Transport problems

Resolution

Hugues Talbot

Centre de Vision Numérique (CVN)

May 4, 2018

Outline

Introduction

- Transport problems are LP associating producers and consumers. consommateurs ;
- We can always balance transport problems so that all the production is consumed, if necessary by introducing extra consumer nodes ;
- Transport problems are easier to solve than standard LP. There is no matrix inversion, only additions and substractions;
- Integer transport problems are no harder to solve than real number ones.

Refresher

- We can represent a transport problem in a *tableau* ;
- A problem with m producers and n consumers is of rank $m + n - 1$ (Q: why ?) ;
- A balanced transport problem only has equality constraints (Q: why ?)
- Normally, an LP with only equality constraints is harder to get started than other problems (i.e. it is harder to find an initial feasible solution basis). Why ?

Example

Transport problem

			4
			5
3	2	4	

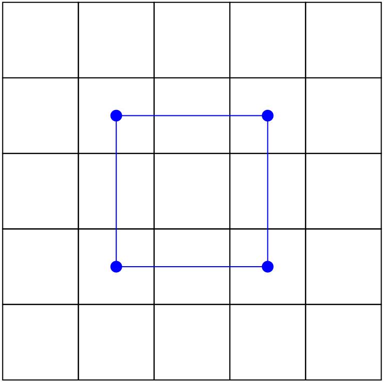
Equivalent LP problem

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 3 \\ 2 \\ 4 \end{bmatrix}$$

- We must eliminate one of the constraints, which is redundant, so the rank of the problem is $m + n - 1 = 4$
- Finding an initial feasible solution basis is not trivial. For instance $\{x_{11}, x_{12}, x_{21}, x_{22}\}$ does not work.

Notion of a loop

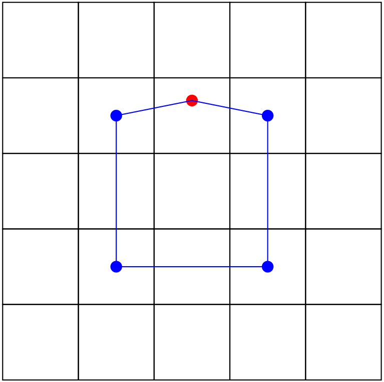
A loop is a sequence of at least 4 cells, such that :



1. Two consecutive cells are either in the same row or in the same column ;

Notion of a loop

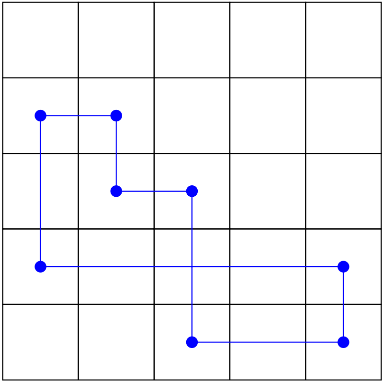
A loop is a sequence of at least 4 cells, such that :



1. Two consecutive cells are either in the same row or in the same column ;
2. Any sub-sequence of three consecutive cells are *never* in the same row or column ;

Notion of a loop

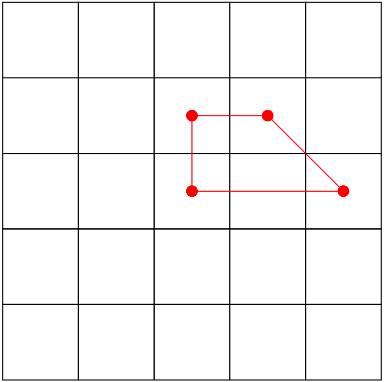
A loop is a sequence of at least 4 cells, such that :



1. Two consecutive cells are either in the same row or in the same column ;
2. Any sub-sequence of three consecutive cells are *never* in the same row or column ;
3. The last cell in the sequence has either a row or a column in common with the first ;

Notion of a loop

A loop is a sequence of at least 4 cells, such that :



1. Two consecutive cells are either in the same row or in the same column ;
2. Any sub-sequence of three consecutive cells are *never* in the same row or column ;
3. The last cell in the sequence has either a row or a column in common with the first ;

The loop theorem

Theorem

Consider a transport problem with m producers and n consumers. The cells corresponding to a set of $m+n-1$ variables do not *contain any loop if and only if these variables form a basis solution.*

Proof.

This theorem derives from the fact that a set of $m+n-1$ cells do not contain any loop if and only if the $m+n-1$ columns of A that correspond to these cells are linearly independent. ☐

Methods for finding an initial FBS

Finding an initial Feasible Basis Solution is much easier for transport problem than for generic Linear Programming problems. There are three classical methods

1. The upper-left corner method ;
2. The minimal cost method ;
3. The VOGEL method (not presented in this course).

The upper-left corner (ULC) method

- We start in the upper left cell with variable x_{11}, and we augment x_{11} as much as possible ;

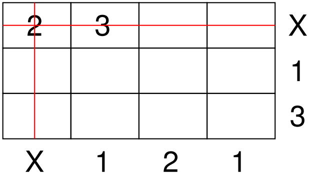
				5
				1
				3
2	4	2	1	

The upper-left corner (ULC) method

2				3
				1
				3
X	4	2	1	

- We start in the upper left cell with variable x_{11}, and we augment x_{11} as much as possible ;
- We eliminate from the tableau the row or column which is saturated. We lower by the value of x_{11} the row or column which is not, if any ;

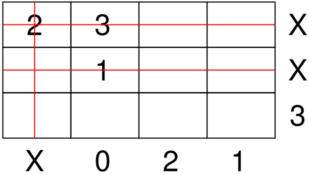
The upper-left corner (ULC) method



2	3			X
				1
				3
X	1	2	1	

- We start in the upper left cell with variable x_{11}, and we augment x_{11} as much as possible ;
- We eliminate from the tableau the row or column which is saturated. We lower by the value of x_{11} the row or column which is not, if any ;
- We continue this procedure iteratively on the remaining sub-tableau ;

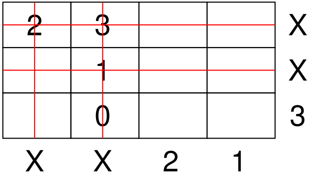
The upper-left corner (ULC) method



2	3			X
	1			X
				3
X	0	2	1	

- We start in the upper left cell with variable x_{11}, and we augment x_{11} as much as possible ;
- We eliminate from the tableau the row or column which is saturated. We lower by the value of x_{11} the row or column which is not, if any ;
- We continue this procedure iteratively on the remaining sub-tableau ;
- In the case where the augmenting procedure saturates both the row and the column, we eliminate only one of them, not both ;

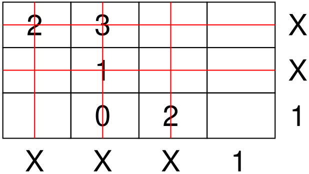
The upper-left corner (ULC) method



2	3			X
	1			X
	0			3
X	X	2	1	

- We start in the upper left cell with variable x_{11}, and we augment x_{11} as much as possible ;
- We eliminate from the tableau the row or column which is saturated. We lower by the value of x_{11} the row or column which is not, if any ;
- We continue this procedure iteratively on the remaining sub-tableau ;
- In the case where the augmenting procedure saturates both the row and the column, we eliminate only one of them, not both ;
- If we have a 0 saturation value (as in the example) the solution is degenerate but valid ;

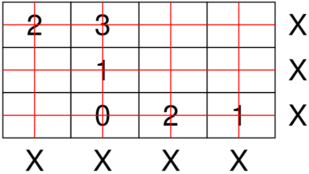
The upper-left corner (ULC) method



2	3			X
	1			X
	0	2		1
X	X	X	1	

- We start in the upper left cell with variable x_{11}, and we augment x_{11} as much as possible ;
- We eliminate from the tableau the row or column which is saturated. We lower by the value of x_{11} the row or column which is not, if any ;
- We continue this procedure iteratively on the remaining sub-tableau ;
- In the case where the augmenting procedure saturates both the row and the column, we eliminate only one of them, not both ;
- If we have a 0 saturation value (as in the example) the solution is degenerate but valid ;

The upper-left corner (ULC) method



2	3			X
	1			X
	0	2	1	X
X	X	X	X	

- We start in the upper left cell with variable x_{11}, and we augment x_{11} as much as possible ;
- We eliminate from the tableau the row or column which is saturated. We lower by the value of x_{11} the row or column which is not, if any ;
- We continue this procedure iteratively on the remaining sub-tableau ;
- In the case where the augmenting procedure saturates both the row and the column, we eliminate only one of them, not both ;
- If we have a 0 saturation value (as in the example) the solution is degenerate but valid ;
- The last case saturates both its row and column. In the end we have a sequence of n+m-1 variables that forms a basis.

Initial FBS = $\{x_{11} = 2, x_{12} = 3, x_{22} = 1, x_{32} = 0, x_{33} = 2, x_{34} = 1\}$

Justification of the ULC method

- All the variables found this way are positive (may be null) ;
- We assign $m + n - 1$ variables;
- The last assignment saturates all $m + n$ constraints, since all rows and columns are saturated;
- The ULC method ensures the sequence of variables assign contains no loop;
- Therefore by the loop theorem, the basis found is a FBS;

Weaknesses of the upper-left corner method

- The ULC method yields a FBS, which may be very far from optimal;
- Often the ULC method produces degenerate FBS (many zeros in the basis) ;
- The cost is not taken into account ;
- Other methods can fix these problems.

The lowest cost (LC) method

2	3	5	6	5
2	1	3	5	10
3	8	4	6	15
12	8	4	6	

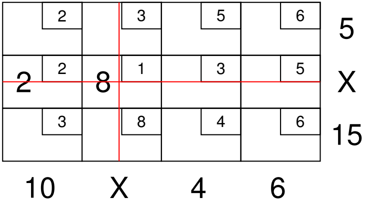
- We start with the variable x_{ij} with the minimum transport cost;

The lowest cost (LC) method

2	3	5	6	5
2	1 8	3	5	2
3	8	4	6	15
12	X	4	6	

- We start with the variable x_{ij} with the minimum transport cost;
- We saturate its value and we close the corresponding row or column, decreasing the constraint accordingly in the non-closed row or column ;

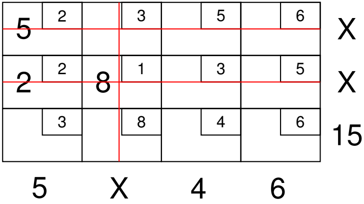
The lowest cost (LC) method



$\boxed{2}$	$\boxed{3}$	$\boxed{5}$	$\boxed{6}$	5
$\boxed{2}$ 2	$\boxed{1}$ 8	$\boxed{3}$	$\boxed{5}$	X
$\boxed{3}$	$\boxed{8}$	$\boxed{4}$	$\boxed{6}$	15
10	X	4	6	

- We start with the variable x_{ij} with the minimum transport cost;
- We saturate its value and we close the corresponding row or column, decreasing the constraint accordingly in the non-closed row or column ;
- We repeat the procedure with all the non-closed cells ;

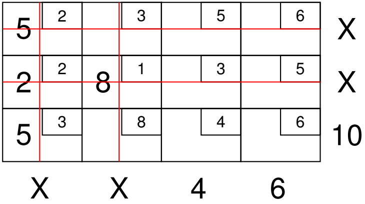
The lowest cost (LC) method



2 5	3	5	6	X
2 2	1 8	3	5	X
3	8	4	6	15
5	X	4	6	

- We start with the variable x_{ij} with the minimum transport cost;
- We saturate its value and we close the corresponding row or column, decreasing the constraint accordingly in the non-closed row or column ;
- We repeat the procedure with all the non-closed cells ;
- If saturating a variable would satisfy both the row and column, we close only one of them;

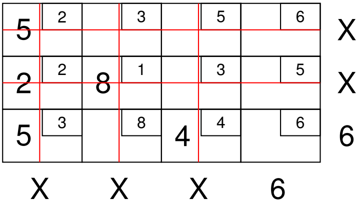
The lowest cost (LC) method



2 5	3	5	6	X
2 2	1 8	3	5	X
3 5	8	4	6	10
X	X	4	6	

- We start with the variable x_{ij} with the minimum transport cost;
- We saturate its value and we close the corresponding row or column, decreasing the constraint accordingly in the non-closed row or column ;
- We repeat the procedure with all the non-closed cells ;
- If saturating a variable would satisfy both the row and column, we close only one of them;

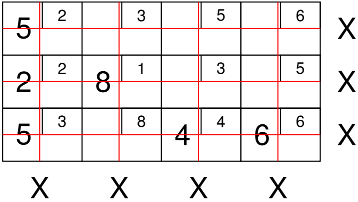
The lowest cost (LC) method



5 2	3	5	6	X
2 2	8 1	3	5	X
5 3	8	4 4	6	6
X	X	X	6	

- We start with the variable x_{ij} with the minimum transport cost;
- We saturate its value and we close the corresponding row or column, decreasing the constraint accordingly in the non-closed row or column ;
- We repeat the procedure with all the non-closed cells ;
- If saturating a variable would satisfy both the row and column, we close only one of them;

The lowest cost (LC) method



2 5	3	5	6	X
2 2	1 8	3	5	X
3	8	4 4	6 6	X
X	X	X	X	

- We start with the variable x_{ij} with the minimum transport cost;
- We saturate its value and we close the corresponding row or column, decreasing the constraint accordingly in the non-closed row or column ;
- We repeat the procedure with all the non-closed cells ;
- If saturating a variable would satisfy both the row and column, we close only one of them;
- When there is only once cell left, close both its row and column.

Justification of the LC method

- Using the same arguments as for the ULC, the solution we find is an initial FBS ;
- This time we can hope that the solution has a lower total cost than the ULC ;
- However it is possible to find unfavourable counter-examples for this method :

6	7	8	10
15	80	78	15
15	5	5	

- The method of VOGEL can avoid these problems, but we don't consider it in this course.

The simplex for transport problems

Steps of the algorithm

1. As long as we have not found the optimum (see below), then iterate:
    1.1 Determine which variable should enter the system (see below);
    1.2 Find the closed loop implicating the new variable and a subset of existing variables ;
    1.3 Enumerate the variables in the loop from the entering variable (with index 0) ;
    1.4 Find the odd index cell with the smallest value θ;
    1.5 Augment by θ all the even index variables in the loop and reduce by θ all the odd index variables ;
    1.6 None of the other variables change.

Illustration on the electricity transport problem

We recall the electricity distribution problem of the previous course :

	Ville 1	Ville 2	Ville 3	Ville 4	Offre
centrale 1	0 $\boxed{8}$	0 $\boxed{6}$	0 $\boxed{10}$	0 $\boxed{9}$	35
centrale 2	0 $\boxed{9}$	0 $\boxed{12}$	0 $\boxed{13}$	0 $\boxed{7}$	50
centrale 3	0 $\boxed{14}$	0 $\boxed{9}$	0 $\boxed{16}$	30 $\boxed{5}$	40
Demande	45	20	30	30	

Resolution of the electricity problem

8	6	10	9	35
9	12	13	7	50
14	9	16	5	40
45	20	30	30	

- Before initialization by the ULC Method

Resolution of the electricity problem

8 35	6	10	9	35
9 10	12 20	13 20	7	50
14	9	16 10	5 30	40
45	20	30	30	

- After initialization by the ULC Method.

Reduced costs

- Recall the formulat from the simplex method $\overline{\mathbf{c}}_e^\intercal = \mathbf{c}_e^\intercal - \mathbf{c}_b^\intercal \mathbf{B}^{-1}\mathbf{E}$.
- Here we need to compute $\mathbf{c}_b^\intercal \mathbf{B}^{-1}$, which has the same length as $\mathbf{c}_b$, i.e. $m + n - 1$.
- We write $\mathbf{c}_b^\intercal \mathbf{B}^{-1} = [u_2 u_3 \dots u_m v_1 v_2 \dots v_n]$, where u_i are the production constraints and the v_i the consumption constraints. Note that we have abandonned one constraints to keep $m + n - 1$ equations.
- The reduced cost of a basis variable is zero, so for each basis variable x_{ij}, we have

$$c_{ij} = \mathbf{c}_b^\intercal \mathbf{B}^{-1} \mathbf{a}_{ij}$$

  where c_{ij} is the cost associated with variable x_{ij} and $\mathbf{a}_{ij}$ the column of $\mathbf{A}$ (minus its first line) associated with the same variable.

The electricity distribution LP problem

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{31} \\ x_{32} \\ x_{33} \\ x_{34} \end{bmatrix} = \begin{bmatrix} 35 \\ 50 \\ 40 \\ 45 \\ 20 \\ 30 \\ 30 \end{bmatrix}$$

NOTE: we must eliminate on line in this matrix, for instance the first !

Illustration on the electricity problem

8 35	6	10	9	35
9 10	12 20	13 20	7	50
14	9	16 10	5 30	40
45	20	30	30	

- $\bar{c}_{11} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - 8 =$

  $v_1 - 8 = 0$

Illustration on the electricity problem

8 35	6	10	9	35
9 10	12 20	13 20	7	50
14	9	16 10	5 30	40
45	20	30	30	

- $v_1 - 8 = 0$
- $\bar{c}_{21} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - 9 =$

  $u_2 + v_1 - 9 = 0$

Illustration on the electricity problem

8 35	6	10	9	35
9 10	12 20	13 20	7	50
14	9	16 10	5 30	40
45	20	30	30	

- $v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $$\bar{c}_{22} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} - 12 =$$

  $u_2 + v_2 - 12 = 0$

Illustration on the electricity problem

8 35	6	10	9	35
9 10	12 20	13 20	7	50
14	9	16 10	5 30	40
45	20	30	30	

- $v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $u_2 + v_2 - 12 = 0$
- $\bar{c}_{23} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} - 13 =$

  $u_2 + v_3 - 13 = 0$

Illustration on the electricity problem

35 8	6	10	9	35
10 9	20 12	20 13	7	50
14	9	10 16	30 5	40
45	20	30	30	

- $v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $u_2 + v_2 - 12 = 0$
- $u_2 + v_3 - 13 = 0$
- $\bar{c}_{33} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} - 16 =$

  $u_3 + v_3 - 16 = 0$

Illustration on the electricity problem

8 35	6	10	9	35
9 10	12 20	13 20	7	50
14	9	16 10	5 30	40
45	20	30	30	

- $v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $u_2 + v_2 - 12 = 0$
- $u_2 + v_3 - 13 = 0$
- $u_3 + v_3 - 16 = 0$

$$\bar{c}_{34} = [u_2 u_3 v_1 v_2 v_3 v_4] \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - 5 =$$

$$u_3 + v_4 - 5 = 0$$

Illustration on the electricity problem

8 35	6	10	9	35
9 10	12 20	13 20	7	50
14	9	16 10	5 30	40
45	20	30	30	

- $v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $u_2 + v_2 - 12 = 0$
- $u_2 + v_3 - 13 = 0$
- $u_3 + v_3 - 16 = 0$
- $u_3 + v_4 - 5 = 0$

We see that if we write $u_1 = 0$, all the equations have the same form, i.e. $u_i + v_j = c_{ij}$ for the basis variable x_{ij}.

Illustration on the electricity problem

8 35	6	10	9	35
9 10	12 20	13 20	7	50
14	9	16 10	5 30	40
45	20	30	30	

- $u_1 = 0$
- $u_1 + v_1 - 8 = 0$
- $u_2 + v_1 - 9 = 0$
- $u_2 + v_2 - 12 = 0$
- $u_2 + v_3 - 13 = 0$
- $u_3 + v_3 - 16 = 0$
- $u_3 + v_4 - 5 = 0$

Very easy to solve !!

We see that if we write $u_1 = 0$, all the equations have the same form, i.e. $u_i + v_j = c_{ij}$ for the basis variable x_{ij}.

Illustration on the electricity problem

35 8	6	10	9	35
10 9	20 12	20 13	7	50
14	9	10 16	30 5	40
45	20	30	30	

- $u_1 = 0$
- $v_1 = 8$
- $u_2 = 1$
- $v_2 = 11$
- $v_3 = 12$
- $u_3 = 4$
- $v_4 = 1$

We see that if we write $u_1 = 0$, all the equations have the same form, i.e. $u_i + v_j = c_{ij}$ for the basis variable x_{ij}.

Computing the remaining reduced costs

- Once we have computed the reduced costs u_i and v_j the rest is easy ;
- Indeed we apply the same column-wise technique and the usual reduced cost formula simplifies to :

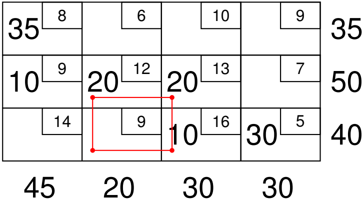
$$\bar{c}_{ij} = c_{ij} - u_i - v_j$$

  note: this formula applies to *all* costs, since reduced costs are zero for the basis variables.
- In our example, this yields :

$$\begin{array}{ll} \bar{c}_{12} = 6 - 11 + 6 = -5 & \bar{c}_{13} = 10 - 0 - 12 = -2 \\ \bar{c}_{14} = 9 + 0 - 1 = 8 & \bar{c}_{24} = 7 - 1 - 1 = 5 \\ \bar{c}_{31} = 14 - 4 - 8 = 2 & \bar{c}_{32} = 9 - 4 - 11 = -6 \end{array}$$

- Here we want to minimize costs, so we choose the most negative reduced cost, i.e. $\bar{c}_{32}$. Therefore x_{32} enters the basis.

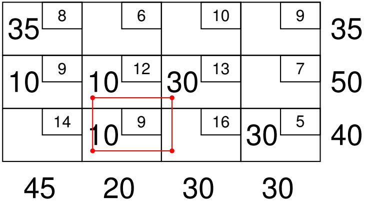
Swapping variables



8 35	6	10	9	35
9 10	12 20	13 20	7	50
14	9	16 10	5 30	40
45	20	30	30	

- Here x_{32} must enter the basis ;
- This would create a unique loop $x_{32} - x_{22} - x_{23} - x_{33}$;
- Numbering the nodes from the entering variable x_{32}, We consider the *even-numbered* nodes of this loop: x_{22} et x_{33}. The smallest value, called θ, is $\theta = 10$;
- We augment the odd-numbered nodes (so here x_{32} and x_{23} from the amount θ and we lower the even-numbered nodes by the same amount ;
- As a result, we have effectively swapped x_{33} with x_{32}.

Swapping variables



35 8	6	10	9	35
10 9	10 12	30 13	7	50
14	10 9	16	30 5	40
45	20	30	30	

- We must recompute the reduced costs
- We must solve

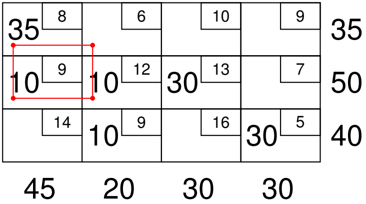
$$\begin{array}{lll} u_1 = 0 & u_1 + v_1 = 8 & u_2 + v_1 = 9 \\ u_2 + v_2 = 12 & u_2 + v_3 = 13 & u_3 + v_2 = 9 \\ u_3 + v_4 = 5 & & \end{array}$$

- We must then recompute $\bar{c}_{ij} = c_{ij} - u_i - v_j$ for all the non-basis variables.
- We find the only negatives are

$$\bar{c}_{12} = -5, \bar{c}_{24} = -1, \bar{c}_{13} = -2,$$

- So x_{12} enters the basis.

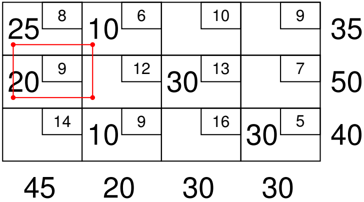
Swapping variables



8 35	6	10	9	35
9 10	12 10	13 30	7	50
14	9 10	16	5 30	40
45	20	30	30	

- Variable x_{12} enters the basis ;
- This would create a unique loop $x_{12} - x_{22} - x_{21} - x_{11}$;
- The even-numbered nodes from the entering variables are x_{22} et x_{11}. The value of θ is the smallest, i.e. 10 ;
- We augment the odd-numbered nodes (x_{12} and x_{21} by the amount θ and we lower the even-numbered nodes by the same amount ;
- Effectively, we swap x_{22} with x_{12}.

Swapping variables



8 25	6 10	10	9	35
9 20	12	13 30	7	50
14	9 10	16	5 30	40
45	20	30	30	

- We recompute the reduced cost one more time
- We must solve

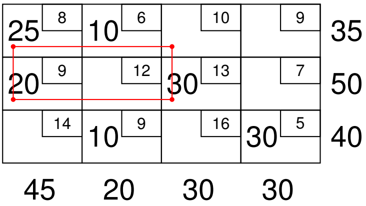
$$\begin{array}{lll} u_1 = 0 & u_1 + v_1 = 8 & u_1 + v_2 = 6 \\ u_2 + v_1 = 9 & u_2 + v_3 = 13 & u_3 + v_2 = 9 \\ u_3 + v_4 = 5 & & \end{array}$$

- We use the formula $\bar{c}_{ij} = c_{ij} - u_i - v_j$ for all non-basis variables.
- The only negative reduced cost is

$$\bar{c}_{13} = -2$$

- Therefore x_{13} enters the basis.

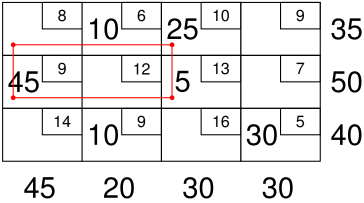
Swapping variables



25 8	10 6	10	9	35
20 9	12	30 13	7	50
14	10 9	16	30 5	40
45	20	30	30	

- Variable x_{13} enters the basis ;
- This creates a unique loop $x_{13} - x_{23} - x_{21} - x_{11}$;
- The even-numbered nodes are x_{23} et x_{11}. The value of θ is 25 (the lowest);
- We augment the odd-numbered nodes (i.e. x_{13} and x_{21} of the amount θ and we lower the even-numbered one by the same amount ;
- Effectively, we swap x_{11} with x_{13}.

Swapping variables



8	6 10	10 25	9	35
9 45	12	13 5	7	50
14	9 10	16	5 30	40
45	20	30	30	

- We recompute the reduced costs
- We must solve

$$\begin{array}{lll} u_1 = 0 & u_1 + v_2 = 6 & u_1 + v_3 = 10 \\ u_2 + v_1 = 9 & u_2 + v_3 = 13 & u_3 + v_2 = 9 \\ u_3 + v_4 = 5 & & \end{array}$$

- We must compute $\bar{c}_{ij} = c_{ij} - u_i - v_j$ for all non-basis variables
- There are no negative reduced costs
- This is the optimum !
- The optimum cost is $z = 6 * 10 + 10 * 25 + 45 * 9 + 5 * 13 + 10 * 9 + 30 * 5 = 1020.$

Notes

Important notes

- There is always only one unique loop that appears when we add a variable to a basis ;
- This loop is not necessarily simple (consisting of 4 variables), contrary to what we have shown here.
- A loop contains an even number of nodes, in which no three nodes or more are consecutive (same row or same column), and does form a loop (starts and ends at the same node)
- We will show an example with a more complex transshipment problem.

Définition of a transshipment problem

- A pure transport problem ships goods, products, etc from the producer to the consumer directly. It is represented by a bipartite graph;
- In a transshipment problem, there might exist intermediary nodes. The representation is an arbitrary directed graph ;
- However we can always transform transshipment problem into pure transport problems.

Example of a transshipment problem

- Consider company W who makes toys. It has producing units in Montpellier, and in Douais. The Montpellier unit can produce 150 toys a day, the Douais one can produce 200 toys a day.
- The toys are sent by road to retailers in Lyon and Brest. Customers in these cities are expected to by 130 toys a day.
- Because of expensive road costs, it might be cheaper to exploit the railroad networks by going through Nevers and/or Castres. The cost matrix is as follows:

	M	D	N	C	L	B
M	0	-	8	13	25	28
D	-	0	15	12	26	25
N	-	-	0	6	16	17
C	-	-	6	0	14	16
L	-	-	-	-	0	-

Transforming into a transport problem

- It is easy to transform this problem into a transport one ;
- Typically, we will split intermediary nodes into consumer and producer, and treat them separately.

Rules for splitting nodes

- Intermediary nodes split in two halves have no transport cost between them;
- Their capacity (either as producer or consumer) is the total capacity of the network before the split.

The balanced equivalent transport problem for the toy transshipment is as follows:

	N	C	L	B	V	
M	8	13	25	28	0	150
D	15	12	26	25	0	200
N	0	6	16	17	0	350
C	6	0	14	16	0	350
	350	350	130	130	90	

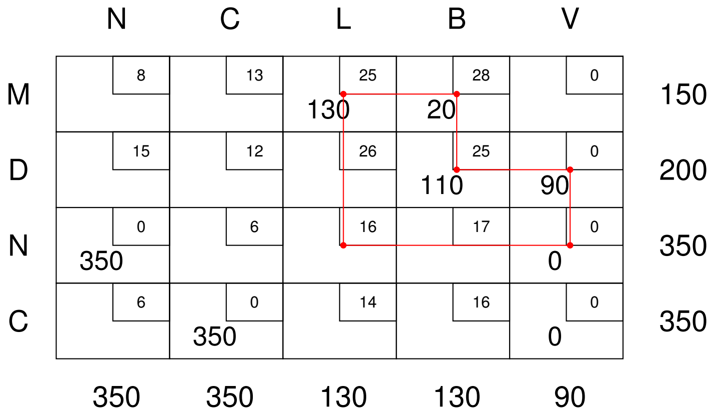
Inital basis for the toy problem

With the lowest-cost method, verify that one finds:

	N	C	L	B	V	
M	8	13	25 130	28 20	0	150
D	15	12	26	25 110	0 90	200
N	0 350	6	16	17	0 0	350
C	6	0 350	14	16	0 0	350
	350	350	130	130	90	

Improving the solution for the toy problem

Compute the u_i, v_j. In this problem, sometime loops are not simple, for instance for variable x_{33}



	N	C	L	B	V	
M	8	13	25 130	28 20	0	150
D	15	12	26	25 110	0 90	200
N	0 350	6	16	17	0 0	350
C	6	0 350	14	16	0 0	350
	350	350	130	130	90	

Augmenting entering variables

- In such cases, we still number nodes in the loop from the entering variable, and find θ, the lowest value of the odd-numbered nodes, then augment the even-numbered variable by θ and lower the odd-numbered nodes by θ, as before (but there are more than two nodes to lower and more than two to augment).
- the number of nodes to lower and augment is the same (3 in this case).
- It may be that $\theta = 0$. In this case, augmentation is impossible with this non-basis variable, try with the next one in order from most negative.

Conclusion

- Transport, assignment and transshipment problems are particular cases of LP.
- They are not solved by the generic simplex algorithm, because more effective methods exist, that does not involve linear algebra;
- There is not extra costs involved in these problems when dealing with integers.

General conclusion

- This course is an introduction to *operations research*;
- This is a very important domain in practice, plenty of jobs in this area, very few trained professionals;
- Application in industry: 20% of containers ship empty to the USA (!)
- New theoretical results: it is possible under some conditions to process signals beyond the sampling limit of Shannon (Emnanuel Candes (France) and Terence Tao (Australia), Fields medalist 2008), opening the field of *compressive sensing*).
- Very few people know optimization, please let me know if you are interested in this area.