Minimum cost transportation problem

Combinatorial optimization

Giovanni Righini

Università degli Studi di Milano

Definitions

The *minimum cost transportation problem* is a special case of the *minimum cost maximum flow problem*.

We are given:

- a set of origins, $\mathcal{O}$;
- a set of destinations, $\mathcal{D}$;
- an amount of flow o_i to be sent from each origin $i \in \mathcal{O}$;
- an amount of flow d_j to be received by each destination $j \in \mathcal{D}$;
- a transportation network, connecting each origin $i \in \mathcal{O}$ with each destination $j \in \mathcal{D}$ at a unit cost c_{ij}.

There are no capacities associated with the arcs between $\mathcal{O}$ and $\mathcal{D}$.

We assume

$$\sum_{i \in \mathcal{O}} o_i = \sum_{j \in \mathcal{D}} d_j .$$

If this does not hold, we add a dummy origin or a dummy destination where necessary, with zero cost arcs incident in it.

A reformulation

By adding a dummy source s connected to each node $i \in \mathcal{O}$ with an arc of capacity o_i and a dummy sink t connected to each node $j \in \mathcal{D}$ with an arc of capacity d_j, we can solve a min cost max flow problem from s to t on the resulting digraph.

This is equivalent to the original problem, because the bottleneck is the cut separating s (or t) from the rest of the digraph and hence the max flow has value $\sum_{i \in \mathcal{O}} o_i = \sum_{j \in \mathcal{D}} d_j$.

A formulation

We use a variable x_{ij} (continuous and non-negative) to indicate the amount of flow on each arc $(i, j) \in \mathcal{A} = \mathcal{O} \times \mathcal{D}$.

A mathematical model of the problem is:

$$
\begin{aligned}
\text{minimize} \;\; z = & \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} & \\
s.t. \;\; & \sum_{j \in \mathcal{D}} x_{ij} = o_i & \forall i \in \mathcal{O} \\
& \sum_{i \in \mathcal{O}} x_{ij} = d_j & \forall j \in \mathcal{D} \\
& x_{ij} \geq 0 & \forall (i, j) \in \mathcal{A}.
\end{aligned}
$$

It is a linear programming model. Duality theory applies.

The dual problem

$$\text{P) minimize } z = \sum_{(i,j)\in\mathcal{A}} c_{ij} x_{ij}$$

$$\text{s.t. } \sum_{j\in\mathcal{D}} x_{ij} = o_i \qquad \forall i \in \mathcal{O}$$

$$\sum_{i\in\mathcal{O}} x_{ij} = d_j \qquad \forall j \in \mathcal{D}$$

$$x_{ij} \geq 0 \qquad \forall (i,j) \in \mathcal{A}.$$

$$\text{D) maximize } w = \sum_{i\in\mathcal{O}} o_i u_i + \sum_{j\in\mathcal{D}} d_j v_j$$

$$\text{s.t. } u_i + v_j \leq c_{ij} \qquad \forall (i,j) \in \mathcal{A}$$

Complementary slackness conditions imply:

$$x_{ij}(c_{ij} - (u_i + v_j)) = 0.$$

Primal-dual algorithm

Begin
Primal initialization;
Dual initialization;
while $\exists(i,j) : u_i + v_j > c_{ij}$ **do**
  Update primal solution;
  Update dual solution;
end while
End

At each iteration the algorithm keeps a primal-dual pair of solutions satisfying the CSCs.

The primal solution is feasible; the dual solution is (super-)optimal.

When the optimal dual solution is also feasible, the feasible primal solution is also optimal.

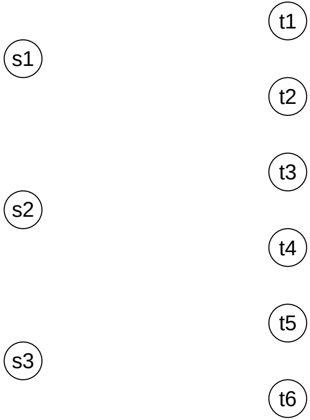
Primal initialization

Begin Primal Initialization
for $i \in \mathcal{O}$ **do**
  $R^o(i) := o_i$;
end for
for $j \in \mathcal{D}$ **do**
  $R^d(j) := d_j$;
end for
for $(i,j) \in \mathcal{A}$ **do**
  $x_{ij} := \min\{R^o(i), R^d(j)\}$;
  $R^o(i) := R^o(i) - x_{ij}$;
  $R^d(j) := R^d(j) - x_{ij}$;
end for
End Primal Initialization

At each step at least one of the residuals R is set to 0. At least in the last iteration a row residual and a column residual are simultaneously set to 0, because o and d are balanced.

The number of arcs with non-zero flow is at most $|\mathcal{O}| + |\mathcal{D}| - 1$.

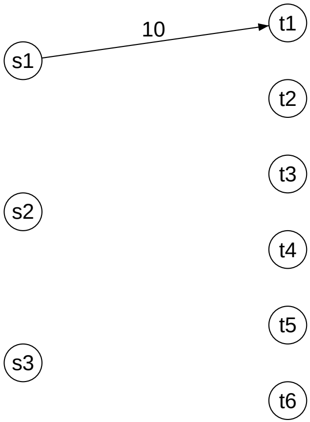
Visualization of primal initialization



x	t1	t2	t3	t4	t5	t6	R^o
s1	0	0	0	0	0	0	30
s2	0	0	0	0	0	0	30
s3	0	0	0	0	0	0	30
R^d	10	12	13	15	20	20	0

$\bar{c}$	t1	t2	t3	t4	t5	t6	u
s1	7	12	4	15	2	0	0
s2	1	8	9	10	3	0	0
s3	6	5	7	2	18	0	0
v	0	0	0	0	0	0	**0**

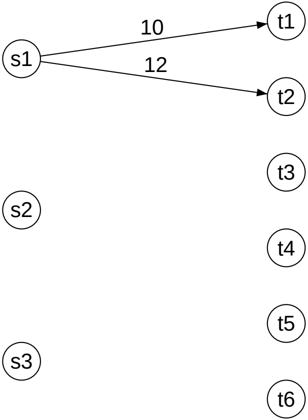
Visualization of primal initialization



x	t1	t2	t3	t4	t5	t6	R^o
s1	10	0	0	0	0	0	**20**
s2	0	0	0	0	0	0	30
s3	0	0	0	0	0	0	30
R^d	**0**	12	13	15	20	20	**70**

$\bar{c}$	t1	t2	t3	t4	t5	t6	u
s1	7	12	4	15	2	0	0
s2	1	8	9	10	3	0	0
s3	6	5	7	2	18	0	0
v	0	0	0	0	0	0	**0**

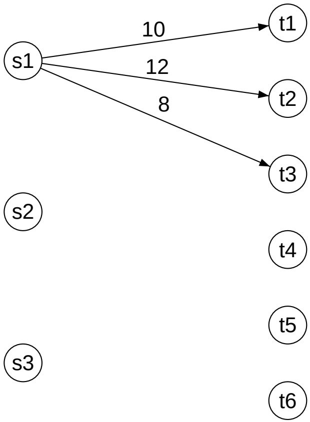
Visualization of primal initialization



x	t1	t2	t3	t4	t5	t6	R^o
s1	10	12	0	0	0	0	**8**
s2	0	0	0	0	0	0	30
s3	0	0	0	0	0	0	30
R^d	0	**0**	13	15	20	20	**214**

$\bar{c}$	t1	t2	t3	t4	t5	t6	u
s1	7	12	4	15	2	0	0
s2	1	8	9	10	3	0	0
s3	6	5	7	2	18	0	0
v	0	0	0	0	0	0	**0**

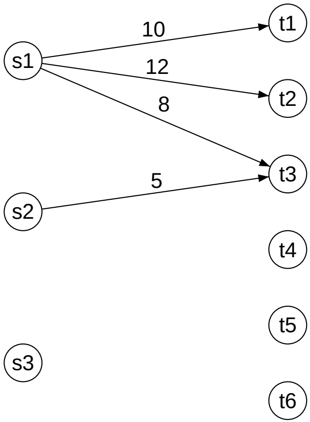
Visualization of primal initialization



x	t1	t2	t3	t4	t5	t6	R^o
s1	10	12	8	0	0	0	**0**
s2	0	0	0	0	0	0	30
s3	0	0	0	0	0	0	30
R^d	0	0	**5**	15	20	20	**246**

$\bar{c}$	t1	t2	t3	t4	t5	t6	u
s1	7	12	4	15	2	0	0
s2	1	8	9	10	3	0	0
s3	6	5	7	2	18	0	0
v	0	0	0	0	0	0	**0**

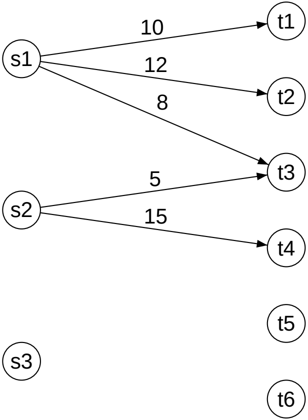
Visualization of primal initialization



x	t1	t2	t3	t4	t5	t6	R^o
s1	10	12	8	0	0	0	0
s2	0	0	5	0	0	0	**25**
s3	0	0	0	0	0	0	30
R^d	0	0	**0**	15	20	20	**291**

$\bar{c}$	t1	t2	t3	t4	t5	t6	u
s1	7	12	4	15	2	0	0
s2	1	8	9	10	3	0	0
s3	6	5	7	2	18	0	0
v	0	0	0	0	0	0	**0**

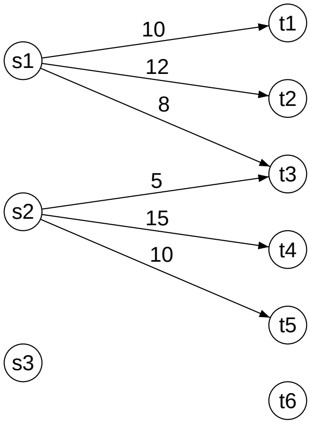
Visualization of primal initialization



x	t1	t2	t3	t4	t5	t6	R^o
s1	10	12	8	0	0	0	0
s2	0	0	5	15	0	0	**10**
s3	0	0	0	0	0	0	30
R^d	0	0	0	**0**	20	20	**441**

$\bar{c}$	t1	t2	t3	t4	t5	t6	u
s1	7	12	4	15	2	0	0
s2	1	8	9	10	3	0	0
s3	6	5	7	2	18	0	0
v	0	0	0	0	0	0	**0**

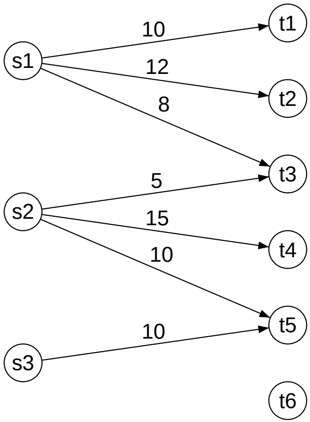
Visualization of primal initialization



x	t1	t2	t3	t4	t5	t6	R^o
s1	10	12	8	0	0	0	0
s2	0	0	5	15	10	0	**0**
s3	0	0	0	0	0	0	30
R^d	0	0	0	0	**10**	20	**471**

$\bar{c}$	t1	t2	t3	t4	t5	t6	*u*
s1	7	12	4	15	2	0	0
s2	1	8	9	10	3	0	0
s3	6	5	7	2	18	0	0
v	0	0	0	0	0	0	**0**

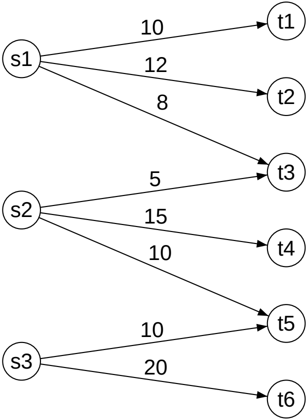
Visualization of primal initialization



x	t1	t2	t3	t4	t5	t6	R^o
s1	10	12	8	0	0	0	0
s2	0	0	5	15	10	0	0
s3	0	0	0	0	10	0	**20**
R^d	0	0	0	0	**0**	20	**651**

$\bar{c}$	t1	t2	t3	t4	t5	t6	*u*
s1	7	12	4	15	2	0	0
s2	1	8	9	10	3	0	0
s3	6	5	7	2	18	0	0
v	0	0	0	0	0	0	**0**

Visualization of primal initialization



x	t1	t2	t3	t4	t5	t6	R^o
s1	10	12	8	0	0	0	0
s2	0	0	5	15	10	0	0
s3	0	0	0	0	10	20	**0**
R^d	0	0	0	0	0	**0**	**651**

$\bar{c}$	t1	t2	t3	t4	t5	t6	u
s1	7	12	4	15	2	0	0
s2	1	8	9	10	3	0	0
s3	6	5	7	2	18	0	0
v	0	0	0	0	0	0	**0**

Dual update

Exploiting CSCs, a dual solution corresponding to the current primal solution is obtained by solving a system of linear equations:

$$u_i + v_j = c_{ij} \quad \forall (i,j) : x_{ij} > 0.$$

Since the number of basic x variables is $|\mathcal{O}| + |\mathcal{D}| - 1$ and the number of dual variables is $|\mathcal{O}| + |\mathcal{D}|$, the system has infinitely many dual solutions.

This is consistent with the degeneracy of the primal problem: any equality constraint can be derived from the others.

Dual update amounts at arbitrarily fixing a dual variable and solving a system of linear equations, with a triangular matrix of coefficients.

This is done both after primal initialization and after each primal update step.

Visualization of dual update

c	t1	t2	t3	t4	t5	t6
s1	7	12	4	15	2	0
s2	1	8	9	10	3	0
s3	6	5	7	2	18	0

$$\begin{cases} u_1 + v_1 = 7 \\ u_1 + v_2 = 12 \\ u_1 + v_3 = 4 \\ u_2 + v_3 = 9 \\ u_2 + v_4 = 10 \\ u_2 + v_5 = 3 \\ u_3 + v_5 = 18 \\ u_3 + v_6 = 0 \end{cases} \quad \begin{cases} u_1 = 0 \\ v_1 = 7 \\ v_2 = 12 \\ v_3 = 4 \\ u_2 = 5 \\ v_4 = 5 \\ v_5 = -2 \\ u_3 = 20 \\ v_6 = -20 \end{cases}$$

$\bar{c}$	t1	t2	t3	t4	t5	t6	u
s1	0	0	0	10	4	20	0
s2	-11	-9	0	0	0	15	5
s3	-21	-27	-17	-23	0	0	20
v	7	12	4	5	-2	-20	**651**

Visualization of dual feasibility test

In our example there are several violated dual constraints: they correspond to arcs with negative reduced cost.

$\overline{c}$	t1	t2	t3	t4	t5	t6	u
s1	0	0	0	10	4	20	0
s2	-11	-9	0	0	0	15	5
s3	-21	**-27**	-17	-23	0	0	20
v	7	12	4	5	-2	-20	**651**

We can choose for example the minimum reduced cost.

We want to enforce $u_3+v_2=c_{32}$ to allow x_{32} taking positive value.

Primal update

The primal solution is changed by sending flow on arc $(\bar{i}, \bar{j})$, corresponding to a violated dual constraint.

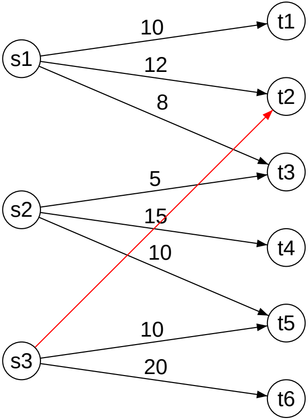
This implies re-balancing the flow on other arcs, in order to maintain primal feasibility.

Hence a cycle containing $(\bar{i}, \bar{j})$ is identified, such that all the arcs in the cycle correspond to active dual constraints, i.e. basic primal variables.

The flow is alternately increased and decreased on the arcs of the cycle. The bottleneck arc determines the amount of flow to be sent along the cycle.

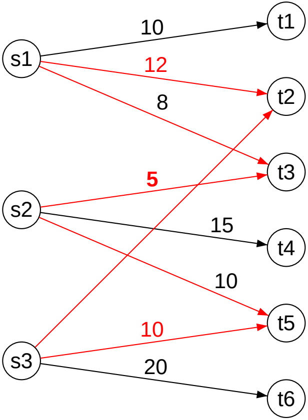
After the flow update, the bottleneck arc has zero flow and its corresponding dual constraint is no longer forced to be active.

Visualization of primal update



x	t1	t2	t3	t4	t5	t6
s1	10	12	8	0	0	0
s2	0	0	5	15	10	0
s3	0	**0**	0	0	10	20

Visualization of primal update



x	t1	t2	t3	t4	t5	t6
s1	10	12	8	0	0	0
s2	0	0	5	15	10	0
s3	0	**0**	0	0	10	20

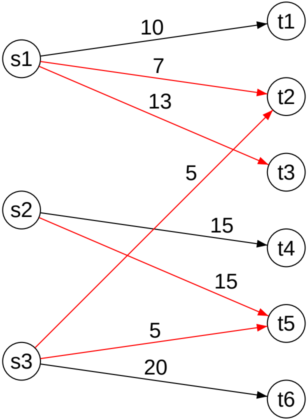
There is a unique cycle joining $s3$ with $t2$ in the flow graph corresponding to the current primal solution.

Flow must be

- increased on $(3, 2)$, $(1, 3)$ and $(2, 5)$;
- decreased on $(1, 2)$, $(2, 3)$ and $(3, 5)$.

$$\delta = \min\{x_{12}, x_{23}, x_{35}\} = 5.$$

Visualization of primal update



x	t1	t2	t3	t4	t5	t6
s1	10	7	13	0	0	0
s2	0	0	0	15	15	0
s3	0	**5**	0	0	5	20

Now $(2,3)$ carries no flow;
it has been replaced by $(3,2)$.

The cost variation is
$\delta\, \overline{c}_{32} = \delta(c_{32} - c_{12} + c_{13} - c_{23} + c_{25} - c_{35}) =$
$5(5-12+4-9+3-18) = 5(-27) = -135.$

Basic arcs always form a spanning tree on the flow network.