Domande di teoria Ricerca operativa 1

1. Definizione di insieme e funzione convessa

Definition 1.1.2 *A set* $\mathrm{X} \subseteq \Re^n$ *is said to be* convex *if* $\forall\ \mathbf{x}, \mathbf{y} \in \mathrm{X}$ *we have that* X *contains all the convex combinations of* $\mathbf{x}$ *and* $\mathbf{y}$*, i.e.:*

$$\mathbf{z} := [\lambda \mathbf{x} + (1-\lambda)\mathbf{y}] \in \mathrm{X} \quad , \quad \forall \lambda \in [0,1].$$

Definition 1.1.3 *A function* $f : \mathrm{X} \to \Re$ *defined on a convex set* $\mathrm{X} \subseteq \Re^n$ *is said to be* convex *if* $\forall\ \mathbf{x}\ ,\ \mathbf{y}\ \in\ \mathrm{X}$ *and* $\forall\ \lambda\ \in\ [0,1]$ *we have that*

$$f(\mathbf{z}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}) \quad \textit{where} \quad \mathbf{z} = \lambda \mathbf{x} + (1-\lambda)\mathbf{y}.$$

2. Si enunci e si dimostri il teorema fondamentale della programmazione convessa (ottimalità locale e globale) ott loc=>ott glob

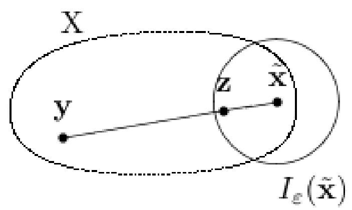
Theorem 1.1.2 *Consider a* convex programming *problem, i.e., a problem* $\min\{f(\mathbf{x}) : \mathbf{x} \in \mathrm{X}\}$ *where* $\mathrm{X} \subseteq \Re^n$ *is a convex set and* $f : \mathrm{X} \to \Re$ *is a convex function. Every locally optimal solution is also a globally optimal solution.*

Proof: Let $\tilde{\mathbf{x}}$ be any locally optimal solution. By the local optimum definition, there exists then $\varepsilon > 0$ such that $f(\tilde{\mathbf{x}}) \leq f(\mathbf{z})$ for all $\mathbf{z} \in I_\varepsilon(\tilde{\mathbf{x}}) := \{\mathbf{x} \in \mathrm{X} : ||\mathbf{x} - \tilde{\mathbf{x}}|| \leq \varepsilon\}$. We have to prove that $f(\tilde{\mathbf{x}}) \leq f(\mathbf{y})$ for all $\mathbf{y} \in \mathrm{X}$.

Given any $\mathbf{y} \in \mathrm{X}$, consider the point $\mathbf{z}$ belonging to the segment that connects $\tilde{\mathbf{x}}$ to $\mathbf{y}$ and defined as $\mathbf{z} := \lambda\tilde{\mathbf{x}} + (1-\lambda)\mathbf{y}$, where $\lambda < 1$ is chosen very close to the value 1 so that $\mathbf{z} \in I_\varepsilon(\tilde{\mathbf{x}})$ and hence $f(\tilde{\mathbf{x}}) \leq f(\mathbf{z})$. By the convexity hypothesis of f it follows that

$$f(\tilde{\mathbf{x}}) \leq f(\mathbf{z}) = f(\lambda\tilde{\mathbf{x}} + (1-\lambda)\mathbf{y}) \leq \lambda f(\tilde{\mathbf{x}}) + (1-\lambda) f(\mathbf{y}),$$

from which, dividing by $1 - \lambda > 0$, we obtain $f(\tilde{\mathbf{x}}) \leq f(\mathbf{y})$, as requested. $\square$



3. Definizione di soluzione base

For min{$c^T x$: Ax = b, x >=0} with A = [B|F] with b basis and x* = (x_B, x_F)

Definition 4.1.6 *The solution obtained imposing* $\mathbf{x}_F = 0$ *and* $\mathbf{x}_B = B^{-1}\mathbf{b}$ *is said to be the* basic solution *associated with basis* B. *The basic solution (and, by extension, basis* B *itself) is said to be* feasible *if* $\mathbf{x}_B = B^{-1}\mathbf{b} \geq 0$.

4. Dimostrare equivalenza tra vertici del poliedro e soluzioni base ammissibili.

Theorem 4.1.3 *A point* $\mathbf{x} \in P$ *is a vertex of the not empty polyhedron* $P := \{\mathbf{x} \geq 0 : A\mathbf{x} = \mathbf{b}\}$ *if and only if* $\mathbf{x}$ *is a basic feasible solution of the system* $A\mathbf{x} = \mathbf{b}$.

Proof: Let us first prove the implication "$\mathbf{x}$ is a basic feasible solution $\Rightarrow$ $\mathbf{x}$ is a vertex".

Let

$$\mathbf{x} = [\underbrace{x_1, \ldots, x_k}_{\text{positive}}, 0, \ldots, 0]^T$$

be any basic feasible solution associated with a basis B of A, where $k \geq 0$ is the number of non-zero (i.e., strictly positive) components of $\mathbf{x}$. It follows that columns $A_1, \ldots, A_k$ must be part of B, possibly together with other columns (in case of degenerate solution). Let us assume by contradiction that $\mathbf{x}$ is not a vertex. There exist thus

$$\begin{aligned} \mathbf{y} &= [y_1, \ldots, y_k, 0, \ldots, 0]^T \in P \\ \mathbf{z} &= [z_1, \ldots, z_k, 0, \ldots, 0]^T \in P \end{aligned}$$

with $\mathbf{y} \neq \mathbf{z}$ such that $\mathbf{x} = \lambda\mathbf{y} + (1-\lambda)\mathbf{z}$ for any $\lambda \in (0,1)$, which implies that $k \geq 1$.

Note that both $\mathbf{y}$ and $\mathbf{z}$ must have the last components set to zero, otherwise their convex combination cannot give $\mathbf{x}$. For the hypotheses, we then have:

$$\begin{aligned} \mathbf{y} \in P &\Rightarrow A\mathbf{y} = \mathbf{b} \Rightarrow A_1 y_1 + \ldots + A_k y_k = \mathbf{b} \\ \mathbf{z} \in P &\Rightarrow A\mathbf{z} = \mathbf{b} \Rightarrow A_1 z_1 + \ldots + A_k z_k = \mathbf{b}. \end{aligned}$$

By subtracting the second equation from the first we obtain

$$(y_1 - z_1)A_1 + \ldots + (y_k - z_k)A_k = \alpha_1 A_1 + \ldots + \alpha_k A_k = 0,$$

where $\alpha_i := y_i - z_i$, $i = 1, \ldots, k$. Hence there exist $\alpha_1, \ldots, \alpha_k$ scalars not all zero (since $\mathbf{y} \neq \mathbf{z}$) such that $\sum_{i=1}^{k} \alpha_i A_i = 0$, thus columns $A_1, \ldots, A_k$ are linearly dependent and cannot be part of the basis B ($\Rightarrow$ contradiction).

We will now prove the implication "$\mathbf{x}$ is a vertex $\Rightarrow$ $\mathbf{x}$ is a basic solution"; the fact that the basic solution is also feasible obviously derives from the hypothesis that $\mathbf{x} \in P$.

Writing, as before, $\mathbf{x} = [x_1, \ldots, x_k, 0, \ldots, 0]^T$ with $x_1, \ldots, x_k > 0$ and $k \geq 0$, we have that:

$$\mathbf{x} \in P \Rightarrow A\mathbf{x} = \mathbf{b} \Rightarrow A_1 x_1 + \ldots + A_k x_k = \mathbf{b}. \tag{4.2}$$

Two cases can occur:

1. columns $A_1, \ldots, A_k$ are linearly independent (or $k = 0$): by arbitrarily selecting other $m - k$ linearly independent columns (which, as is well known, is always possible), we obtain a basis $B = [A_1, \ldots, A_k, \ldots]$ whose basic associated solution is indeed $\mathbf{x}$ (which satisfies $A\mathbf{x} = \mathbf{b}$ and has non-basic components all equal to zero), thus concluding the proof.

2. columns $A_1, \ldots, A_k$ are linearly dependent: we will prove that this case cannot actually happen. Indeed, if the columns were linearly dependent, then there would exist $\alpha_1, \ldots, \alpha_k$ not all zero and such that

$$\alpha_1 A_1 + \ldots + \alpha_k A_k = 0, \tag{4.3}$$

The sum of (4.2) and (4.3) multiplied by $\varepsilon > 0$ would give:

$$(x_1 + \varepsilon\alpha_1)\mathrm{A}_1 + \ldots + (x_k + \varepsilon\alpha_k)\mathrm{A}_k \;=\; \mathbf{b}.$$

Similarly, the subtraction of (4.3) from (4.2) multiplied by ε would give:

$$(x_1 - \varepsilon\alpha_1)\mathrm{A}_1 + \ldots + (x_k - \varepsilon\alpha_k)\mathrm{A}_k \;=\; \mathbf{b}.$$

By defining

$$\mathbf{y} := [x_1 - \varepsilon\alpha_1, \ldots, x_k - \varepsilon\alpha_k, 0, \ldots, 0]^T$$
$$\mathbf{z} := [x_1 + \varepsilon\alpha_1, \ldots, x_k + \varepsilon\alpha_k, 0, \ldots, 0]^T,$$

we would have $\mathrm{A}\mathbf{y} = \mathbf{b}$ and $\mathrm{A}\mathbf{z} = \mathbf{b}$, while choosing a sufficiently small ε we would have $\mathbf{y}, \mathbf{z} \geq 0$ and thus $\mathbf{y}, \mathbf{z} \in P$, $\mathbf{y} \neq \mathbf{z}$. But since by construction

$$\mathbf{x} = \frac{1}{2}\mathbf{y} + \frac{1}{2}\mathbf{z},$$

this would mean that vertex $\mathbf{x}$ can be expressed as the strict convex combination of two distinct points of P ($\Rightarrow$ contradiction).

□

5. Dato il problema di PL min {c^T x : A x = b, x >= 0} ed una base ammissibile B, si dia la formula per il calcolo del vettore dei costi ridotti associati a B e si dimostri la correttezza della corrispondente condizione sufficiente di ottimalità.

Definition 4.2.1 *The vector*

$$\overline{\mathbf{c}}^T \;:=\mathbf{c}^T - \mathbf{c}_{\mathrm{B}}^T\mathrm{B}^{-1}\mathrm{A} \;=\; [\underbrace{\mathbf{c}_{\mathrm{B}}^T - \mathbf{c}_{\mathrm{B}}^T\mathrm{B}^{-1}\mathrm{B}}_{=0^T}, \mathbf{c}_{\mathrm{F}}^T - \mathbf{c}_{\mathrm{B}}^T\mathrm{B}^{-1}\mathrm{F}]$$

is called the reduced cost vector *with respect to basis* B.

Theorem 4.2.1 *Let* B *be a feasible basis. If* $\overline{\mathbf{c}}^T := \mathbf{c}^T - \mathbf{c}_{\mathrm{B}}^T\mathrm{B}^{-1}\mathrm{A} \geq 0^T$, *then the basic solution associated with* B *is optimal.*

Proof: Rewriting the objective function as $\mathbf{c}^T\mathbf{x} = \mathbf{c}_{\mathrm{B}}^T\mathrm{B}^{-1}\mathbf{b} + \overline{\mathbf{c}}^T\mathbf{x}$, in the hypothesis $\overline{\mathbf{c}}^T \geq 0^T$ we have that $\mathbf{c}^T\mathbf{x} \geq \mathbf{c}_{\mathrm{B}}^T\mathrm{B}^{-1}\mathbf{b}$ for all $\mathbf{x} \geq 0$ (and thus for all $\mathbf{x} \in P$), where $\mathbf{c}_{\mathrm{B}}^T\mathrm{B}^{-1}\mathbf{b} = [\mathbf{c}_{\mathrm{B}}^T, \mathbf{c}_{\mathrm{F}}^T]\begin{bmatrix}\mathrm{B}^{-1}\mathbf{b}\\ 0\end{bmatrix}$ is the value of the objective function corresponding to the basic feasible solution associated with B. □

6. Quali casi si possono presentare dopo la fase 1 del simplesso?

1. $w^* > 0$. In this case, there exists no solution to the artificial problem with $y_i = 0 \; \forall\, i \in \{1, \ldots, m\}$ (such solution would have cost $w = 0$), hence the system $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$, does not admit a solution: the original problem is hence infeasible.

2. $w^* = 0$. In this case we have $\mathbf{y}^* = 0$ (since $\mathbf{y}^* \geq 0$ and $\sum_{i=1}^{m} y_i^* = 0$), hence $\mathbf{x}^*$ is a feasible solution of the original problem. With respect to the optimal tableau for the artificial problem, two subcases may occur:

   (a) All artificial variables are non-basic variables: eliminating the columns of the tableau corresponding to the artificial variables, we have then a tableau (equivalent to the initial one) for the original problem, already in canonical form with respect to a basis. Now the artificial objective function has to be replaced with the original one, the objective function is converted in canonical form (by means of a substitution of variables), and the original problem is solved with the simplex algorithm.

   (b) There exists a basic variable y_h on a row i: in this case we have $y_h^* = 0$ (since $w^* = 0$), hence we are in the presence of degeneracy. The situation is as follows:

		x_1	$\ldots$	x_j	$\ldots$	x_n	y_1	$\ldots$	y_h	$\ldots$	y_m
$-w$	0								0		
									0		
									0		
y_h	0	$\overline{a}_{i1}$	$\ldots$	$\overline{a}_{ij}$	$\ldots$	$\overline{a}_{in}$			1		
									0		
									0		

   If there exist a value $\overline{a}_{ij} \neq 0$ in correspondence to a variable x_j ($j \leq n$), then it is sufficient to perform a pivot operation on element (i, j). Since $\overline{b}_i = 0$, this operation is admissible even if $\overline{a}_{ij} < 0$ and does not entail variations to the objective function w. In this way, y_h leaves the basis. Repeating the procedure for all the basic y_h leads us back to the non-degenerate case (a).

   One last possibility remains: all values $\overline{a}_{i1}, \ldots, \overline{a}_{in}$ are equal to zero. In this case, eliminating the artificial columns (an operation that is always admissible) we obtain a tableau equivalent to the original one, but with a zero row. This implies that the matrix A does not have full row rank m, and thus the i-th row of the current tableau may be eliminated without problems (the corresponding equation of system the $\mathbf{A}\mathbf{x} = \mathbf{b}$ is linearly dependent from the others).

7. regola di Bland, solo enunciato

Bland's Rule: *Whenever it is possible to choose, always choose the entering/leaving variable x_j with the smallest index j.*

In particular, we have to:

- choose variable x_h to enter the basis defining $h := \arg\min\{j : \overline{c}_j < 0\}$;
- among all rows t with $\overline{b}_t/\overline{a}_{th} = \vartheta$ that are eligible for the pivot operation, choose the one with minimum $\beta[t]$ so as to force the smallest-index variable $x_{\beta[t]}$ to leave the basis.

8. Dimostrare che con Bland l'algoritmo converge

Theorem 4.2.2 *Using Bland's rule, the simplex algorithm converges after, at most, $\binom{n}{m}$ iterations.*

Proof: Let us suppose by contradiction that the thesis is false and let us consider as counterexample the *smallest* LP problem for which there is no convergence. As already seen, in this case the simplex algorithm has to "go through" a cyclic sequence $B_1, B_2, \dots, B_k = B_1$ of bases. During this sequence, pivot operations are performed on *all* rows and *all* columns of the tableau, otherwise eliminating the not involved rows/columns we would obtain a smaller counterexample. It follows that *all* variables enter and leave, in turn, the current basis. Moreover, we must have $\overline{b}_t = 0$ for all rows $t \in \{1, \dots, m\}$, otherwise in the iteration in which the pivot operation is performed on row t, we would

have $\vartheta > 0$, hence the value of the objective function would change—preventing cycling in the sequence of bases.

Consider now tableau T in which variable x_n leaves the current basis to let a given non-basic variable x_h enter the basis. Let us indicate with $x_{\beta[i]}$ the basic variable in row $i \in \{1, \dots, m\}$, and with t the row in which x_n is in the basis (i.e., $\beta[t] = n$).

$$
\text{Tableau } T: \quad
\begin{array}{r|c|ccccccc|l}
 & & \dots & x_{\beta[i]} & \dots & x_h & \dots & x_n & & \\
\hline
-z & & & 0 & & - & & 0 & & \\
\hline
\dots & 0 & & 0 & & - & & 0 & & \dots \\
x_{\beta[i]} & 0 & & 1 & & - & & 0 & & \mu_i \geq 0 \\
\dots & 0 & & 0 & & - & & 0 & & \dots \\
x_{\beta[t]} & 0 & & 0 & & + & & 1 & & \mu_t < 0 \\
\dots & 0 & & 0 & & - & & 0 & & \dots \\
\hline
\end{array}
$$

The main features of this tableau are:

- $\overline{b}_i = 0 \ \forall i$ (as already seen)
- $\overline{c}_h < 0$ (since x_h enters the basis)
- $\overline{c}_{\beta[i]} = 0 \ \forall i$ (reduced costs basic variables)
- $\overline{a}_{th} > 0$ (pivot element)
- $\overline{a}_{ih} \leq 0 \ \forall i \neq t$ (Bland's rule).

Note that if there existed $i \neq t$ with $\overline{a}_{ih} > 0$, then Bland's rule would certainly have preferred to let variable $x_{\beta[i]}$ (instead of x_n) leave the basis.

Consider now tableau $\tilde{T}$ in which x_n re-enters the basis:

$$
\text{Tableau } \tilde{T} : \quad
\begin{array}{c|cccccc|}
 & \ldots & x_{\beta[i]} & \ldots & x_h & \ldots & x_n \\
\hline
 & + & + & + & + & + & - \\
\hline
 & & & & & & \\
 & & & & & & \\
 & & & & & & \\
\hline
\end{array}
$$

Indicating with $\tilde{c}_j$ the reduced costs in row 0 of $\tilde{T}$, we must have

- $\tilde{c}_n < 0$ (since x_n enters the basis)

- $\tilde{c}_j \geq 0\ \forall j \neq n$ (Bland's rule).

Now, $\tilde{T}$ has been obtained from T by means of a sequence of pivot operations, hence there exist appropriate multipliers $\mu_1, \ldots, \mu_m$ such that:

$$[\text{row } 0 \text{ of } \tilde{T}] = [\text{row } 0 \text{ of } T] + \sum_{i=1}^{m} \mu_i [\text{row } i \text{ of } T].$$

But then:

- $\underbrace{\tilde{c}_{\beta[t]}}_{=\tilde{c}_n < 0} = \underbrace{\overline{c}_{\beta[t]}}_{=0} + \mu_t \quad \Rightarrow \quad \mu_t < 0$
- $\underbrace{\tilde{c}_{\beta[i]}}_{\geq 0} = \underbrace{\overline{c}_{\beta[i]}}_{=0} + \mu_i \quad \Rightarrow \quad \mu_i \geq 0\ \forall i \neq t$

hence there is a contradiction:

$$\underbrace{\tilde{c}_h}_{\geq 0} = \underbrace{\overline{c}_h}_{<0} + \sum_{i \neq t} \underbrace{\overline{a}_{ih}}_{\leq 0} \underbrace{\mu_i}_{\geq 0} + \underbrace{\overline{a}_{th}}_{>0} \underbrace{\mu_t}_{<0} < 0.$$

$\square$

9. Enunciare e dimostrare lemma di Farkas (per il poliedro P non vuoto)

Theorem 5.1.1 (Farkas' Lemma) *The inequality $\mathbf{c}^T\mathbf{x} \geq c_0$ is valid for the non-empty polyhedron $P := \{\mathbf{x} \geq 0 \ : \ \mathrm{A}\mathbf{x} = \mathbf{b}\}$ if and only if $\mathbf{u} \in \Re^m$ exists such that*

$$\mathbf{c}^T \geq \mathbf{u}^T\mathrm{A} \tag{5.3}$$

$$c_0 \leq \mathbf{u}^T\mathbf{b}. \tag{5.4}$$

Proof: As already seen, the fact that the condition is sufficient is trivially true, given that for all $\mathbf{x} \geq 0$ such that $\mathrm{A}\mathbf{x} = \mathbf{b}$ we have:

$$\mathbf{c}^T\mathbf{x} \geq \mathbf{u}^T\mathrm{A}\mathbf{x} = \mathbf{u}^T\mathbf{b} \geq c_0.$$

We will now prove that the condition is also necessary, i.e., that "$\mathbf{c}^T\mathbf{x} \geq c_0$ valid for $P \neq \emptyset \Rightarrow \exists\mathbf{u} \in \Re^m \ : \ \mathbf{c}^T \geq \mathbf{u}^T\mathrm{A}\,,\ c_0 \leq \mathbf{u}^T\mathbf{b}$". By the hypothesis of validity, we have that

$$c_0 \leq z^* := \min\{\mathbf{c}^T\mathbf{x} \ : \ \mathrm{A}\mathbf{x} = \mathbf{b}\,,\ \mathbf{x} \geq 0\}, \tag{5.5}$$

which excludes $z^* = -\infty$. Let then $\mathbf{x}^*$ be an optimal basic feasible solution found by the simplex algorithm applied to problem (5.5). This solution exists by the convergence property of the simplex algorithm. In addition, let B be an optimal basis associated with $\mathbf{x}^*$, and let us partition as usual $\mathrm{A} = [\mathrm{B}, \mathrm{F}]$, $\mathbf{c}^T = [\mathbf{c}_\mathrm{B}^T, \mathbf{c}_\mathrm{F}^T]$, and $\mathbf{x}^* = (\mathbf{x}_\mathrm{B}^*, \mathbf{x}_\mathrm{F}^*)$ with $\mathbf{x}_\mathrm{B}^* = \mathrm{B}^{-1}\mathbf{b}$ and $\mathbf{x}_\mathrm{F}^* = 0$. We will prove that vector

$$\mathbf{u}^T := \mathbf{c}_\mathrm{B}^T\mathrm{B}^{-1}$$

verifies conditions (5.3) and (5.4), hence that the thesis is valid. Recalling the reduced cost expression computed in correspondence of the optimal basis B we have:

$$\bar{\mathbf{c}}^T := \mathbf{c}^T - \underbrace{\mathbf{c}_\mathrm{B}^T\mathrm{B}^{-1}}_{\mathbf{u}^T}\mathrm{A} \geq 0^T \ \Rightarrow \ \mathbf{c}^T \geq \mathbf{u}^T\mathrm{A}$$

and thus (5.3) is valid. In addition, for (5.5) we have that

$$c_0 \leq z^* = \mathbf{c}^T\mathbf{x}^* = \mathbf{c}_\mathrm{B}^T\mathbf{x}_\mathrm{B}^* + \mathbf{c}_\mathrm{F}^T\mathbf{x}_\mathrm{F}^* = \mathbf{c}_\mathrm{B}^T\mathrm{B}^{-1}\mathbf{b} = \mathbf{u}^T\mathbf{b},$$

and thus also (5.4) is verified. □

10. Si enunci e dimostri qui sotto il teorema della dualità debole per la programmazione lineare

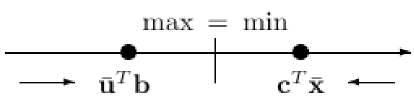
Theorem 5.3.2 (Weak Duality) *Let $P := \{\mathbf{x} \geq 0 \ : \ \mathrm{A}\mathbf{x} \geq \mathbf{b}\} \neq \emptyset$ and $D := \{\mathbf{u} \geq 0 \ : \ \mathbf{c}^T \geq \mathbf{u}^T\mathrm{A}\} \neq \emptyset$. For all pairs of points $\bar{\mathbf{x}} \in P$ and $\bar{\mathbf{u}} \in D$ we have that*

$$\bar{\mathbf{u}}^T\mathbf{b} \leq \mathbf{c}^T\bar{\mathbf{x}}.$$

Proof: Given $\bar{\mathbf{x}} \in P$ and $\bar{\mathbf{u}} \in D$, we have $\mathrm{A}\bar{\mathbf{x}} \geq \mathbf{b}$, $\bar{\mathbf{x}} \geq 0$, $\bar{\mathbf{u}} \geq 0$ and $\mathbf{c}^T \geq \bar{\mathbf{u}}^T\mathrm{A}$, from which

$$\bar{\mathbf{u}}^T\mathbf{b} \leq \bar{\mathbf{u}}^T\mathrm{A}\bar{\mathbf{x}} \leq \mathbf{c}^T\bar{\mathbf{x}}.$$

□



11. Analisi di sensitività PL: scrivere le condizioni di ottimalità per la base B a fronte delle variazioni dei termini noti, dei costi delle variabili fuori base e dei costi delle variabili in base.

$$\mathrm{B}^{-1}\mathbf{b} \geq -\mathrm{B}^{-1}\Delta\mathbf{b}. \qquad \Delta\mathbf{c}_{\mathrm{F}} \geq -\bar{\mathbf{c}}_{\mathrm{F}}. \qquad \Delta\mathbf{c}_{\mathrm{B}}^{T}\mathrm{B}^{-1}\mathrm{F} \leq \bar{\mathbf{c}}_{\mathrm{F}}^{T}.$$

12. Algoritmo di Dijkstra per i cammini minimi in O(n^2) e sua dimostrazione, meglio con pseudocodice.

Dijkstra's algorithm ($O(n^2)$ **version**);

```
    begin
1.    for j := 1 to n do /* initialization */
        begin
          flag[j] := 0 ;
          pred[j] := s ;
          L[j] := c[s, j]
        end ;
2.    flag[s] := 1; /* source vertex */
      L[s] := 0 ;
3.    for k := 1 to n − 1 do /* select a new arc */
        begin
4.        min := +∞; /* identify h = arg min{L[j] : j ∉ S} */
          for j := 1 to n do
            if (flag[j] = 0) and (L[j] < min) then
              begin
                min := L[j] ;
                h := j
              end ;
5.        flag[h] := 1; /* update S := S ∪ {h} */
6.        for j := 1 to n do /* update L[j] and pred[j] for all j ∉ S */
            if (flag[j] = 0) and (L[h] + c[h, j] < L[j]) then
              begin
                L[j] := L[h] + c[h, j] ;
                pred[j] := h
              end
        end
    end .
```

Theorem 7.6.2 *Assume that the cost L_i of the shortest paths from s to each vertex i belonging to a given set $S \subset V$ with $s \in S$ ($L_s := 0$) is known, and let $(v, h) := \arg\min\{L_i + c_{ij} \;:\; (i,j) \in \delta^+(S)\}$. If $c_{ij} \geq 0$ for all $(i,j) \in A$, then $L_v + c_{vh}$ is the cost of the shortest path from s to h.*

Proof: $L_v + c_{vh}$ is the cost of a path from s to h formed by the shortest path from s to v (with cost L_v) followed by arc (v, h). We have to prove that any other path P (say) from s to h has cost $c(P) \geq L_v + c_{vh}$. Let (i, j) be the first arc in $P \cap \delta^+(S)$, and as shown in Figure 7.21 partition P in $P_1 \cup \{(i,j)\} \cup P_2$, where P_1 and P_2 are two paths from s to i and from j to h, respectively. We have then

$$c(P) \;=\; \underbrace{c(P_1)}_{\geq L_i} + c_{ij} + \underbrace{c(P_2)}_{\geq 0} \geq L_i + c_{ij} \geq L_v + c_{vh}.$$

□

13. Algoritmo di Kruskal

```
Kruskal's Algorithm;
  begin
1.  sort the edges of G according to the increasing costs, obtaining:
      Edge[1].cost ≤ Edge[2].cost ≤ . . . ≤ Edge[m].cost ;
    k := 0 ;
    h := 0 ;
2.  for i := 1 to n do /* a vertex in each component */
      comp[i] := i ;
3.  while (k < n − 1) and (h < m) do
      begin /* consider the h-th edge */
4.      h := h + 1 ;
        i := Edge[h].From ;
        j := Edge[h].To ;
        C1 := comp[i] ;
        C2 := comp[j] ;
5.      if C1 ≠ C2 then /* select [i, j] */
          begin
6.          k := k + 1 ;
            Tree[k] := Edge[h] ;
7.          for q := 1 to n do /* combine components C1 and C2 */
              if comp[q] = C2 then
                comp[q] := C1
          end
      end ;
8.  if k ≠ n − 1 then "graph G is not connected"
  end .
```

14. Definire problema del massimo flusso su rete e dimostrare che la matrice dei vincoli è TUM sfruttando la nota proprietà.

Definition 7.7.1 *Given a flow network, a* feasible flow *from s to t is a function* $\mathbf{x} : A \to \Re$ *such that:*

$$0 \leq x_{ij} \leq k_{ij} \ , \ (i,j) \in A \tag{7.3}$$

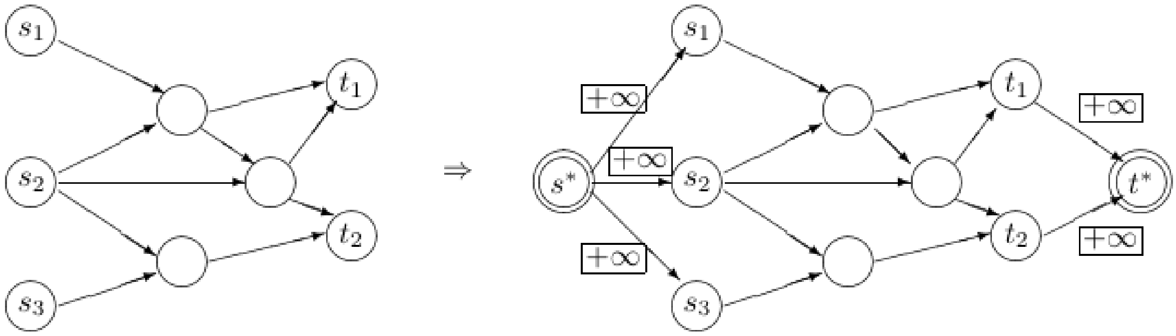
$$\underbrace{\sum_{(h,j) \in \delta^+(h)} x_{hj}}_{\text{flow leaving from } h} - \underbrace{\sum_{(i,h) \in \delta^-(h)} x_{ih}}_{\text{flow entering into } h} = 0 \ , \ h \in V \setminus \{s,t\}. \tag{7.4}$$

$$\mathbf{MAX-FLOW}: \ \max\{\varphi_0 := \sum_{(s,j)\in\delta^+(s)} x_{sj} - \sum_{(i,s)\in\delta^-(s)} x_{is} \ : \ \text{constraints } (7.3)-(7.4)\}.$$

Note that the constraint matrix of these linear programming problems is totally unimodular, given that constraint matrix (7.4) is composed by the rows of the node-arc incidence matrix of graph G associated with vertices $h \neq s, t$. It follows that a flow problem always has an optimal integer solution when capacities k_{ij} are all integer.

15. definire come si deve procedere in presenza di sorgenti/terminali multipli per ottiene una rete con una sola sorgente ed un solo terminale;

If there is a network with more than one source vertex and/or more than one sink vertex, it is possible to reduce to the regular case by creating a dummy source s^* from which arcs with infinite capacity go towards the real sources, and a dummy sink vertex t^* to which arcs with infinite capacity come from the real sinks. Note that the transformation cannot be used if the "goods" transported from the various sources to the various sinks are different from each other (leading to an NP-hard problem known as the *multi-commodity flow* problem).



16. Definizione di sezione

Definition 7.7.2 *We call* cut *a partition* $(S, V \setminus S)$ *of the set of vertices such that* $s \in S$ *and* $t \in V \setminus S$.

17. Definizione di flusso attraverso una sezione

Definition 7.7.3 *Given a feasible flow* $\mathbf{x}$, *we call* flow through a cut $(S, V \setminus S)$ *the quantity:*

$$\varphi(S) := \underbrace{\sum_{(i,j)\,\in\,\delta^+(S)} x_{ij}}_{\text{flow leaving from } S} - \underbrace{\sum_{(i,j)\,\in\,\delta^-(S)} x_{ij}}_{\text{flow entering into } S} .$$

18. Definizione di capacità della sezione

Definition 7.7.4 *The* capacity of the cut $(S, V \setminus S)$ *is the quantity:*

$$k(S) := \sum_{(i,j)\in\delta^+(S)} k_{ij}.$$

Note that arcs $(i,j) \in \delta^-(S)$ *entering* into S do not contribute to the capacity of the cut. The following theorems apply:

19. Dimostrare che il flusso attraverso una sezione è costante

Theorem 7.7.1 *Let* $\mathbf{x}$ *be a feasible flow. For every cut* $(S, V \setminus S)$ *one has* $\varphi(S) = \varphi_0$*, i.e., the flow through every cut is a constant.*

Proof: Consider an arbitrary cut $(S, V \setminus S)$. Then:

$$\varphi_0 := \varphi(\{s\}) := \sum_{(s,j)\in\delta^+(s)} x_{sj} - \sum_{(i,s)\in\delta^-(s)} x_{is} = \sum_{h \in S} \underbrace{\left[\sum_{(h,j)\in\delta^+(h)} x_{hj} - \sum_{(i,h)\in\delta^-(h)} x_{ih}\right]}_{=0 \text{ for all } h \neq s}$$

$$= \sum_{h \in S} \sum_{(h,j)\in\delta^+(h)} x_{hj} - \sum_{h \in S} \sum_{(i,h)\in\delta^-(h)} x_{ih}$$

$$= \left[\underbrace{\sum_{(i,j)\in A(S)} x_{ij}}_{=} + \sum_{(i,j)\in\delta^+(S)} x_{ij}\right] - \left[\underbrace{\sum_{(i,j)\in A(S)} x_{ij}}_{=} + \sum_{(i,j)\in\delta^-(S)} x_{ij}\right] =: \varphi(S).$$

□

20. Dimostrare che il flusso attraverso una sezione è sempre <= alla capacità della sezione

Theorem 7.7.2 *For every feasible flow* $\mathbf{x}$ *and every cut* $(S, V \setminus S)$*, one has*

$$\varphi(S) \leq k(S).$$

Proof:

$$\varphi(S) := \sum_{(i,j)\in\delta^+(S)} \underbrace{x_{ij}}_{\leq k_{ij}} - \sum_{(i,j)\in\delta^-(S)} \underbrace{x_{ij}}_{\geq 0} \leq \sum_{(i,j)\in\delta^+(S)} k_{ij} =: k(S).$$

□

21. Enunciare e dimostrare il teorema fondamentale relativo all'ottimalità di un flusso ammissibile su rete.

Theorem 7.7.3 *A feasible flow* $\mathbf{x}$ *is optimal for the MAX-FLOW problem if and only if vertex* t *is not reachable from vertex* s *in the residual network* $\overline{G} = (V, \overline{A})$ *associated with* $\mathbf{x}$*.*

Proof: Let φ_0 be the value of flow $\mathbf{x}$. If t is reachable from s in $\overline{G}$, there exists then an augmenting path P from s to t in $\overline{G}$. Setting $\delta := \min\{\overline{k}_{uv} \ : \ (u,v) \in P\} > 0$, for all $(u,v) \in P$ it is possible to update $x_{uv} := x_{uv} + \delta$ if (u,v) is a forward arc; $x_{vu} := x_{vu} - \delta$ if (u,v) is a backward arc. It is easy to verify that the new vector $\mathbf{x}$ defines a feasible flow of value $\varphi_0 + \delta$ in the original network, which proves that the starting solution $\mathbf{x}$ was not optimal for the MAX-FLOW problem.

Let us now assume that t is not reachable from s in $\overline{G}$. There hence exists a cut $(S^*, V \backslash S^*)$ in the *residual* network $\overline{G}$ such that $\delta^+_{\overline{G}}(S^*) = \emptyset$. By the definition of residual network we have that, in the *original* network G:

- each arc $(i,j) \in \delta^+_G(S^*)$ is saturated
- each arc $(i,j) \in \delta^-_G(S^*)$ is empty.

It follows that

$$\varphi(S^*) := \underbrace{\sum_{(i,j)\in\delta^+_G(S^*)} x_{ij}}_{\text{all saturated}} - \underbrace{\sum_{(i,j)\in\delta^-_G(S^*)} x_{ij}}_{\text{all empty}} = \sum_{(i,j)\in\delta^+_G(S^*)} k_{ij} =: k(S^*)$$

hence the optimality of $\mathbf{x}$ derives from Theorem 7.7.2, which also guarantees that $(S^*, V \setminus S^*)$ is a cut of minimum capacity in the original network. □

22. Definizione del problema dello zaino di Knapsnack, complessità, modello PLI, algoritmo di Danzig per la risoluzione del rilassamento continuo.

Let a set of items $\{1, \ldots, n\}$ be given, each with a *profit* p_j and a *weight* w_j, and a container of *capacity* W.

Without loss of generality, we can assume that all values p_j, w_j and W are positive integers, with $w_j < W$ for all $j = 1, \ldots, n$, and $\sum_{j=1}^n w_j > W$.

The *Knapsack Problem* (KP) consists in selecting an item subset with maximum profit to load into the container. This is a fundamental problem that arises whenever we want to optimally load any container (truck, cargo ship, etc...).

Introducing the decision variables

$$x_j = \begin{cases} 1 & \text{if item } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

we get the ILP model:

$$z^* := \max \sum_{j=1}^n p_j x_j \tag{8.1a}$$

example, $x_j = 0.3$ means selecting 30% of item j. This interpretation suggests an algorithm (due to G. Dantzig) to solve the continuous relaxation of model (8.1a)-(8.1c) by:

1. split each item j into w_j *subitems* of unitary weight and profit p_j/w_j;
2. fully load the container selecting the subitems of maximal profit p_j/w_j.

In other words, the algorithm sorts the items $j \in \{1, \ldots, n\}$ by non-increasing relative profit

$$\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \ldots \geq \frac{p_n}{w_n},$$

and finds the *critical item* $s \in \{1, \ldots, n\}$, defined as the one with the property:

$$\sum_{j=1}^{s-1} w_j < W \leq \sum_{j=1}^{s} w_j.$$

The optimal solution $\mathbf{x}^*$ of the continuous relaxation can then be obtained by:

1. fully selecting the first $s-1$ items $\Rightarrow$ $x_1^* = \ldots = x_{s-1}^* := 1$;
2. partially selecting the critical item s $\Rightarrow$ $x_s^* := \left(W - \sum_{j=1}^{s-1} w_j\right)/w_s$;
3. discarding the following items $\Rightarrow$ $x_{s+1}^* = \ldots = x_n^* := 0$.

This way it is possible to solve the continuous relaxation in $O(n \log n)$ time, or even in $O(n)$ time by using a so-called *partial sorting* algorithm.

23. Modello di programmazione lineare del TSP.
Si descriva un algoritmo di separazione dei vincoli di connessione per il modello.

$$\min \underbrace{\sum_{(i,j)\in A} c_{ij}x_{ij}}_{\text{circuit cost}} \tag{8.2a}$$

$$\underbrace{\sum_{(i,j)\in\delta^-(j)} x_{ij}}_{\text{one arc entering } j} = 1\,,\; j \in V \tag{8.2b}$$

$$\underbrace{\sum_{(i,j)\in\delta^+(i)} x_{ij}}_{\text{one arc leaving } i} = 1\,,\; i \in V \tag{8.2c}$$

$$\underbrace{\sum_{(i,j)\in\delta^+(S)} x_{ij} \geq 1}_{\text{reachability from 1}}\,,\; S \subset V \;:\; 1 \in S \tag{8.2d}$$

$$x_{ij} \geq 0 \text{ integer}\,,\; (i,j) \in A. \tag{8.2e}$$

Separation Problem for constraints (8.2d) *Given a point* $\mathbf{x}^* \geq 0$*, identify a set* $S^* \subset V$ *such that* $1 \in S^*$ *and* $\gamma^* := \sum_{(i,j)\in\delta^+(S^*)} x^*_{ij}$ *is minimum. If* $\gamma^* \geq 1$*, then no violated constraints (8.2d) exist; otherwise,* S^* *corresponds to a constraint (8.2d) most violated by* $\mathbf{x}^*$*.*

24. Si scriva qui sotto un modello di PLI per problema dell'albero di Steiner su di un grafo orientato G=(V,A) con radice r ed insieme target T:
Si descriva un algoritmo di separazione dei vincoli di connessione per il modello.

$$\min \underbrace{\sum_{(i,j)\in A} c_{ij}x_{ij}}_{\text{solution cost}} \tag{8.3a}$$

$$\underbrace{\sum_{(i,j)\in\delta^-(j)} x_{ij}}_{\text{n. of arcs entering } j} \begin{cases} = 1 & \text{for all } j \in T \\ = 0 & \text{for } j = r \\ \leq 1 & \text{for all } j \in \mathrm{V} \setminus (T \cup \{r\}) \end{cases} \tag{8.3b}$$

$$\sum_{(i,j)\in\delta^+(S)} x_{ij} \geq \sum_{(i,t)\in\delta^-(t)} x_{it}\ , \text{ for all } S \subset V : r \in S, \text{ and for all } t \in V \setminus S \tag{8.3c}$$

$$x_{ij} \geq 0 \text{ integer },\ (i,j) \in A. \tag{8.3d}$$

As in the TSP case, model (8.3a)-(8.3d) can be solved by means of the *branch-and-cut* technique using a separation scheme for constraints (8.3c) based on the computation, for each set $t \in V \setminus \{r\}$, of a cut with minimum capacity $(S^*_t, V \setminus S^*_t)$ with $r \in S^*_t$ and $t \notin S^*_t$ on the network with capacity x^*_{ij} on the arcs: if the capacity of this cut is lower than $\sum_{(i,t)\in\delta^-(t)} x^*_{it}$, then the corresponding constraint (8.3c) is violated.