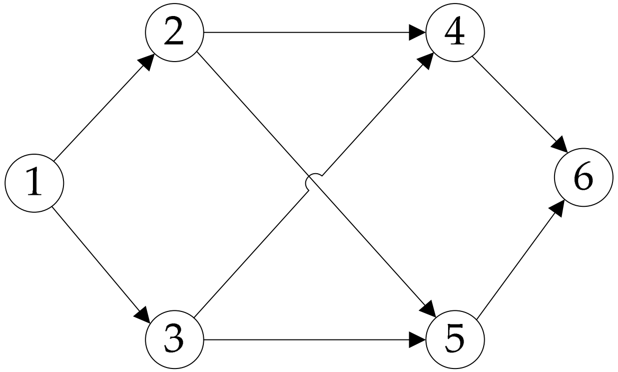
Modelação e Optimização de Sistemas em Rede

1. Introdução
2. Representação em Rede e Representação Algébrica de alguns Problemas de Optimização
    - i. Problema de Transportes
    - ii. Problema de Afectação
    - iii. Problema de Caminho Mais Curto
    - iv. Problema de Fluxo Máximo
3. Representação em Folhas de Cálculo e Resolução (*Solver* do MS Excel)
    - i. Modelos de Programação Linear
    - ii. Resolução com o Solver do MS Excel dos problemas de optimização em redes (exemplos)
4. Algoritmos de Optimização e Heurísticas
    - i. Método Simplex para Transportes …(Problema de Transportes)
    - ii. Algoritmo Húngaro …………………..(Problema de Afectação)
    - iii. Algoritmo de Dijkstra ………………..(Problema de Caminho Mais Curto)
    - iv. Algoritmo de Kruskal ………………..(Problema da Árvore de Suporte de Custo Mínimo)
    - v. Heurística do Vizinho Mais Próximo (Problema do Caixeiro Viajante)

Cláudio M. Martins Alves
Departamento de Produção e Sistemas
Escola de Engenharia – Universidade do Minho
Gualtar – 4710-057 Braga – PORTUGAL
tel.: 253 604765
gabinete: EEII156
claudio@dps.uminho.pt :: pessoais.dps.uminho.pt/claudio

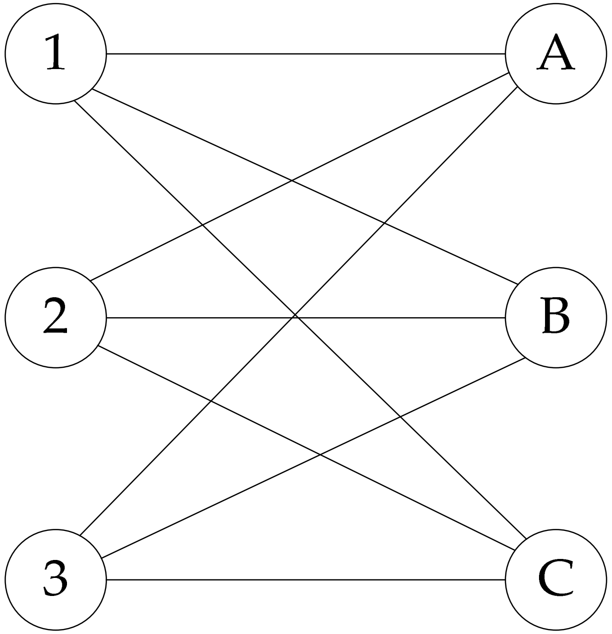
1. Introdução

- Os modelos de fluxos em rede são ferramentas que permitem formular e resolver vários problemas de optimização
- Esses modelos são definidos a partir de *grafos*: estruturas constituídas por vértices, e arcos que interligam pares de vértices definindo assim relações entre eles
- Formalmente, um grafo G é definido através de um conjunto V de vértices e A de arcos
- No exemplo seguinte o grafo G é constituído por um conjunto de vértices V={1,2,3,4,5,6}, e um conjunto de arcos A={(1,2),(1,3),(2,4),(2,5),(3,4),(3,5),(4,6),(5,6)}.

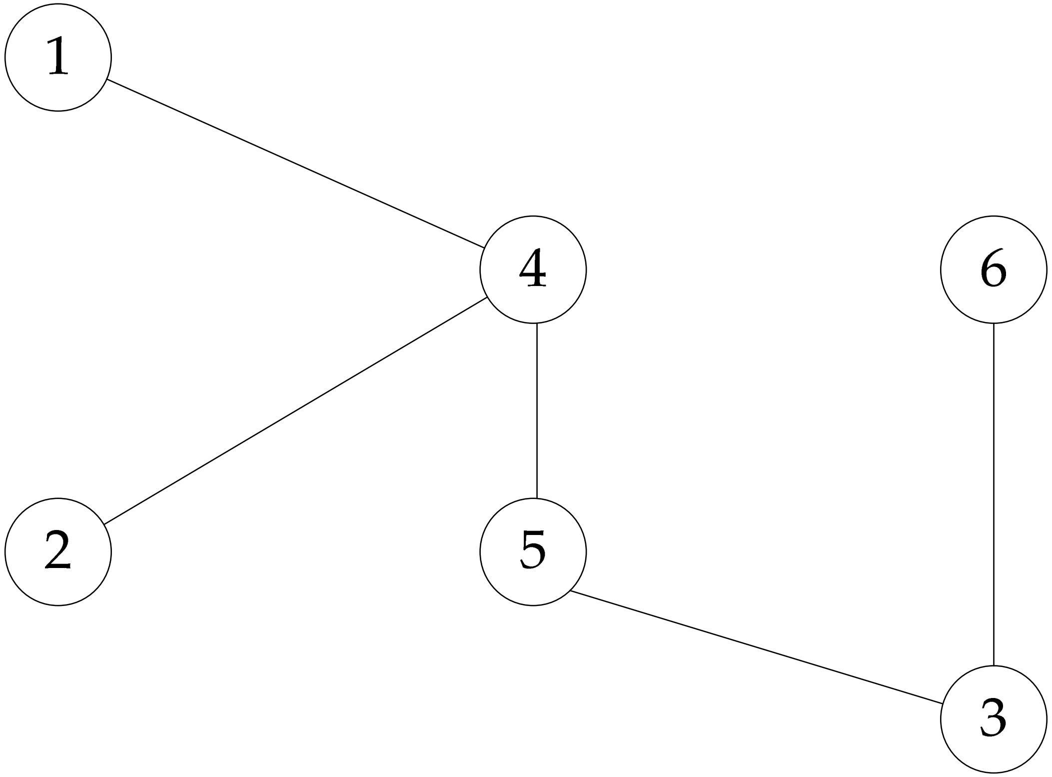


- Algumas definições associadas aos grafos:
  - Grafo orientado: grafo em que os arcos têm uma direcção (o grafo acima é um grafo orientado); num arco (*a*,*b*)∈ A, o vértice *a* corresponde à origem do arco e *b* ao seu destino;
  - Grafo não-orientado: grafo em que os arcos não têm direcção;
  - Caminho: sequência de arcos que ligam dois vértices (no grafo acima (1,2), (2,4), (4,6) é um caminho entre os vértices 1 e 6) ;
  - Caminho simples: caminho que não passa mais de uma vez pelo mesmo vértice;
  - Ciclo: caminho com origem e destino num mesmo vértice do grafo;
  - Ciclo orientado (circuito): num grafo orientado, consiste num ciclo em que todos os arcos são percorridos no sentido definido pela sua direcção (por oposição aos ciclos não-orientados; no grafo acima, o ciclo (1,2), (2,5), (5,3), (3,1) é não orientado)

- Algumas famílias (tipos) de grafos:
  - Grafo bipartido: grafo $G=(V,A)$ em que é possível dividir o conjunto de vértices em dois subconjuntos V_1 e V_2 de tal modo que $V_1 \cup V_2 = V$, e qualquer arco $(i,j) \in A$ é tal que $i \in V_1$ e $j \in V_2$, ou seja todos os arcos do grafo têm forçosamente um vértice em V_1 e outro em V_2

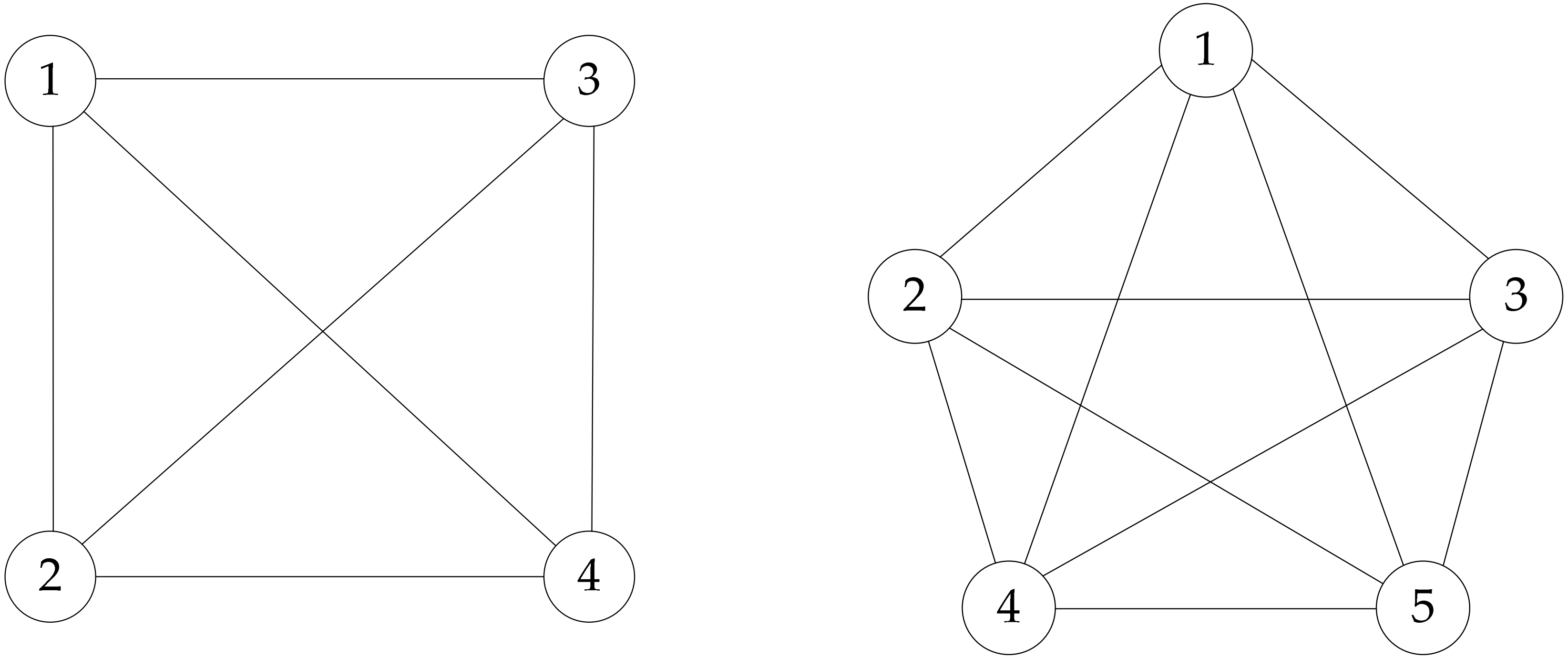


- Árvore: grafo $G=(V,A)$ em que cada par de vértices que pertence a V está ligado por um e um único caminho



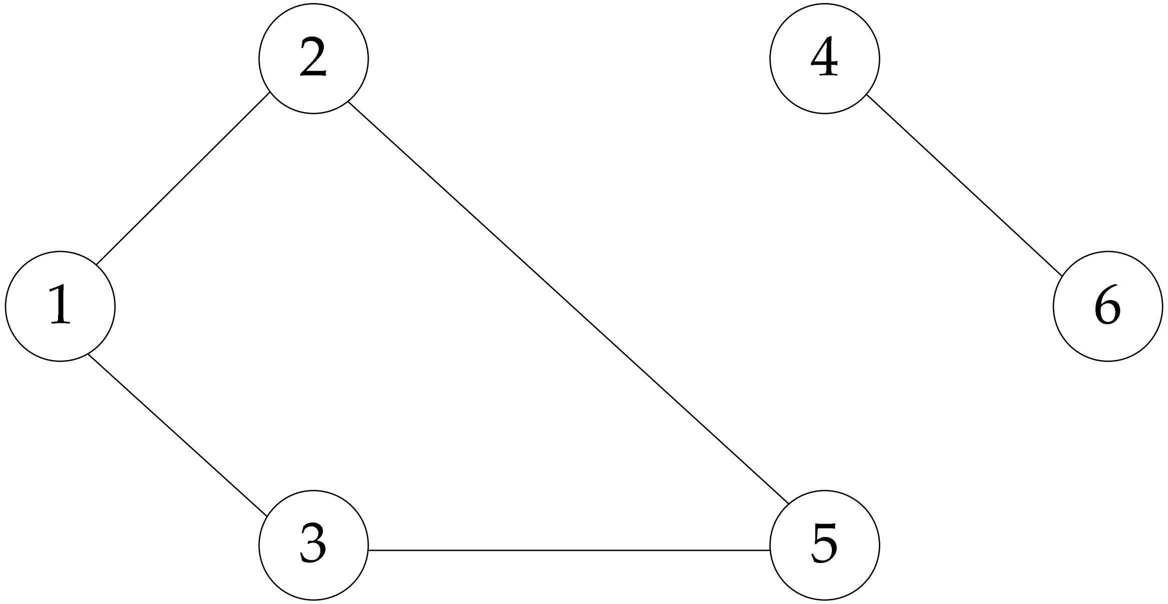
Uma árvore de suporte de um grafo $G=(V,A)$ é uma árvore que inclui todos os vértices do conjunto V.

- Grafo completo: grafo em que cada par de vértices é ligado por um arco



Num grafo completo $G=(V,A)$, o número de arcos $|A|$ é sempre igual a $|V|(|V|-1)/2$.

- Grafo ligado e grafo não-ligado: grafo em que cada par de vértices está ligado por um caminho (por oposição aos chamados grafos não-ligados; todos os grafos acima são grafos ligados; o grafo abaixo é um grafo não-ligado)



- Fluxos em redes:
  - Em alguns casos, podem existir vantagens em associar determinados valores de fluxo aos arcos de um grafo; esse fluxo representa quantidades que são enviadas de uma extremidade à outra de um arco (pessoas, ou produtos, por exemplo)
  - Nesses casos, é comum existirem restrições associadas aos fluxos que podem ser enviados pela rede:
    1. Restrições de capacidade (associadas aos arcos): o fluxo num arco não poderá exceder a capacidade desse arco;
    2. Restrições de conservação de fluxo: aquilo que entra num vértice, deve voltar a sair desse vértice, excepto se o vértice for um vértice de origem ou destino
  - O conceito de *fluxo em redes* está na origem dos chamados modelos de fluxos em redes a partir dos quais é possível formular vários problemas de optimização

2. Representação em Rede e Algébrica de alguns Problemas de Optimização

i. Problema de Transportes

- O problema de transportes consiste em determinar a melhor forma de distribuir produtos, pessoas, ou qualquer outra entidade, a partir de uns pontos de origem até determinados pontos de destino;
- O problema é definido num grafo bipartido, em que um dos conjuntos de vértices identifica as origens do problema, e o outro conjunto os destino do problema;
- A formulação clássica do problema de transportes define-o como um problema de minimização no qual se procura determinar a distribuição dos fluxos pela rede que minimize o custo total de transportes. Contudo, o problema de transportes pode ser visto também como um modelo extensível a outros casos (problemas de maximização de lucros, por exemplo) desde que esses tenham a mesma estrutura.

- Dados do problema:
    - m origens com um determinado nível de oferta s_i (i=1,…,m) (disponibilidade);
    - n destinos com uma determinada procura d_j, $j=1,\ldots,n$;
    - um conjunto de arcos A que define as ligações entre os vértices que representam as origens e os vértices associados aos destinos;
    - custos unitários de transporte c_{ij} associados aos arcos $(i,j)\in A$.
- Objectivo do problema:
    - Determinar o fluxo entre origens e destinos de modo a esgotar a disponibilidade nas origens, e garantir que todas as procuras sejam satisfeitas
    - Minimizar o custo total de transporte

- Condição para a existência de uma solução válida: a soma das disponibilidades deve ser igual à soma das procuras

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$$

- Mesmo quando essa condição não se verifica à partida, é sempre possível proceder à reformulação seguinte:
  - Se $\sum_{i=1}^{m} s_i > \sum_{j=1}^{n} d_j$, isto é, se a disponibilidade total for superior à procura, deve-se adicionar um destino fictício ao problema, com uma procura igual a $\sum_{i=1}^{m} s_i - \sum_{j=1}^{n} d_j$;
  - Se $\sum_{i=1}^{m} s_i < \sum_{j=1}^{n} d_j$, a disponibilidade nas origens é inferior à procura total, e nesse caso, deve-se adicionar uma origem fictícia com uma disponibilidade igual a $\sum_{j=1}^{n} d_j - \sum_{i=1}^{m} s_i$.

- Exemplo de um problema de transportes:

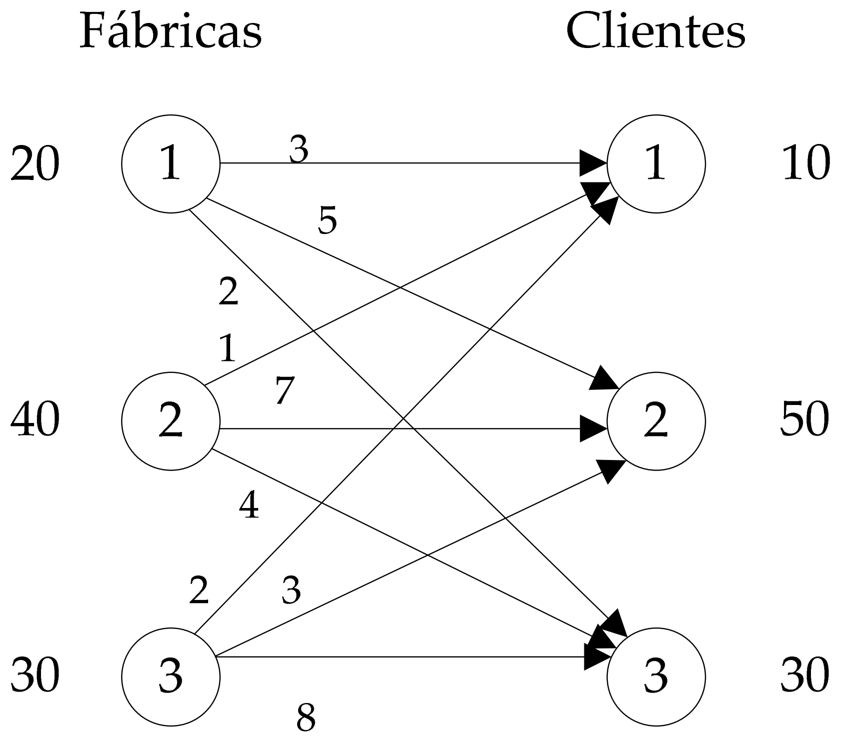
Uma empresa possui 3 fábricas distribuídas pelo país. A capacidade de produção de cada uma dessas fábricas é respectivamente de 20, 40 e 30 unidades. A empresa acaba de assinar um contrato de fornecimento com 3 clientes, no qual se compromete a entregar 10 unidades do seu produto ao 1º cliente, 50 ao 2º cliente, e 30 ao 3º cliente.

A empresa pretende determinar quais devem ser as fábricas que irão fornecer os produtos aos clientes, e em que quantidades, de modo a que o custo total de transporte seja o menor possível. A empresa estimou os custos unitários de transporte (em Unidades Monetárias/artigo), e chegou aos valores indicados na tabela seguinte.

	Cliente 1	*Cliente 2*	*Cliente 3*
Fábrica 1	*3*	*5*	*2*
Fábrica 2	*1*	*7*	*4*
Fábrica 3	*2*	*3*	*8*

- Dados do problema:
  - Origens: 3 fábricas (*m*=3, fábrica 1, 2 e 3) com disponibilidades respectivamente iguais a 20, 40 e 30 unidades;
  - Destinos: 3 clientes (*n*=3, cliente 1, 2 e 3) com procuras iguais respectivamente a 10, 50, e 30 unidades;
  - Custos de transporte iguais às distâncias dadas na tabela;
  - Verifica-se a condição de existência de uma solução válida: $\sum_{i=1}^{3} s_i = 90$, e $\sum_{j=1}^{3} d_j$=90.
- Representação em Rede:
  - Grafo bipartido *G*=(*V*,*A*): *V* é composto por 6 vértices: 3 associados às origens do problema, e 3 associados aos destinos; e existe um arco entre cada par origem/destino (|*A*|=o grafo é um grafo bipartido completo).

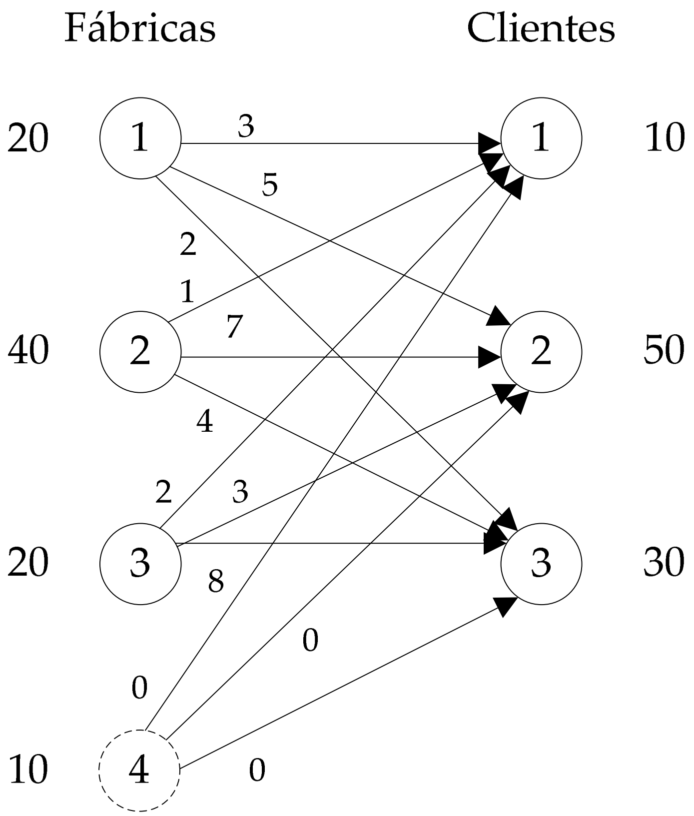
- Rede:



Nota: os valores junto aos vértices representam respectivamente a disponibilidade nas origens e a procura nos destinos

- Alterações aos dados do problema:
  - ❖CASO 1: Vamos supor que a fábrica 3 apenas podia fornecer 20 unidades (nesse caso a procura excede a oferta em 10 unidades)
    - É necessário adicionar uma origem fictícia com uma disponibilidade suficiente para compensar essa diferença (na figura abaixo, o vértice 4 com contorno a tracejado representa essa origem fictícia);
    - Os fluxos que saem desse vértice têm um significado particular: representam procura não satisfeita no respectivo destino (por exemplo, se o fluxo no arco (4,1) for igual a 5, isso significa que o cliente 1 não receberá 5 das unidades que pediu).

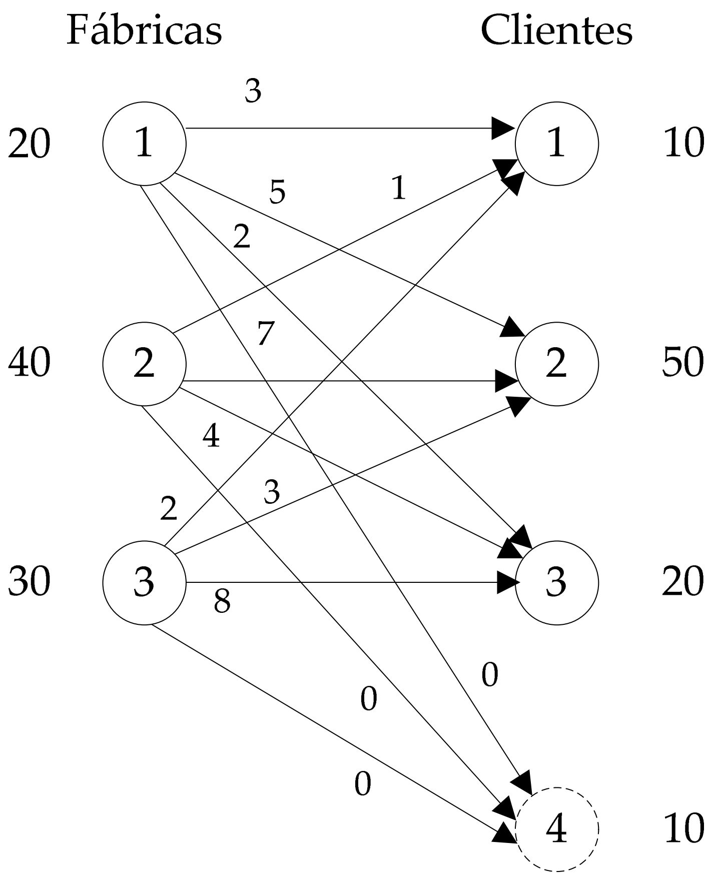
- A rede:



Nota: nem sempre os custos de transporte nos arcos que saem de uma origem fictícia são iguais a 0. Pense por exemplo no caso em que o cliente impõe uma penalidade (custo) por cada unidade que não lhe é fornecida.

- CASO 2: Partindo dos dados do problema original, vamos agora supor que o cliente 3 apenas apenas pretendia adquirir 20 unidades (nesse caso a oferta excede a procura em 10 unidades)
    - Nesse caso, é necessário adicionar um destino fictício com uma procura igual a 10 unidades (na figura abaixo, o vértice 4 com contorno a tracejado representa esse destino fictício);
    - Os fluxos que têm por destino esse vértice são na realidade unidades que não chegam a sair das fábricas (por exemplo, se o fluxo no arco (2,4) for igual a 10, isso significa que a fábrica 2 não envia 10 unidades para qualquer um dos clientes).

- A rede:



Nota: a observação feita atrás relativamente aos custos associados a arcos ligados a origens fictícias, também se aplica aos casos em que é necessário adicionar destinos fictícios

- Representação algébrica:
    - As quantidades que devem ser transportadas em cada arco (os fluxos) são as incógnitas do problema; para qualquer arco $(i,j)\in A$, vamos designar por x_{ij} o valor do fluxo entre a origem i e o destino j;
    - O critério de optimização num problema de transportes (a *função objectivo*) consiste em minimizar o custo total de transporte, o qual corresponde à soma das contribuições em cada arco:
        - Custo Total = $\sum_{(i,j)\in A} c_{ij}x_{ij}$
        - Para o exemplo acima, o custo total é igual a

$$\sum_{(i,j)\in A} c_{ij}x_{ij} = 3x_{11} + 5x_{12} + 2x_{13} + x_{21} + 7x_{22} + 4x_{23} + 2x_{31} + 3x_{32} + 8x_{33}$$

- Existem dois grandes conjuntos de restrições num problema de transportes:
  - Restrições nas origens: que obrigam a que o total das quantidades que saem de uma origem sejam iguais à disponibilidade dessa origem
    - $\sum_{(i,j)\in A} x_{ij} = s_i, \quad i = 1, \dots, m;$
  - Restrições nos destinos: que obrigam a que a soma das quantidades que fluem em direcção a um determinado destino sejam iguais à procura nesse destino
    - $\sum_{(i,j)\in A} x_{ij} = d_j, \quad j = 1, \dots, n;$
- Uma vez que as quantidades transportadas são não-negativas, as variáveis x_{ij} também terão de o ser:
    - $x_{ij} \geq 0, \quad \forall (i,j) \in A.$

- Se juntarmos todos esses elementos, obtemos um modelo que é uma representação algébrica do problema de transportes;
- Para o caso geral, temos o seguinte modelo:

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

sujeito a

$$\sum_{(i,j)\in A} x_{ij} = s_i\,, i = 1, \dots, m,$$

$$\sum_{(i,j)\in A} x_{ij} = d_j\,, j = 1, \dots, n,$$

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

- Esse modelo é um modelo de Programação Linear (todas as expressões são lineares nas variáveis)

- No caso do problema do exemplo acima, esse modelo toma a seguinte forma:

Minimizar $z = 3x_{11} + 5x_{12} + 2x_{13} + x_{21} + 7x_{22} + 4x_{23} + 2x_{31} + 3x_{32} + 8x_{33}$

sujeito a

$$x_{11} + x_{12} + x_{13} = 20$$
$$x_{21} + x_{22} + x_{23} = 40$$
$$x_{31} + x_{32} + x_{33} = 30$$

Restrições nas origens
(ou de disponibilidade)

$$x_{11} + x_{21} + x_{31} = 10$$
$$x_{12} + x_{22} + x_{32} = 50$$
$$x_{13} + x_{23} + x_{33} = 30$$

Restrições nos destinos
(ou de procura)

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33} \geq 0$$

- Uma vez que num problema de transportes se assume que a disponibilidade total é igual à procura total, uma das restrições desse modelo será sempre redundante

- É interessante observar a estrutura matricial dos coeficientes do modelo:
  - Para o modelo do exemplo:

	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	x_{31}	x_{32}	x_{33}		
Fábrica 1	1	1	1							=	20
Fábrica 2				1	1	1				=	40
Fábrica 3							1	1	1	=	30
Cliente 1	1			1			1			=	10
Cliente 2		1			1			1		=	50
Cliente 3			1			1			1	=	30
	3	5	2	1	7	4	2	3	8		

- Na coluna associada a uma variável x_{ij}, existem apenas dois coeficientes não-nulos (iguais a 1) nas linhas associadas às restrições, um na restrição associada à origem i, e outro na restrição associada ao destino j;

- Para o caso geral:

$$
\begin{array}{c|cccccccccccccccc|c|c}
 & x_{11} & x_{12} & x_{13} & \dots & x_{1n} & x_{21} & x_{22} & x_{23} & \dots & x_{2n} & \dots & x_{m1} & x_{m2} & x_{m3} & \dots & x_{mn} & & \\
\hline
 & 1 & 1 & 1 & \dots & 1 & & & & & & & & & & & & = & s_1 \\
 & & & & & & 1 & 1 & 1 & \dots & 1 & & & & & & & = & s_2 \\
 & & & & & & & & & & & \dots & & & & & & & \\
 & & & & & & & & & & & & 1 & 1 & 1 & \dots & 1 & = & s_m \\
 & 1 & & & & & 1 & & & & & & 1 & & & & & = & b_1 \\
 & & 1 & & & & & 1 & & & & & & 1 & & & & = & b_2 \\
 & & & 1 & \dots & & & & 1 & \dots & & \dots & & & 1 & & & = & \\
 & & & & & 1 & & & & & 1 & & & & & & 1 & & b_n \\
\hline
 & c_{11} & c_{12} & c_{13} & \dots & c_{1n} & c_{21} & c_{22} & c_{23} & \dots & c_{2n} & \dots & c_{m1} & c_{m2} & c_{m3} & \dots & c_{mn} & &
\end{array}
$$

- Essa estrutura confere ao problema de transportes a chamada *propriedade da integralidade*: se todas as disponibilidades s_i e procuras b_j forem valores inteiros, a solução óptima do problema será sempre uma solução em que todas as variáveis assumem valores inteiros.

ii. Problema de Afectação

- O problema de afectação é um caso especial do problema de transportes: os dois problemas partilham a maior parte das suas características; o problema de afectação é mais restritivo relativamente a um dos seus parâmetros;
- O problema de afectação consiste em determinar a melhor forma de afectar entidades (pessoas ou máquinas, por exemplo) de um conjunto às entidades (tarefas, por exemplo) de outro conjunto;
- Num problema de afectação, a disponibilidade em qualquer uma das origens é sempre igual a 1; de igual forma, a procura num destino terá sempre o valor 1;
- A aplicação mais comum dos problemas de afectação consiste em determinar como devem ser associadas pessoas a tarefas; à semelhança do que acontece com o problema de transportes, o problema de afectação pode ser aplicado a muitos outros casos.

- Dados do problema (formulação clássica):
    - m entidades (pessoas, máquinas, por exemplo);
    - n tarefas;
    - um conjunto de arcos A entre entidades do primeiro conjunto e tarefas;
    - custos de afectação c_{ij} associados aos arcos $(i,j) \in A$; se uma entidade não puder realizar uma tarefa, poder-se-á associar um custo muito elevado ao respectivo arco
- Objectivo do problema:
    - Associar entidades a tarefas de tal modo que cada entidade só seja associada a uma tarefa, e cada tarefa só seja realizada (esteja associada a) por uma entidade
    - Minimizar o custo total de afectação

- Condição para a existência de uma solução válida: semelhante à condição de existência de uma solução válida para o problema de transportes, mas adaptada às especificidades do problema de afectação

$$\sum_{i=1}^{m} 1 = \sum_{j=1}^{n} 1 \Rightarrow m = n,$$

isto é, o número de entidades a afectar deve ser igual ao número de tarefas.

- Também aqui, é possível reformular o problema caso essa condição não se verifique:
  - Se $m > n$, deve-se adicionar *m*-*n* tarefas fictícias;
  - Se $m < n$, deve-se adicionar *n*-*m* entidades fictícias.

- Exemplo de um problema de afectação:

Uma empresa pretende contratar 3 novos colaboradores para os seus quadros, um para cada um dos seus departamentos. Após análise dos currículos, a empresa seleccionou 3 candidatos, e pretende agora determinar em que departamentos irão trabalhar cada um deles. Os 3 colaboradores possuem as mesmas competências, podendo assumir funções com o mesmo nível de qualidade de serviço em qualquer um dos departamentos. Por outro lado, todos eles concorreram aos 3 departamentos onde tinham sido abertas vagas.

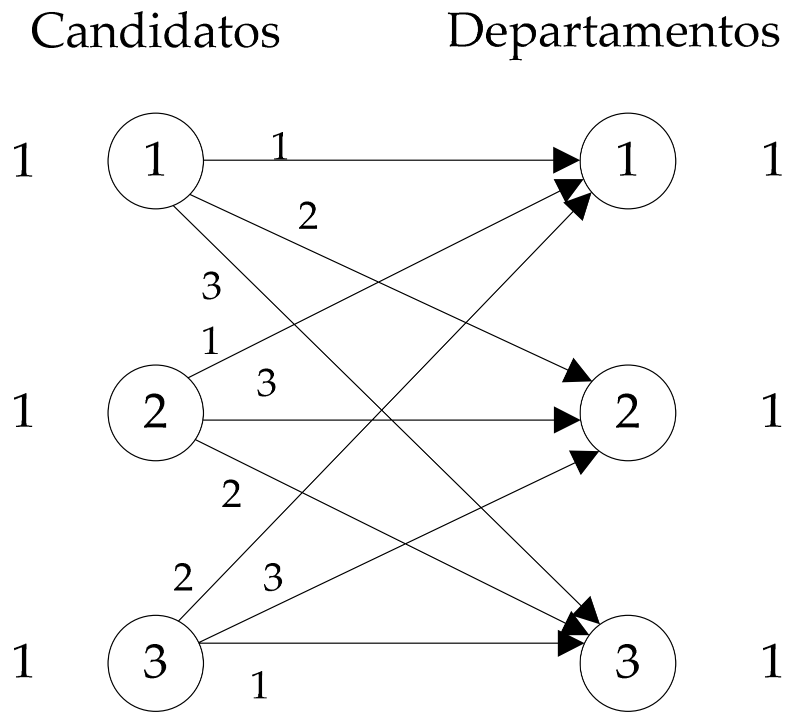
A empresa pediu aos candidatos para que ordenasse os departamentos por ordem decrescente de preferência (o 1º departamento da lista é o preferido pelo candidato). A tabela seguinte ilustra as respostas que recebeu:

	Dept 1	*Dept 2*	*Dept 3*
Candidato 1	*1*	*2*	*3*
Candidato 2	*1*	*3*	*2*
Candidato 3	*2*	*3*	*1*

A afectação dos candidatos aos departamentos será feita de forma a maximizar a satisfação global.

- Dados do problema:
  - Entidades: 3 candidatos (m=3, candidato 1, 2 e 3);;
  - "Tarefas": 3 departamentos (n=3, dept 1, 2 e 3);
  - Custos de afectação: índice de ordenação de cada um dos departamentos para cada um dos candidatos;
  - O número de candidatos é igual ao número de tarefas, e não é por isso necessário reformular o problema.
- Representação em Rede:
  - Grafo bipartido G=(V,A): V é composto por 6 vértices: 3 vértices associados aos candidatos (origens), e 3 associados aos departamentos (destinos); existe um arco entre cada par origem/destino.

- A rede:



Nota: a rede é em tudo semelhante à rede que é usada para representar um problema de transportes; Num problema de afectação, as “disponibilidades” nas origens e as “procuras” nos destinos são sempre iguais a 1.

- Representação algébrica:
  - A estrutura do modelo algébrico que representa o problema de afectação é muito semelhante à do modelo usado para representar o problema de transportes;
  - Diferenças:
    - As variáveis x_{ij}, com $(i,j) \in A$, só podem tomar o valor 0 (se i não for associado a j), e 1 (caso contrário);
    - O termo independente de todas as restrições do modelo é sempre igual a 1;
    - Em consequência do ponto anterior, o número de restrições associadas às origens (entidades) é igual ao número de restrições associadas aos destinos (tarefas).

- Para o caso geral:

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

sujeito a

$$\sum_{(i,j)\in A} x_{ij} = 1\,, i = 1, \dots, m,$$

$$\sum_{(i,j)\in A} x_{ij} = 1\,, j = 1, \dots, n,$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in A$$

- Dado que esse modelo tem a mesma estrutura que o modelo usado para o problema de transportes, as variáveis terão sempre um valor inteiro;
- A última restrição pode por isso ser substituída pela seguinte:

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

- Para o problema do exemplo acima:

Minimizar z $= x_{11} + 2x_{12} + 3x_{13} + x_{21} + 3x_{22} + 2x_{23} + 2x_{31} + 3x_{32} + 1x_{33}$

sujeito a

$$x_{11} + x_{12} + x_{13} = 1$$
$$x_{21} + x_{22} + x_{23} = 1$$
$$x_{31} + x_{32} + x_{33} = 1$$

Restrições nas origens (candidatos)

$$x_{11} + x_{21} + x_{31} = 1$$
$$x_{12} + x_{22} + x_{32} = 1$$
$$x_{13} + x_{23} + x_{33} = 1$$

Restrições nos destinos (departamentos)

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33} \in \{0,1\}$$

- A última restrição pode por isso ser substituída pela seguinte:

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33} \geq 0$$

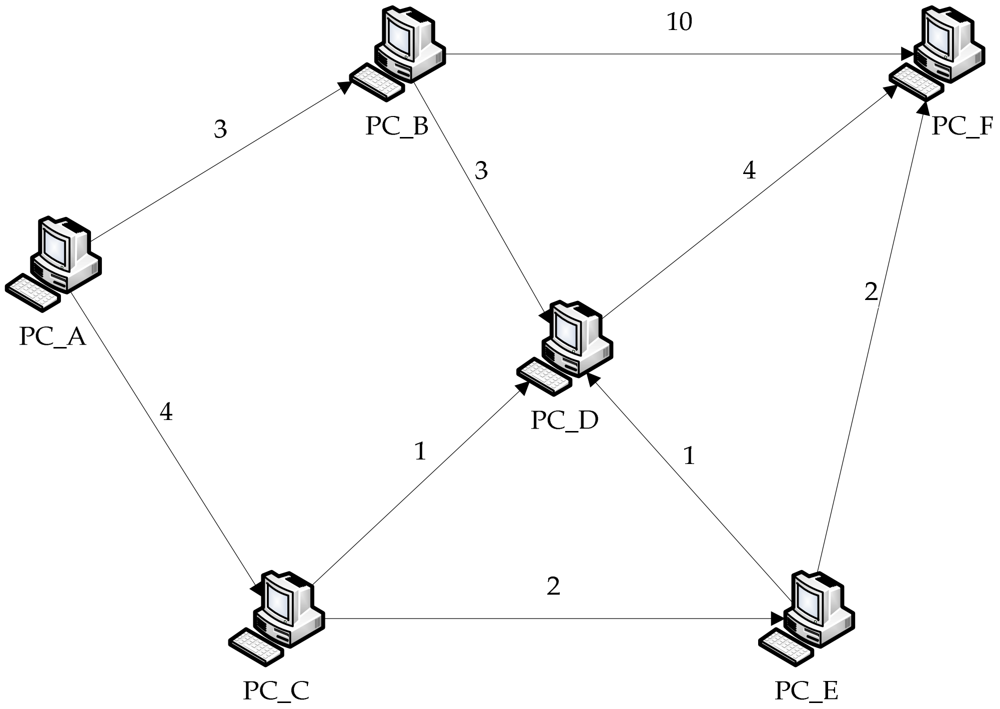
- O modelo com esta última restrição é teoricamente mais fácil de resolver.

iii. Problema de Caminho Mais Curto

- O problema de caminho mais curto consiste em determinar a sequência de arcos num grafo que conduzem de um vértice de origem o a um vértice de destino d ao menor custo (ou distância);
- Algumas variações do problema:
    - Determinar o caminho mais curto entre um vértice e todos os outros vértices;
    - Determinar o caminho mais curto entre cada par de vértices do grafo.
- O problema de caminho mais curto tem um grande número de aplicações, desde os casos mais imediatos como os problemas de encaminhamento, passando por problemas de sequenciamento de actividades ou de localização de facilidades;
- O problema de caminho mais curto aparece frequentemente como um subproblema quando se procura resolver outros problemas de optimização mais complexos.

- Exemplo de um problema de caminho mais curto:

Numa rede de computadores, pacotes de dados têm de ser enviados desde o PC_A assinalado na figura abaixo até ao PC_F. Os pacotes devem seguir todos pelo mesmo caminho. Os valores junto aos arcos definem o tempo que é necessário para transmitir esses pacotes entre cada uma das extremidades da ligação. Pretende-se determinar o caminho que minimize o tempo total de transmissão dos pacotes entre esses dois computadores.



- Representação algébrica:
  - O problema de caminho mais curto consiste em escolher uma sequência de arcos desde a origem até ao destino; o problema pode ser visto também como um problema de fluxo em que se pretende enviar 1 unidade de fluxo desde a origem até ao destino;
  - Para um arco $(i,j) \in A$, vamos designar por x_{ij} o facto do arco (i,j) ser ou não incluído no caminho; temos assim que $x_{ij} \in \{0,1\}$;
  - O critério de optimização (a *função objectivo*) consiste em minimizar o custo total dos arcos que são escolhidos, que corresponde à soma dos custos em cada arco:

$$Minimizar \sum_{(i,j)\in A} c_{ij}x_{ij} = 4x_{AC} + 3x_{AB} + x_{CD} + 2x_{CE} + 3x_{BD} + 10x_{BF} + 4x_{DF} + x_{DE} + 2x_{EF}$$

- Restrições do problema:
  - Na origem: um dos arcos com origem no PC_A deve ser escolhido
    - $x_{AB} + x_{AC} = 1$
  - No destino: um dos arcos com destino no PC_F deve ser escolhido
    - $x_{BF} + x_{DF} + x_{EF} = 1$
  - Nos outros vértices: para um vértice i, se um dos arcos com destino em i for escolhido, deve ser obrigatoriamente escolhido um outro arco com origem em i (o fluxo que sai de i deve ser igual ao fluxo que entra em i)
    - $x_{BD} + x_{BF} = x_{AB}$ (PC_B)
    - $x_{CD} + x_{CE} = x_{AC}$ (PC_C)
    - $x_{DF} = x_{BD} + x_{CD} + x_{ED}$ (PC_D)
    - $x_{ED} + x_{EF} = x_{CE}$ (PC_E)

- Modelo completo para o exemplo:

$$Min\ z = 4x_{AC} + 3x_{AB} + x_{CD} + 2x_{CE} + 3x_{BD} + 10x_{BF} + 4x_{DF} + x_{DE} + 2x_{EF}$$

sujeito a

$$x_{AB} + x_{AC} = 1$$

$$-x_{AB} + x_{BD} + x_{BF} = 0$$

$$-x_{AC} + x_{CD} + x_{CE} = 0$$

$$-x_{BD} - x_{CD} - x_{ED} + x_{DF} = 0$$

$$-x_{CE} + x_{ED} + x_{EF} = 0$$

$$x_{BF} + x_{DF} + x_{EF} = 1$$

$$x_{AC}, x_{AB}, x_{CD}, x_{BD}, x_{BF}, x_{DF}, x_{DE}, x_{EF} \ \in \{0,1\}$$

- Estrutura matricial dos coeficientes do modelo (exemplo acima):

	x_{AB}	x_{AC}	x_{BD}	x_{BF}	x_{CD}	x_{CE}	x_{DE}	x_{DF}	x_{EF}		
PC_A	1	1								=	1
PC_B	-1		1	1						=	0
PC_C		-1			1	1				=	0
PC_D			-1		-1		1	1		=	0
PC_E						-1	-1		1	=	0
PC_F				-1				-1	-1	=	-1
	3	4	3	10	1	2	1	4	2		

- Em cada coluna, nas linhas que dizem respeito às restrições do modelo, existem apenas dois coeficientes não-nulos: um é igual a +1 (na restrição associada à origem do arco), e outro igual a -1 (na restrição associada ao destino do arco)

(No modelo anterior a última restrição pode ser multiplicada por -1)

- Modelo geral:

$$\text{Min. z} = \sum_{(i,j)\in A} c_{ij}x_{ij}$$

sujeito a

$$-\sum_{(i,j)\in A} x_{ij} + \sum_{(j,i)\in A} x_{ji} = \begin{cases} 1, & se\ j = o\ (vértice\ de\ origem), \\ 0, & se\ j \neq o\ e\ j \neq d, \\ -1, & se\ j = d\ (vértice\ de\ destino), \end{cases}$$

$$x_{ij} \in \{0,1\}, \forall(i,j) \in A$$

- Esse modelo também possui a propriedade da integralidade, razão pela qual ó último conjunto de restrições pode ser substituído pelo seguinte:

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

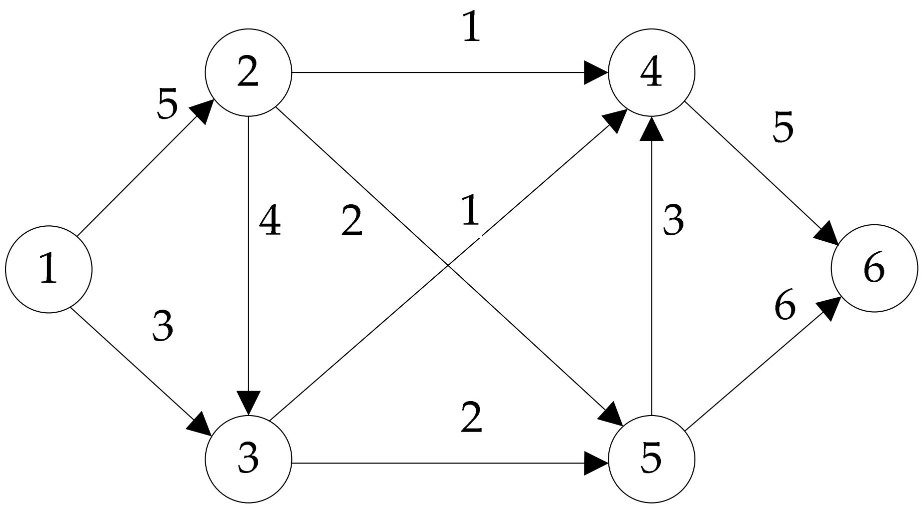
iv. Problema de Fluxo Máximo

- O problema de fluxo máximo consiste em determinar o valor máximo de fluxo que pode ser enviado de um vértice de origem (a fonte) do grafo até um vértice de destino do grafo;
- Nos arcos, existem capacidades que limitam o valor máximo do fluxo que pode passar por ele;
- O problema de fluxo máximo permite medir a quantidade máxima de produtos (ou qualquer outra entidade) que uma rede pode suportar;
- Além das aplicações típicas dos problemas de fluxo máximo, como determinar o número máximo de veículos que podem transitar numa rede viária, por exemplo, esse problema pode ser aplicado a outros problemas de planeamento, ou de escalonamento, entre outros.

- Dados do problema:
  - Um grafo orientado $G=(V,A)$ em que o conjunto V de vértices é composto por:
    - Um vértice que representa a *fonte* da rede (vértice o)
    - Um vértice que representa o destino da rede (vértice d)
    - Vértices que são pontos de transbordo da rede
  - Um conjunto de arcos orientados $(i,j)\in A$ aos quais estão associados capacidades cap_{ij};
- Objectivo do problema:
  - Maximizar o fluxo que pode ser enviado desde a fonte de uma rede até ao seu destino sem exceder a capacidade dos arcos, e respeitando a orientação dos arcos.

- Exemplo de um problema de fluxo máximo:

Considere o grafo representado na figura abaixo. O vértice 1 corresponde à fonte, e o vértice 6 ao destino. As capacidades dos arcos estão indicadas junto aos arcos.



O fluxo máximo que é possível enviar do vértice1 ao vértice 6 é igual a 6. Note que à saída do vértice 1, é possível enviar até 8 unidades de fluxo, contudo não é possível fazer chegar todo esse fluxo até ao vértice 6, dada a estrutura da rede, e uma vez que dos vértices 2 e 3 só podem sair no máximo 6 unidades de fluxo.

- Representação algébrica:
  - O valor do fluxo num arco (*i*,*j*)∈ *A* é uma das incógnitas do problema: a cada arco, associamos uma variável $x_{ij} \geq 0$, que representa o valor desse fluxo;
  - Outra das incógnitas é o valor do fluxo máximo, isto é, o fluxo que pode sair da fonte e chegar até ao destino: a esse fluxo, associamos a variável y.
  - Modelo geral:

Max. y

sujeito a

$$-\sum_{(i,j)\in A} x_{ij} + \sum_{(j,i)\in A} x_{ji} = \begin{cases} y, & se\ j = o\ (fonte), \\ 0, & se\ j \neq o\ e\ j \neq d, \\ -y, & se\ j = d\ (destino), \end{cases}$$

$$x_{ij} \leq cap_{ij}, \ \ \forall (i,j) \in A$$

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

3. Representação em Folhas de Cálculo e Resolução (*Solver* do MS Excel)

i. Modelos de Programação Linear

- É possível usar folhas de cálculo (em particular o MS Excel) para representar os modelos de Programação Linear que vimos acima
- A resolução desses modelos pode ser feita com recurso a um módulo opcional do Excel: o *Solver*
- O *Solver* pode ser acedido através do menu “Tools” (MS Excel < 2007), ou no tab “Data” (MS Excel 2007)
- Antes de invocar o *Solver* é necessário representar o modelo na folha de cálculo: nesse passo, todos os dados do modelo são introduzidas em células da folha de cálculo

- Exemplo:

Considere o caso de um problema de produção no qual se pretende determinar a quantidade a produzir de dois artigos diferentes. O primeiro artigo tem um lucro unitário igual a 6 Unidades Monetárias (UM), enquanto o segundo tem um lucro unitário de 3 UM. As variáveis são sujeitas a 3 restrições diferentes. O modelo de Programação Linear que representa o problema é o seguinte:

$$\text{Max.}\, z = 6x_S + 3x_E$$

sujeito a

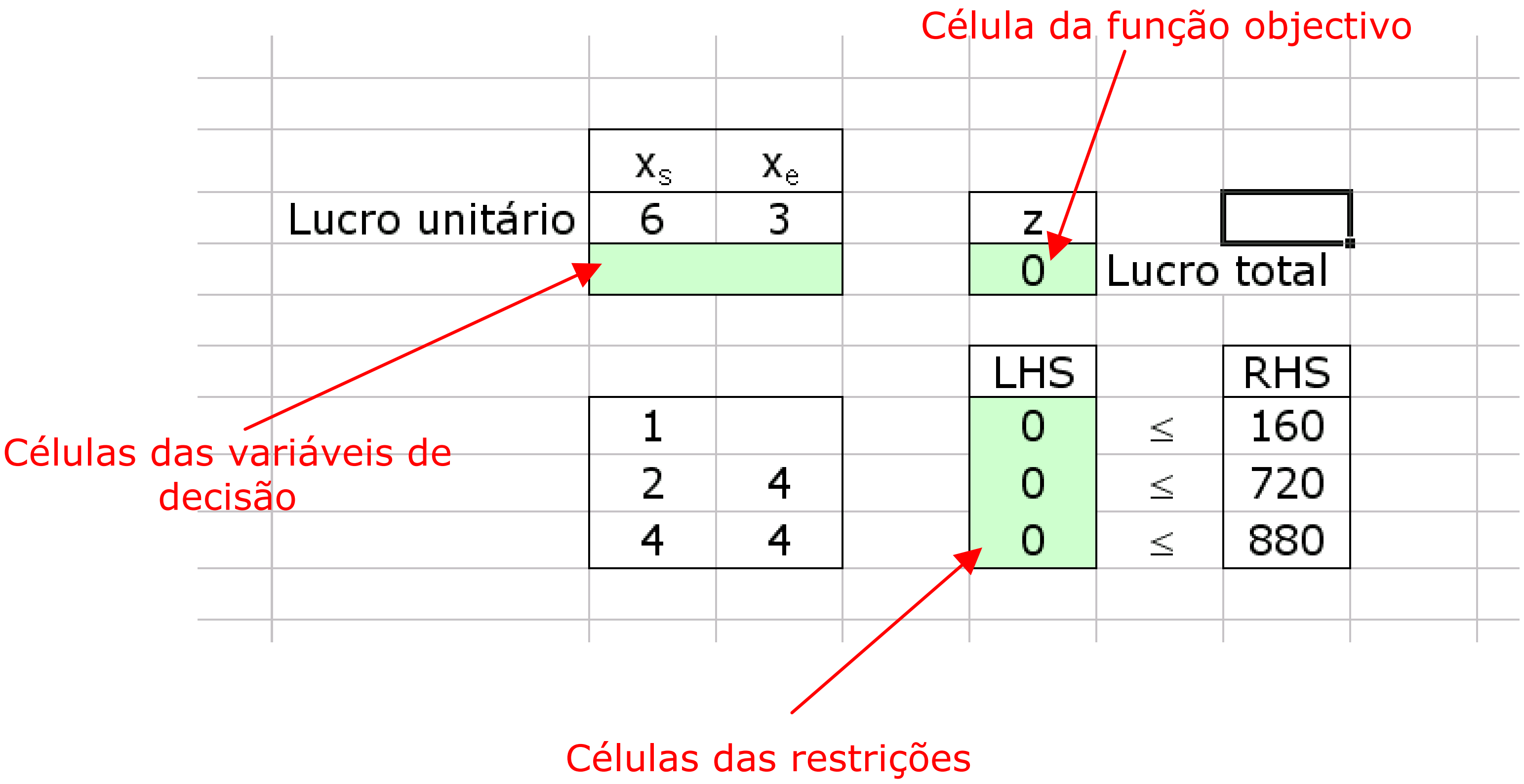
$$x_S \leq 160$$

$$2x_S + 4x_E \leq 720$$

$$4x_S + 4x_E \leq 880$$

$$x_S, x_E \geq 0$$

- Representação / Primeiro passo: copiar os dados do modelo para a folha de cálculo



	x_s	x_e			
Lucro unitário	6	3	z		
			0	Lucro total	
			LHS		RHS
	1		0	≤	160
	2	4	0	≤	720
	4	4	0	≤	880

Célula da função objectivo

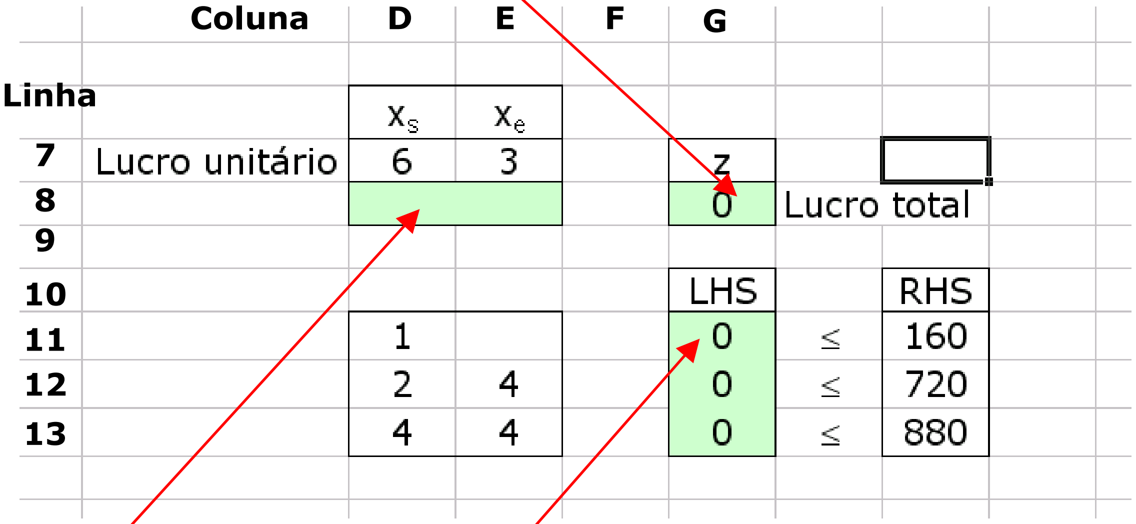
Células das variáveis de decisão

Células das restrições

- As variáveis de decisão são células da folha de cálculo escolhidas pelo utilizador.

- Representação / Segundo passo: converter as expressões algébricas do modelo em funções Excel

Função do Excel: SUMPRODUCT(D8:E8;D7:E7), ou SUMPRODUCT(quantidadesProduzidas;D7:E7), se tiver sido atribuído o nome "quantidadesProduzidas" às células das variáveis de decisão

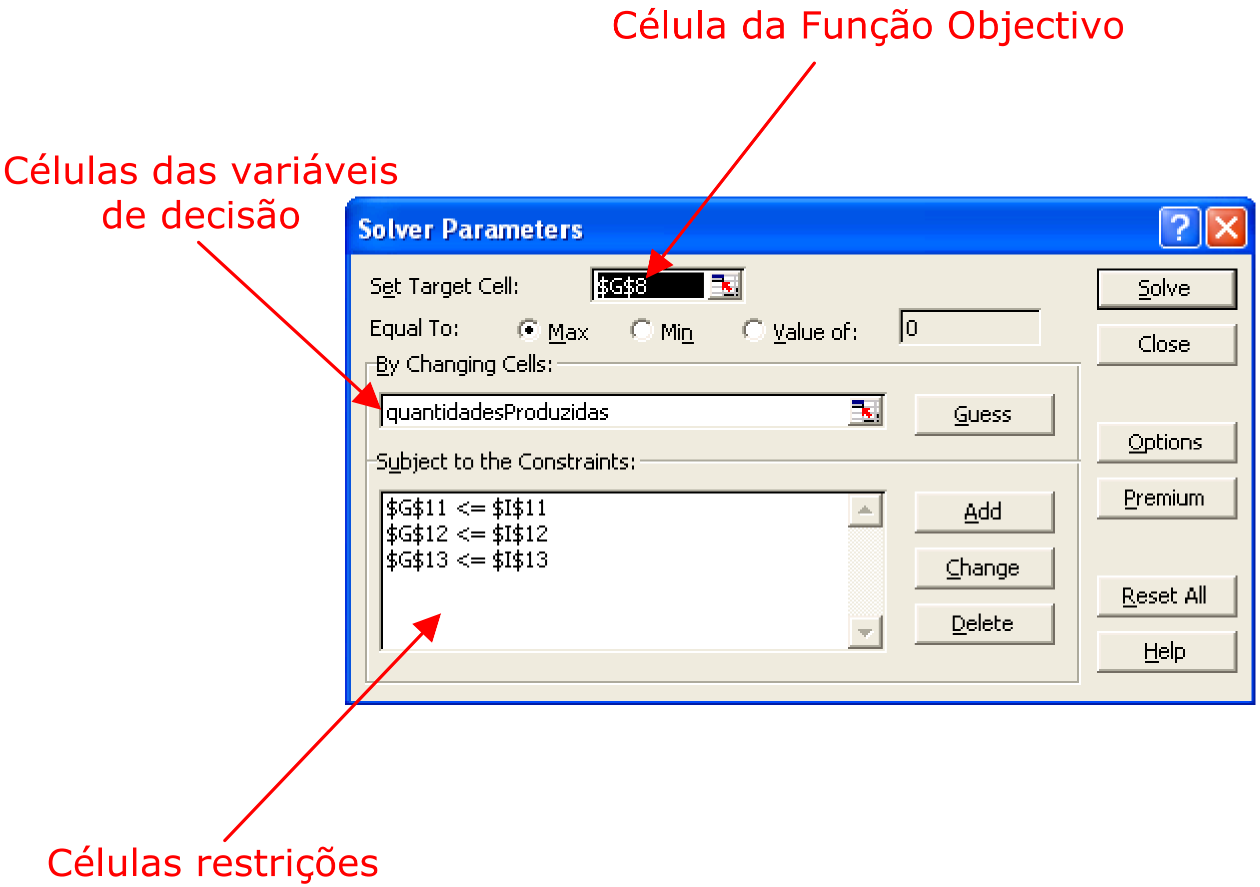


	Coluna	D	E	F	G		
Linha		x_s	x_e				
7	Lucro unitário	6	3		z		
8					0	Lucro total	
9							
10					LHS		RHS
11		1			0	≤	160
12		2	4		0	≤	720
13		4	4		0	≤	880

Função do Excel: nenhuma

Função Excel: SUMPRODUCT(D8:E8;D11:E11), ou SUMPRODUCT(quantidadesProduzidas;D11:E11), se tiver sido atribuído o nome "quantidadesProduzidas" às células das variáveis de decisão

- Resolução com o *Solver*
  - preencher os campos da janela do *Solver:*



Célula da Função Objectivo

Células das variáveis de decisão

Células restrições

Solver Parameters

Set Target Cell: $G$8

Equal To: ● Max ○ Min ○ Value of: 0

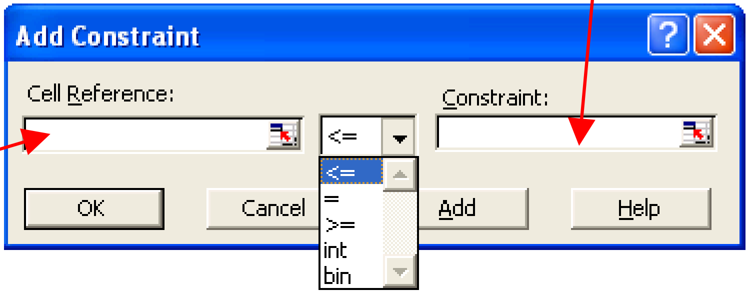
By Changing Cells: quantidadesProduzidas

Subject to the Constraints:

$G$11 <= $I$11
$G$12 <= $I$12
$G$13 <= $I$13

- Adicionar restrições (“Add”):

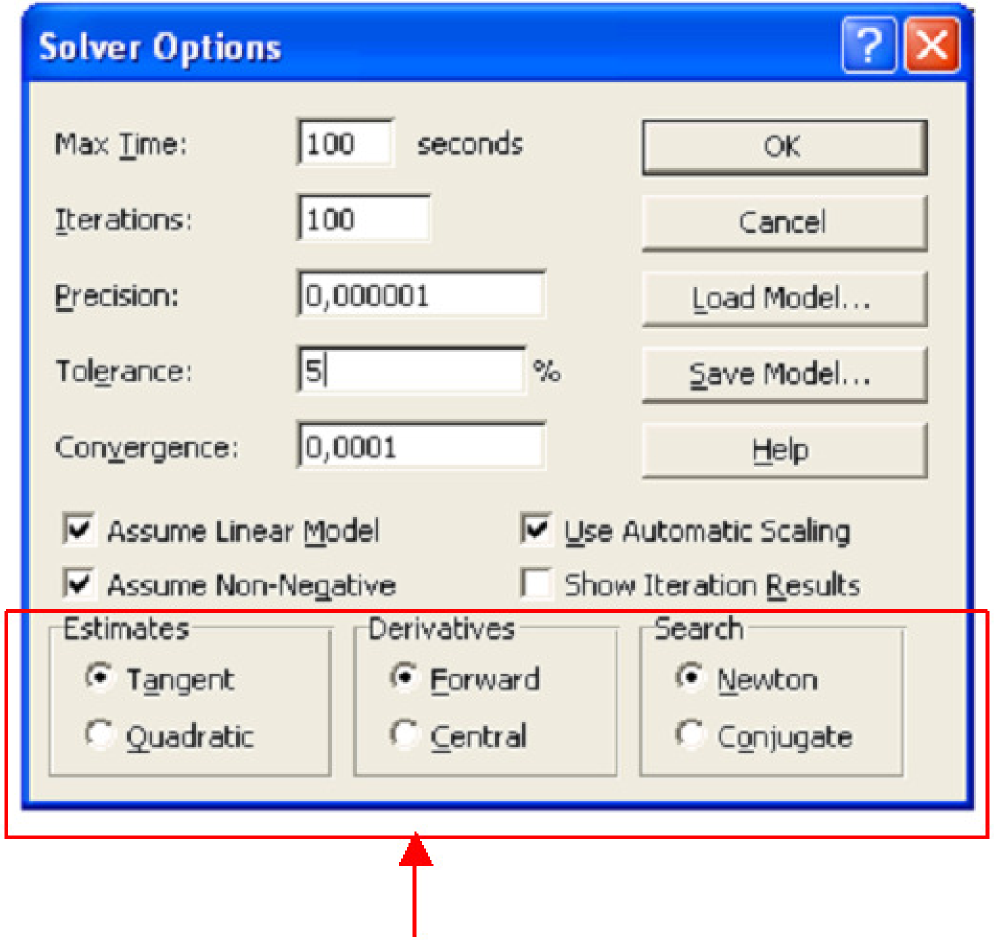
Células RHS (Right Hand Side: lado direito das restrições)



Células LHS (Left Hand Side: lado esquerdo das restrições)

Sinal da restrição (menor ou igual, maior ou igual, igual), ou restrição que obriga a(s) variável referenciadas a assumir valores inteiros ou binários

- Definir opções (“Options” na janela principal):



Max Time: 100 seconds

Iterations: 100

Precision: 0,000001

Tolerance: 5 %

Convergence: 0,0001

☑ Assume Linear Model

☑ Assume Non-Negative

☑ Use Automatic Scaling

☐ Show Iteration Results

Estimates
● Tangent
○ Quadratic

Derivatives
● Forward
○ Central

Search
● Newton
○ Conjugate

não se aplica

- Significado das principais opções:

Max. Time: tempo limite de resolução; se não tiver determinado a solução óptima, devolve a melhor solução que conseguiu nesse intervalo de tempo;

Iterations: nº máximo de iterações;

Precision: violação máxima das restrições;

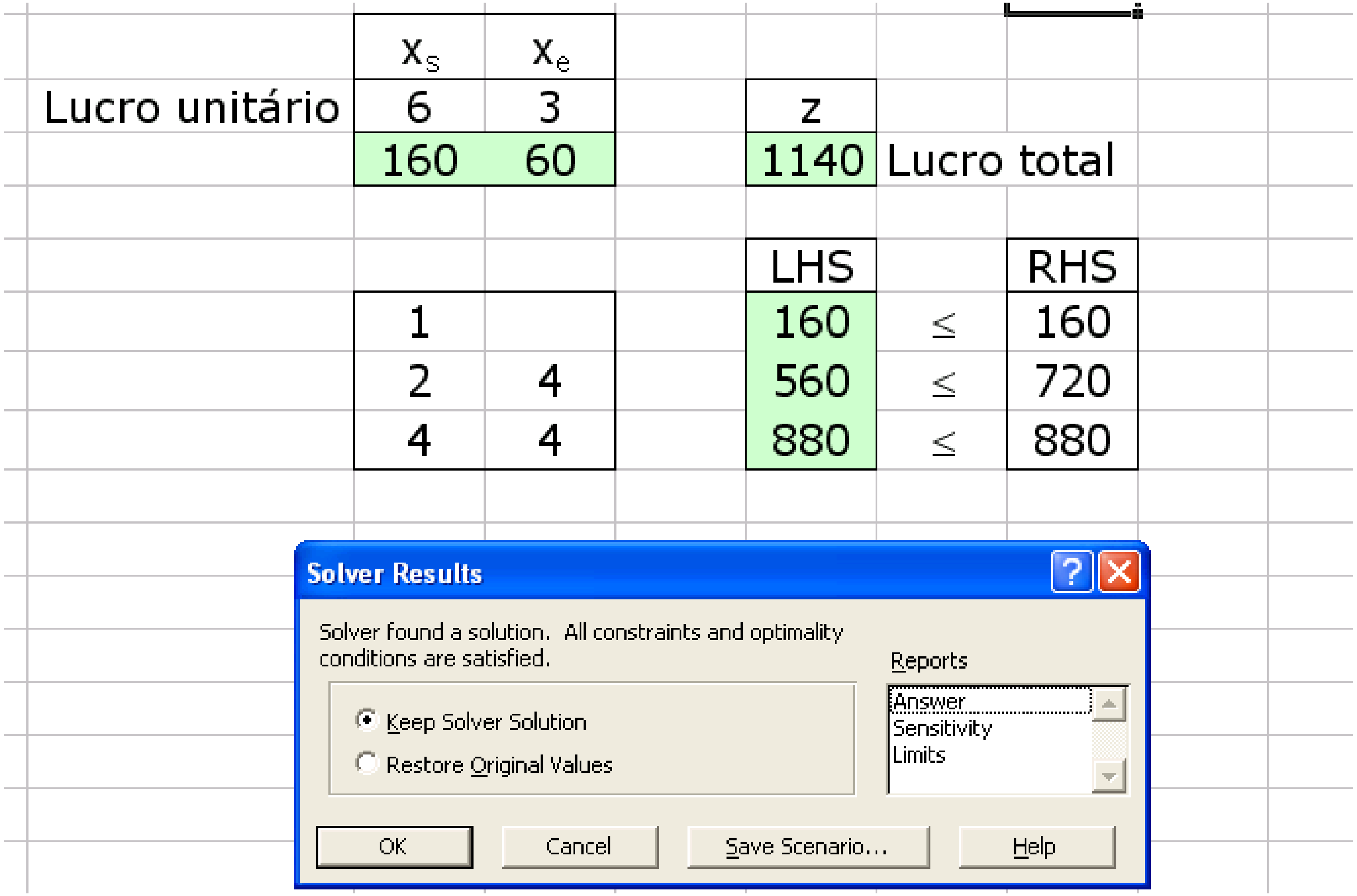
Assume Non-Negative: se as variáveis de decisão forem não negativas;

Show Iterations Results: a execução é parada no fim de cada iteração

- Para resolver o modelo, fazer "Solve" na janela principal:
  - Uma janela indica o estado da optimização:

    1. *Encontrou a solução óptima;*
    2. *O problema é impossível;*
    3. *O problema é ilimitado.*

  - Se uma solução tiver sido encontrada, o valor das variáveis, da função objectivo, e dos termos do lado esquerdo das restrições são actualizados na folha de cálculo;
  - Além de poder guardar a solução, o utilizador pode também editar vários relatórios (ver apenas o relatório "Answer"- "Resposta")



	x_s	x_e			
Lucro unitário	6	3	z		
	160	60	1140	Lucro total	
			LHS		RHS
	1		160	≤	160
	2	4	560	≤	720
	4	4	880	≤	880

Solver Results

Solver found a solution. All constraints and optimality conditions are satisfied.

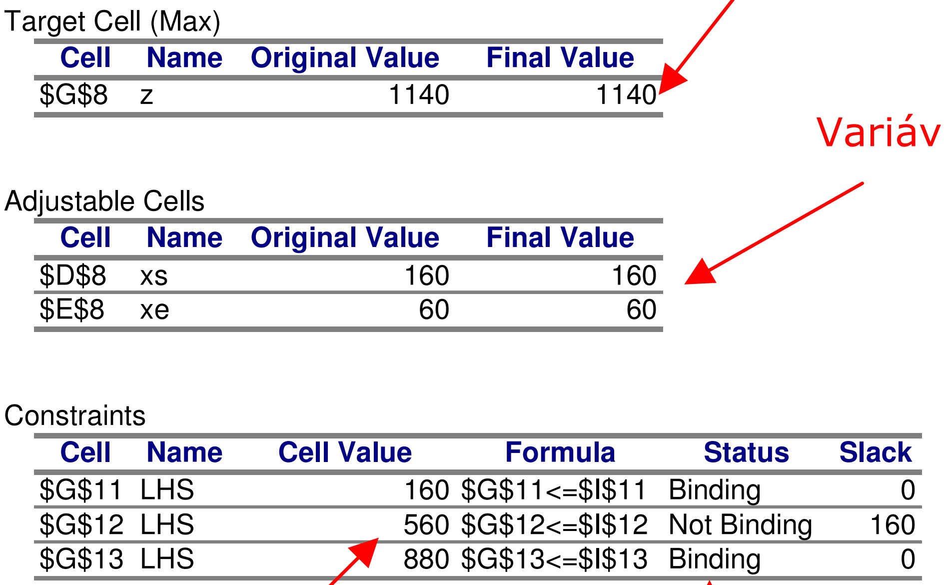
● Keep Solver Solution

○ Restore Original Values

Reports

Answer
Sensitivity
Limits

- Relatório de resposta:



Target Cell (Max)

Cell	Name	Original Value	Final Value
$G$8	z	1140	1140

Adjustable Cells

Cell	Name	Original Value	Final Value
$D$8	xs	160	160
$E$8	xe	60	60

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$G$11	LHS	160	$G$11<=$I$11	Binding	0
$G$12	LHS	560	$G$12<=$I$12	Not Binding	160
$G$13	LHS	880	$G$13<=$I$13	Binding	0

Função Objectivo: valor óptimo

Variáveis de decisão: valor óptimo

Valor do lado esquerdo das restrições (LHS)

Restrições activas: binding
Restrição não activa (com folga): not binding

ii. Resolução com o *Solver* do MS Excel dos problemas de optimização em redes (exemplos)

- Como vimos, os problemas de optimização que abordámos podem ser expressos algebricamente através de modelos de Programação Linear, e podem por isso ser resolvidos usando o *Solver* do Excel;

- Vamos resolver cada um dos exemplos que vimos atrás usando essa aplicação:
  - ➢ Problema de Transportes;
  - ➢ Problema de Afectação;
  - ➢ Problema de Caminho Mais Curto;
  - ➢ Problema de Fluxo Máximo.

- Problema de Transportes

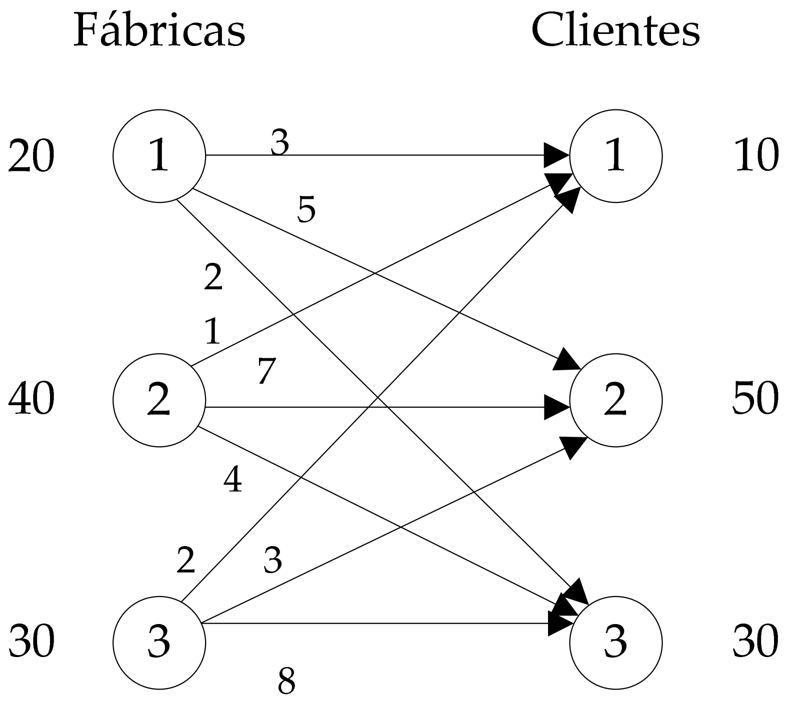
Exemplo

Uma empresa possui 3 fábricas distribuídas pelo país. A capacidade de produção de cada uma dessas fábricas é respectivamente de 20, 40 e 30 unidades. A empresa acaba de assinar um contrato de fornecimento com 3 clientes, no qual se compromete a entregar 10 unidades do seu produto ao 1º cliente, 50 ao 2º cliente, e 30 ao 3º cliente.

A empresa pretende determinar quais devem ser as fábricas que irão fornecer os produtos aos clientes, e em que quantidades, de modo a que o custo total de transporte seja o menor possível. A empresa estima que os custos de transporte serão directamente proporcionais à distância que separa as fábricas dos clientes. Essas distâncias são indicadas (em centenas de km) na tabela seguinte:

	Cliente 1	*Cliente 2*	*Cliente 3*
Fábrica 1	3	5	2
Fábrica 2	1	7	4
Fábrica 3	2	3	8

Rede



Modelo

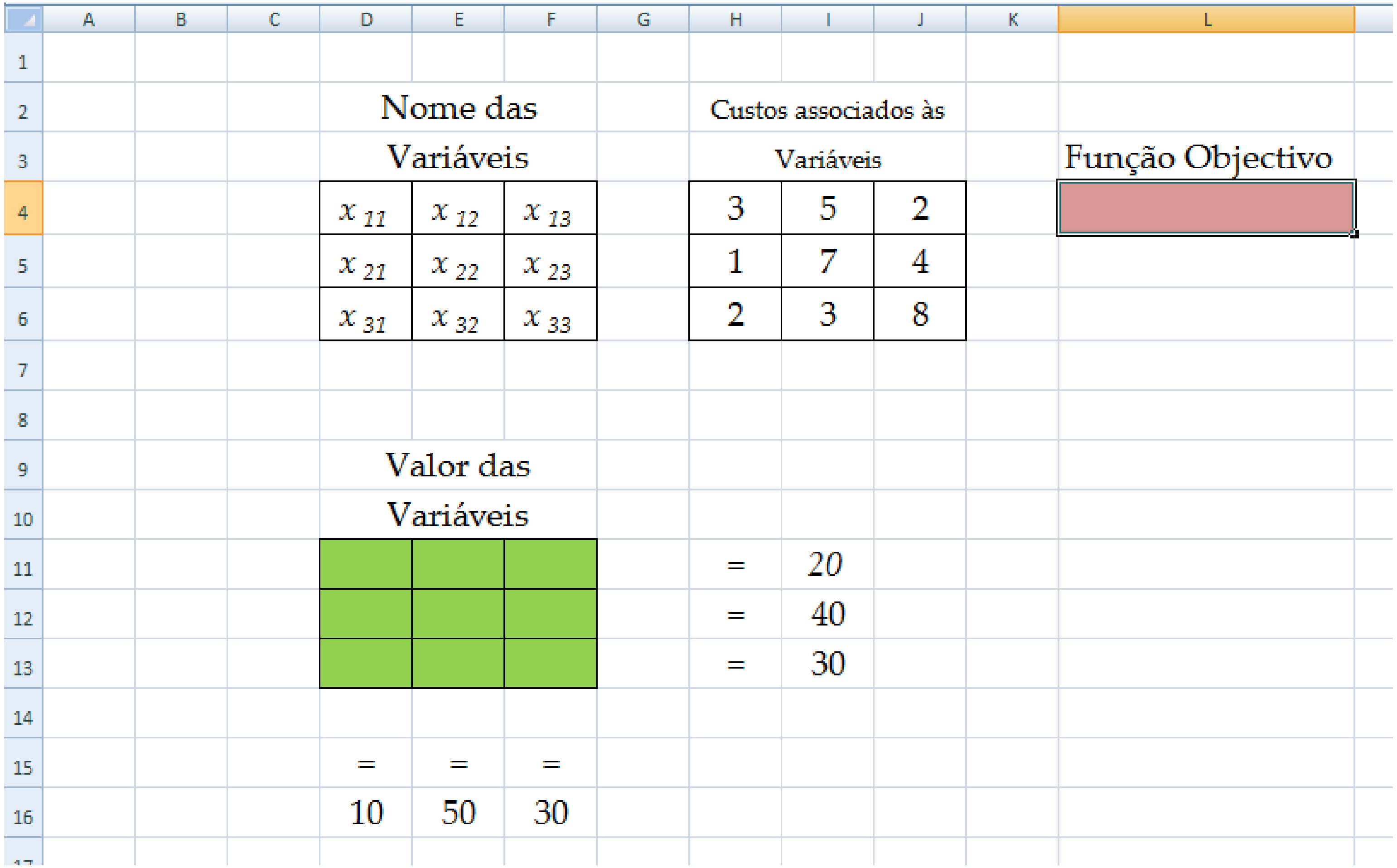
$$
\begin{aligned}
\text{Minimizar } z = {} & 3x_{11} + 5x_{12} + 2x_{13} + x_{21} + 7x_{22} \\
& + 4x_{23} + 2x_{31} + 3x_{32} + 8x_{33}
\end{aligned}
$$

sujeito a

$$
\begin{aligned}
x_{11} + x_{12} + x_{13} &= 20 \\
x_{21} + x_{22} + x_{23} &= 40 \\
x_{31} + x_{32} + x_{33} &= 30 \\
x_{11} + x_{21} + x_{31} &= 10 \\
x_{12} + x_{22} + x_{32} &= 50 \\
x_{13} + x_{23} + x_{33} &= 30 \\
x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33} &\geq 0
\end{aligned}
$$

Representação numa folha de cálculo

- Passar dados do problema



Nome das Variáveis

x_{11}	x_{12}	x_{13}
x_{21}	x_{22}	x_{23}
x_{31}	x_{32}	x_{33}

Custos associados às Variáveis

3	5	2
1	7	4
2	3	8

Função Objectivo

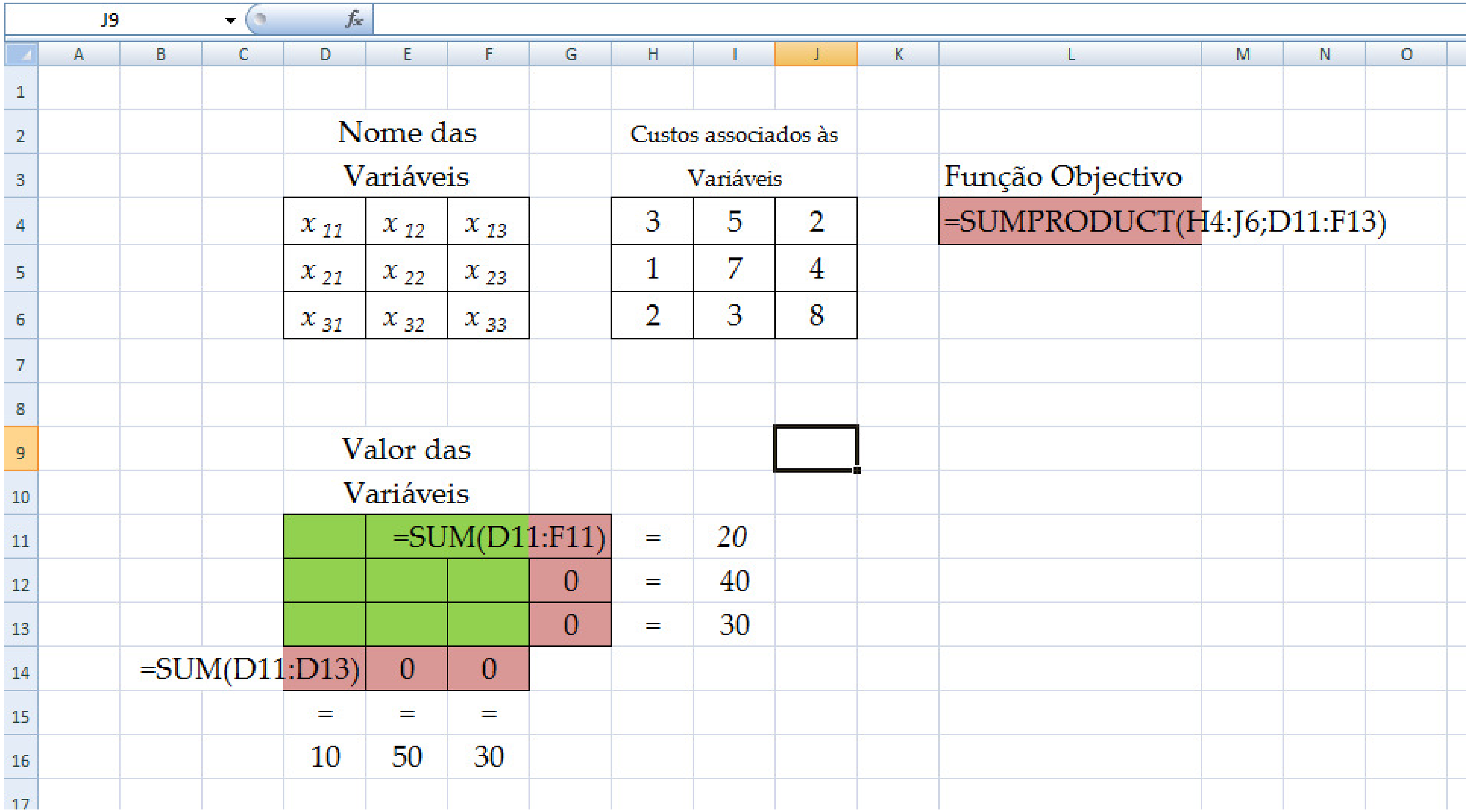
🟥

Valor das Variáveis

🟩	🟩	🟩	=	20
🟩	🟩	🟩	=	40
🟩	🟩	🟩	=	30
=	=	=		
10	50	30		

Representação numa folha de cálculo

- Converter expressões algébricas em funções Excel



Nome das Variáveis

x_{11}	x_{12}	x_{13}
x_{21}	x_{22}	x_{23}
x_{31}	x_{32}	x_{33}

Custos associados às Variáveis

3	5	2
1	7	4
2	3	8

Função Objectivo

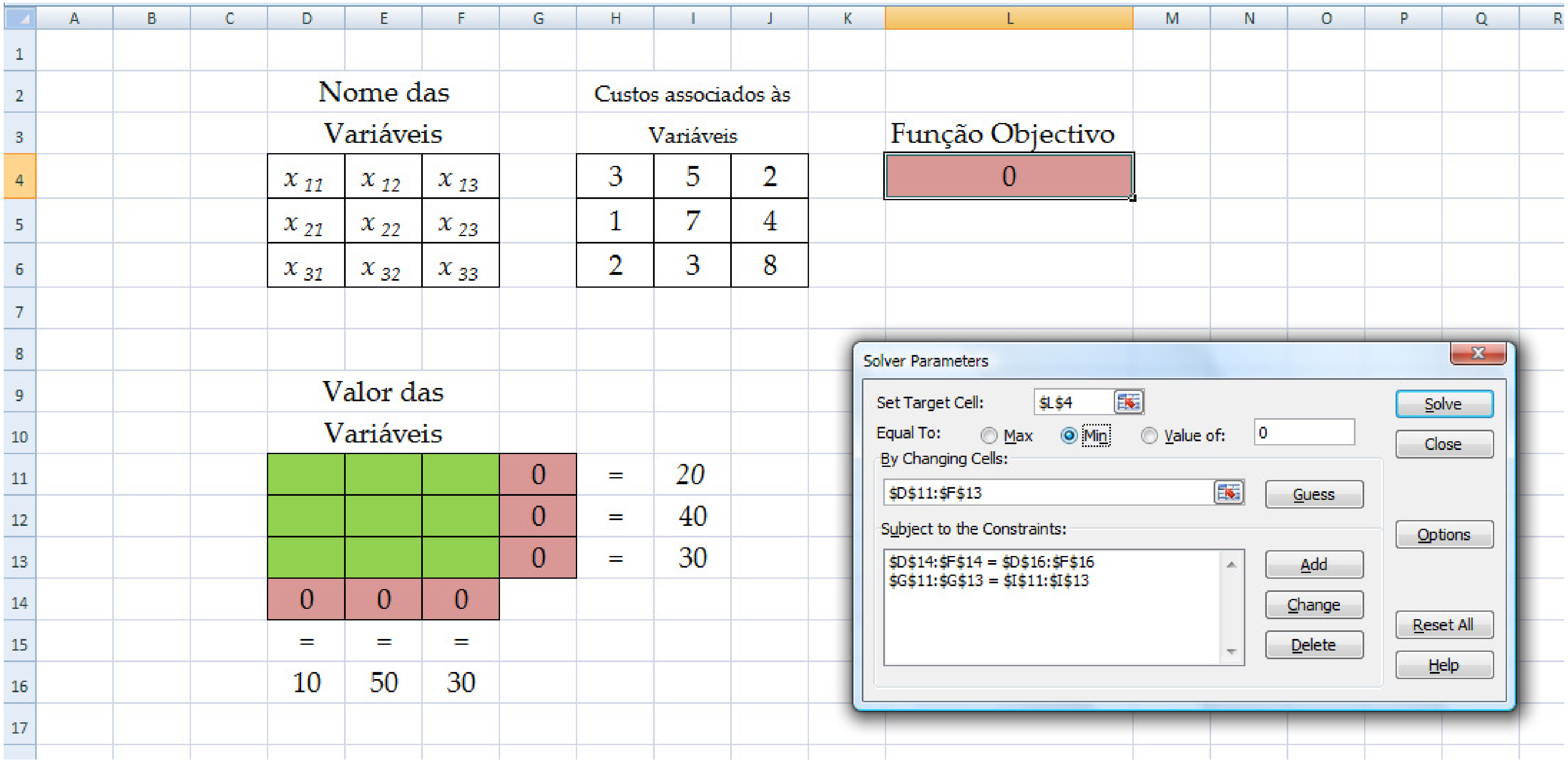
=SUMPRODUCT(H4:J6;D11:F13)

Valor das Variáveis

			=SUM(D11:F11)	=	20
			0	=	40
			0	=	30
=SUM(D11:D13)	0	0			
=	=	=			
10	50	30			

Resolução com o *Solver*

- preencher os campos da janela do *Solver:*



	D	E	F	G	H	I	J	K	L
1									
2	Nome das				Custos associados às				
3	Variáveis				Variáveis				Função Objectivo
4	x_{11}	x_{12}	x_{13}		3	5	2		0
5	x_{21}	x_{22}	x_{23}		1	7	4		
6	x_{31}	x_{32}	x_{33}		2	3	8		
7									
8									
9	Valor das								
10	Variáveis								
11				0	=	20			
12				0	=	40			
13				0	=	30			
14	0	0	0						
15	=	=	=						
16	10	50	30						
17									

Solver Parameters

Set Target Cell: $L$4

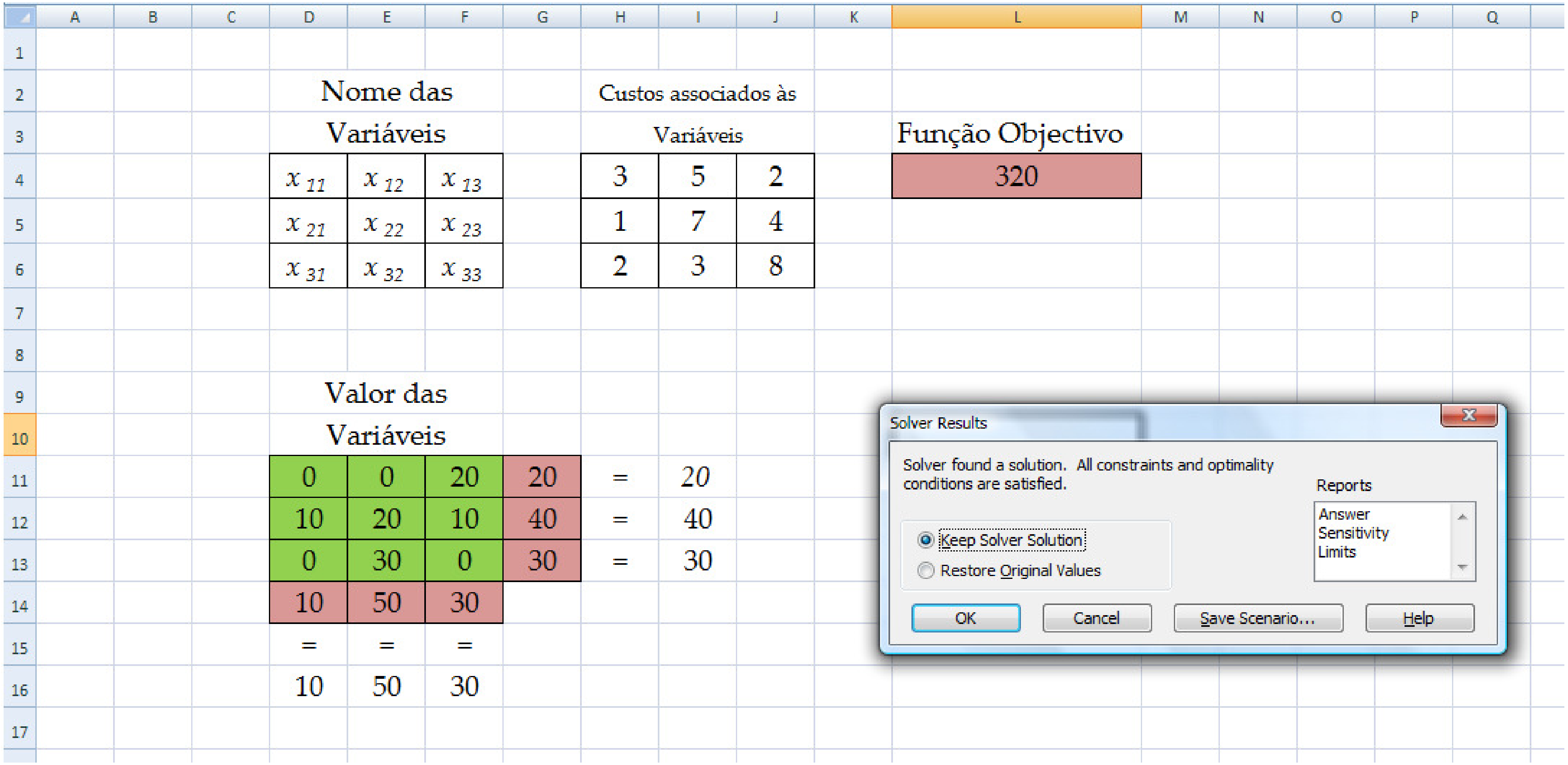
Equal To: ○ Max ● Min ○ Value of: 0

By Changing Cells: $D$11:$F$13

Subject to the Constraints:

$D$14:$F$14 = $D$16:$F$16
$G$11:$G$13 = $I$11:$I$13

Interpretação dos resultados



Nome das Variáveis

x_{11}	x_{12}	x_{13}
x_{21}	x_{22}	x_{23}
x_{31}	x_{32}	x_{33}

Custos associados às Variáveis

3	5	2
1	7	4
2	3	8

Função Objectivo

320

Valor das Variáveis

0	0	20	20	=	20
10	20	10	40	=	40
0	30	0	30	=	30
10	50	30			
=	=	=			
10	50	30			

Solver Results

Solver found a solution. All constraints and optimality conditions are satisfied.

● Keep Solver Solution
○ Restore Original Values

- Problema de Afectação

Exemplo

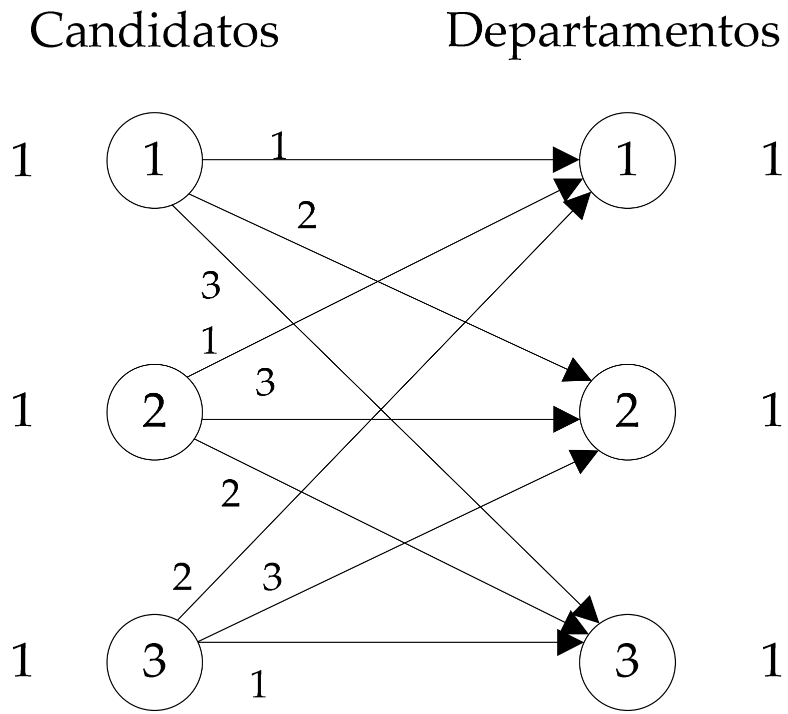
Uma empresa pretende contratar 3 novos colaboradores para os seus quadros, um para cada um dos seus departamentos. Após análise dos currículos, a empresa seleccionou 3 candidatos, e pretende agora determinar em que departamentos irão trabalhar cada um deles. Os 3 colaboradores possuem as mesmas competências, podendo assumir funções com o mesmo nível de qualidade de serviço em qualquer um dos departamentos. Por outro lado, todos eles concorreram aos 3 departamentos onde tinham sido abertas vagas.

A empresa pediu aos candidatos para que ordenasse os departamentos por ordem decrescente de preferência (o 1º departamento da lista é o preferido pelo candidato). A tabela seguinte ilustra as respostas que recebeu:

	Dept 1	*Dept 2*	*Dept 3*
Candidato 1	*1*	*2*	*3*
Candidato 2	*1*	*3*	*2*
Candidato 3	*2*	*3*	*1*

A afectação dos candidatos aos departamentos será feita de forma a maximizar a satisfação global.

Rede



Modelo

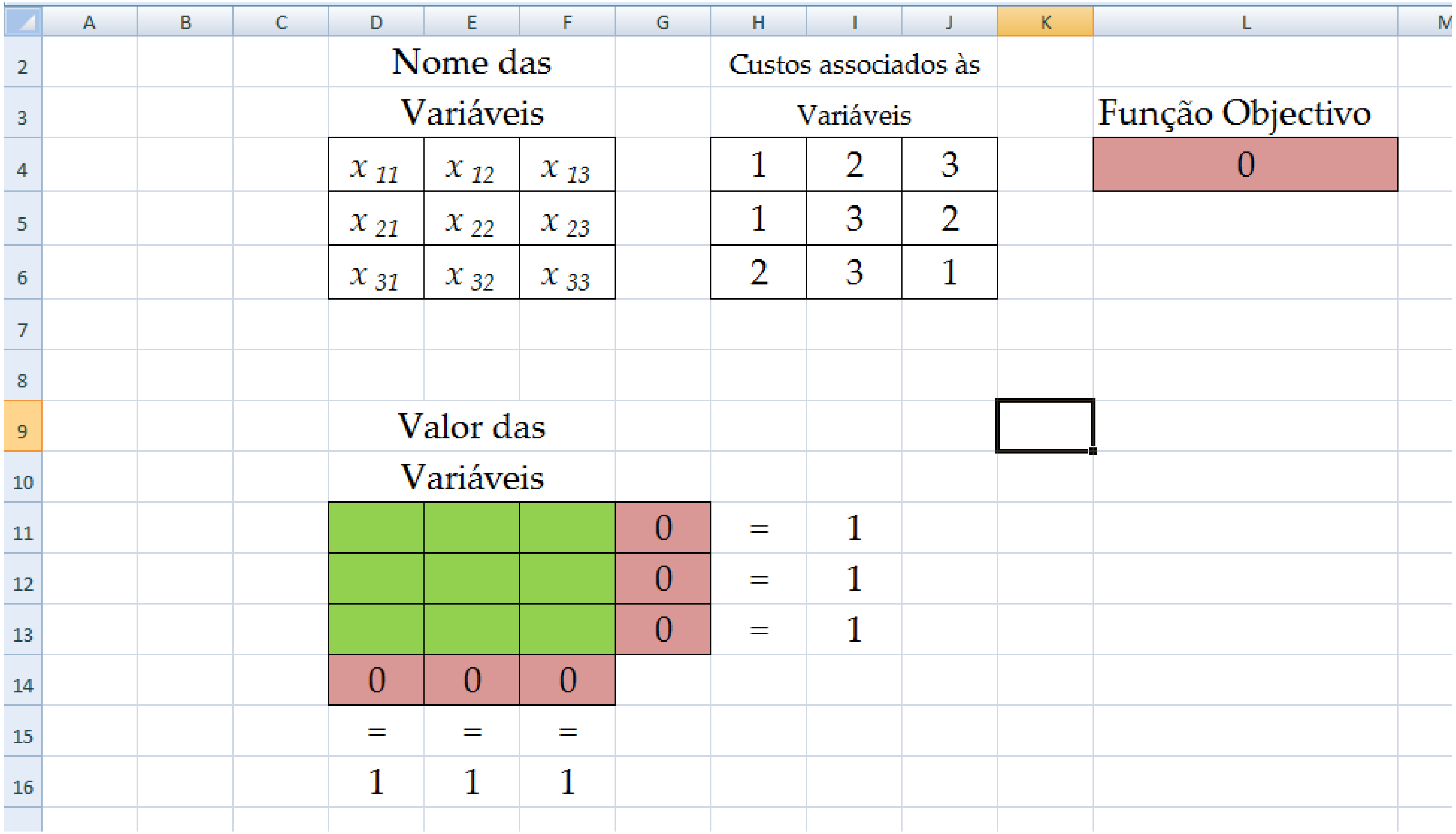
$$\text{Min. } z = x_{11} + 2x_{12} + 3x_{13} + x_{21} + 3x_{22} + 2x_{23} + 2x_{31} + 3x_{32} + 1x_{33}$$

sujeito a

$$x_{11} + x_{12} + x_{13} = 1$$
$$x_{21} + x_{22} + x_{23} = 1$$
$$x_{31} + x_{32} + x_{33} = 1$$
$$x_{11} + x_{21} + x_{31} = 1$$
$$x_{12} + x_{22} + x_{32} = 1$$
$$x_{13} + x_{23} + x_{33} = 1$$
$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33} \in \{0,1\}$$

Representação numa folha de cálculo

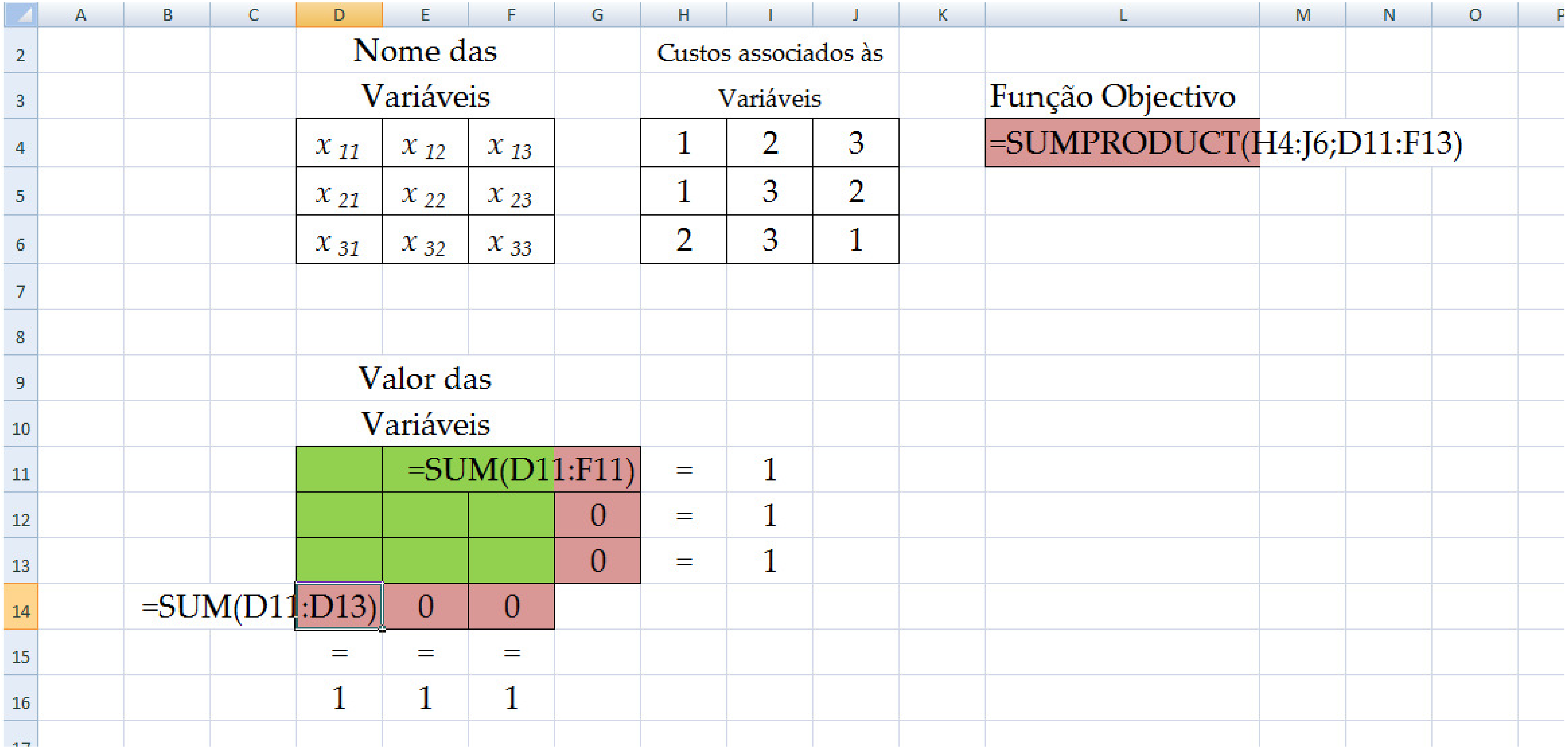
- Passar dados do problema



	A	B	C	D	E	F	G	H	I	J	K	L	M
2				Nome das				Custos associados às					
3				Variáveis				Variáveis				Função Objectivo	
4				x_{11}	x_{12}	x_{13}		1	2	3		0	
5				x_{21}	x_{22}	x_{23}		1	3	2			
6				x_{31}	x_{32}	x_{33}		2	3	1			
7													
8													
9				Valor das									
10				Variáveis									
11							0	=	1				
12							0	=	1				
13							0	=	1				
14				0	0	0							
15				=	=	=							
16				1	1	1							

Representação numa folha de cálculo

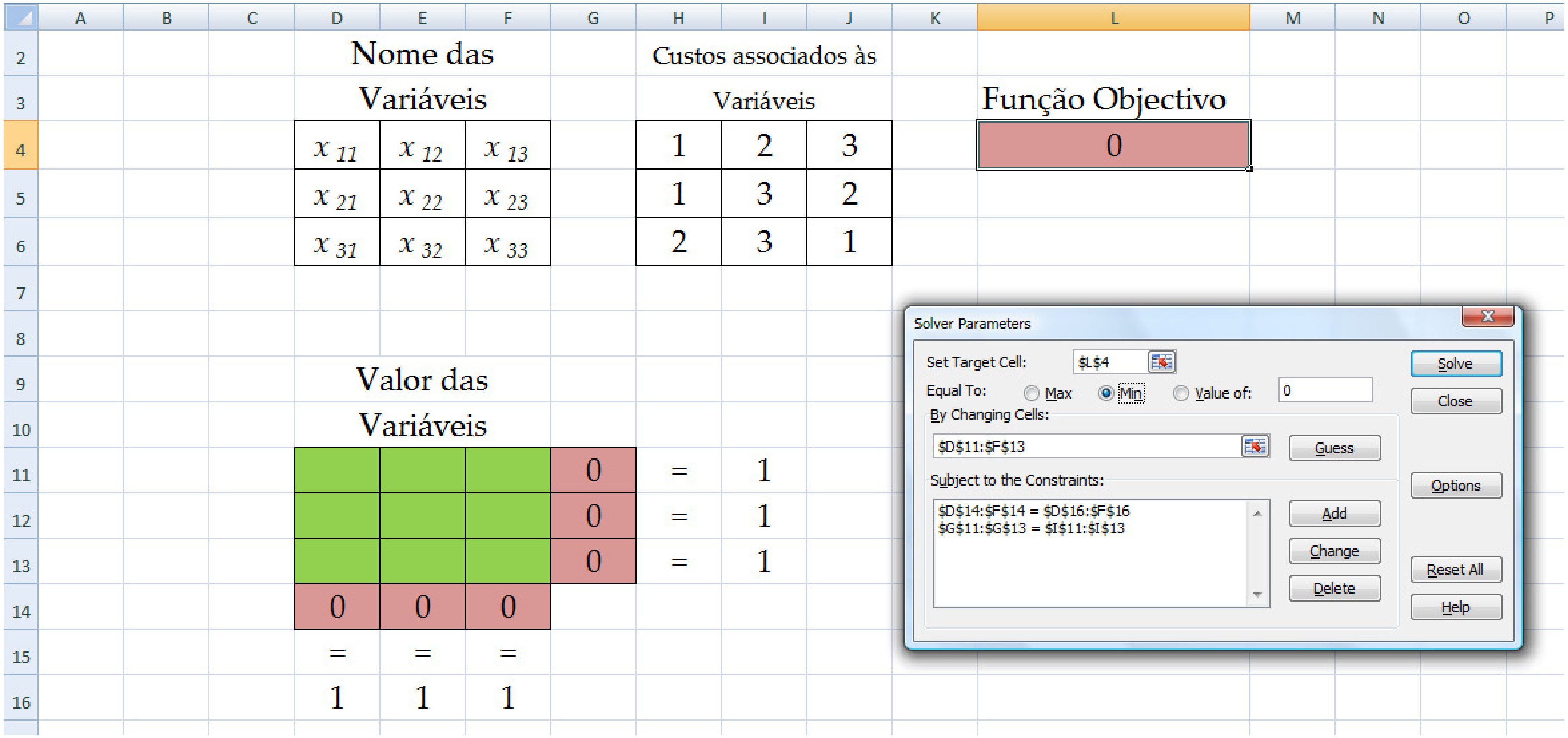
- Converter expressões algébricas em funções Excel



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
2					Nome das				Custos associados às						
3					Variáveis				Variáveis			Função Objectivo			
4				x_{11}	x_{12}	x_{13}		1	2	3		=SUMPRODUCT(H4:J6;D11:F13)			
5				x_{21}	x_{22}	x_{23}		1	3	2					
6				x_{31}	x_{32}	x_{33}		2	3	1					
7															
8															
9					Valor das										
10					Variáveis										
11							=SUM(D11:F11)	=	1						
12							0	=	1						
13							0	=	1						
14				=SUM(D11:D13)	0	0									
15				=	=	=									
16				1	1	1									

Resolução com o *Solver*

- preencher os campos da janela do *Solver:*



	D	E	F	G	H	I	J	L
2		Nome das				Custos associados às		
3		Variáveis				Variáveis		Função Objectivo
4	x_{11}	x_{12}	x_{13}		1	2	3	0
5	x_{21}	x_{22}	x_{23}		1	3	2	
6	x_{31}	x_{32}	x_{33}		2	3	1	
7								
8								
9		Valor das						
10		Variáveis						
11				0	=	1		
12				0	=	1		
13				0	=	1		
14	0	0	0					
15	=	=	=					
16	1	1	1					

Solver Parameters

Set Target Cell: $L$4

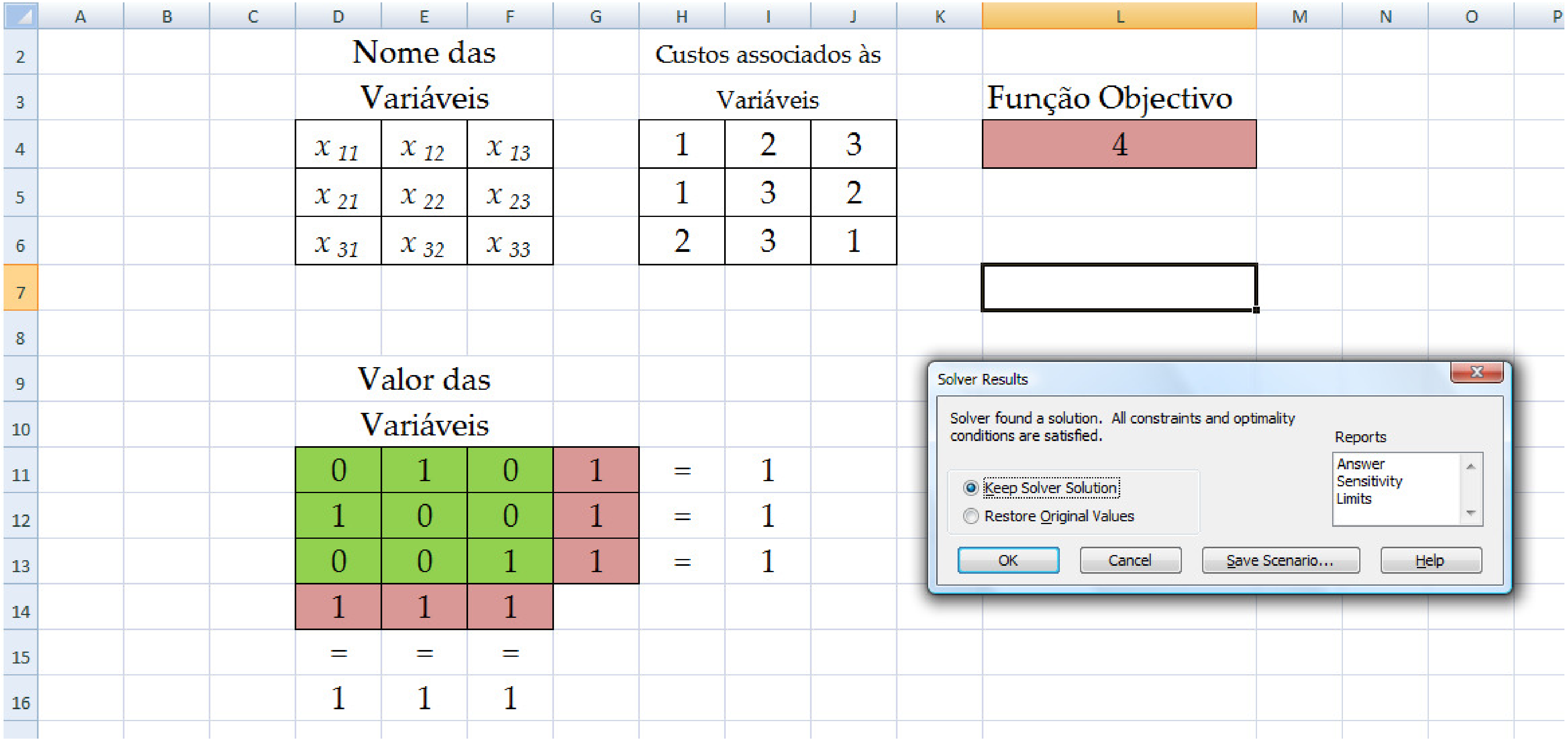
Equal To: ○ Max ● Min ○ Value of: 0

By Changing Cells: $D$11:$F$13

Subject to the Constraints:

$D$14:$F$14 = $D$16:$F$16
$G$11:$G$13 = $I$11:$I$13

Interpretação dos resultados



Nome das Variáveis

x_{11}	x_{12}	x_{13}
x_{21}	x_{22}	x_{23}
x_{31}	x_{32}	x_{33}

Custos associados às Variáveis

1	2	3
1	3	2
2	3	1

Função Objectivo

4

Valor das Variáveis

0	1	0	1	=	1
1	0	0	1	=	1
0	0	1	1	=	1
1	1	1			
=	=	=			
1	1	1			

Solver Results

Solver found a solution. All constraints and optimality conditions are satisfied.

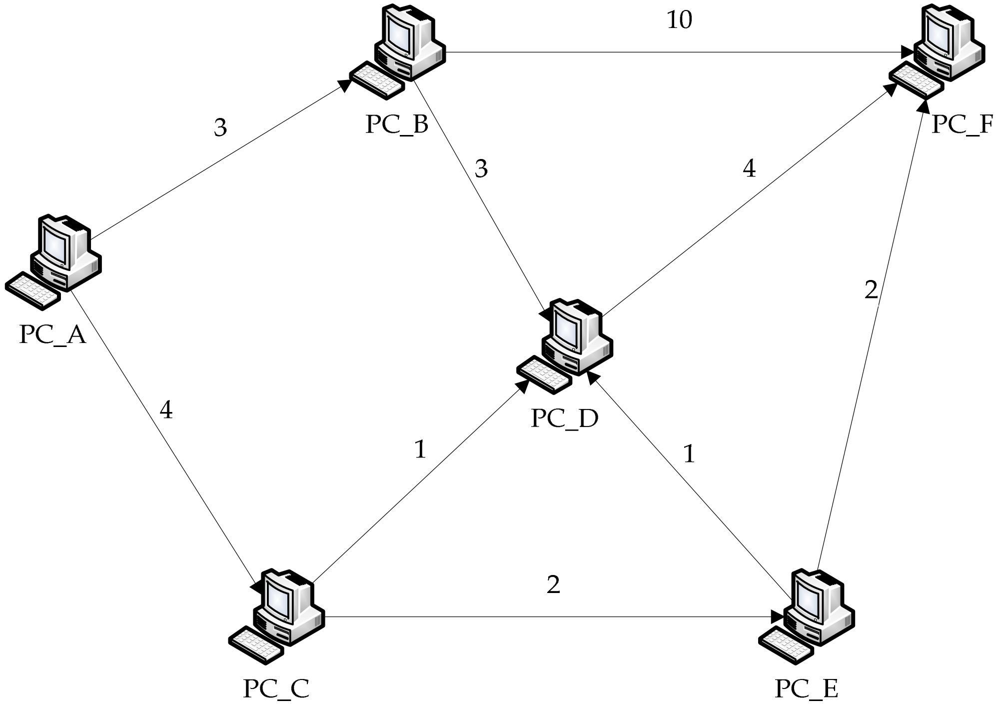
● Keep Solver Solution

○ Restore Original Values

- Problema de Caminho Mais Curto

Exemplo

Numa rede de computadores, pacotes de dados têm de ser enviados desde o PC_A assinalado na figura abaixo até ao PC_F. Os pacotes devem seguir todos pelo mesmo caminho. Os valores junto aos arcos definem o tempo que é necessário para transmitir esses pacotes entre cada uma das extremidades da ligação. Pretende-se determinar o caminho que minimize o tempo total de transmissão dos pacotes entre esses dois computadores.



Modelo

$$Min\ z = 4x_{AC} + 3x_{AB} + x_{CD} + 2x_{CE} + 3x_{BD} + 10x_{BF} + 4x_{DF} + x_{DE} + 2x_{EF}$$

sujeito a

$$x_{AB} + x_{AC} = 1$$

$$-x_{AB} + x_{BD} + x_{BF} = 0$$

$$-x_{AC} + x_{CD} + x_{CE} = 0$$

$$-x_{BD} - x_{CD} - x_{ED} + x_{DF} = 0$$

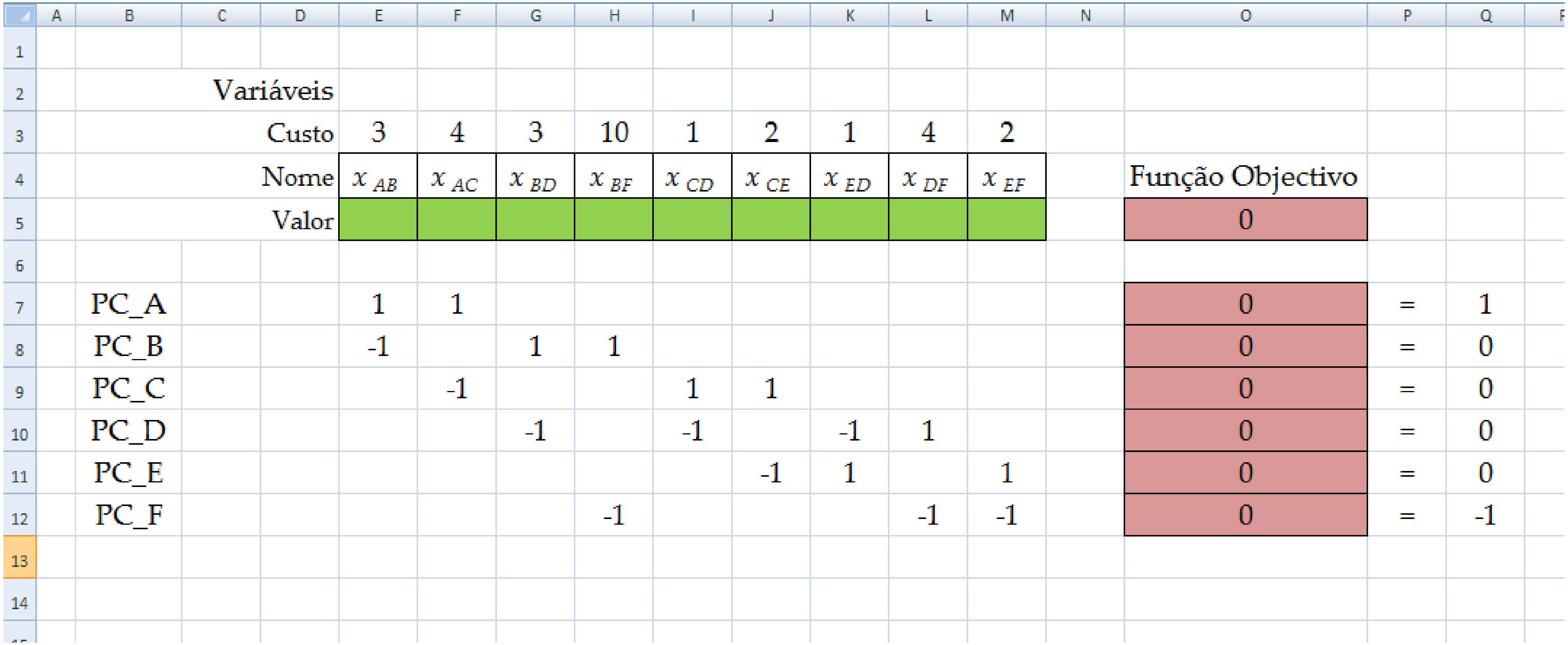
$$-x_{CE} + x_{ED} + x_{EF} = 0$$

$$x_{BF} + x_{DF} + x_{EF} = 1$$

$$x_{AC}, x_{AB}, x_{CD}, x_{BD}, x_{BF}, x_{DF}, x_{DE}, x_{EF} \ \in \{0,1\}$$

Representação numa folha de cálculo

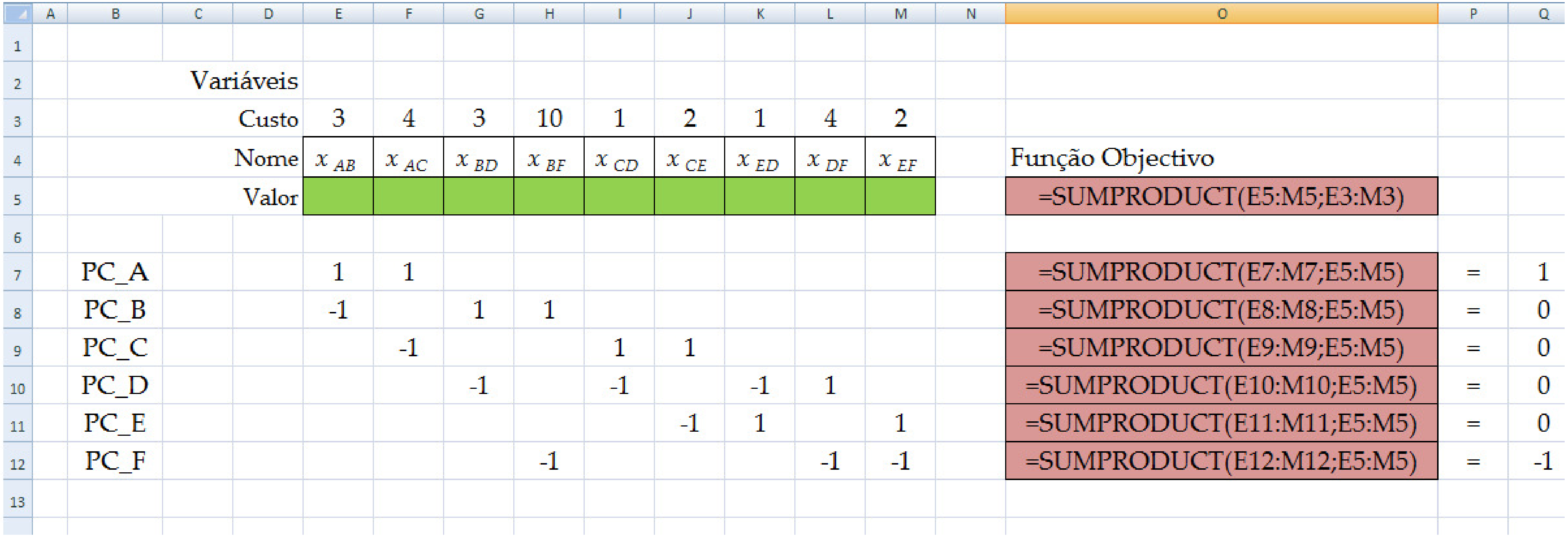
- Passar dados do problema



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2				Variáveis													
3				Custo	3	4	3	10	1	2	1	4	2				
4				Nome	x_{AB}	x_{AC}	x_{BD}	x_{BF}	x_{CD}	x_{CE}	x_{ED}	x_{DF}	x_{EF}		Função Objectivo		
5				Valor											0		
6																	
7		PC_A			1	1									0	=	1
8		PC_B			-1		1	1							0	=	0
9		PC_C				-1			1	1					0	=	0
10		PC_D					-1		-1		-1	1			0	=	0
11		PC_E								-1	1		1		0	=	0
12		PC_F						-1				-1	-1		0	=	-1
13																	
14																	

Representação numa folha de cálculo

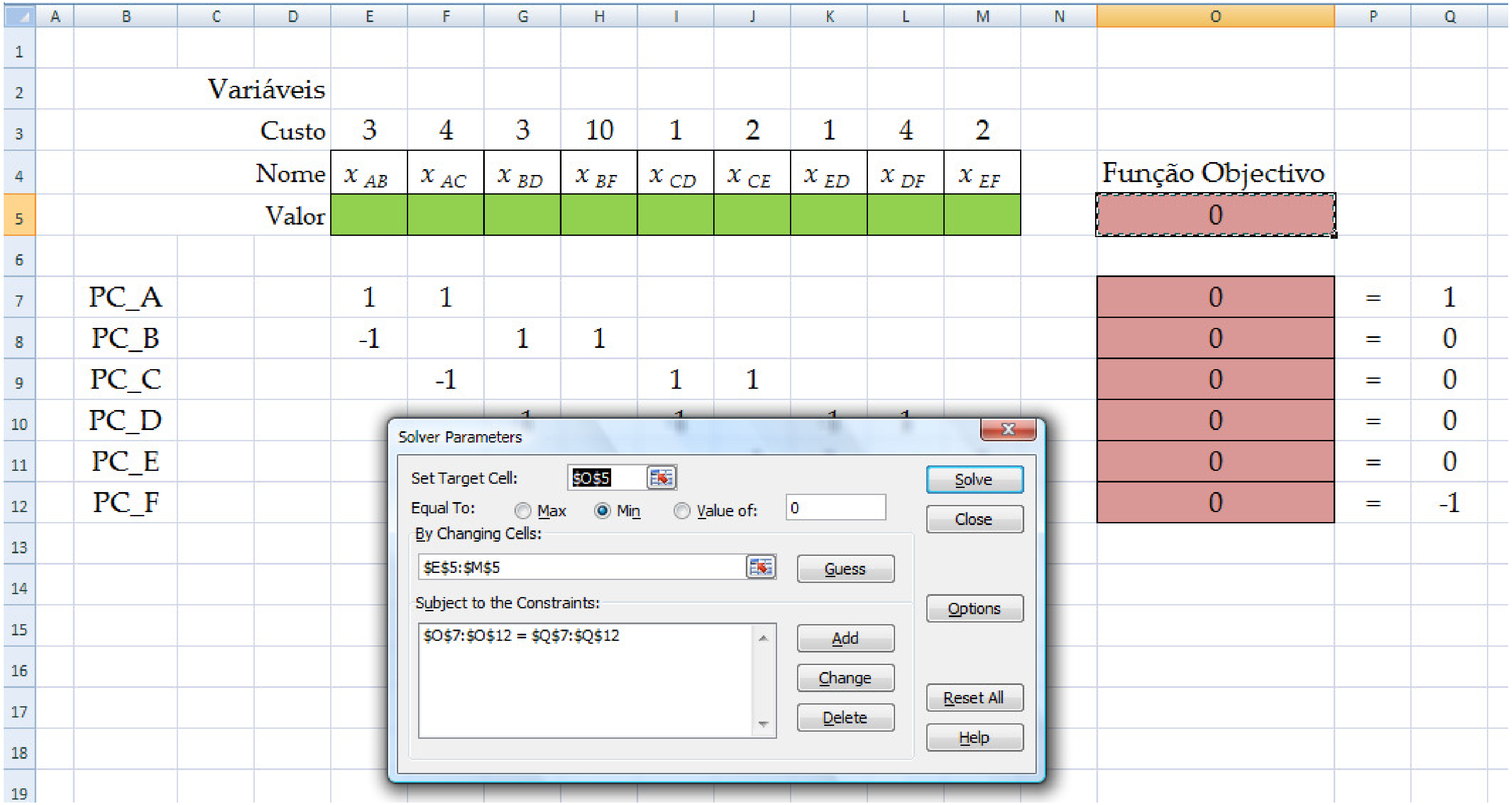
- Converter expressões algébricas em funções Excel



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2				Variáveis													
3				Custo	3	4	3	10	1	2	1	4	2				
4				Nome	x_{AB}	x_{AC}	x_{BD}	x_{BF}	x_{CD}	x_{CE}	x_{ED}	x_{DF}	x_{EF}		Função Objectivo		
5				Valor											=SUMPRODUCT(E5:M5;E3:M3)		
6																	
7		PC_A			1	1									=SUMPRODUCT(E7:M7;E5:M5)	=	1
8		PC_B			-1		1	1							=SUMPRODUCT(E8:M8;E5:M5)	=	0
9		PC_C				-1			1	1					=SUMPRODUCT(E9:M9;E5:M5)	=	0
10		PC_D					-1		-1		-1	1			=SUMPRODUCT(E10:M10;E5:M5)	=	0
11		PC_E								-1	1		1		=SUMPRODUCT(E11:M11;E5:M5)	=	0
12		PC_F						-1				-1	-1		=SUMPRODUCT(E12:M12;E5:M5)	=	-1
13																	

Resolução com o *Solver*

- preencher os campos da janela do *Solver:*



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2				Variáveis													
3				Custo	3	4	3	10	1	2	1	4	2				
4				Nome	x_{AB}	x_{AC}	x_{BD}	x_{BF}	x_{CD}	x_{CE}	x_{ED}	x_{DF}	x_{EF}		Função Objectivo		
5				Valor											0		
6																	
7		PC_A			1	1									0	=	1
8		PC_B			-1		1	1							0	=	0
9		PC_C				-1			1	1					0	=	0
10		PC_D													0	=	0
11		PC_E													0	=	0
12		PC_F													0	=	-1
13																	
14																	
15																	
16																	
17																	
18																	
19																	

Solver Parameters

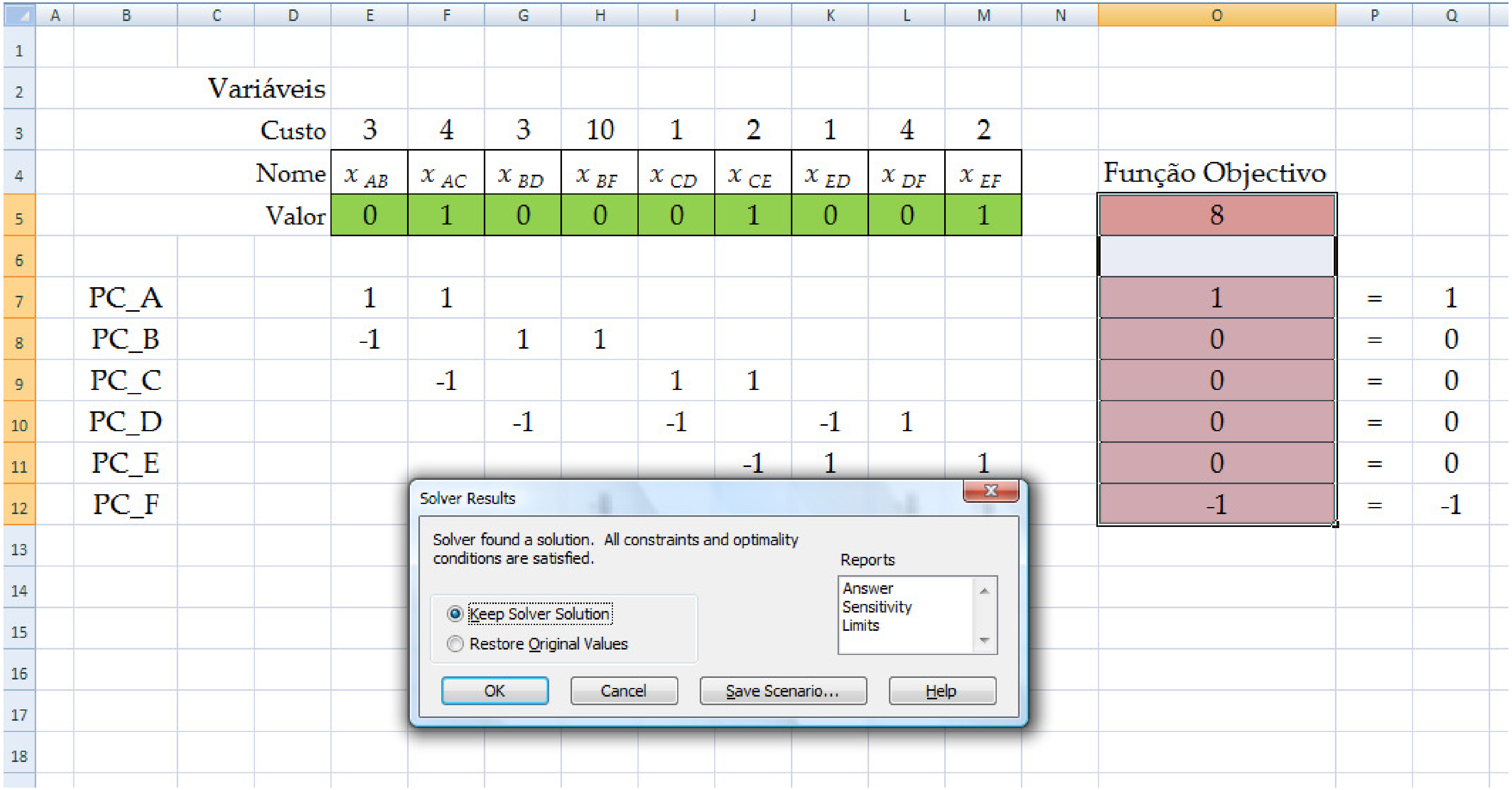
Set Target Cell: $O$5

Equal To: ○ Max ● Min ○ Value of: 0

By Changing Cells: $E$5:$M$5

Subject to the Constraints:
$O$7:$O$12 = $Q$7:$Q$12

Interpretação dos resultados



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2				Variáveis													
3				Custo	3	4	3	10	1	2	1	4	2				
4				Nome	x_{AB}	x_{AC}	x_{BD}	x_{BF}	x_{CD}	x_{CE}	x_{ED}	x_{DF}	x_{EF}		Função Objectivo		
5				Valor	0	1	0	0	0	1	0	0	1		8		
6																	
7		PC_A			1	1									1	=	1
8		PC_B			-1		1	1							0	=	0
9		PC_C				-1			1	1					0	=	0
10		PC_D					-1		-1		-1	1			0	=	0
11		PC_E								-1	1		1		0	=	0
12		PC_F													-1	=	-1

Solver Results

Solver found a solution. All constraints and optimality conditions are satisfied.

● Keep Solver Solution

○ Restore Original Values

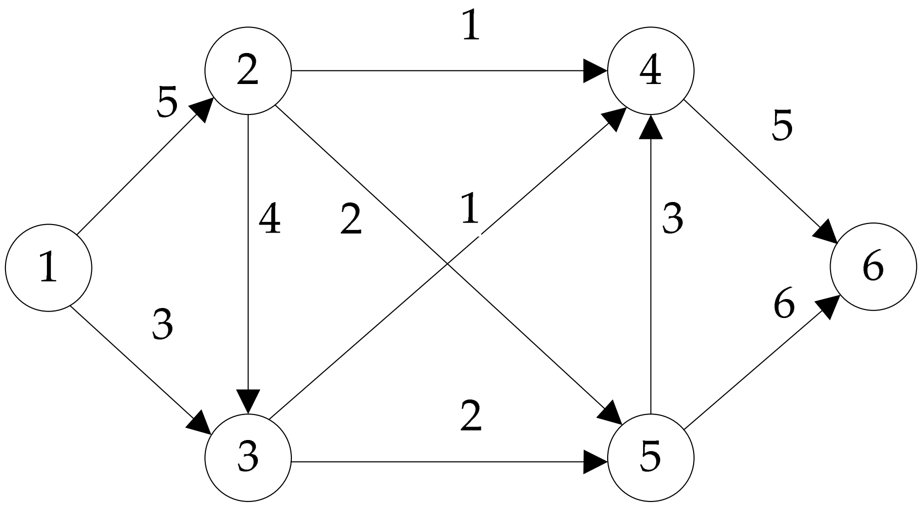
Reports

Answer
Sensitivity
Limits

- Problema de Fluxo Máximo

Exemplo

Considere o grafo representado na figura abaixo. O vértice 1 corresponde à fonte, e o vértice 6 ao destino. As capacidades dos arcos estão indicadas junto aos arcos.



O fluxo máximo que é possível enviar do vértice1 ao vértice 6 é igual a 6. Note que à saída do vértice 1, é possível enviar até 8 unidades de fluxo, contudo não é possível fazer chegar todo esse fluxo até ao vértice 6, dada a estrutura da rede, e uma vez que dos vértices 2 e 3 só podem sair no máximo 6 unidades de fluxo.

Modelo

$Max\ y$

sujeito a

$$x_{12} + x_{13} = y$$

$$-x_{12} + x_{23} + x_{24} + x_{25} = 0$$

$$-x_{13} - x_{23} + x_{34} + x_{35} = 0$$

$$-x_{24} - x_{34} - x_{54} + x_{46} = 0$$

$$-x_{25} - x_{35} + x_{54} + x_{56} = 0$$

$$-x_{46} - x_{56} = -y$$

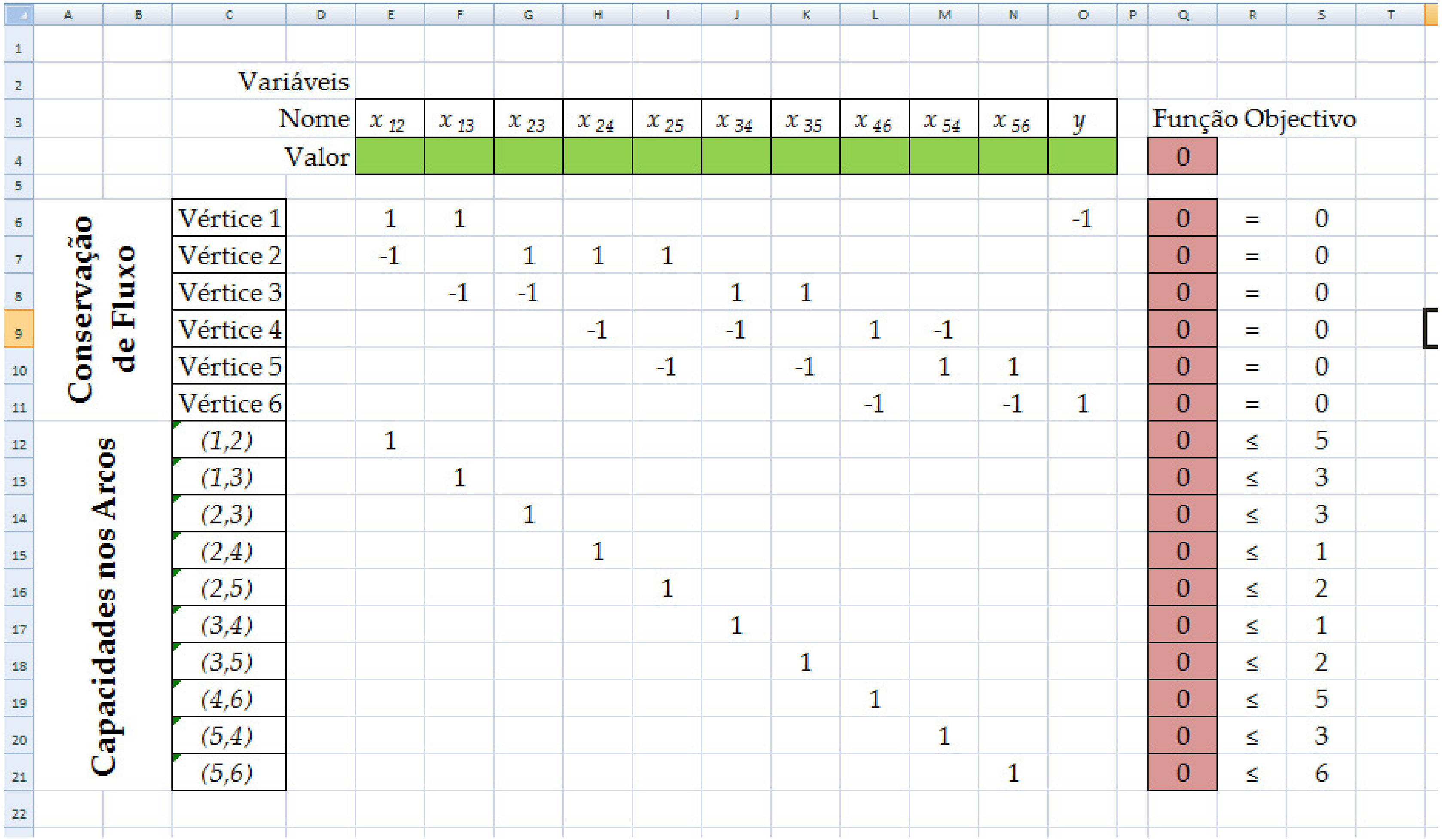
$$x_{12} \leq 5, x_{13} \leq 3, x_{23} \leq 4, x_{24} 1 \leq, x_{25} \leq 2$$

$$x_{34} \leq 1, x_{35} \leq 2, x_{46} \leq 5, x_{54} \leq 3, x_{56} \leq 6$$

$$x_{12}, x_{13}, x_{23}, x_{24}, x_{25}, x_{34}, x_{35}, x_{46}, x_{54}, x_{56} \ \geq 0$$

Representação numa folha de cálculo

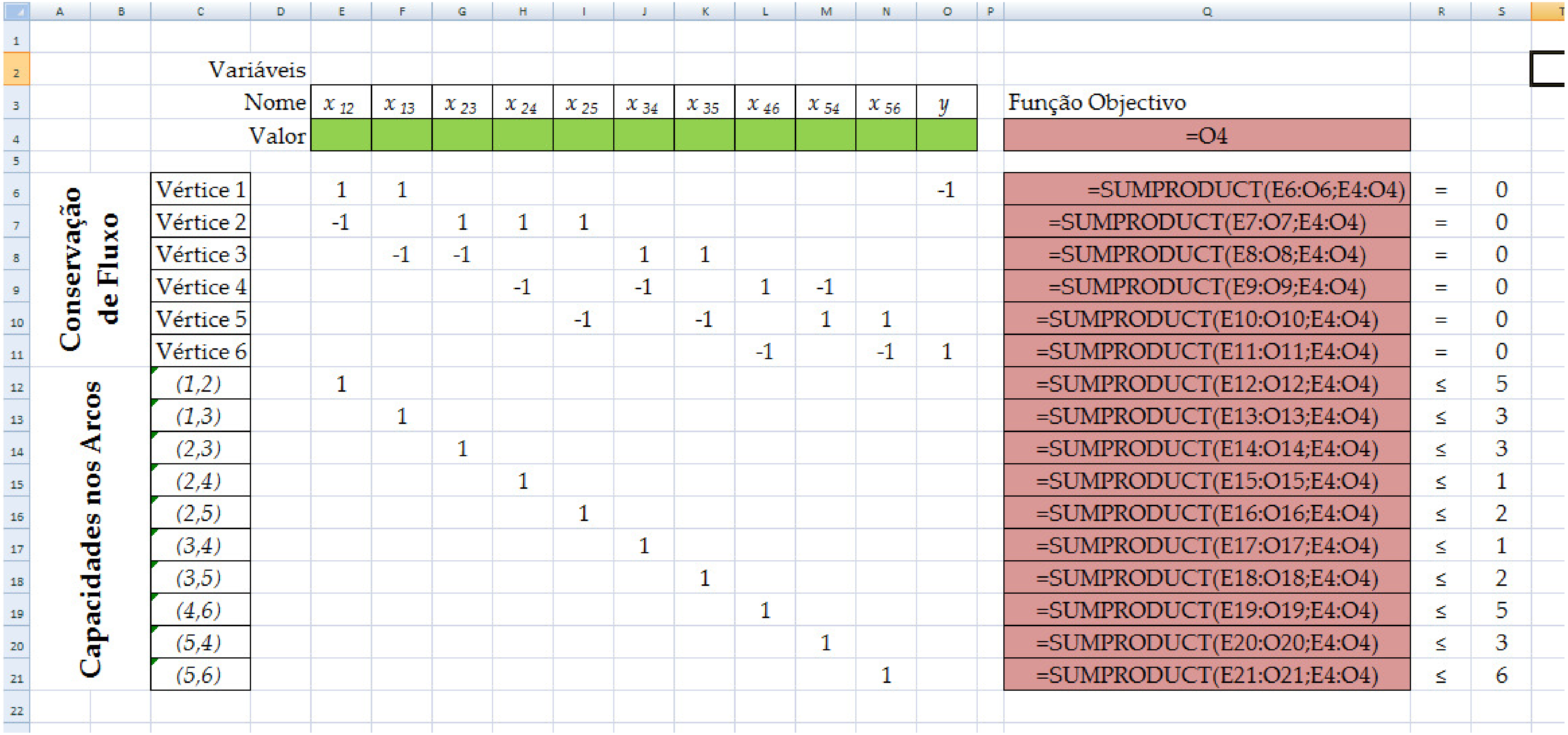
- Passar dados do problema



	Variáveis													
	Nome	x_{12}	x_{13}	x_{23}	x_{24}	x_{25}	x_{34}	x_{35}	x_{46}	x_{54}	x_{56}	y	Função Objectivo		
	Valor												0		
Conservação de Fluxo	Vértice 1	1	1									-1	0	=	0
	Vértice 2	-1		1	1	1							0	=	0
	Vértice 3		-1	-1			1	1					0	=	0
	Vértice 4				-1		-1		1	-1			0	=	0
	Vértice 5					-1		-1		1	1		0	=	0
	Vértice 6								-1		-1	1	0	=	0
Capacidades nos Arcos	*(1,2)*	1											0	≤	5
	(1,3)		1										0	≤	3
	(2,3)			1									0	≤	3
	(2,4)				1								0	≤	1
	(2,5)					1							0	≤	2
	(3,4)						1						0	≤	1
	(3,5)							1					0	≤	2
	(4,6)								1				0	≤	5
	(5,4)									1			0	≤	3
	(5,6)										1		0	≤	6

Representação numa folha de cálculo

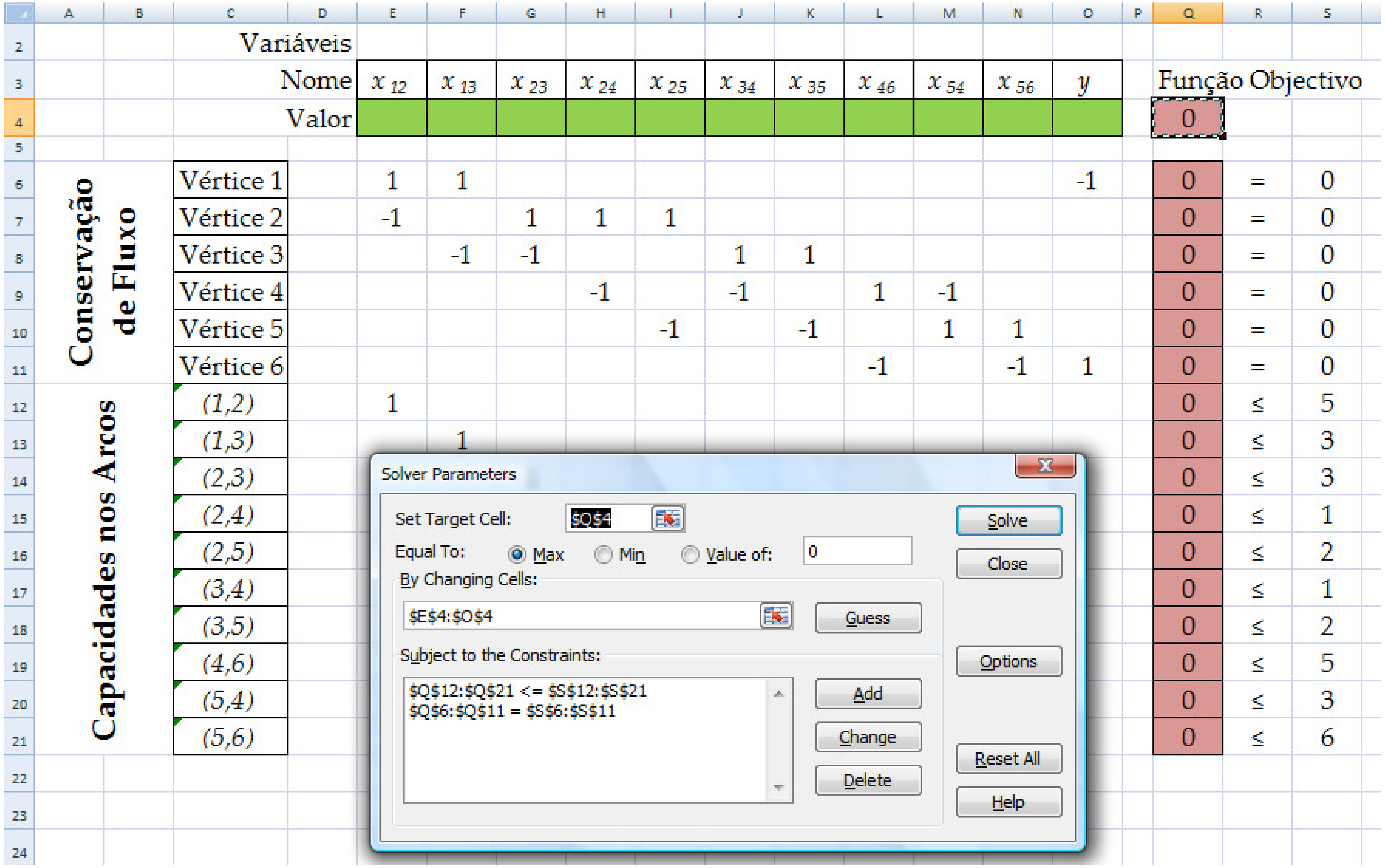
- Converter expressões algébricas em funções Excel



	Variáveis														
	Nome	x_{12}	x_{13}	x_{23}	x_{24}	x_{25}	x_{34}	x_{35}	x_{46}	x_{54}	x_{56}	y	Função Objectivo		
	Valor												=O4		
Conservação de Fluxo	Vértice 1	1	1									-1	=SUMPRODUCT(E6:O6;E4:O4)	=	0
	Vértice 2	-1		1	1	1							=SUMPRODUCT(E7:O7;E4:O4)	=	0
	Vértice 3		-1	-1			1	1					=SUMPRODUCT(E8:O8;E4:O4)	=	0
	Vértice 4				-1		-1		1	-1			=SUMPRODUCT(E9:O9;E4:O4)	=	0
	Vértice 5					-1		-1		1	1		=SUMPRODUCT(E10:O10;E4:O4)	=	0
	Vértice 6								-1		-1	1	=SUMPRODUCT(E11:O11;E4:O4)	=	0
Capacidades nos Arcos	*(1,2)*	1											=SUMPRODUCT(E12:O12;E4:O4)	≤	5
	(1,3)		1										=SUMPRODUCT(E13:O13;E4:O4)	≤	3
	(2,3)			1									=SUMPRODUCT(E14:O14;E4:O4)	≤	3
	(2,4)				1								=SUMPRODUCT(E15:O15;E4:O4)	≤	1
	(2,5)					1							=SUMPRODUCT(E16:O16;E4:O4)	≤	2
	(3,4)						1						=SUMPRODUCT(E17:O17;E4:O4)	≤	1
	(3,5)							1					=SUMPRODUCT(E18:O18;E4:O4)	≤	2
	(4,6)								1				=SUMPRODUCT(E19:O19;E4:O4)	≤	5
	(5,4)									1			=SUMPRODUCT(E20:O20;E4:O4)	≤	3
	(5,6)										1		=SUMPRODUCT(E21:O21;E4:O4)	≤	6

Resolução com o *Solver*

- preencher os campos da janela do *Solver:*



	Variáveis													
	Nome	x_{12}	x_{13}	x_{23}	x_{24}	x_{25}	x_{34}	x_{35}	x_{46}	x_{54}	x_{56}	y	Função Objectivo	
	Valor												0		
Conservação de Fluxo	Vértice 1	1	1									-1	0	=	0
	Vértice 2	-1		1	1	1							0	=	0
	Vértice 3		-1	-1			1	1					0	=	0
	Vértice 4				-1		-1		1	-1			0	=	0
	Vértice 5					-1		-1		1	1		0	=	0
	Vértice 6								-1		-1	1	0	=	0
Capacidades nos Arcos	(1,2)	1											0	≤	5
	(1,3)		1										0	≤	3
	(2,3)												0	≤	3
	(2,4)												0	≤	1
	(2,5)												0	≤	2
	(3,4)												0	≤	1
	(3,5)												0	≤	2
	(4,6)												0	≤	5
	(5,4)												0	≤	3
	(5,6)												0	≤	6

Solver Parameters

Set Target Cell: $Q$4

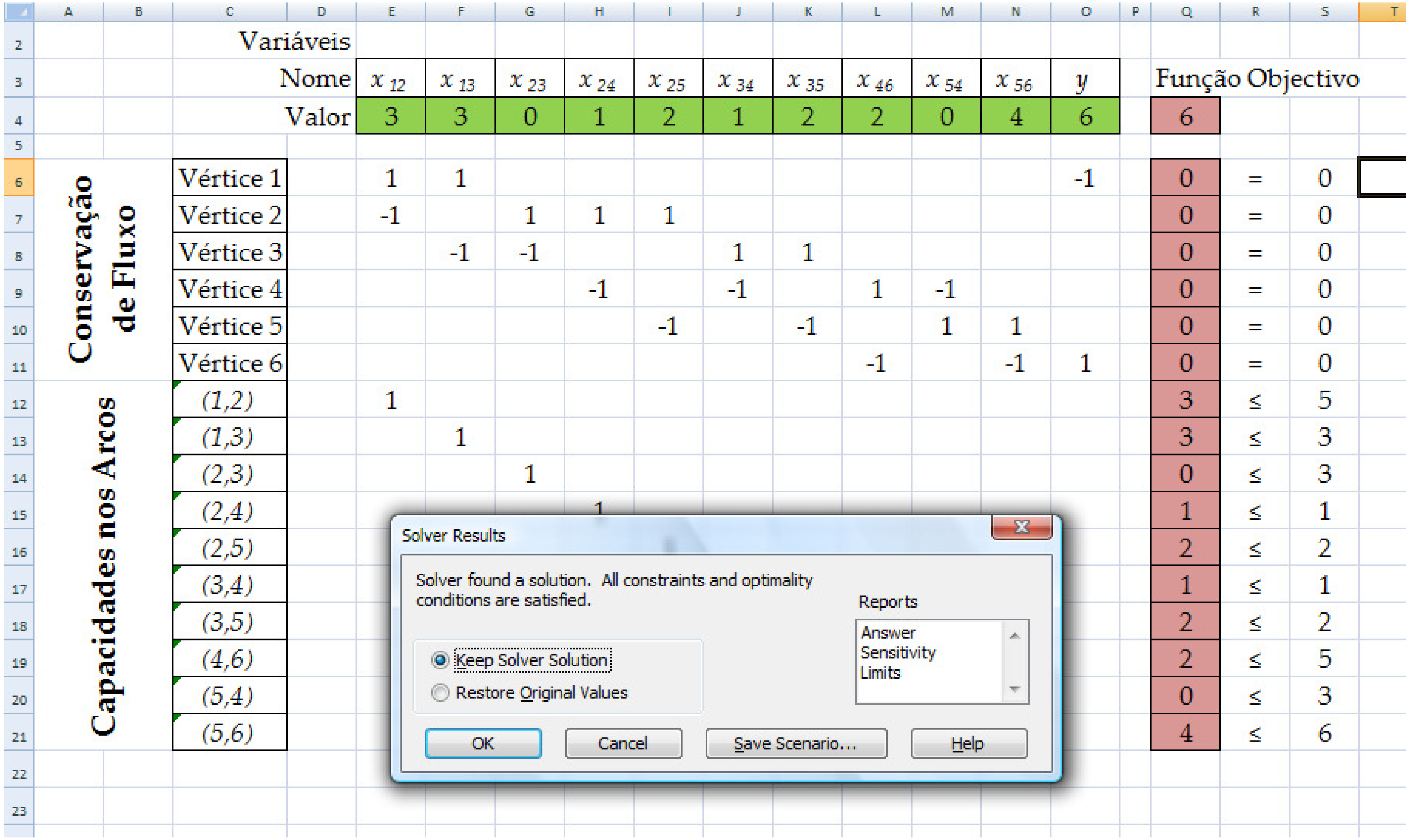
Equal To: ● Max ○ Min ○ Value of: 0

By Changing Cells: $E$4:$O$4

Subject to the Constraints:

$Q$12:$Q$21 <= $S$12:$S$21
$Q$6:$Q$11 = $S$6:$S$11

Interpretação dos resultados



	Variáveis														
	Nome	x_{12}	x_{13}	x_{23}	x_{24}	x_{25}	x_{34}	x_{35}	x_{46}	x_{54}	x_{56}	y	Função Objectivo		
	Valor	3	3	0	1	2	1	2	2	0	4	6	6		
Conservação de Fluxo	Vértice 1	1	1									-1	0	=	0
	Vértice 2	-1		1	1	1							0	=	0
	Vértice 3		-1	-1			1	1					0	=	0
	Vértice 4				-1		-1		1	-1			0	=	0
	Vértice 5					-1		-1		1	1		0	=	0
	Vértice 6								-1		-1	1	0	=	0
Capacidades nos Arcos	*(1,2)*	1											3	≤	5
	(1,3)		1										3	≤	3
	(2,3)			1									0	≤	3
	(2,4)				1								1	≤	1
	(2,5)												2	≤	2
	(3,4)												1	≤	1
	(3,5)												2	≤	2
	(4,6)												2	≤	5
	(5,4)												0	≤	3
	(5,6)												4	≤	6

Solver Results

Solver found a solution. All constraints and optimality conditions are satisfied.

● Keep Solver Solution

○ Restore Original Values

4. Algoritmos de Optimização e Heurísticas

i. Método Simplex para Transportes (Problema de Transportes)

- O Simplex é um método de optimização que permite resolver modelos de Programação Linear; uma vez que o problema de transportes pode ser formulado através de um modelo de Programação Linear, ele também pode ser resolvido usando esse método;
- É possível aproveitar a estrutura do problema de transportes, e melhorar assim a execução desse método geral de optimização;
- O método resultante é designado na literatura por método Simplex para transportes

- A estrutura do problema de transportes faz com que ele possa ser representado através de um quadro com a forma seguinte:

Destinos

Origens	1	2	3	...	n	Disponibilidades
1	c_{11}	c_{12}	c_{13}	...	c_{1n}	d_1
2	c_{21}	c_{22}	c_{23}	...	c_{2n}	d_2
3	c_{31}	c_{32}	c_{33}	...	c_{3n}	d_3
...	...	...	...	...	...	...
m	c_{m1}	c_{m2}	c_{m3}	...	c_{mn}	d_m
Procuras	s_1	s_2	s_3	...	s_n	

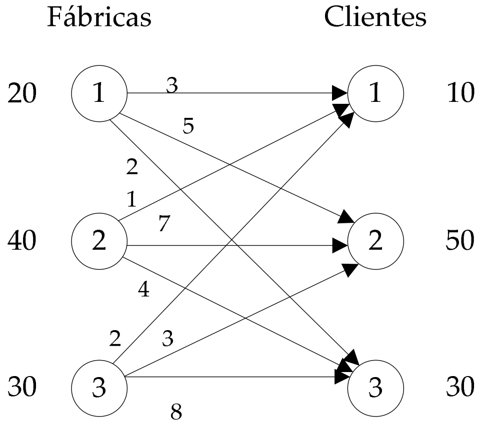
- As origens do problema estão associadas às linhas do quadro;
- Os destinos correspondem às colunas do quadro;
- Os custos de transporte são assinalados no canto superior direito das células;
- Cada célula do quadro representa uma variável do problema;
- Células vazias correspondem a variáveis com valor 0;
- Células com valores representam quantidades não-nulas transportadas entre a origem e o destino associados respectivamente à linha e à coluna da célula (essas células são designadas por *células/variáveis básicas*);
- Uma solução para o problema de transportes é sempre constituída por $m+n-1$ variáveis básicas;
- Em casos especiais, poderão existir variáveis básicas ao nível 0.

- Passos do método Simplex para transportes:

  1. **INICIALIZAÇÃO:**

     - ➢ Procurar uma primeira solução válida;
     - ➢ Essa solução é construída preenchendo células do quadro sequencialmente até obter uma solução que satisfaça todas as restrições do problema;
     - ➢ Essa solução deve ser uma solução básica (m+n-1 variáveis básicas);
     - ➢ As células a preencher são escolhidos segundo um determinado critério (menor custo, maior lucro, posição no quadro);
     - ➢ Ao preencher uma célula, deve-lhe ser atribuído o maior valor possível tendo em conta as restrições a que está sujeita;

- Regras para a selecção da próxima célula a ser preenchida:
  - Regra do canto noroeste: a próxima célula que deve ser preenchida é aquela que se encontra mais acima e á esquerda no quadro;
  - Regra do custo mínimo: a próxima célula a preencher é aquela que possui o menor custo.
- Exemplo: problema de transportes representado pela rede abaixo



❖ Inicialização usando a regra do canto noroeste:

	1	2	3	
1	3 10	5 10	2 —	~~20~~ ~~10~~ 0
2	1 —	7 40	4 —	~~40~~ 0
3	2 —	3 0	8 30	~~30~~ 0
	~~10~~ 0	~~50~~ 0	~~30~~	

➢ Solução tem um custo igual a $10 \times 3 + 10 \times 5 + 40 \times 7 + 30 \times 8 = 600$

- ❖ Inicialização usando a regra do custo mínimo:

	1	2	3	
1	3	5	2 20	~~20~~ 0
2	1 10	7 20	4 10	~~40~~ ~~30~~ ~~20~~ 0
3	2	3 30	8	~~30~~ 0
	~~10~~ 0	~~50~~ ~~20~~	~~30~~ ~~10~~	

- ➢ Solução tem um custo igual a $20 \times 2 + 10 + 20 \times 7 + 4 \times 10 + 30 \times 3 = 320$

2. VERIFICAR SE A SOLUÇÃO É ÓPTIMA:

- A cada linha i do quadro associa-se uma variável u_i, $i=1,\ldots,m$;
- A cada coluna j do quadro associa-se uma variável v_j, $j=1,\ldots,n$;
- Para as células básicas, temos:

$$u_i+v_j=c_{ij}$$

- Para as células não-básicas, a quantidade $c_{ij}-(u_i+v_j)$ mede a atractividade dessa célula:
  - Num problema de minimização, se todos esses valores forem não-negativos, a solução é óptima;
  - Num problema de maximização, se todos esses valores forem não-positivos, a solução é óptima.

- Exemplo: partindo da solução inicial obtida pela regra do canto noroeste
  - Cálculo dos u_i e dos v_j

Para as células básicas

$$u_1 + v_1 = 3$$

$$u_1 + v_2 = 5$$

$$u_2 + v_2 = 7$$

$$u_3 + v_2 = 3$$

$$u_3 + v_3 = 8$$

Solução válida actual

	1	2	3	
1	3 10	5 10	2	20
2	1	7 40	4	40
3	2	3 0	8 30	30
	10	50	30	

- Existem 5 equações (*m*+*n*-*1*, uma por cada célula básica) e 6 variáveis (*m*+*n*); podemos arbitrar o valor de uma dessas variáveis
- Fazendo u_1=0

Para as células básicas

$$u_1 = 0; v_1 = 3$$

$$v_2 = 5$$

$$u_2 = 2$$

$$u_3 = -2$$

$$v_3 = 10$$

Solução válida actual

	1	2	3	
1	3 10	5 10	2	20
2	1	7 40	4	40
3	2	3 0	8 30	30
	10	50	30	

- Análise das células não-básicas

Para as células não-básicas

(1,3)	$c_{13} - (u_1 + v_3)$
(2,1)	$c_{21} - (u_2 + v_1)$
(2,3)	$c_{23} - (u_2 + v_3)$
(3,1)	$c_{31} - (u_3 + v_1)$

Solução válida actual

	1	2	3	
1	3 10	5 10	2	20
2	1	7 40	4	40
3	2	3 0	8 30	30
	10	50	30	

- Análise das células não-básicas

Para as células não-básicas

(1,3)	$2-(0+10)=-8$
(2,1)	$1-(2+3)=-4$
(2,3)	$4-(2+10)=-8$
(3,1)	$2-(-2+3)=1$

Solução válida actual

	1	2	3	
1	3 10	5 10	2	20
2	1	7 40	4	40
3	2	3 0	8 30	30
	10	50	30	

➢ Existem $c_{ij}-(u_i+v_j)$ negativos, logo a solução não é óptima

3. SE A SOLUÇÃO NÃO É ÓPTIMA: DETERMINAR UMA SOLUÇÃO VÁLIDA DE MENOR CUSTO (PROBLEMAS DE MINIMIZAÇÃO); VOLTAR AO PASSO 2

- Escolher a célula não-básica mais atractiva:
  - Em problemas de MIN, aquela com o valor de c_{ij}-(u_i+v_j) mais negativo;
  - Em problemas de MAX, aquela com o valor de c_{ij}-(u_i+v_j) maior.
- Determinar o maior valor que essa célula pode tomar:
  - Construir um circuito no quadro composto apenas por células básicas (excepto a célula não-básica cujo valor pretendemos determinar);
  - Atribuir o maior valor à célula não-básica de forma a que nenhuma das outras células passe a ter um valor negativo

- Exemplo: vamos escolher a célula (1,3)

	1	2	3	
1	$\overset{3}{10}$	$\overset{5}{10_{-\theta}}$	$\overset{2}{+\theta}$	20
2	$\overset{1}{}$	$\overset{7}{40}$	$\overset{4}{}$	40
3	$\overset{2}{}$	$\overset{3}{0_{+\theta}}$	$\overset{8}{30_{-\theta}}$	30
	10	50	30	

- O maior valor que θ pode tomar é 10;

- O quadro é actualizado com a nova solução; custo dea solução: 520;

	1	2	3	
1	3 10	5	2 10	20
2	1	7 40	4	40
3	2	3 10	8 20	30
	10	50	30	

- **Atenção**: em cada iteração, uma célula passa a básica, e apenas uma passa a não-básica; em alguns casos, algumas células básicas passam a estar ao nível 0;
- O quadro tem sempre *(m+n)-1* células básicas.

- O processo é repetido desde o passo 2.

Iteração 2

		$v_1=3$	$v_2=-3$	$v_3=2$	
		1	2	3	
$u_1=0$	1	3 10 $-\theta$	8 5	2 10 $+\theta$	20
$u_2=10$	2	-12 1 $+\theta$	7 40 $-\theta$	-8 4	40
$u_3=6$	3	-7 2	3 10 $+\theta$	8 20 $-\theta$	30
		10	50	30	

$\theta=10$

Iteração 3

		$v_1=-9$	$v_2=-3$	$v_3=2$	
		1	2	3	
$u_1=0$	1	12 3	8 5	2 20	20
$u_2=10$	2	1 10	7 30 $-\theta$	-8 4 $+\theta$	40
$u_3=6$	3	5 2	3 20 $+\theta$	8 10 $-\theta$	30
		10	50	30	

$\theta=10$

- A solução obtida no final da 3ª iteração é óptima; Custo=320.

		$v_1=-1$	$v_2=5$	$v_3=2$	
		1	2	3	
$u_1=0$	1	4 3	0 5	2 20	20
$u_2=2$	2	1 10	7 20	4 10	40
$u_3=-2$	3	5 2	3 30	8 8	30
		10	50	30	

- Síntese do método Simplex para transportes:

  Pré-requisito: total da disponibilidade é igual ao total da procura;

  1. INICIALIZAÇÃO: determinar uma primeira solução básica válida;
  2. TESTE DE OPTIMALIDADE:
     - ➢ Calcular u_i e v_j;
     - ➢ Calcular cij-(u_i+v_j) para as células não-básicas; a solução é óptima se:
       - Num problema MIN: esses valores forem todos não-negativos;
       - Num problema MAX: esses valores forem todos não positivos.
  3. SE A SOLUÇÃO É ÓPTIMA, PARAR,

     SENÃO DETERMINAR UMA MELHOR SOLUÇÃO BÁSICA VÁLIDA:
     - Escolher a célula mais atractiva, e determinar o maior valor que pode tomar;
     - Actualizar o quadro, e voltar ao passo2.

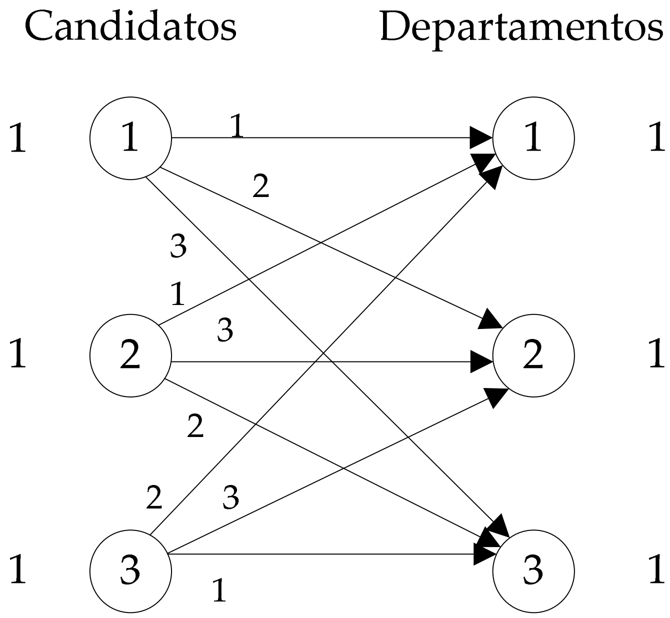
ii. Algoritmo Húngaro (Problema de Afectação)

- Sendo um caso especial do problema de transportes, o problema de afectação pode ser resolvido usando o método Simplex para transportes;
- Contudo, o método Simplex para transportes é um método geral que não tira partido das particularidades do problema de afectação;
- Por outro lado, ao usar o método Simplex para transportes, verificamos que muitas iterações são iterações onde na prática a solução não é alterada (problema de existirem muitas soluções básicas ao nível 0);
- O algoritmo Húngaro é um método especializado que permite tirar partido da estrutura do problema de afectação.

- A estrutura do problema de afectação faz com que ele possa ser representado através de um quadro com a forma seguinte:

				Tarefas		
		1	2	3	...	n
Entidades	1	c_{11}	c_{12}	c_{13}	...	c_{1n}
	2	c_{21}	c_{22}	c_{23}	...	c_{2n}
	3	c_{31}	c_{32}	c_{33}	...	c_{3n}
	...	...	...	...	...	...
	m	c_{m1}	c_{m2}	c_{m3}	...	c_{mn}

- Exemplo: problema de afectação representado pela rede abaixo; o quadro associado está representado à esquerda;



	1	2	3
1	1	2	3
2	1	3	2
3	2	3	1

- Note que somar ou subtrair qualquer quantidade aos elementos (custos) de uma linha ou coluna não altera o problema; a solução óptima continuará a ser a mesma;
- Quadros equivalentes para o exemplo acima:

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

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

- Se num quadro for possível identificar uma solução de custo 0, então essa solução é necessariamente óptima (assumindo que todos os custos são não-negativos);
- O algoritmo Húngaro procura converter o quadro noutro em que é possível identificar uma solução de custo 0;
- Essa conversão é feita somando e/ou subtraindo valores aos elementos das linhas e colunas do quadro;
- Logo que exista pelo menos uma solução de custo 0 no quadro, uma dessas soluções é construída passo a passo.

- Síntese do algoritmo Húngaro para problemas de afectação:

  Pré-requisito: número de origens (entidades) ser igual ao número de tarefas;

  1. Aos elementos de cada linha, subtrair o menor elemento dessa linha;
  2. Aos elementos de cada coluna, subtrair o menor elemento dessa coluna;
  3. VERIFICAR SE EXISTE UMA SOLUÇÃO DE CUSTO 0:
     - Cobrir os 0 do quadro com o **menor** número de traços verticais e/ou horizontais;
     - Se o número de traços for igual à dimensão do problema, existe uma solução de custo 0; passar ao ponto 5, senão ir ao ponto 4;
  4. CONTINUAR A SIMPLIFICAR O QUADRO:
     - Dos elementos não traçados no ponto 3., escolher o menor;
     - Subtrair esse valor a todos os elementos do quadro;

- Somar esse valor a todos os elementos das linhas traçadas;
- Somar esse valor a todos os elementos das colunas traçadas.

Nota: Esses três últimos passos são equivalentes aos seguintes:

- Subtrair esse valor a todos os elementos não traçados do quadro;
- Somar esse valor a todos os elementos traçados duas vezes;

5. CONSTRUIR UMA SOLUÇÃO DE CUSTO 0:

- Começar com as colunas e linhas que têm apenas um 0;
- Escolher as células de custo 0 passo a passo, de modo a que não haja mais de uma célula escolhida por linha e por coluna.

- Exemplo: Considere o problema de afectação representado pelo quadro à esquerda

	1	2	3
1	3	5	2
2	1	7	4
3	2	6	8

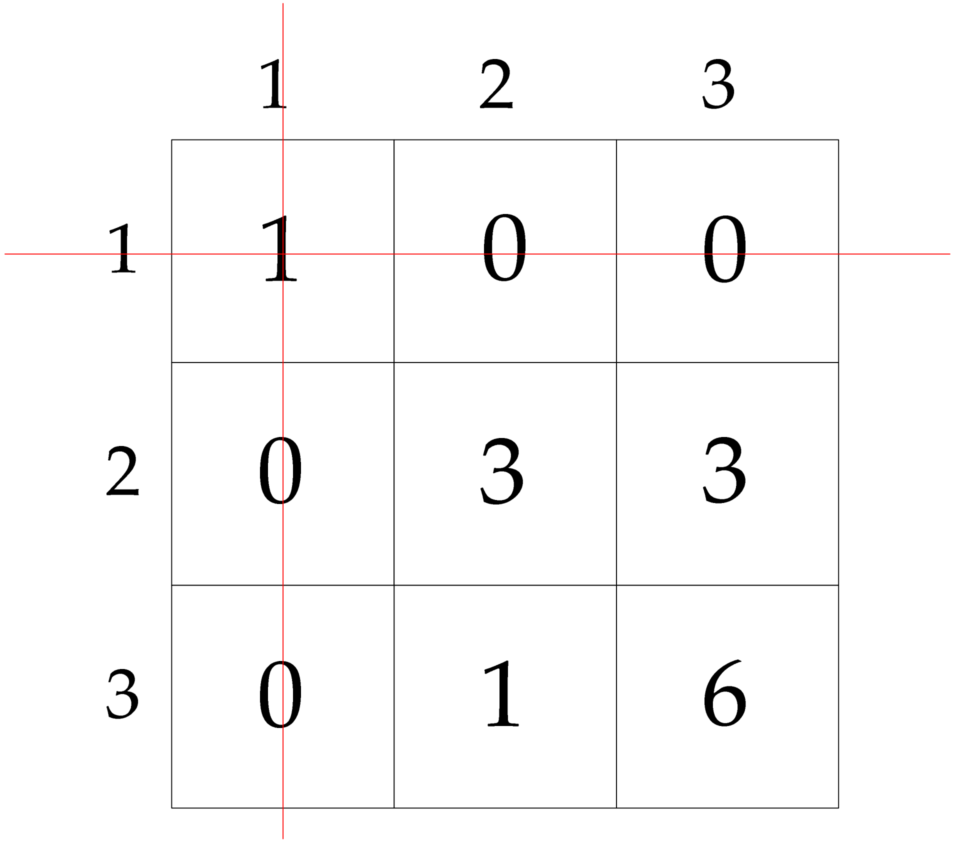
1.

	1	2	3
1	1	3	0
2	0	6	3
3	0	4	6

2.

	1	2	3
1	1	0	0
2	0	3	3
3	0	1	6

3. São necessários apenas 2 traços; 2 < 3, logo não é possível ainda determinar uma solução de custo 0;

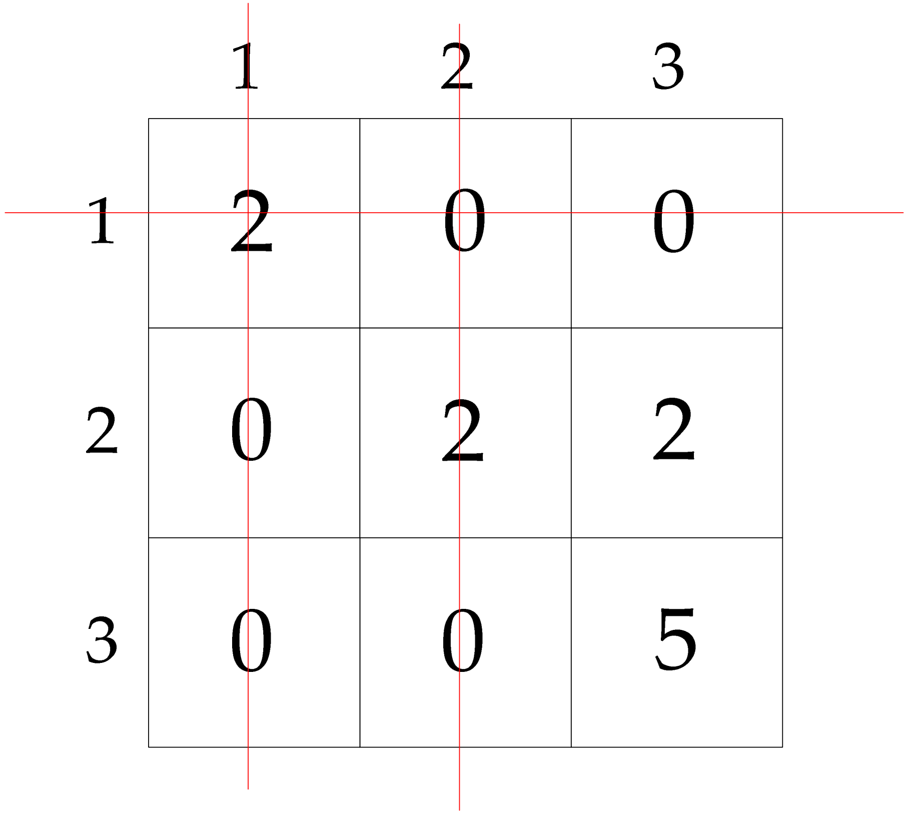


$$\begin{array}{c|c|c|c}
 & 1 & 2 & 3 \\
\hline
1 & 1 & 0 & 0 \\
\hline
2 & 0 & 3 & 3 \\
\hline
3 & 0 & 1 & 6
\end{array}$$

4. O menor elemento não traçado é igual a 1; esse valor é subtraído aos elementos não traçados, e somado aos elementos traçados 2 vezes;

	1	2	3
1	2	0	0
2	0	2	2
3	0	0	5

3. São necessários apenas 3 traços; já é possível construir uma solução de custo 0;



$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 2 & 0 & 0 \\ 2 & 0 & 2 & 2 \\ 3 & 0 & 0 & 5 \end{array}$$

5. A solução é ilustrada no quadro seguinte: consiste nas associações 1→3, 2→1 e 3→2; custo da solução = 9.

No quadro final simplificado

	1	2	3
1	2	0	**0**
2	**0**	2	2
3	0	**0**	5

No quadro original

	1	2	3
1	3	5	**2**
2	**1**	7	4
3	2	**6**	8

iii. Algoritmo de Dijkstra (Problema de Caminho Mais Curto)

- O algoritmo de Dijkstra permite calcular o caminho mais curto entre um vértice de um grafo (a fonte) e todos os outros vértices do grafo;
- O algoritmo pode ser usado em grafos onde não existem arcos com custos negativos;
- Existem outros algoritmos para este problema que permitem abordar casos mais genéricos;
- Os caminhos determinados através do algoritmo de Dijkstra formam uma árvore de caminhos mais curtos: qualquer subcaminho de um caminho mais curto é por sua vez o caminho mais curto entre as respectivas extremidades desse subcaminho ;
- O algoritmo de Dijkstra baseia-se na atribuição de rótulos aos vértices do grafo; à diferença de outros algoritmos esses rótulos, uma vez fixados, não voltam a ser alterados.

- Definições:
    - Dados de entrada: um grafo $G=(V,A)$; custos nos arcos c_{ij}, $(i,j)\in A$
    - O rótulo de um vértice i é constituída por dois elementos:
        - *d(i)*: distância desde a fonte até ao vértice i seguindo pelo caminho mais curto;
        - *pred(i)*: predecessor do vértice i no caminho mais curto entre a fonte e o vértice i.
    - N: conjunto de vértices cujo rótulo está fixada definitivamente;
    - $\overline{N}$: conjunto de vértices com rótulo temporário (temos que $\overline{N} \cup N = V$).

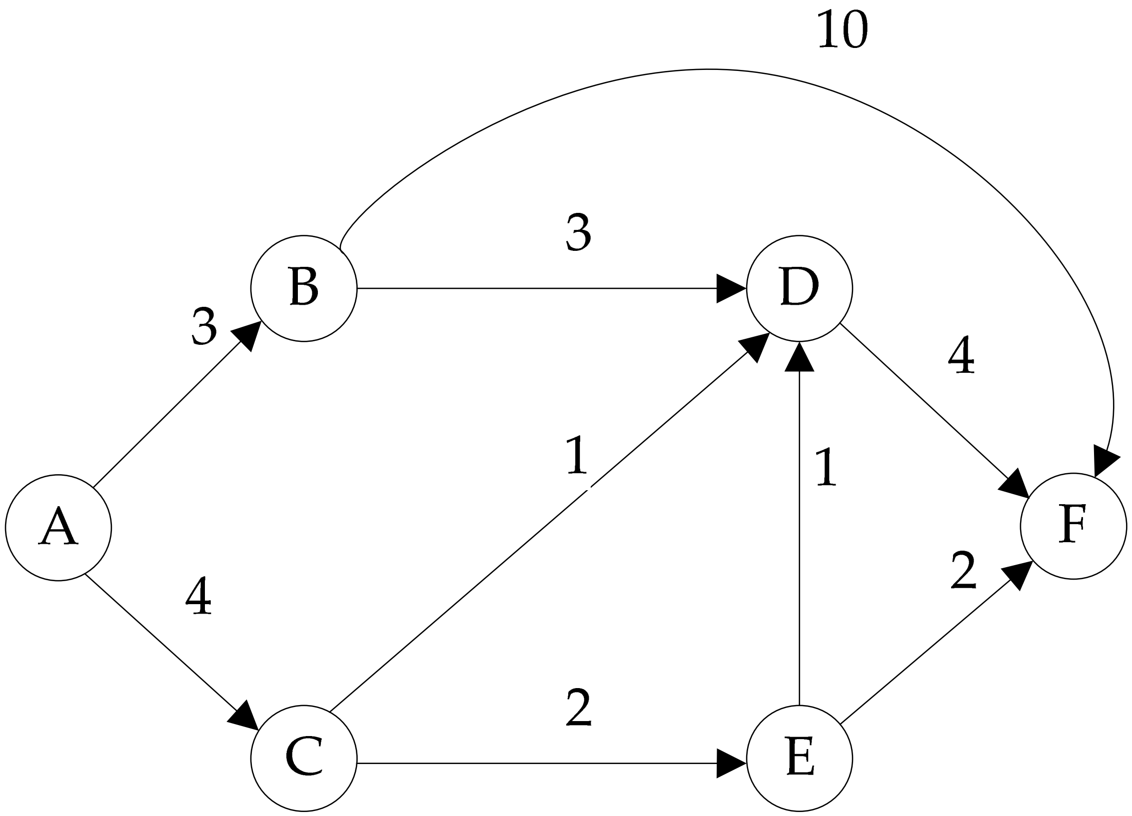
- O objectivo do algoritmo de Dijkstra está em atribuir um rótulo definitivo a todos os vértices do grafo; quando $N=V$, o algoritmo termina.

- Síntese do algoritmo de Dijkstra para problemas de caminho mais curto:

Seja "o" o vértice de origem (a fonte);

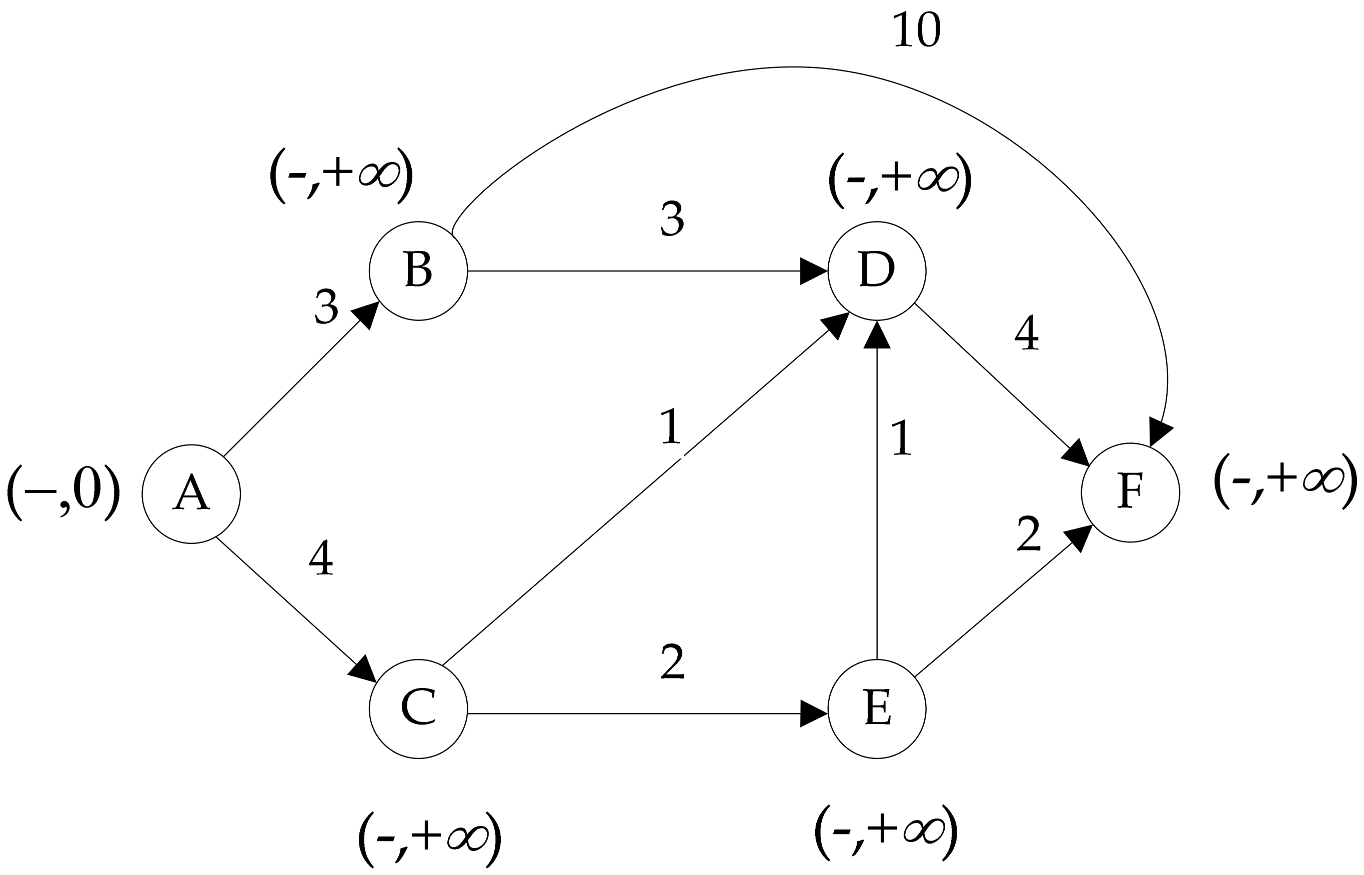
1. INICIALIZAÇÃO
    - $N=\{\}$; $d(o)=0$; $pred(o)=-$;
    - $d(i)=+\infty$,e $pred(i)=-$ para todos os $i \in V \setminus \{o\}$;
2. ACTUALIZAÇÃO DOS RÓTULOS:
    - Escolher um vértice i de $\overline{N}$ com o menor valor $d(i)$;
    - Fazer $\overline{N} = \overline{N} \setminus \{i\}$ e $N = N \cup \{\{i\}\}$;
    - Para todos os $j \in \overline{N}$ tal que $(i,j) \in A$, se $d(i)+c_{ij}<d(j)$, então $d(j)= d(i)+c_{ij}$ e $pred(j)=i$.
3. SE $N=V$ ($\overline{N} = \emptyset$), parar, SENÃO voltar ao passo 2.

- Exemplo:
  - Origem: Vértice A; Destino: Vértice F;



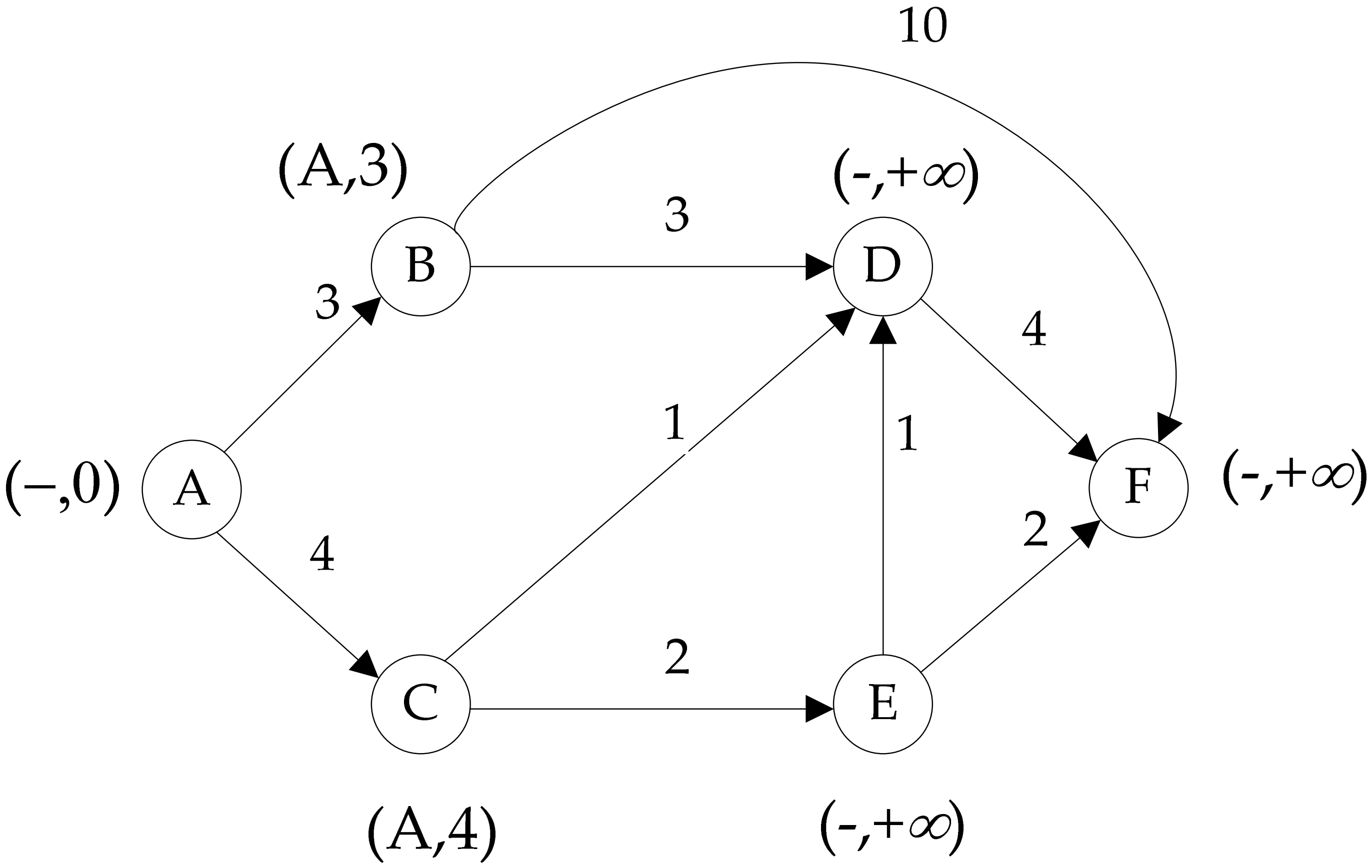
1. INICIALIZAÇÃO

$N = \{\}; \overline{N} = \{A, B, C, D, E, F\}.$



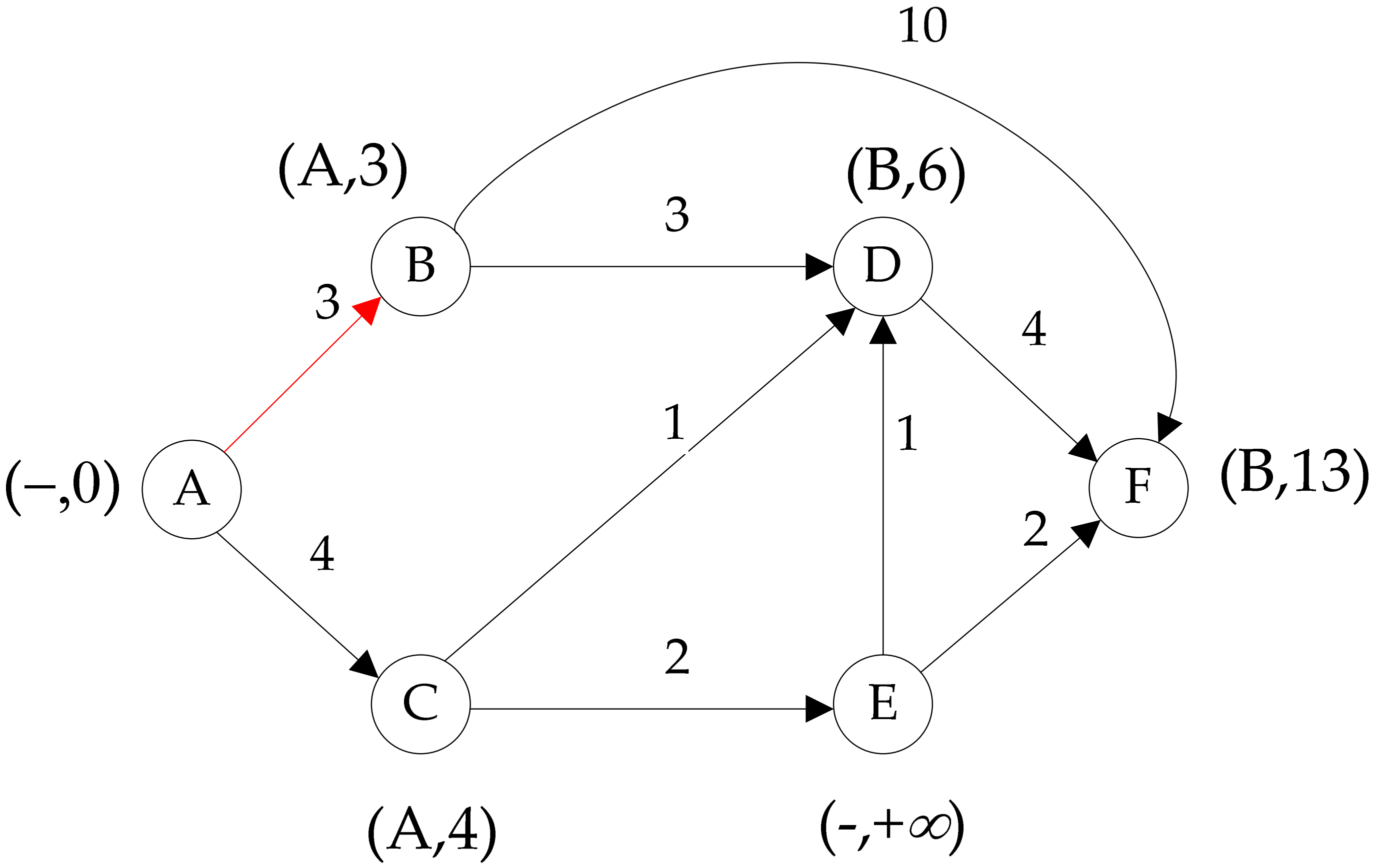
2. ACTUALIZAÇÃO DOS RÓTULOS

$N = \{A\}; \overline{N} = \{B, C, D, E, F\}.$



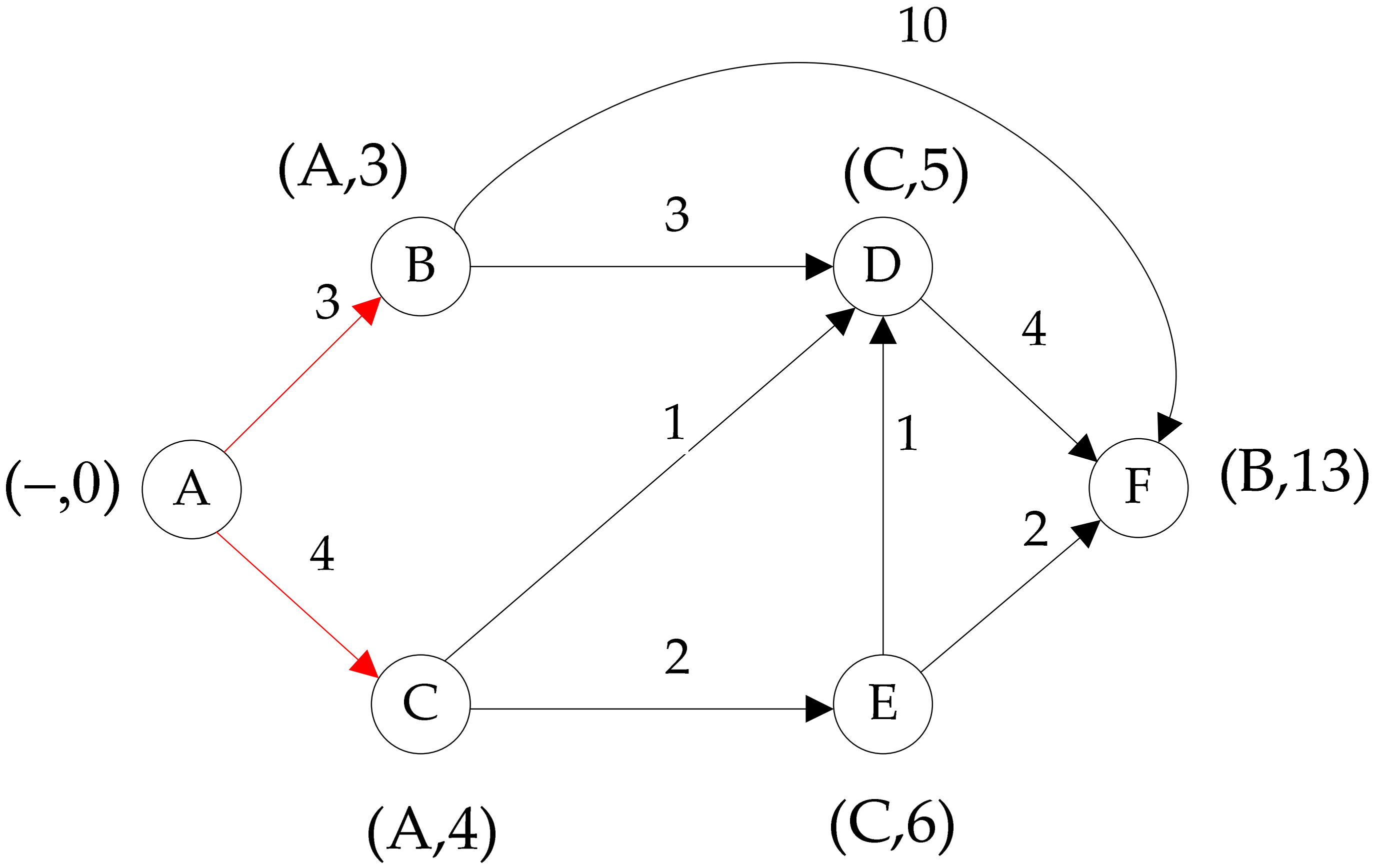
2. ACTUALIZAÇÃO DOS RÓTULOS

$N = \{A, B\}; \overline{N} = \{C, D, E, F\}.$



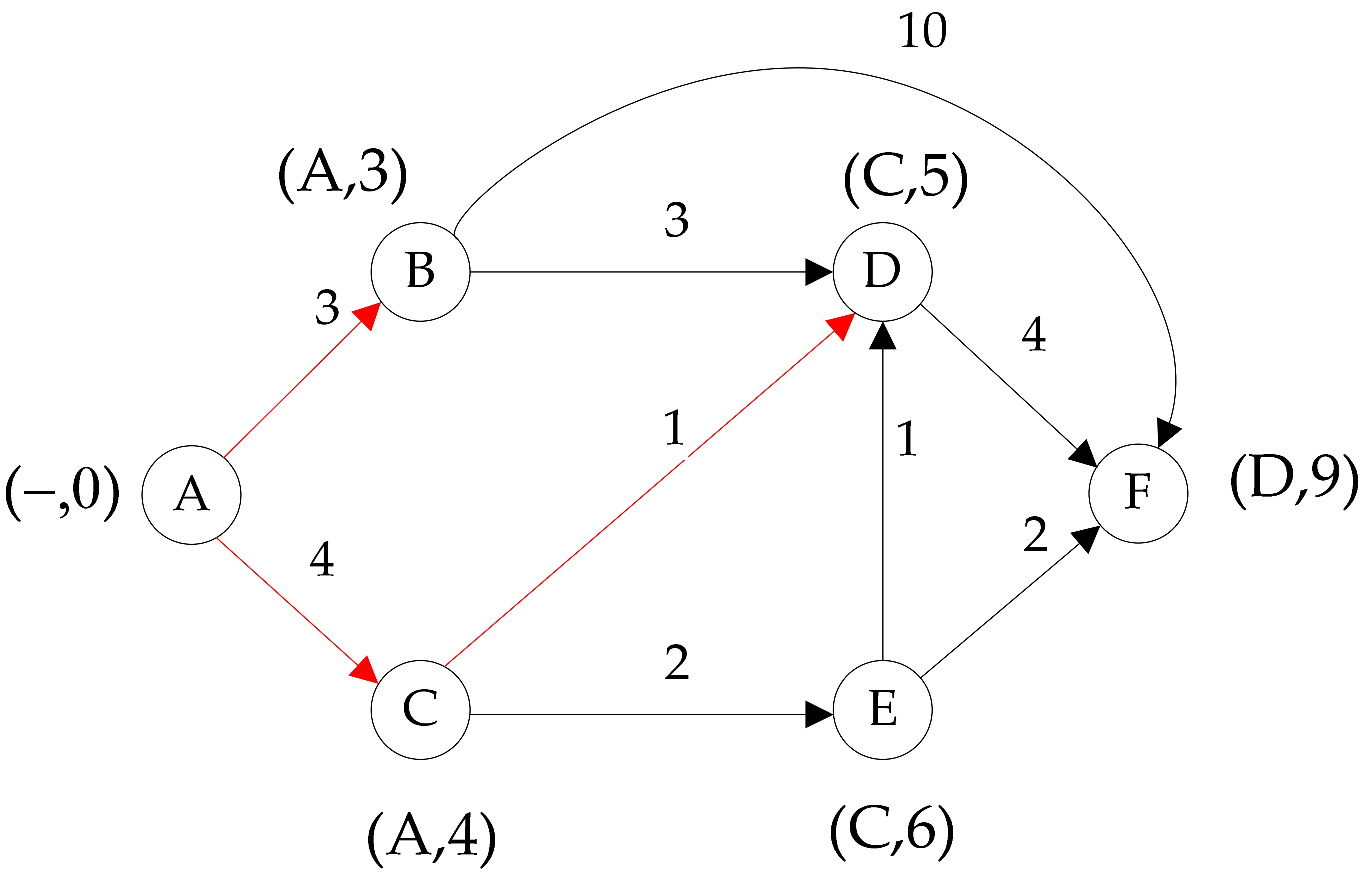
2. ACTUALIZAÇÃO DOS RÓTULOS

$$N = \{A, B, C\}; \overline{N} = \{D, E, F\}.$$



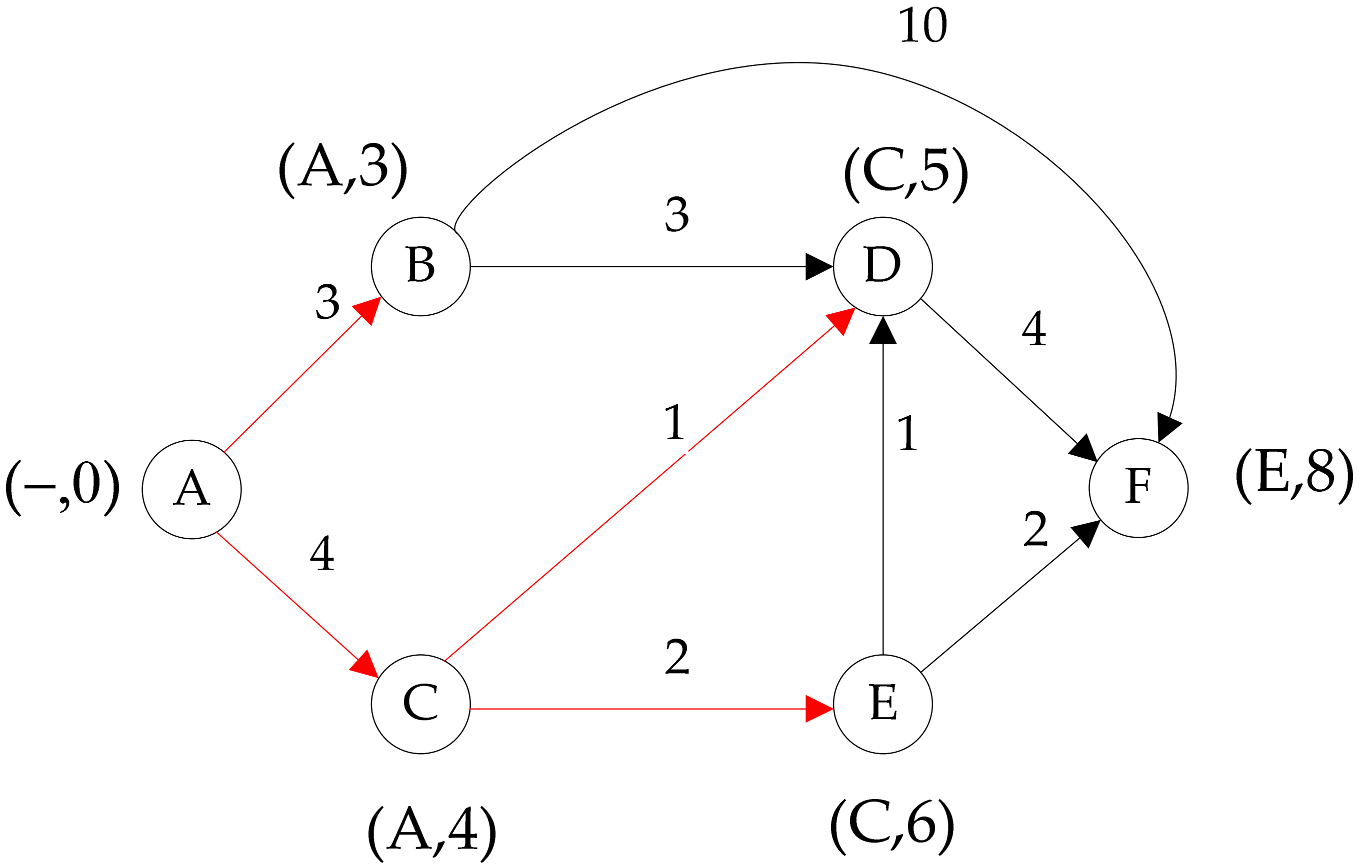
2. ACTUALIZAÇÃO DOS RÓTULOS

$N = \{A, B, C, D\}; \overline{N} = \{E, F\}.$



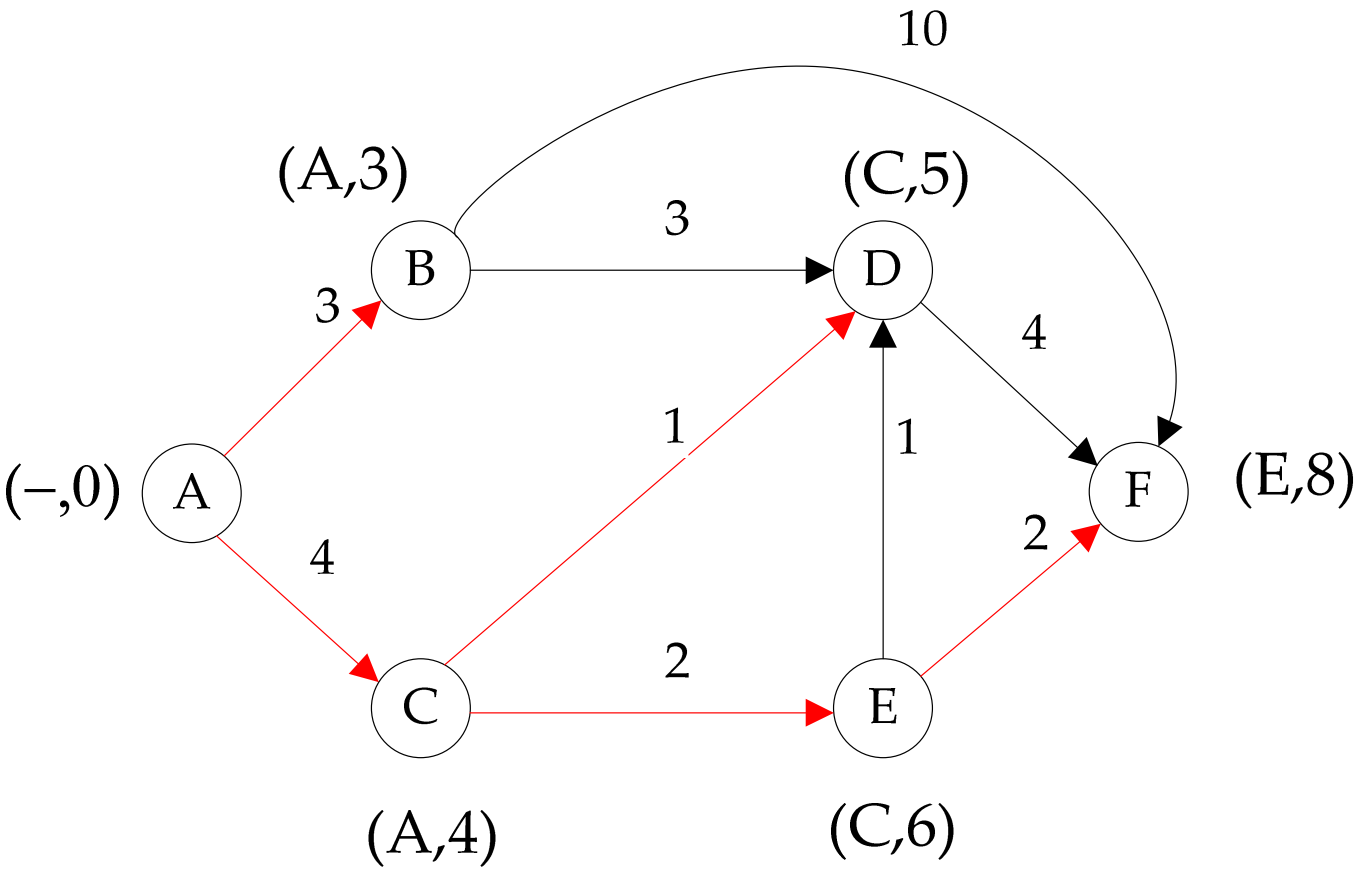
2. ACTUALIZAÇÃO DOS RÓTULOS

$N = \{A, B, C, D, E\}; \overline{N} = \{F\}.$



2. ACTUALIZAÇÃO DOS RÓTULOS

$N = \{A, B, C, D, E, F\}; \overline{N} = \emptyset.$



3. N**=**V **(**$\overline{N} = \emptyset$ **), O ALGORITMO PÁRA.**

OUTPUT DO ALGORITMO DE DIJKSTRA :

Caminho mais curto entre A e F: A-C-E-F; CUSTO=8

Árvore de caminhos mais curtos (a vermelho na última figura):

ORIGEM	DESTINO	CAMINHO	CUSTO
A	B	A-B	3
	C	A-C	4
	D	A-C-D	5
	E	A-C-E	6
	F	A-C-E-F	8

iv. Algoritmo de Kruskal (Problema da Árvore de Suporte de Custo Mínimo)

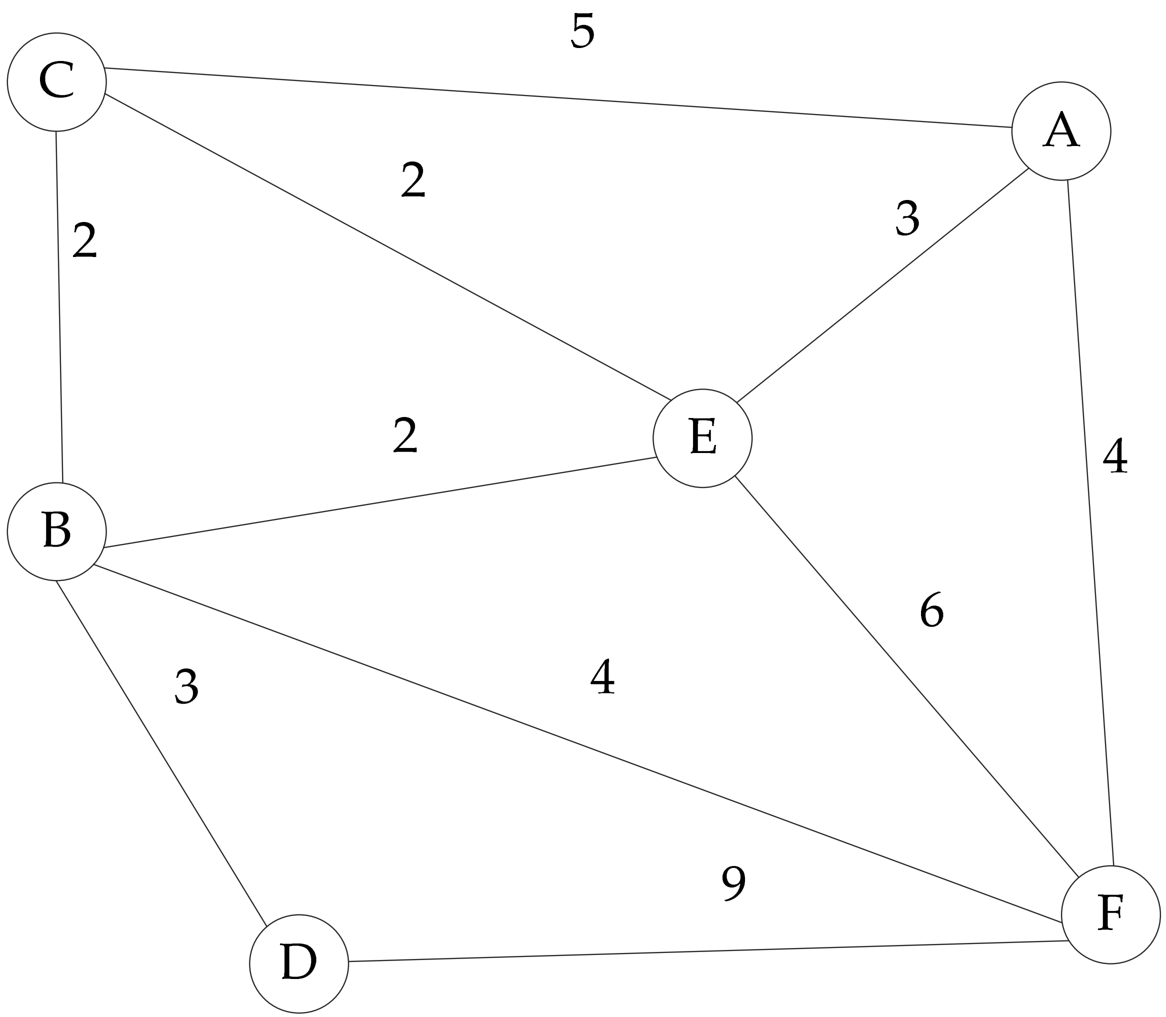
- Uma árvore de suporte de um grafo $G=(V,A)$ é um grafo $G'=(V,A')$ tal que $A' \subseteq A$, e existe um caminho entre qualquer par de vértices de V;
- Um grafo $G=(V,A)$ é uma árvore se tiver exactamente |V|-1 arcos, e não tiver ciclos;
- O problema da árvore de suporte de custo mínimo consiste em determinar o subgrafo de um grafo G que garante que todos os vértices do grafo estão ligados entre eles usando um conjunto de arcos cujo custo total seja o menor possível;
- O problema tem muitas aplicações na área das telecomunicações, por exemplo;
- O problema pode ser resolvido de forma muito eficiente através de diferentes algoritmos;
- O algoritmo de Kruskal é um desses algoritmos; apesar da sua simplicidade o algoritmo retorna sempre a solução óptima do problema.

- Síntese do algoritmo de Kruskal para problemas da árvore de suporte de custo mínimo:

Considere um grafo não-orientado G=(V,A);

1. INICIALIZAÇÃO
    - Fazer $S = \emptyset$; $R = A$;
2. INSERÇÃO DE UM ARCO
    - Escolher o arco *(i,j)* de *R* de menor custo e tal que $S \cup \{i,j\}$ seja um grafo sem ciclos;
    - Fazer $S = S \cup \{(i,j)\}$ e $R=R \setminus \{(i,j)\}$
3. SE $|S|=|V|-1$, parar, SENÃO voltar ao passo 2.

- Exemplo: grafo *G=(V,A)* com |V|=6

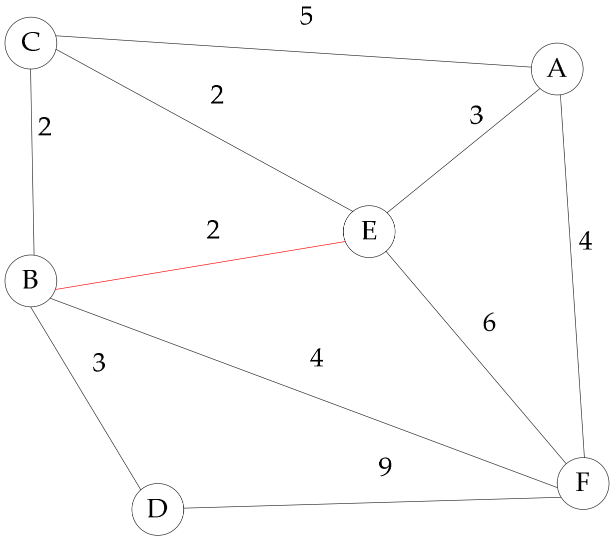


1. INICIALIZAÇÃO

$$S = \emptyset;\ R = \{(A,C),(A,E),(A,F),(B,C),(B,E),(B,D),(B,F),(C,E),(D,F),(E,F)\}$$

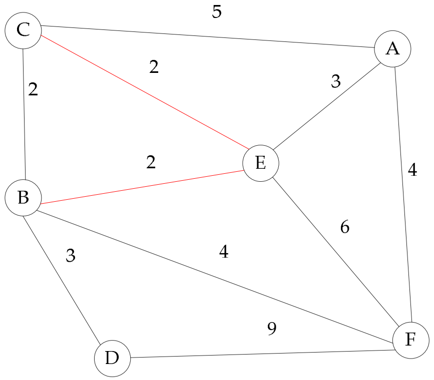
2. INSERÇÃO DE UM ARCO

$$S = \{(B,E)\};\ R = \{(A,C),(A,E),(A,F),(B,C),(B,D),(B,F),(C,E),(D,F),(E,F)\}$$



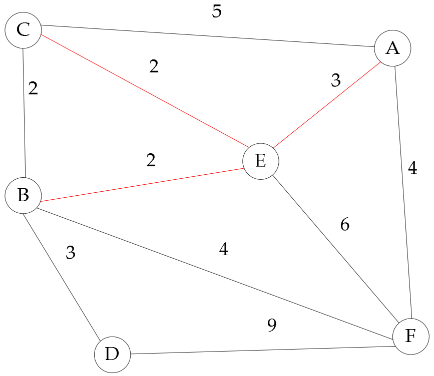
$|S| = 1 \neq 5;$

$$S = \{(B,E),(C,E)\};\ R = \{(A,C),(A,E),(A,F),(B,C),(B,D),(B,F),(D,F),(E,F)\}$$



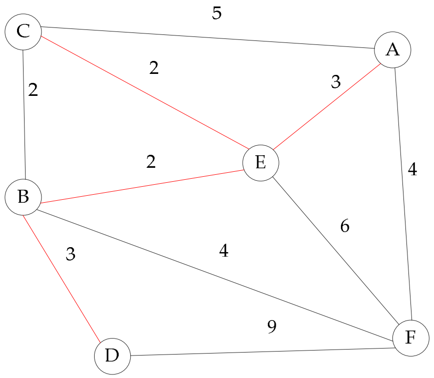
$|S| = 2 \neq 5;$

$$S = \{(A,E),(B,E),(C,E)\};\ R = \{(A,C),(A,F),(B,C),(B,D),(B,F),(D,F),(E,F)\}$$



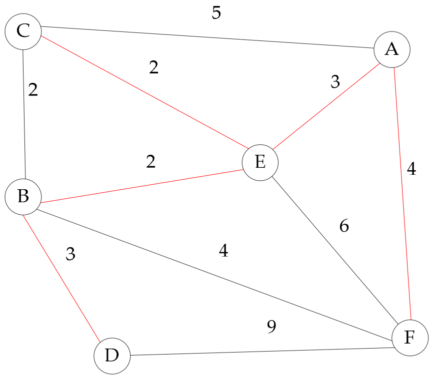
$|S| = 3 \neq 5;$

$S = \{(A,E),(B,D),(B,E),(C,E)\};\ R = \{(A,C),(A,F),(B,C),(B,F),(D,F),(E,F)\}$



|S| = 4 ≠ 5;

$S = \{(A,E),(B,D),(B,E),(C,E)\};\ R = \{(A,C),(A,F),(B,C),(B,F),(D,F),(E,F)\}$



|S| = 5; O ALGORITMO PÁRA.

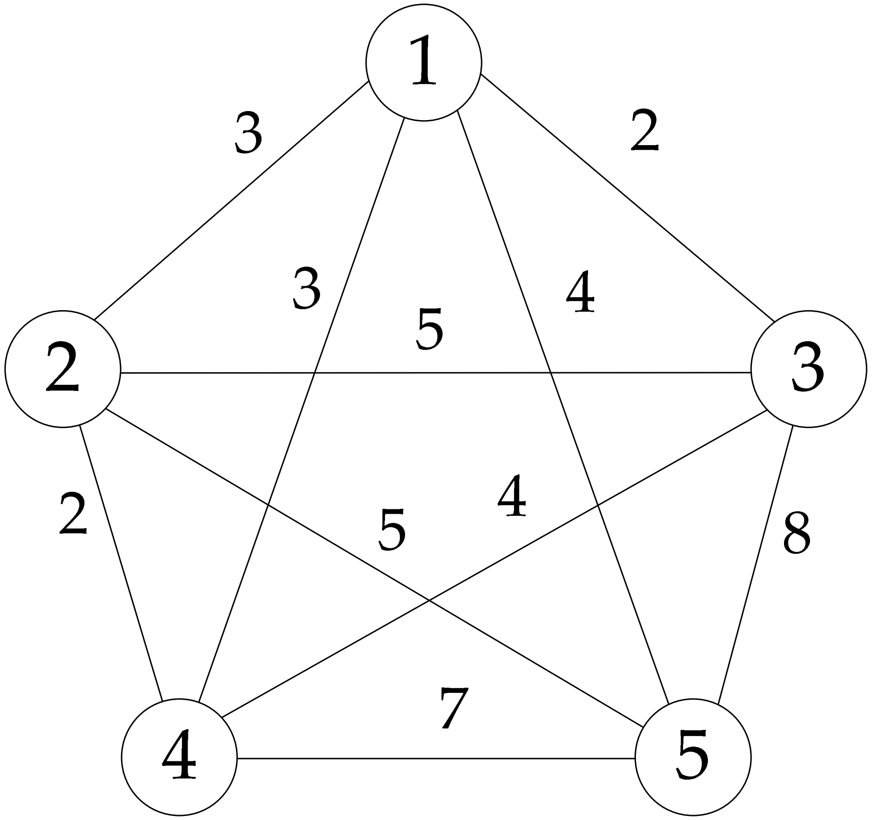
v. Heurística do Vizinho Mais Próximo (Problema do Caixeiro Viajante)

- O problema de caixeiro viajante consiste em determinar um circuito que passe por todos os vértices de um grafo, e cujo custo total seja o menor possível;
- O problema é mais complexo que os últimos problemas de optimização que abordámos;
- Não existe nenhum algoritmo eficiente que permita obter a solução óptima do problema qualquer que seja a instância considerada;
- Foram desenvolvidas heurísticas para este problema (uma heurística é um algoritmo de resolução que não garante que a solução óptima seja obtida no final);
- A heurística do vizinho mais próximo é uma das heurísticas mais simples para este problema; trata-se de uma heurística construtiva *miópica* porque, em cada iteração, procura a melhor alternativa sem ter em atenção as iterações seguintes.

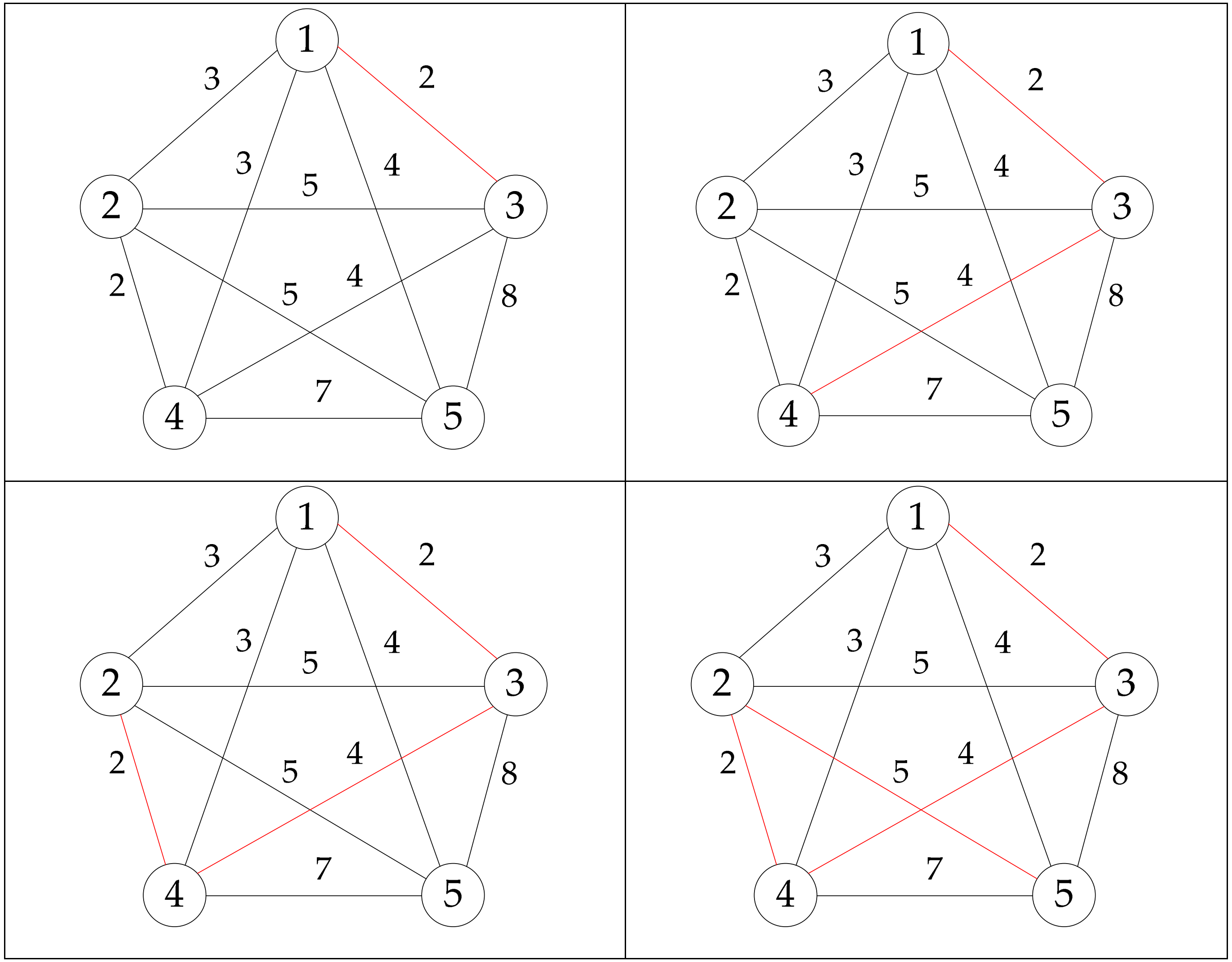
- Síntese da heurística do vizinho mais próximo para o problema de caixeiro viajante:

1. INICIALIZAÇÃO
    - Escolher um vértice i do grafo arbitrariamente;
    - Fazer $S = \{i\}$; $\overline{S} = V\backslash\{i\}$; $R = \emptyset$;
2. INSERÇÃO DE UM ARCO
    - Seja j o último vértice inserido em S;
    - Escolher o arco $(j,k) \in A$ de menor custo e tal que $k \in \overline{S}$;
    - Fazer $S = S \cup \{k\}$; $\overline{S} = \overline{S}\backslash\{k\}$; e $R=R\cup\{(j,k)\}$
3. SE $|\overline{S}| \neq 0$, voltar ao passo 2, SENÃO
    - Fazer $R=R\cup\{(j,i)\}$; Parar. (*Conjunto de arcos do circuito* = R)

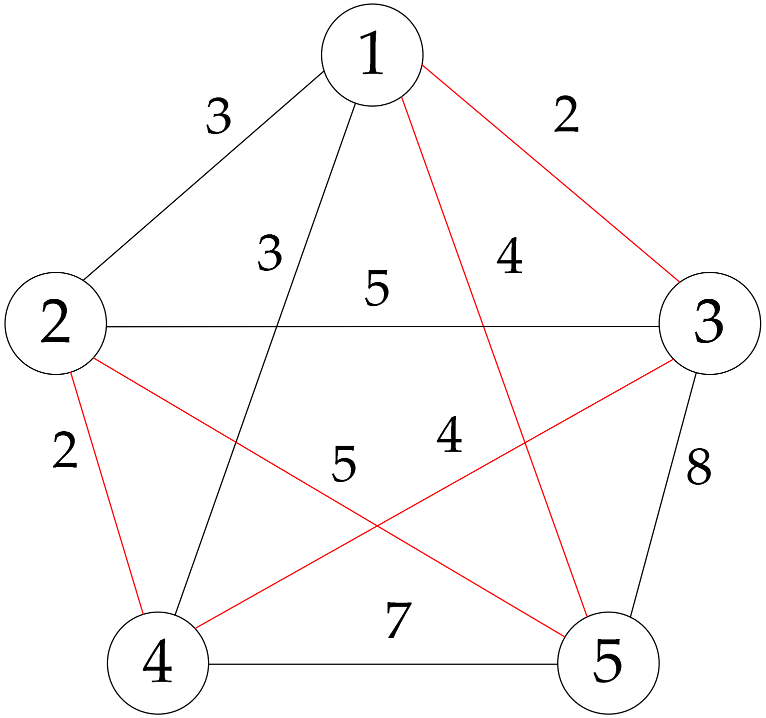
- Exemplo:



- Iterações: *Partindo do vértice 1*



- Solução:



➢ Custo da solução = 25.

- Em muitos casos, a heurística do vizinho mais próximo permite obter soluções de boa qualidade (não muito afastadas do óptimo);
- Contudo, não é difícil arranjar casos em que a heurística retorna soluções de muito má qualidade;
- É possível obter melhores soluções variando o vértice de origem, e escolhendo no final o circuito de menor custo (o preço a pagar é um aumento no número de iterações);
- A solução dada pela heurística é um limite superior para o valor do óptimo (no melhor caso, esse limite coincide com o próprio valor do óptimo);
- A heurística destaca-se por ser simples de implementar, e pela sua eficiência.