondo grado

The coefficients of each cut are obtained by solving the scenario subproblems for given first-stage decisions.

- Ozioni dualità →
- Due soluzioni =
  - Unbounded
  - Nessuno dei due è ammissibile

## L-shaped decomposition: Optimality cuts

→ E' fissato ed e' una proposta di soluzione rilassata al primo stadio

Let  $\hat{x}$  be the optimal solution of the initial master problem and consider the dual of the second-stage problem for scenario  $s$ :

- Il problema e' che questo potrebbe andare ad  $\infty$
- Ha se il problema di minimo non ha soluzione, allora questo problema  $Q_s(\bar{z})$  puo' essere ammissibile o non ammissibile (n)

$$Q_s(\hat{x}) \equiv \max \quad (\mathbf{h}_s - \mathbf{T}_s \hat{x})^\top \boldsymbol{\pi}_s$$

↓  
Non sono presenti le variabili  $y$  del secondo stadio perché per i tagli servono i  $\Pi$  Duali di min-problemi.

s.t.  $\mathbf{W}^\top \boldsymbol{\pi}_s \leq \mathbf{q}_s$ .

Risolviamo tanti problemi (1 per scenario) indipendenti  
Ottimizzo rispetto a queste variabili duali  
→ La  $\bar{z}$  entra in gioco quando si sceglie la direzione  
Se il problema e' non ammissibile (8)  
allora la regione di ammissibilita' di questo problema e' vuota (perche' non c'e' z nei vicoli)

Given an optimal dual solution  $\hat{\boldsymbol{\pi}}_s$ , it is easy to see that the following relationships hold:

Bisogna dunque accettare che primale non ammissibile, duale unbounded

Disegualanza che mi serve per trovare l'iperpiano di supporto

Non puo' essere meglio dell'ottimo ↗

Idea su cui si basano gli iperpiani di supporto

$$Q_s(\hat{x}) = (\mathbf{h}_s - \mathbf{T}_s \hat{x})^\top \hat{\boldsymbol{\pi}}_s \rightarrow \text{Uguaglianza che vale per un } \bar{z} \text{ fissato} \quad (9)$$

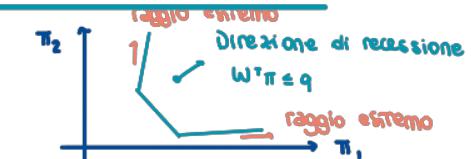
$$Q_s(x) \geq (\mathbf{h}_s - \mathbf{T}_s x)^\top \hat{\boldsymbol{\pi}}_s \quad \forall x. \rightarrow \text{Rilassa l'ugualanza che ora vale per } \bar{z} \text{ generici} \quad (10)$$

The inequality (10) derives from the fact that  $\hat{\boldsymbol{\pi}}_s$  is the optimal dual solution corresponding to  $\hat{x}$ , but not to a generic  $x$ .

Summing (10) over the scenarios, we get

$$Q(x) = \sum_{s \in S} p_s Q_s(x) \geq \sum_{s \in S} p_s (\mathbf{h}_s - \mathbf{T}_s x)^\top \hat{\boldsymbol{\pi}}_s.$$

Subgradiente



Il poliedro non e' limitato se ci sono direzioni di recessione che formano un cono

Hence, we may add the following optimality cut to the relaxed master

$$\theta \geq \sum_{s \in S} p_s (\mathbf{h}_s - \mathbf{T}_s x)^\top \hat{\boldsymbol{\pi}}_s. \rightarrow \text{E' il taglio di ottimalita' che viene aggiunto al master}$$

Dal 2° stadio fisso questa variabile

## L-shaped decomposition: Feasibility cuts

If the recourse is not complete, some of scenario subproblems may be infeasible for a certain first-stage decision  $\hat{x}$ . In this case, we may again exploit the dual of the scenario subproblem to find a feasibility cut, eliminating  $\hat{x}$  from further consideration.

Note that the feasibility region of this dual does not depend on the first-stage decisions, since  $\hat{x}$  does not occur in constraints (8). Thus, if a dual problem is infeasible, it means that the second-stage problem for the corresponding scenario will be infeasible for any first-stage decision.

Ruling out this case, which is likely to be due to a modeling error, when the primal problem is infeasible, the dual will be unbounded.

Hence, there is an extreme ray of the dual feasible set along which the optimal dual solution goes to infinity. → Le direzioni estreme sono combinazioni lineari dei raggi estremi

If we consider the dual of the second-stage subproblem, we see that it is unbounded if we find dual variables  $\pi^*$  such that

Caratterizza le direzioni di recessione che sono ammissibili  $\rightarrow W^T \pi^* \leq 0$ ,  $(h_s - T_s \hat{x})^T \pi^* > 0$

• Direzione di recessione lungo questa direzione l'obiettivo sale  
 lungo cui l'obiettivo aumenta

We may derive a condition on  $x$ , preventing this to occur. Primal feasibility requires

$$W y = h_s - T_s x \Rightarrow \underbrace{\pi^{*T} W}_{\leq 0} \underbrace{y}_{\geq 0} = \pi^{*T} (h_s - T_s x) \leq 0$$

Taglio non aggregato sugli scenari.  $\hookrightarrow$  Vincolo che collega le variabili del 2° stadio con quelle al primo stadio

In questo modo escludo la condizione precedente (b)

Hence  $\pi^{*T} (h_s - T_s x) \leq 0$  is a valid cut that is added to the relaxed master.



## L-shaped decomposition

---

L-shaped decomposition iterates between:

1. the solution of the relaxed master problem, which yields  $\hat{\theta}$  and  $\hat{x}$ ; → Risolvo il problema rilassato e ottengo un lower bound dei costi del secondo livello
2. the solution of the corresponding scenario subproblems.

At each iteration, cuts are added to the master problem.

The algorithm stops when the optimal solution of the master problem satisfies

$$\hat{\theta} \leq \underline{Q}(x).$$

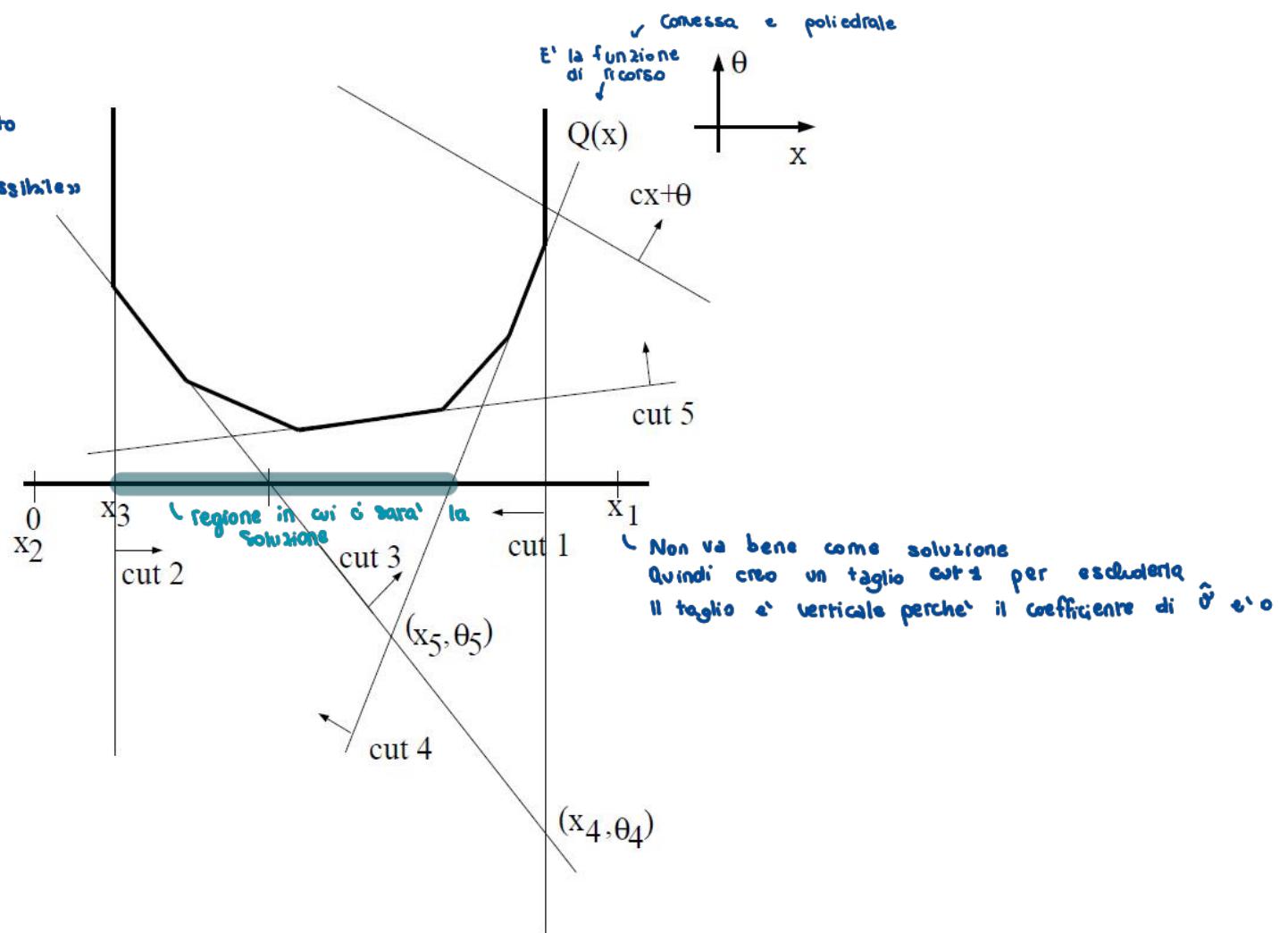
Sempre verificata, non puo' essere la stop condition

$$\underline{Q}(\underline{x}) - \hat{\theta} < \epsilon \quad \text{con } \epsilon \text{ piccolo}$$
$$\hat{\theta} \geq \overline{Q}(\hat{x})$$

This condition may be relaxed if a near-optimal solution is good enough for our purposes.

An illustration, from (Kall, Wallace, 1994)

Con un numero finito di scenari, in tempo finito  
si raggiunge una soluzione  
Anche se «infinito» non implica «ammisibilità»  
per noi



## L-shaped decomposition: variations on the theme

---

- There are variants of L-shaped decomposition to manage cuts and to improve convergence (e.g., regularized L-shaped decomposition).  
→ Ci sono trucchi per rendere più stabile e per farlo andare a convergenza
- The idea can be generalized to multi-stage problems:
  - nested Benders decomposition
  - abridged nested Benders decomposition
  - stochastic dual dynamic programming → Vis di meno tra campionare e ottimizzare insieme
- In stochastic decomposition methods (Higle and Sen), there is an interplay between sampling and optimization: the idea is to build asymptotically valid cuts.
- Alternative strategies rely on bounding the recourse function from above and below, partitioning the domain in order to get tight bounds.
- When simplex or interior point methods manage to solve a stochastic LP, there is usually no advantage in using decomposition; however, decomposition is needed for really large-scale problems.

- Ripassare →
- Base
  - Costo ridotto
  - Simplex
  - Teorema del poliedro
  - Ricopertura convessa + corp

la generazione di colonna

Si usa per problemi di routing di veicoli o di trasporto

- Negli algoritmi di generazione di colonna aggiungo nuove colonne, senza togliere mai
- Negli algoritmi come il simplex, solitamente si sostituiscono le nuove colonne a quelle precedenti meno significative

## Part 3

convessi

In questo caso l'idea è di approssimare gli insiemi dal di dentro → additive manufacturing

# Dantzig–Wolfe decomposition and column generation

! e' l'ideatore dell'algoritmo del simplex

↳ Generiamo dinamicamente solo le colonne di cui hai realmente bisogno

L'idea è utile quando si hanno un gran numero di variabili, che sono le colonne, e un numero minore di vincoli

Supponiamo vincoli di uguaglianza  $Az=b$



Saranno solo 10.000 colonne, non tutte quelle che realmente ho e mi servono per creare la base della soluzione del simplex

Problema: bisogna tagliare dei rotoli  
e vuoi utilizzare uno schema  
di taglio che, seguendo la domanda,  
si abbia residuo/specie minore

## Cutting stock by column generation

→ Versione accademica del problema

→ Li ipotizziamo tutti uguali

We must cut rolls (all with width  $L$ ) of some material into shorter rolls, in order to satisfy a given demand. Say that we introduce a set of  $n$  items, featuring demand  $d_i$  and width  $w_i \leq L$ ,  $i \in [n]$ . Our aim is to pack items in such a way that we use the minimum amount of rolls.

↳ Ogni item ha una sua certa lunghezza / larghezza  
in base alla direzione di taglio che si vuole scegliere

This is a stylized example of a large class of problems known under the common umbrella of cutting-stock problems.

Say that  $m$  rolls are available, indexed by  $j \in [m]$ , and let us introduce decision variables:

↳ li ipotizziamo tutti uguali

↳ Indiciamo i rotoli

- $\delta_j \in \{0, 1\}$ , set to 1 if we use roll  $j$ ;

- $x_{ij} \in \mathbb{Z}_+$ , the number of items  $i$  cut from roll  $j$ . → Quanti oggetti di tipo  $i$  si ricavano dal rotolo  $j$

Then, we could consider the following MILP model:

Questa formulazione prende in considerazione tutte le possibili permutazioni perché considera i rotoli come diversi e distinguibili, ma sono tutti uguali!

Ovvero: ad esempio il branch and bound percorre tutti i possibili percorsi andando poi però a trovare tutti otimi equivalenti

=> Problema di simmetria, bisogna riformulare

il modello con la generazione di colonne

$$\min \sum_{j \in [m]} \delta_j \quad \rightarrow \text{Vogliamo minimizzare il numero di rotoli da utilizzare}$$

$$\text{s.t. } \sum_{i \in [n]} w_i x_{ij} \leq L \delta_j, \quad j \in [m] \quad \rightarrow \text{Per ogni rotolo quanti piatti tagli}$$

$$\sum_{i \in [n]} x_{ij} = d_i, \quad i \in [n] \quad \rightarrow \text{Soddisfacimento della domanda}$$

=> Un vincolo da aggiungere per evitare le simmetrie è  $\delta_{j_1} = \delta_j$

Vincolo della grande M, bound deboli

$$\delta_j \in \{0, 1\}, \quad x_{ij} \in \mathbb{Z}_+, \quad i \in [n], j \in [m].$$

While there is nothing theoretically wrong with the model, it suffers from a few difficulties related with big- $M$  constraints and symmetry issues (multiple optima may have a detrimental effect on commercial solvers).

Combiamo  
Totale  
mente  
la prospettiva

Se usi la  
fucetta « $K_n$   
quanti petti  
di « $i$ » ricavi

→ Supponiamo che qualcuno ci dia già pronti degli schemi di taglio che dice quanti petti produci con un determinato rotolo  
We may resort to a different modeling framework based on a set of prespecified cutting patterns. Each cutting pattern, indexed by  $k$ , is associated with the number  $a_{ik}$  of items  $i$  that are obtained from the roll. Given a set  $\mathcal{K}$  of cutting patterns, we may introduce the integer number  $\gamma_k$  or rolls cut according to that pattern, and solve the problem:

Vantaggio → Non c'è il vincolo della grande  $M$   
Non soffre di simmetrie  
E' un problema di set covering con variabili intere

Nel «problema vero» c'è il collegamento con il magazzino e i petti tagliati di scarto possono essere recuperati mandandoli in fonderia

$$\begin{aligned} \min \quad & \sum_{k \in \mathcal{K}} \gamma_k && \xrightarrow{\text{quante volte applichi lo schema di taglio } k} \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}} a_{ik} \gamma_k = d_i, \quad i \in [n] && \xrightarrow{\text{vantaggio: la cardinalità di } \mathcal{K} \text{ è molto grande}} \\ & \gamma_k \in \mathbb{Z}_+, \quad k \in \mathcal{K}. && \xrightarrow{\geq \text{formulazione equivalente}} \end{aligned}$$

Qui non ci sono più problemi  
di simmetria

This solves the symmetry issue, but the cardinality of set  $\mathcal{K}$  is huge. In fact, only a limited number of cutting patterns are really useful, but there is no obvious way to restrict set  $\mathcal{K}$  a priori.

Thus, we may start with a limited number of patterns, and dynamically generate cutting patterns, following a column generation scheme.

## Algoritmo del Simplex

Consideriamo il problema di minimo.

$$\min c^T z \quad \begin{matrix} \text{valori di } c^T \text{ in base} \\ C_B^T z + C_N^T z_N \end{matrix}$$

$$A z = b \rightarrow A_B z_B + A_N z_N = b \quad \begin{matrix} \text{valori di } c^T \text{ non in base} \\ \approx 0 \end{matrix}$$

Quindi si ottiene,  $z_B = A_B^{-1}(b - A_N z_N) = A_B^{-1}b - A_B^{-1}A_N z_N$ . Da qui e' inseribile il partizionamento, oltre dei vincoli, anche della funzione obiettivo.

tra base e fuori base, cioe':  $C_B^T(A_B^{-1}b - A_B^{-1}A_N z_N) + C_N^T z_N = C_B^T A_B^{-1}b + (C_N^T - C_B^T A_B^{-1}A_N)z_N$

$\hookrightarrow$  e' il pricing di una colonna

$\hat{c}_N^T$  e' il valore dei costi ridotti che descrive il cambiamento della funzione obiettivo se vado a portare nuovi variabili in base

$\hookrightarrow$  Comincia al vertice delle variabili duali corrispondenti a quella base Vengono forniti dall'algoritmo del Simplex

Test di ottimalita'. Ci dice se  $\hat{c}_j < 0$ ,  $j \in N$  oppure no. Se la soluzione non e' ottima trovi una variabile per cui  $\hat{c}_j < 0$

Quindi potrai portare la variabile "j" in base e ne devi togliere una (che devi scegliere in qualche modo).

Per capire se una colonna e' interessante  $\Rightarrow C_j - \pi^T A_j < 0$

## Generating a new column by pricing

Let us consider an LP-relaxation of a restricted version of the cutting stock model:

$$\begin{aligned}
 \min \quad & \sum_{k \in \mathcal{K}'} \gamma_k \\
 \text{s.t.} \quad & \sum_{k \in \mathcal{K}'} a_{ik} \gamma_k = d_i, \quad i \in [n] \\
 & \gamma_k \geq 0, \quad k \in \mathcal{K}' ,
 \end{aligned} \tag{11}$$

*↳ E' un problema a numeri interi*

where  $\mathcal{K}' \subset \mathcal{K}$ . How can we improve the solution by introducing a new column?

Let  $\pi_i$ ,  $i \in [n]$ , denote the dual variables of constraints (11). We know from the theory behind the primal simplex algorithm that we should bring into the basis a variable with negative reduced cost.

In our case, the reduced cost for an alternative pattern  $k$  is

$$\hat{c}_k = 1 - \sum_{i \in [n]} \pi_i a_{ik},$$

OCCHIO! Questi coefficienti non li hai, ma proprio qui il problema  
 di generazione di colonna.  
 → Vado a trovare dei coefficienti di  $a_{ik}$  t.c. È u  
 il più negativo possibile, cioè vado a minimizzare

Io voglio generare  
 dinamicamente le colonne

and we should find the column  $q$  with minimum reduced cost,  $\hat{c}_q = \min_k \hat{c}_k$ . If  $\hat{c}_q \geq 0$ , then the current basis is optimal. Otherwise, we introduce the non-basic variable  $\gamma_q$  into the basis and we iterate by pivoting to another basis.

Considero un sottoinsieme degli schemi di taglio. Ogni schema e' associato a una colonna della matrice

→ Generare una colonna corrispondente allora ad aggiungere uno schema di taglio

Tirare fuori un sottoinsieme di colonne, anche se non furbissimo, e generare una colonna per ogni oggetto, cioè una matrice diagonale / Schemi di taglio

NB Rilassiamo il problema nel continuo

Se parto con un K trovato con euristiche si faranno meno giri di formulazioni di colonna

NP-Hard

preddi →	<table border="1"><tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr><tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr></table>	:	:	:	:	:	:	-	-	-	-	-	-	-	-	-	-	-	-
:	:	:	:	:	:														
-	-	-	-	-	-														
-	-	-	-	-	-														

→ Non e' solo binario, hai piu' repliche dello stesso oggetto

The minimization can be accomplished by solving the integer knapsack problem

Problema di generazione di colonna

Supponiamo che tagliate seguendo uno schema o un altro  
abbia sempre lo stesso costo

$$\begin{aligned} \max \quad & \sum_{i \in [n]} \pi_i y_i && \text{E' il beneficio di inserire un petto di tipo } i \text{ come schema di taglio} \\ \text{s.t.} \quad & \sum_{i \in [n]} w_i y_i \leq L && \text{e' il prezzo ombra, e' la priorita' del petto} \\ & y_i \in \mathbb{Z}_+ && \text{Co' dipende da quanto quel pattern puo' aiutarti a soddisfare la domanda} \\ & && \text{Sono i coefficienti che diventeranno } a_{ik} \\ & && \rightarrow \text{Abbiamo rilassato l'assestaggio di colonne, ma non possiamo farlo} \end{aligned}$$

La generazione di colonne  
si fa risolvendo un problema Knapsack

This gives a new pattern, which is introduced into the set of columns if the value of the optimal solution is larger than 1.

This provides a lower bound for a branch-and-price algorithm. Alternatively, we may generate a set of columns, then restore integrality of the decision variables  $\gamma_k$  and solve a single integer program (which is a heuristic).

Note that, in a less stylized case, the complexity of constraints may be hidden in the column generation subproblem. The column generation scheme is at the heart of Dantzig–Wolfe decomposition.

IDEA. 1. Scelgo un sottoinsieme di colonne

2. Applichli il simplex → ottengo i coefficienti delle variabili duali  $\pi$

3. Calcoliamo  $C_j^T - \pi^T A_j$  e verifichiamo che sia il più negativo possibile  $\Rightarrow$  Più e' negativo meglio è!

→ e' 1 in quanto i coefficienti di  $y_k$  sono settati a 1

## Dantzig–Wolfe decomposition

Consider an LP model

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{D}\mathbf{x} \geq \mathbf{f} \\ & \mathbf{x} \geq \mathbf{0}_n. \end{aligned}$$

→ "Vincoli cattivi": cioè messi in una maniera scomoda  
→ E' il vincolo che compre le scatole perché  $\mathbf{A}$  è diagonale a blocchi (12)

→ "Vincoli buoni": cioè ben organizzati (13)

We split the constraints, as we are going to interpret (12) as the nasty, complicating constraints, whereas (13), along with the non-negativity restriction, is the set of easy constraints. This means that minimizing any linear function over the polyhedron

$$\mathcal{X} \doteq \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{D}\mathbf{x} \geq \mathbf{f}, \mathbf{x} \geq \mathbf{0}_n\}$$

→ Ricopertura convessa dei pt estremi  
Ricopertura conica direzione estrema

is easily accomplished. This may happen because of a decomposition into independent subproblems (i.e.,  $\mathbf{D}$  is block-diagonal) or because of favorable structure, e.g., a network flow structure. Whatever the reason, we know from Minkowski–Weyl theorem that we may represent  $\mathcal{X}$  as the sum of a polytope and a cone:

$$\mathcal{X} = \text{conv}\{\mathcal{V}\} + \text{cone}\{\mathcal{D}\}$$

→ Ricopertura convessa del set di vertici

→ Se il poliedro è limitato, non ci sono direzioni estreme e  $\text{cone}(\emptyset) = \emptyset$

→ Ricopertura delle direzioni di recessione

where  $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_Q\}$  is the set of extreme points (vertices) of  $\mathcal{X}$  and  $\mathcal{D} = \{\mathbf{d}_1, \dots, \mathbf{d}_R\}$  is the set of its extreme directions.

Hence, the original model

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} \geq \mathbf{b} \end{aligned}$$

(\*)  $\mathbf{x} \in \mathcal{X}$ ,  $\rightarrow$  poliedro che racchiude i vincoli precedenti

may be rewritten as follows. We express the constraint  $\mathbf{x} \in \mathcal{X}$  as

Il vincolo (\*) è così riscrivibile     $\mathbf{x} = \sum_{q \in \mathcal{V}} \lambda_q \mathbf{v}_q + \sum_{r \in \mathcal{D}} \mu_r \mathbf{d}_r$   $\xrightarrow{\text{Combinazione convessa dei vertici}}$  Cono finitamente generato dalle direzioni estreme    (14)

$\mathbf{x} =$  combinazione convessa dei vertici     $\xrightarrow{\text{Se il poliedro non è limitato, comuni le direzioni di recessione coniche}}$

$$\sum_{q \in \mathcal{V}} \lambda_q = 1 \quad \rightarrow \lambda_q \text{ non negativi e sommano a } 1 \quad (15)$$

$$\lambda_q \geq 0, q \in \mathcal{V}; \quad \mu_r \geq 0, r \in \mathcal{D}, \quad (16)$$

where (15) is the convexity constraint. We may plug (14) into the original model and obtain

Formulazione finale del problema di partenza che è del tutto equivalente a questo

$$\begin{aligned} \min \quad & \mathbf{c}^T \left( \sum_{q \in [Q]} \lambda_q \mathbf{v}_q + \sum_{r \in [R]} \mu_r \mathbf{d}_r \right) \\ \text{s.t.} \quad & \mathbf{A} \left( \sum_{q \in [Q]} \lambda_q \mathbf{v}_q + \sum_{r \in [R]} \mu_r \mathbf{d}_r \right) \geq \mathbf{b} \end{aligned}$$

(15), (16).

Note that this is an optimization model with respect to the decision variables  $\lambda_q$  and  $\mu_r$ . If we define the following scalars and vectors,

$$\begin{aligned} c_q &\doteq \mathbf{c}^T \mathbf{v}_q, & \mathbf{a}_q &\doteq \mathbf{A} \mathbf{v}_q, & q \in \mathcal{V} \\ c_r &\doteq \mathbf{c}^T \mathbf{d}_r, & \mathbf{a}_r &\doteq \mathbf{A} \mathbf{d}_r, & r \in \mathcal{D}, \end{aligned}$$

we may recast the original problem in the following form, referred to as the **master problem MP**:

*Ma questo problema puo' avere tantissimi vertici! La soluzione e' generarli dinamicamente* (MP)

*Approssimiamo dal di dentro vertici o direzioni estreme che e' equivalente a generare delle colonne*

$$\begin{aligned} \min \quad & \sum_{q \in \mathcal{V}} c_q \lambda_q + \sum_{r \in \mathcal{D}} c_r \mu_r \\ \text{s.t.} \quad & \sum_{q \in \mathcal{V}} \mathbf{a}_q \lambda_q + \sum_{r \in \mathcal{D}} \mathbf{a}_r \mu_r \geq \mathbf{b} \\ & \sum_{q \in \mathcal{V}} \lambda_q = 1 \\ & \lambda_q \geq 0, \quad q \in \mathcal{V}; \quad \mu_r \geq 0, \quad r \in \mathcal{D}. \end{aligned}$$

Ovvero sempre rilevanti perche' possono dare ottimi  
 Da tenere in considerazione per evitare che spari a +oo

→ Numero di vertici e di direzioni di recessione molto elevate

The clear trouble with the master problem is that it may involve a huge number of decision variables. Moreover, it may not be possible, given the description of  $\mathcal{X}$  in terms of inequality constraints, to find its full set of extreme points and extreme directions.

The solution strategy relies on column generation. We initialize a **Restricted Master Problem RMP** with a subset  $\mathcal{V}' \subset \mathcal{V}$  of extreme points and a subset  $\mathcal{D}' \subset \mathcal{D}$  of extreme directions:

(RMP)

$$\begin{aligned} \min \quad & \sum_{q \in \mathcal{V}'} c_q \lambda_q + \sum_{r \in \mathcal{D}'} c_r \mu_r \\ \text{s.t.} \quad & \sum_{q \in \mathcal{V}'} \mathbf{a}_q \lambda_q + \sum_{r \in \mathcal{D}'} \mathbf{a}_r \mu_r \stackrel{(17)}{\geq} \mathbf{b} \rightarrow \text{variabile duale } \pi \\ & \sum_{q \in \mathcal{V}'} \lambda_q = 1 \rightarrow \text{e' un vincolo solo, ha una unica variabile duale } \pi_0 \\ & \lambda_q \geq 0, q \in \mathcal{V}, \mu_r \geq 0, r \in \mathcal{D}. \end{aligned}$$

↳ La combinazione delle direzioni estreme e' conica

Let us denote the dual variables associated with constraints (17) and (18) by  $\pi$  and  $\pi_0$ , respectively.

→ Per capire se una colonna e' da portare in base devo calcolare il costo ridotto

In the standard simplex method, we would bring into the basis a column if its reduced cost is negative. For extreme points, this means checking whether there is any  $q \in \mathcal{Q}$  such that

$$\bar{c}_q = c_q - [\pi^\top \ \pi_0] \begin{bmatrix} \mathbf{a}_q \\ 1 \end{bmatrix} = c_q - \pi^\top \mathbf{a}_q - \pi_0 \stackrel{\text{e' una costante, nella funzione obiettivo del problema di pricing non conta}}{=} c_q - \pi^\top \mathbf{A} \mathbf{v}_q - \pi_0 < 0.$$

→ Se il costo ridotto per i vertici e' negativo, si mette in base la nuova colonna

A similar check should be carried out for extreme directions:

$$\bar{c}_r = c_r - [\pi^\top \ \pi_0] \begin{bmatrix} \mathbf{a}_r \\ 0 \end{bmatrix} = c_r - \pi^\top \mathbf{a}_r = c_r - \pi^\top \mathbf{A} \mathbf{d}_r < 0.$$

↳ combinazione conica

→ Discorso equivalente per le direzioni estreme

Hence, we should solve the following auxiliary problem,

$$\min \left\{ \min_{q \in \mathcal{V}} \bar{c}_q, \min_{r \in \mathcal{D}} \bar{c}_r, \right\}$$

→ Per capire quale colonna aggiungere alla base dovremmo risolvere questo problema

→ Risolviamo il problema di pricing

Clearly, this cannot be tackled by complete enumeration, but we may recast it into the form of the following **pricing problem** PP:

Il mio obiettivo è scoprire vertici o direzioni estreme. I valori che si generano dipendono dalle direzioni  $(c^r - \pi^T A)$  che provengono dal problema master.

(PP)

$$\begin{aligned} \min & \quad (c^T - \pi^T A)x \\ \text{s.t.} & \quad Dx \geq d \\ & \quad x \geq 0_n. \end{aligned}$$

Vettore duale che esce dal problema master  
e la variabile decisionale

Note that we are disregarding the constant term  $\pi_0$ , and use the same problem to find a vector  $x \in \mathbb{R}^n$  that will play the role of an extreme point or extreme direction, depending on the following cases, where  $z_{\text{PP}}^*$  denotes the optimal value of PP:

- Il problema di pricing è illimitato inferiormente, quindi si è trovata una direzione estrema
  - If  $z_{\text{PP}}^* = -\infty$ , i.e., PP is unbounded below, we have found an extreme direction of  $\mathcal{X}$ , say  $x^*$ , with negative reduced cost (commercial LP solvers provide us with this information). Then, we include a new variable  $\mu_r$  into RMP, associated with the with column → aggiungiamo una colonna a (+)

$$\begin{bmatrix} Ax^* \\ 0 \end{bmatrix}$$

→ Perche' μr va a formare una direzione conica  
μr non entra nel vicolo di connessità

- In questo caso pp e' limitato, quindi ho trovato un vertice che e' utile da aggiungere per migliorare il risultato
  - If  $-\infty < z_{PP}^* < \pi_0$ , i.e., the PP is bounded and its value is less than the dual variable of the convexity constraint, we have discovered a new extreme point, say  $x^*$ , with negative reduced cost. Then, we include a new variable  $\lambda_q$  into RMP, associated with column

$$\begin{bmatrix} Ax^* \\ 1 \end{bmatrix}. \rightarrow \lambda_q \text{ deve entrare nel v. colo di convessita'}$$

- Portare la colonna in base non fa migliorare la soluzione
  - If  $z_{PP}^* \geq \pi_0$ , then there is no useful extreme point or direction to add to RMP, and we may stop.

In intuitive terms, the optimal dual variables from solving RMP suggest a direction  $(c - A^\top \pi)$  along which we should “probe” the polyhedron  $\mathcal{X}$ . Clearly, if this is a polytope, we can only find an extreme point.

In the case of an unbounded polyhedron, we may find an extreme direction.

We do not need to find all of them, as only those who have a negative reduced cost are interesting, as they can improve the value we obtain from solving RMP.

Per ottenere la soluzione ottima si utilizza l'algoritmo branch-and-price che sfrutta il rilassamento continuo nel branch-and-bound integrando generazioni di colonne + tagli: