Lecture 7

Lecture 7

Previously in Opt 2 …

The Project Selection Problem

- Input:
  - m projects: $\mathscr{P} = \{1, 2, \ldots, m\}$
  - n tools: $\mathscr{T} = \{1, 2, \ldots, n\}$
  - For each project i,
    - T_i = the set of tools needed for completing project i
    - b_i = the benefit from completing project i
  - For each tool j,
    - c_j = the cost of purchasing tool j
- Objective:
  To select projects to complete and tools to buy, as to minimize

  net profit = total benefit from selected projects – total cost of selected tools

- Constraints:
  If project i is selected to be completed, then all tools in T_i must also be selected to be purchased

The Project Selection Problem

- Input:
  - m projects: $\mathscr{P}$ = {1, 2, …, m}
  - n tools: $\mathscr{T}$ = {1, 2, …, n}
  - For each project i,
    - T_i = the set of tools needed for completing project i
    - b_i = the benefit from completing project i
  - For each tool j,
    - c_j = the cost of purchasing tool j
- Objective:
  To select P’ (subset of $\mathscr{P}$) and T’ (subset of $\mathscr{T}$) to minimize
  net profit = $\sum_{i \in P'} b_i + \sum_{j \in T'} c_j$
- Constraints:
  If project i is in P’, then the set of tools T_i is a subset of T’

The Project Selection Problem

Example

$\mathscr{P} = \{1, 2, 3\}$

$\mathscr{T} = \{1, 2, 3, 4\}$

$T_1 = \{1, 2, 3\}$, $T_2 = \{2, 3\}$, $T_3 = \{3, 4\}$

$b_1 = 15$, $b_2 = 20$, $b_3 = 10$

$c_1 = 5$, $c_2 = 10$, $c_3 = 5$, $c_4 = 15$

The Project Selection Problem: Mincut Formulation

Step 1: Construct the Mincut input

A directed graph $G = (N, E)$, where

$N = \{s, t\} \cup \mathcal{P} \cup \mathcal{T}$

$E = \{(s, i) \mid$ for all project nodes i in $\mathcal{P}\} \cup$

$\{(i, j) \mid$ for j in T_i, for all i in $\mathcal{P}\} \cup$

$\{(j, t) \mid$ for all tool nodes j in $\mathcal{T}\} \cup$

Edge capacities: for (i, j) in E,

$u_{ij} = ?$

The Project Selection Problem

Example

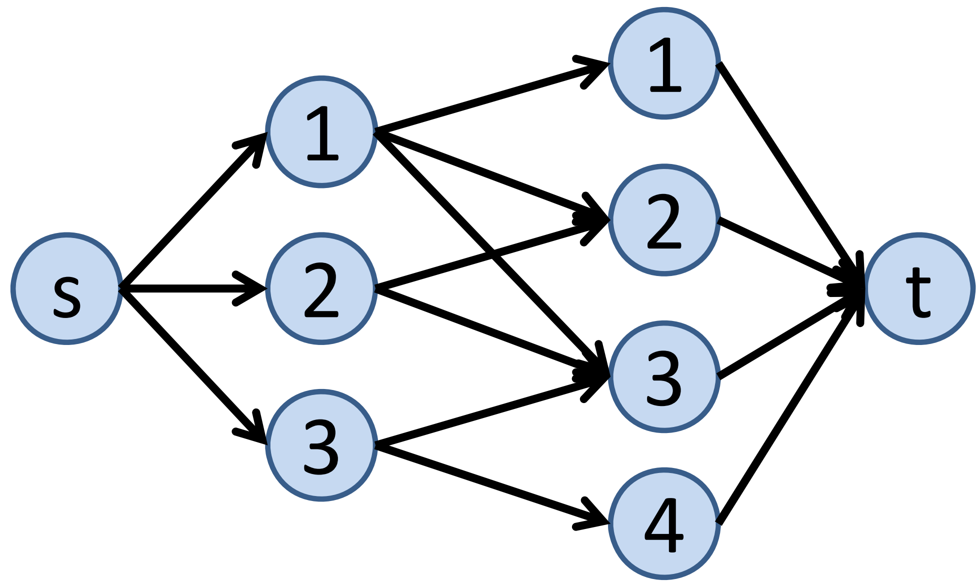
$\mathscr{P} = \{1, 2, 3\}$

$\mathscr{T} = \{1, 2, 3, 4\}$

$T_1 = \{1, 2, 3\}$, $T_2 = \{2, 3\}$,
$T_3 = \{3, 4\}$

$b_1 = 15$, $b_2 = 20$, $b_3 = 10$

$c_1 = 5$, $c_2 = 10$, $c_3 = 5$,
$c_4 = 15$



The Project Selection Problem: Mincut Formulation

Step 1: Construct the Mincut input

A directed graph G = (N, E), where

$N = \{s\} \cup P \cup T$

$E = \{(s, i) \mid \text{for all } i \text{ in } \mathscr{P}\}$

$\cup \ \{(i, j) \mid \text{for } j \text{ in } T_i, \text{ for all } i \text{ in } \mathscr{P}\}$

$\cup \ \{(j, t) \mid \text{for all } j \text{ in } \mathscr{T}\}$

Edge capacities:

$u_{si} = b_i$ for all projects I; $u_{jt} = c_j$ for all tools j;

$u_{ij} = +\infty$ for all i in $\mathscr{P}$ and j in T_i

The Project Selection Problem

Example

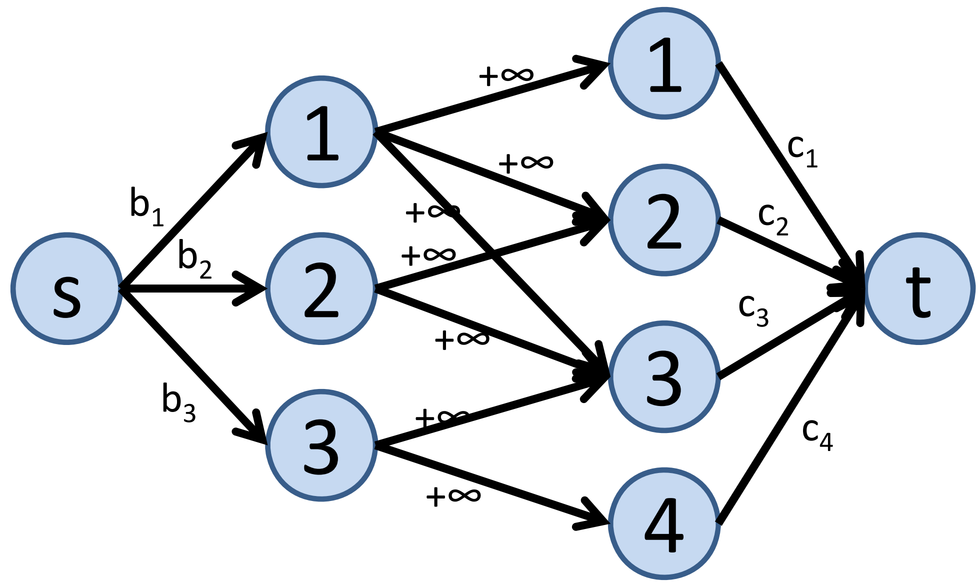
$\mathscr{P} = \{1, 2, 3\}$

$\mathscr{T} = \{1, 2, 3, 4\}$

$T_1 = \{1, 2, 3\}$, $T_2 = \{2, 3\}$,
$T_3 = \{3, 4\}$

$b_1 = 15$, $b_2 = 20$, $b_3 = 10$

$c_1 = 5$, $c_2 = 10$, $c_3 = 5$,
$c_4 = 15$



The Project Selection Problem: Mincut Formulation

Step 2: Correspondence of feasible solutions

Goal: to show the correspondence:

Feasible selections of projects and tools	$\longleftrightarrow$	Cuts of finite capacities

The Project Selection Problem

Example

$\mathscr{P} = \{1, 2, 3\}$

$\mathscr{T} = \{1, 2, 3, 4\}$

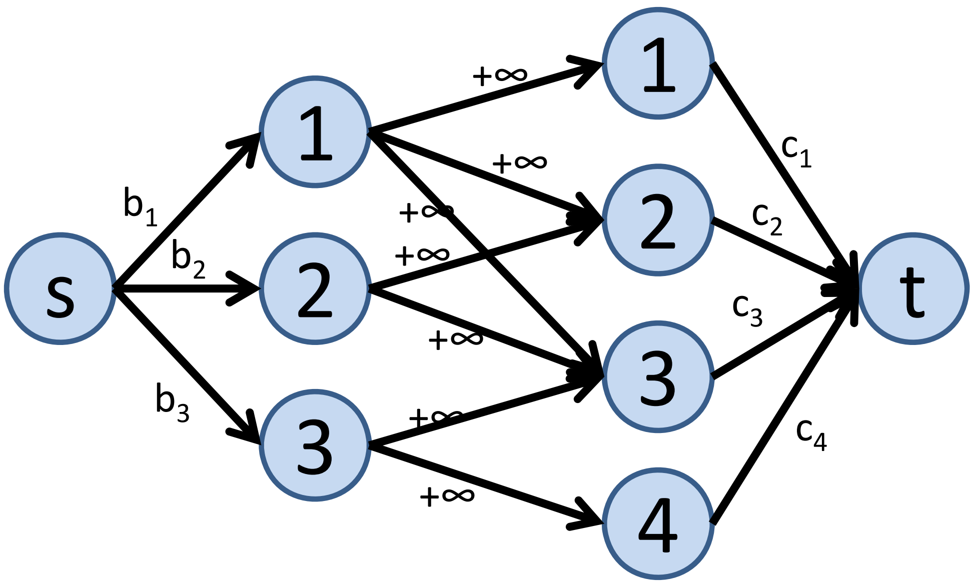
$T_1 = \{1, 2, 3\}$, $T_2 = \{2, 3\}$,
$T_3 = \{3, 4\}$

$b_1 = 15$, $b_2 = 20$, $b_3 = 10$

$c_1 = 5$, $c_2 = 10$, $c_3 = 5$,
$c_4 = 15$

Feasible selection:

P' = {1, 2}, T' = {1, 2, 3}



The Project Selection Problem

Example

$\mathscr{P} = \{1, 2, 3\}$

$\mathscr{T} = \{1, 2, 3, 4\}$

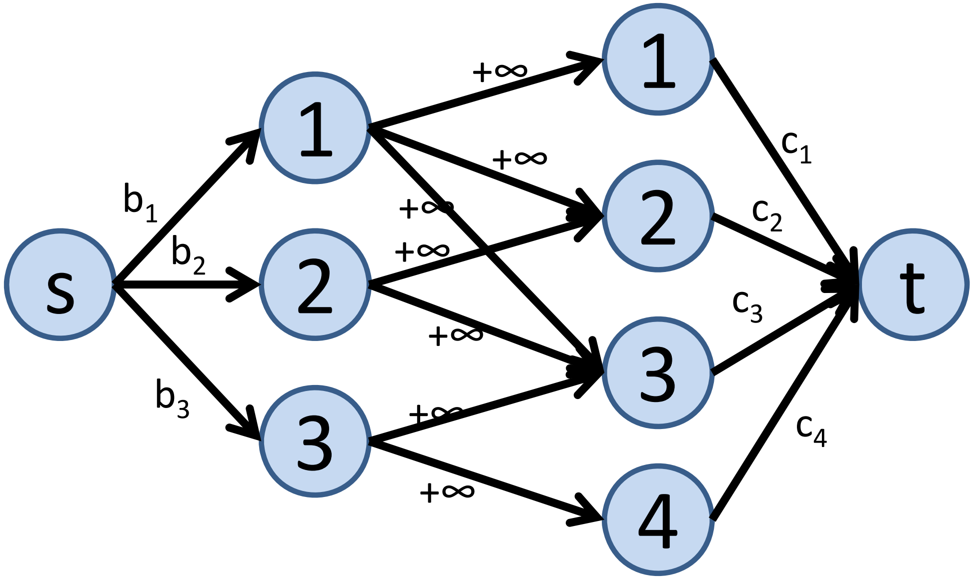
$T_1 = \{1, 2, 3\}$, $T_2 = \{2, 3\}$,
$T_3 = \{3, 4\}$

$b_1 = 15$, $b_2 = 20$, $b_3 = 10$

$c_1 = 5$, $c_2 = 10$, $c_3 = 5$,
$c_4 = 15$

Infeasible selection:

P' = ?, T' = ?



The Project Selection Problem: Mincut Formulation

Step 2: Correspondence of feasible solutions

First, show the correspondence:

Feasible selections of projects and tools → Cuts of finite capacities

Suppose that we have a feasible selections of projects and tools (P’, T’) …

The Project Selection Problem: Mincut Formulation

Step 2: Correspondence of feasible solutions

Suppose (P’, T’) is feasible → Construct the cut:

S = {s} U P’ U T’,

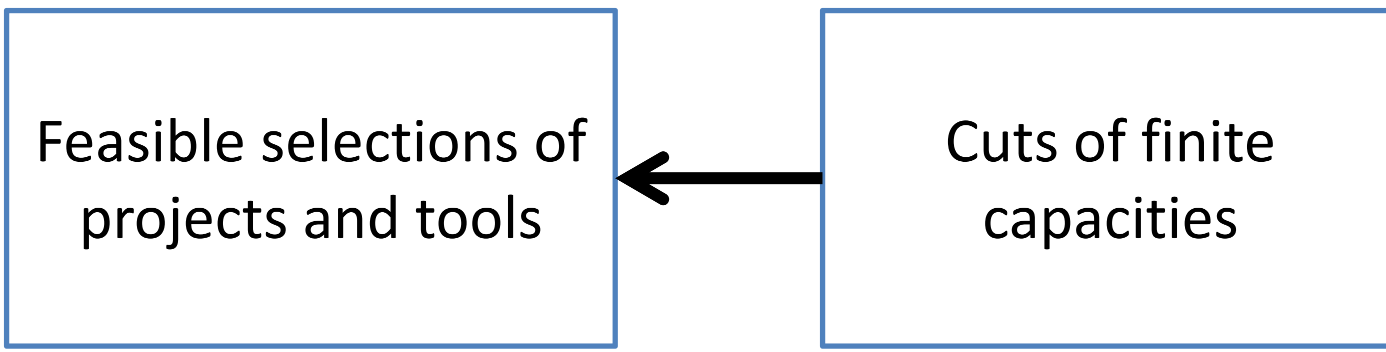
T = all other nodes

Then (S, T) has finite capacity

The Project Selection Problem: Mincut Formulation

Step 2: Correspondence of feasible solutions

Next, show the correspondence:



Suppose that we have a cut (S, T) of finite capacity…

The Project Selection Problem: Mincut Formulation

Step 2: Correspondence of feasible solutions

Construct (P', T'):
P' = projects that are in S
T' = tools that are in S
Then (P', T') is a feasible selection

←

Suppose
(S, T) is a cut of finite capacity

The Project Selection Problem: Mincut Formulation

Step 3: Correspondence of optimal solutions

Goal: to show the correspondence:

“Best” selection of projects and tools $\longleftrightarrow$ “Best” cut of finite capacity

The Project Selection Problem: Mincut Formulation

Step 3: Correspondence of optimal solutions

Goal: to show the correspondence:

Selection of projects and tools with maximum net profit	$\longleftrightarrow$	Cut of minimum finite capacity

The Project Selection Problem: Mincut Formulation

Step 3: Correspondence of optimal solutions

Goal: to show the correspondence:

Selection of projects and tools with maximum net profit $\longleftrightarrow$ Cut of minimum capacity

The Project Selection Problem: Mincut Formulation

Step 3: Correspondence of optimal solutions

So, maximizing the net profit of (P’,T’) is equivalent to minimizing the capacity of (S, T)

$$\max_{(P',T')\text{ feas}} \left(\sum_{i \in P'} b_i - \sum_{j \in T'} c_j \right) \longleftrightarrow \min_{(S,T),\text{feas cut}} \sum_{i \in S,\, j \in T} u_{ij}$$

where (S, T) is the cut that corresponds to (P’, T’):

S = {s} U P’ U T’ and T = all the other nodes

i>clicker

(after the break)

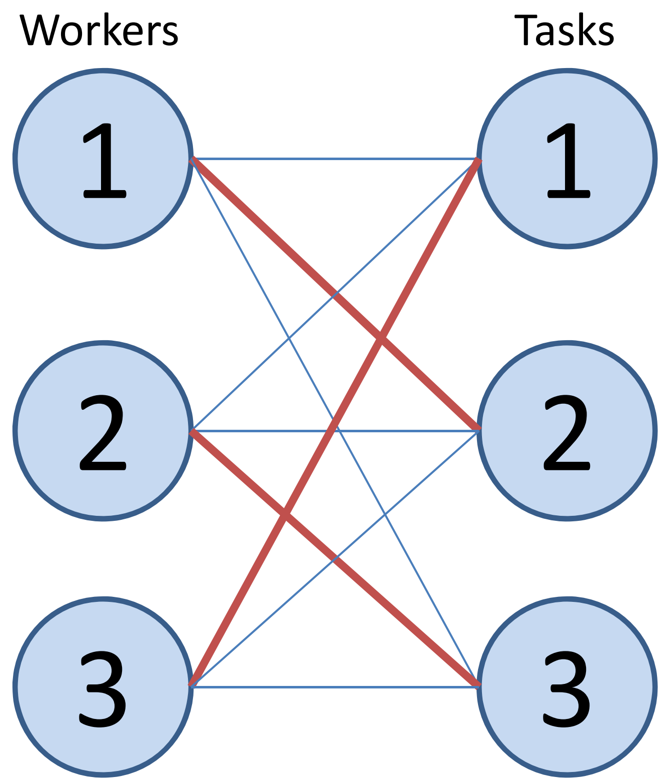
Q1: The Assignment Problem

n = 3

With cost table:

(Row i = worker i; Column j = task j)

1	0	3
0	4	0
0	2	0



A feasible assignment:
(1, 2), (2, 3), (3, 1).
Is this optimal?

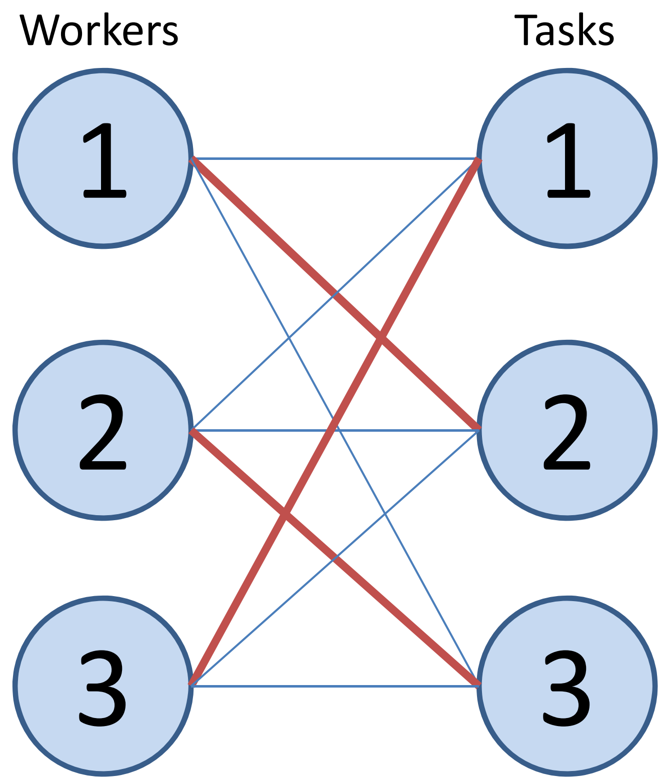
Q1: The Assignment Problem

n = 3

With cost table:

(Row i = worker i; Column j = task j)

1	0	3
0	4	0
0	2	0



A feasible assignment:
(1, 2), (2, 3), (3, 1).

Is this optimal?

A. Yes
B. No
C. Maybe
D. Some of the above
E. None of the above

The assignment problem

(revisited!)

The Assignment Problem

- Input:
  - A set of n workers: W = {1, 2, .., n}
  - A set of n tasks: T = {1, 2, …, n}
  - For each pair of worker i and task j:
    t_{ij} = time (or cost) for worker i to finish task j
- Objective:
  - To minimize the total time (or cost) for finishing all n tasks
- Constraint:
  - Each worker is assigned to do one task
  - Each task is assigned to one worker

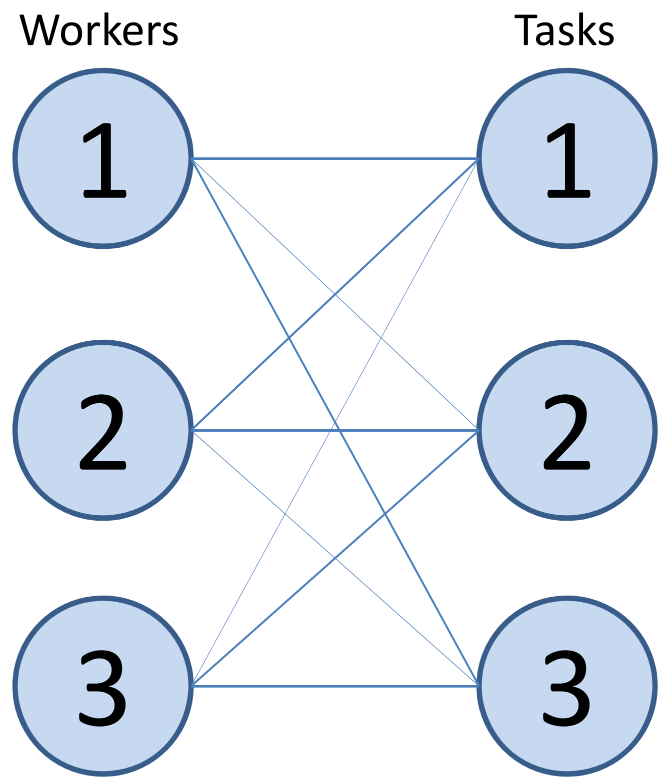
The Assignment Problem: Example 0

n = 3

With cost table:

(Row i = worker i; Column j = task j)

1	0	3
0	4	0
0	2	0



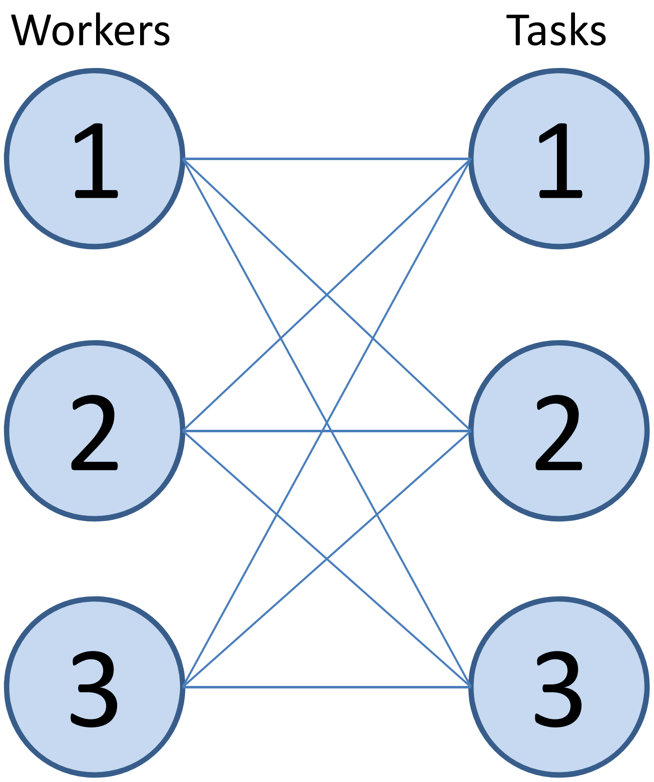
The Assignment Problem: Example 1

n = 3

With cost table:

(Row i = worker i; Column j = task j)

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



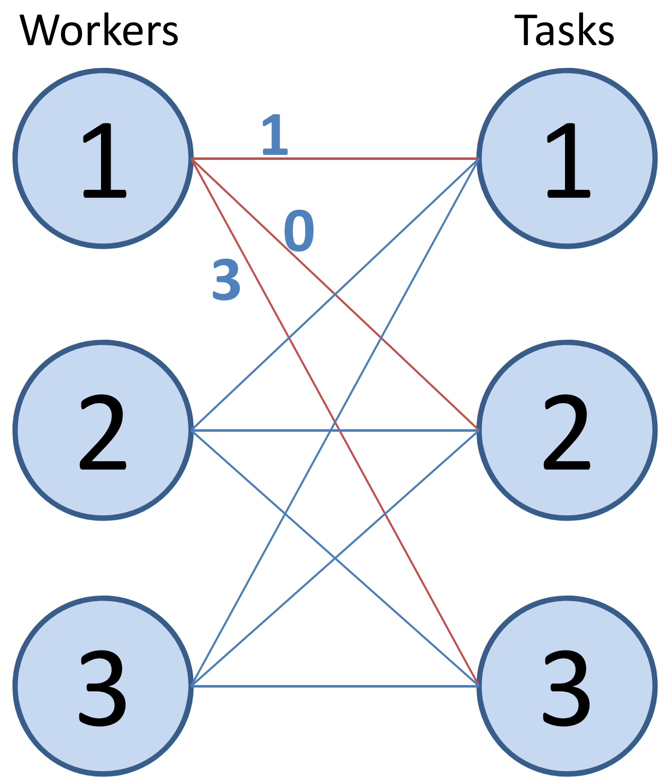
The Assignment Problem: Example 1

n = 3

With cost table:

(Row i = worker i; Column j = task j)

	j = 1	j = 2	j = 3
i = 1	1	0	3
	0	4	0
	0	2	0



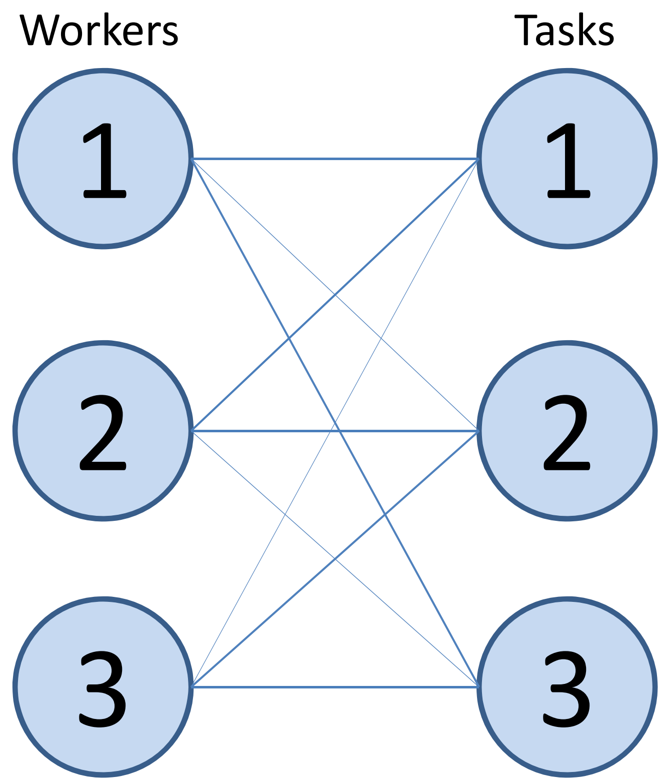
The Assignment Problem: Example 0

n = 3

With cost table:

(Row i = worker i; Column j = task j)

1	0	3
0	4	0
0	2	0



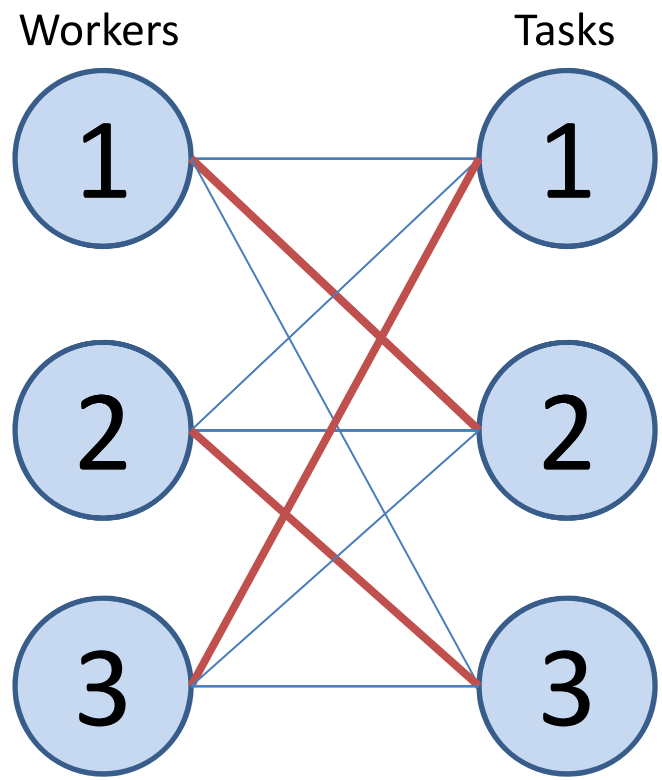
The Assignment Problem: Example 0

n = 3

With cost table:

(Row i = worker i; Column j = task j)

1	0	3
0	4	0
0	2	0



The assignment:
(1, 2), (2, 3), (3, 1)
Is this optimal?

The Assignment Problem: An observation

Observation

If $t_{ij} \geq 0$ for all pairs (i, j), and if we have an assignment of zero total cost,

then this assignment is optimal.

Moreover, each pairing in this assignment must have zero cost.

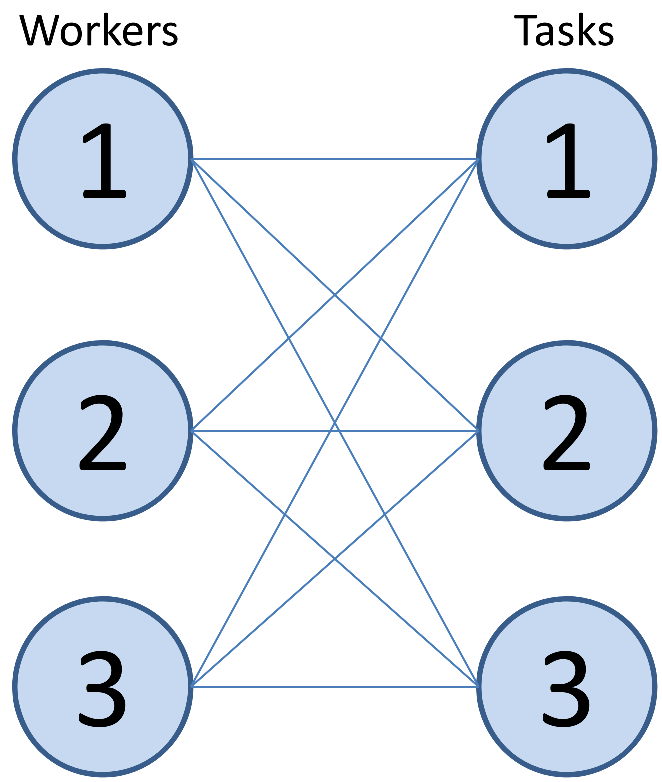
The Assignment Problem: Example 1

n = 3

With cost table:

(Row i = worker i; Column j = task j)

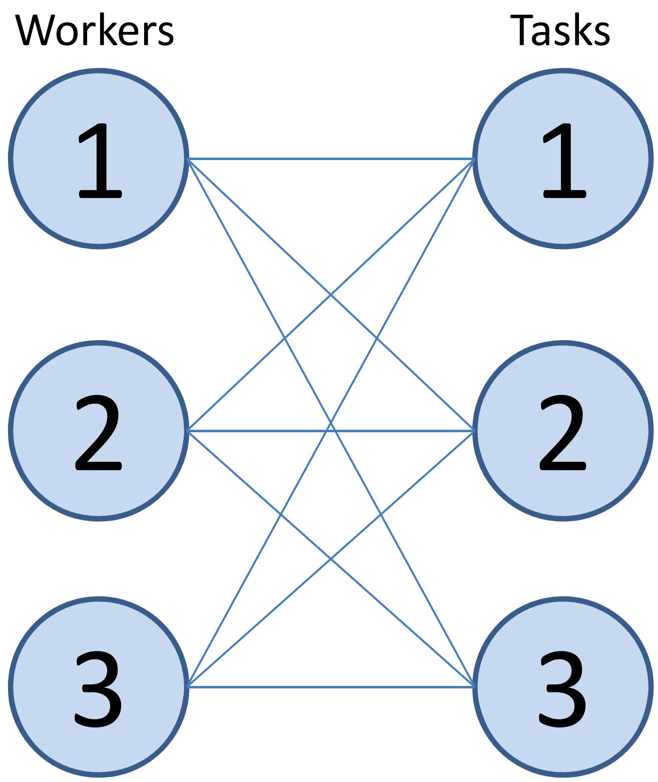
6	4	7
2	5	1
3	4	2



The Assignment Problem: Example 1

Subtract 4 from each entry in row 1

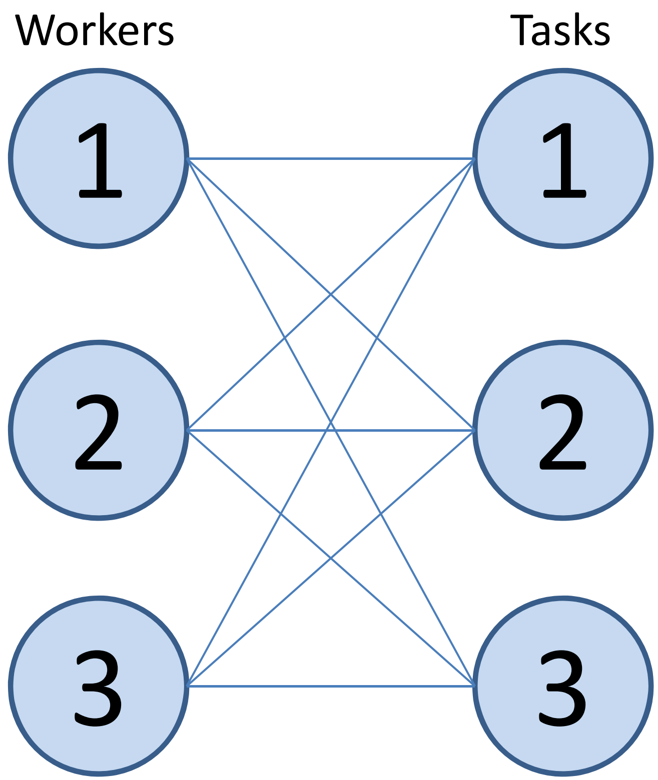
6	4	7
2	5	1
3	4	2



The Assignment Problem: Example 1

Subtract 4 from each entry in row 1

2	0	3
2	5	1
3	4	2



The Assignment Problem: Another observation

Claim

Suppose x denote an optimal assignment to an input with edge costs t_{ij}.

Suppose that we subtract δ from each entry of row i (or from each entry of column j).

(i.e., subtract δ from the cost of each edge adjacent to worker-node i, or from the cost of each edge adjacent to task-node j, respectively)

Then x is still optimal for the modified input.

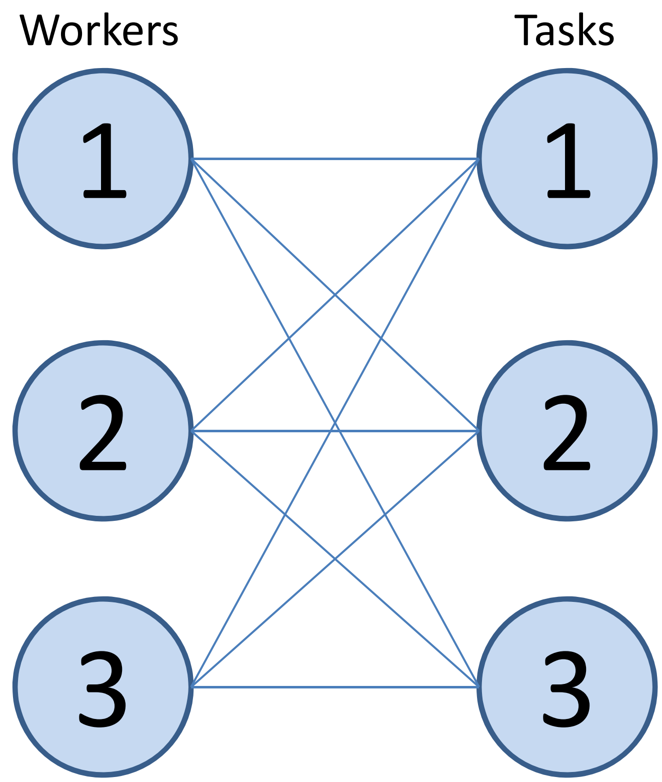
The Assignment Problem: Example 1

Subtract 4 from each entry in row 1

Subtract 1 from each entry in row 2

Subtract 2 from each entry in row 3

2	0	3
2	5	1
3	4	2



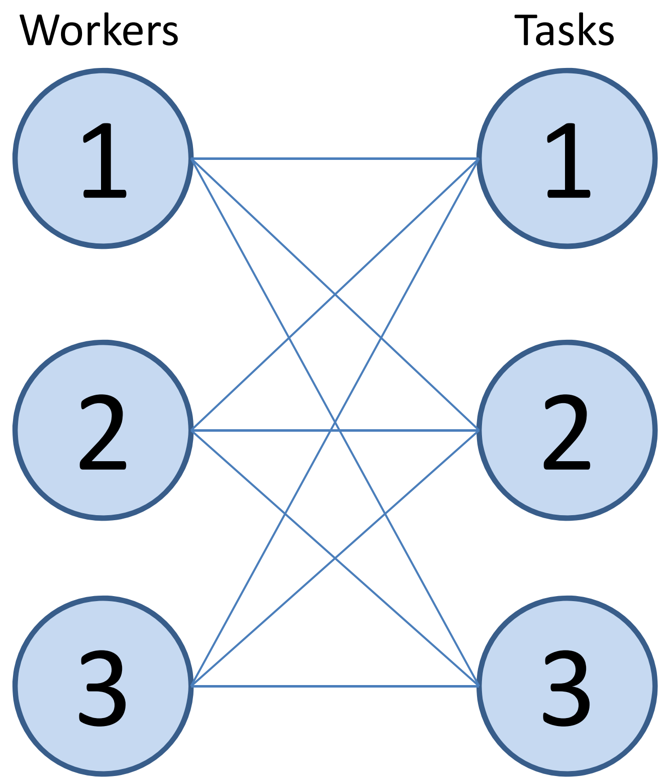
The Assignment Problem: Example 1

Subtract 4 from each entry in row 1

Subtract 1 from each entry in row 2

Subtract 2 from each entry in row 3

2	0	3
1	4	0
1	2	0



The Assignment Problem: Example 1

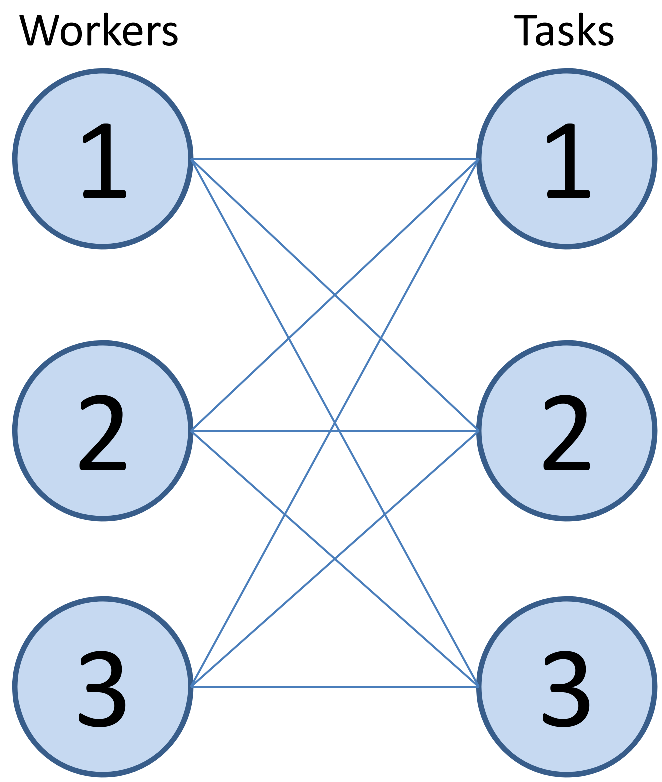
Subtract 4 from each entry in row 1

Subtract 1 from each entry in row 2

Subtract 2 from each entry in row 3

Subtract 1 from each entry in col 1

2	0	3
1	4	0
1	2	0



The Assignment Problem: Example 1

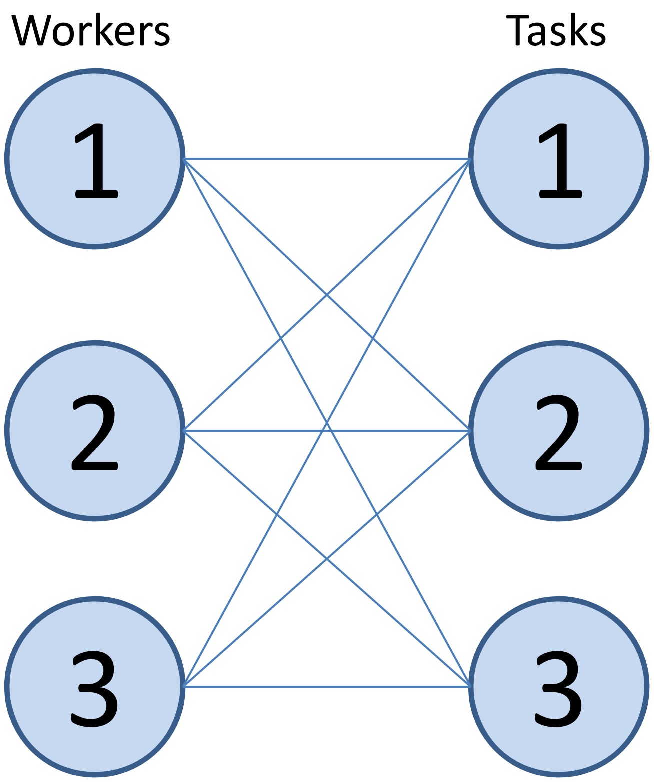
Subtract 4 from each entry in row 1

Subtract 1 from each entry in row 2

Subtract 2 from each entry in row 3

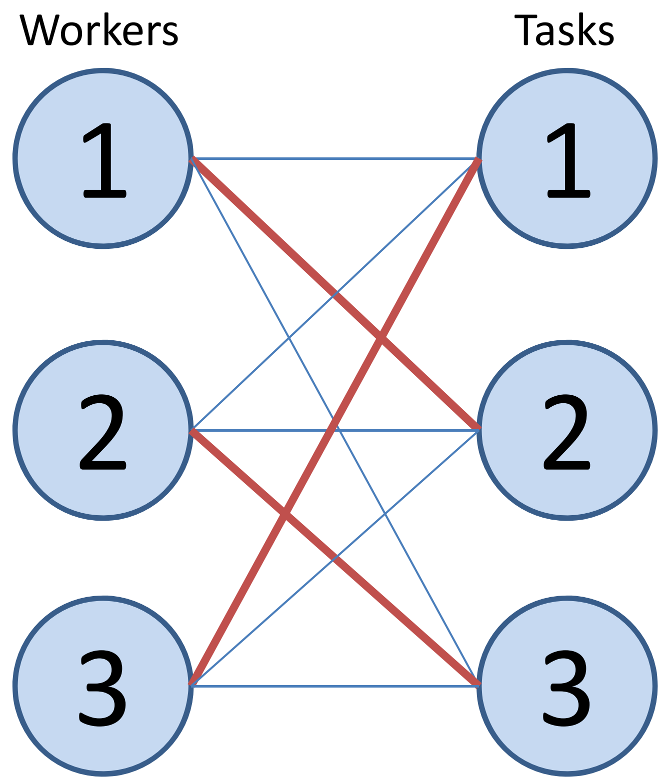
Subtract 1 from each entry in col 1

1	0	3
0	4	0
0	2	0



The Assignment Problem: Example 1

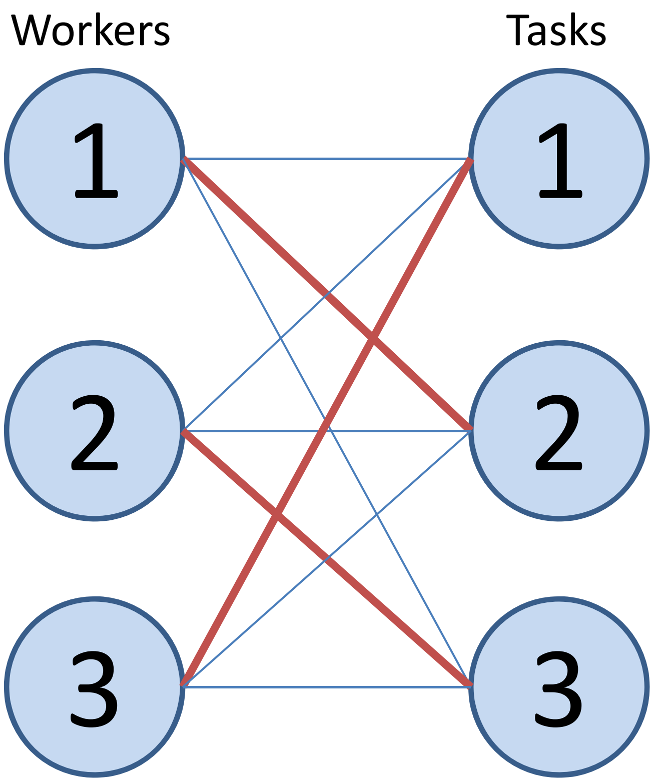
1	0	3
0	4	0
0	2	0



The assignment:
(1, 2), (2, 3), (3, 1)
is an ***all-zero-assignment***.

The Assignment Problem: Example 1

6	4	7
2	5	1
3	4	2



The assignment:

(1, 2), (2, 3), (3, 1)

is also optimal for the original problem, with total cost = 4 + 1 + 3 = 8

The Assignment problem: A first algorithm

0. Set up a cost table. Assume all costs are ≥ 0.
1. For each row $i = 1, \ldots, n$
   let α_i = the smallest entry in row i
   update: $t_{ij} \leftarrow t_{ij} - \alpha_i$ for all entries (i,j) in row i
2. For each col $j = 1, \ldots, n$
   let β_j = the smallest entry in column j
   update: $t_{ij} \leftarrow t_{ij} - \beta_j$ for all entries (i,j) in col j
3. Check if there is an all-zero assignment
   If there is one, this assignment is optimal

The Assignment problem: A first algorithm

0. Set up a cost table. Assume all costs are ≥ 0.
1. For each row $i = 1, \ldots, n$
   let α_i = the smallest entry in row i
   update: $t_{ij} \leftarrow t_{ij} - \alpha_i$ for all entries (i,j) in row i
2. For each col $j = 1, \ldots, n$
   let β_j = the smallest entry in column j
   update: $t_{ij} \leftarrow t_{ij} - \beta_j$ for all entries (i,j) in col j
3. Check if there is an all-zero assignment
   If there is one, this assignment is optimal
   If there isn't one: …?

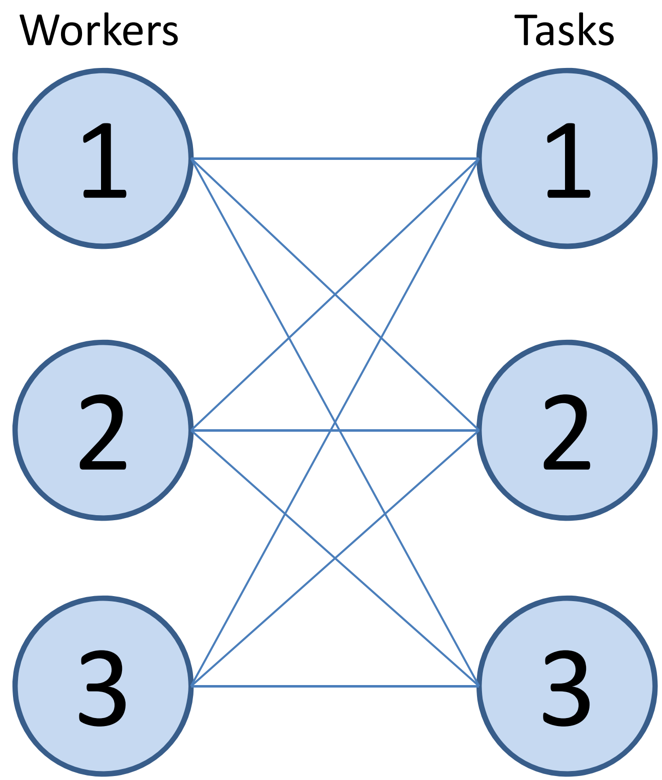
The Assignment Problem: Example 2

n = 3

With cost table:

(Row i = worker i; Column j = task j)

8	10	7
10	11	8
5	6	7



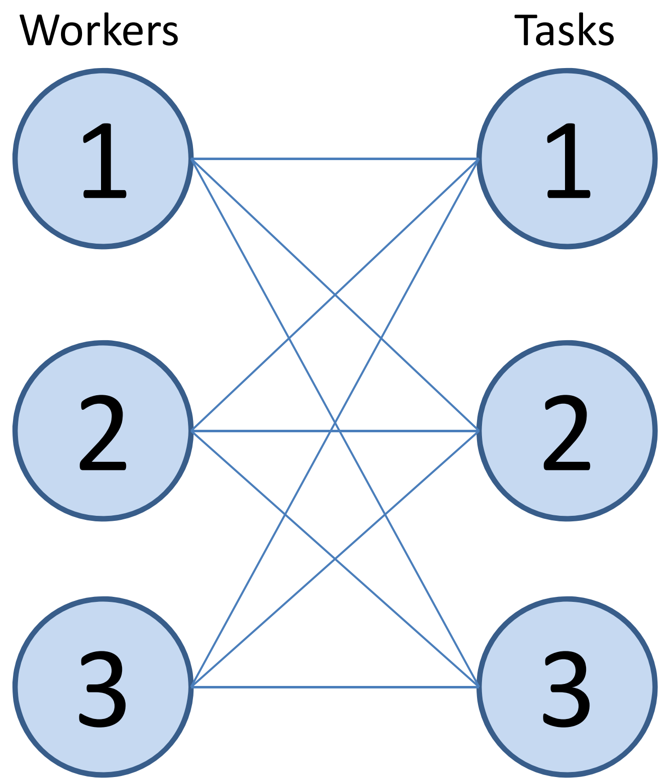
The Assignment Problem: Example 2

Subtract 7 from each entry in row 1

Subtract 8 from each entry in row 2

Subtract 5 from each entry in row3

8	10	7
10	11	8
5	6	7



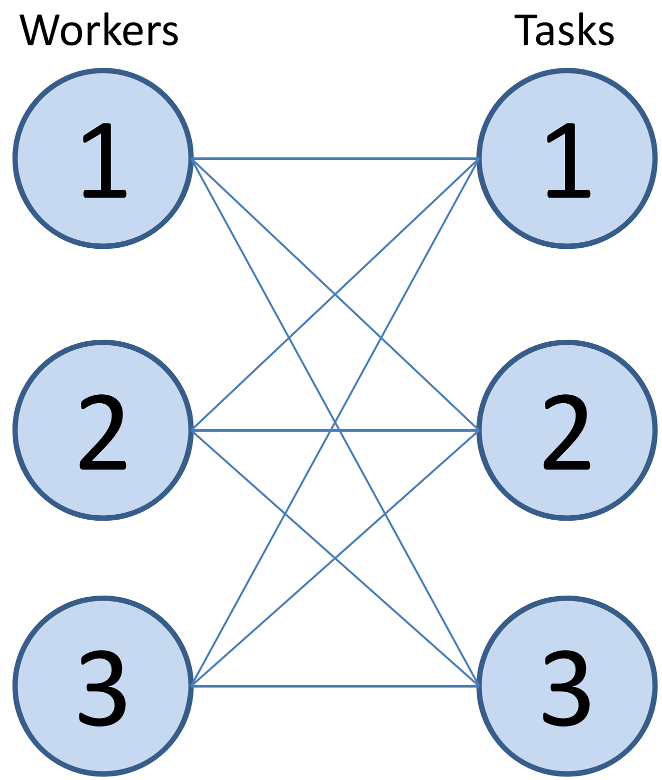
The Assignment Problem: Example 2

Subtract 7 from each entry in row 1

Subtract 8 from each entry in row 2

Subtract 5 from each entry in row3

1	3	0
2	3	0
0	1	2



The Assignment Problem: Example 2

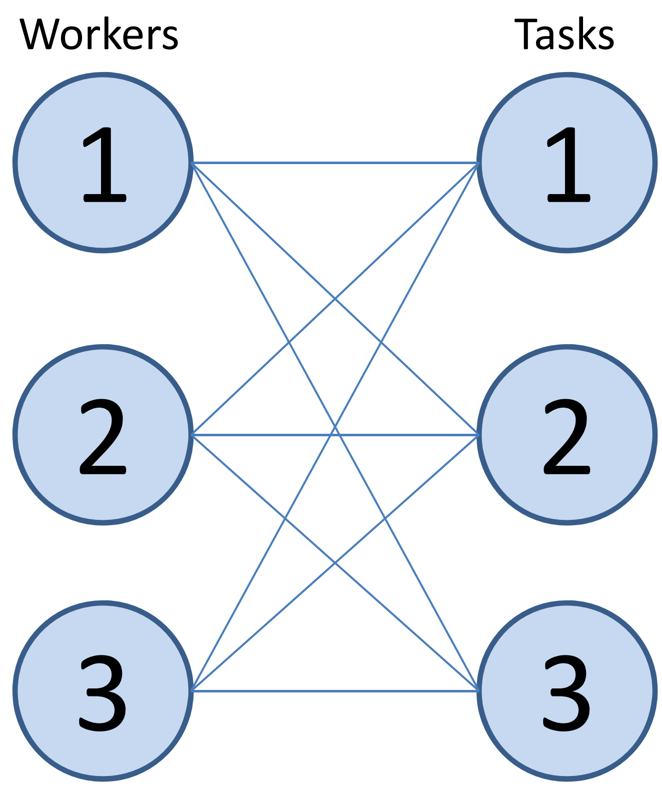
Subtract 7 from each entry in row 1

Subtract 8 from each entry in row 2

Subtract 5 from each entry in row3

Subtract 1 from each entry in col 2

1	3	0
2	3	0
0	1	2



The Assignment Problem: Example 2

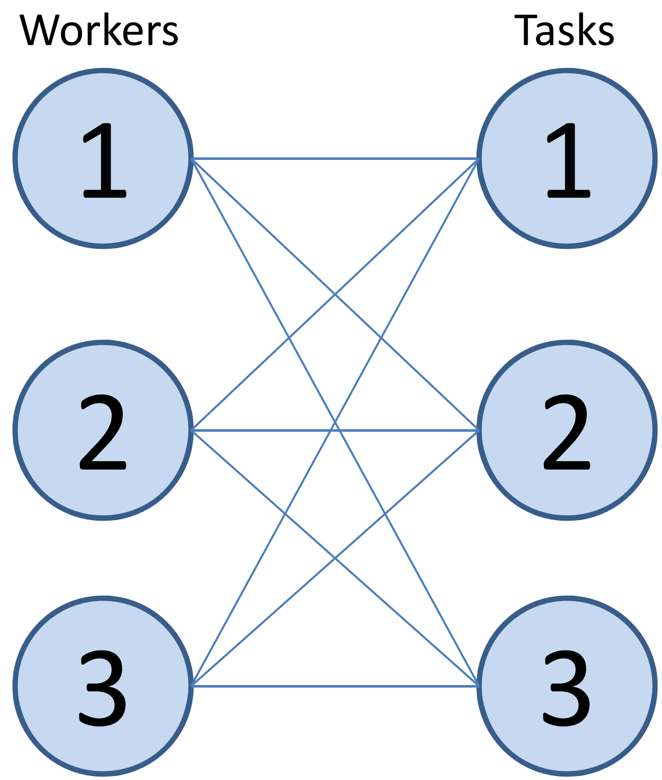
Subtract 7 from each entry in row 1

Subtract 8 from each entry in row 2

Subtract 5 from each entry in row3

Subtract 1 from each entry in col 2

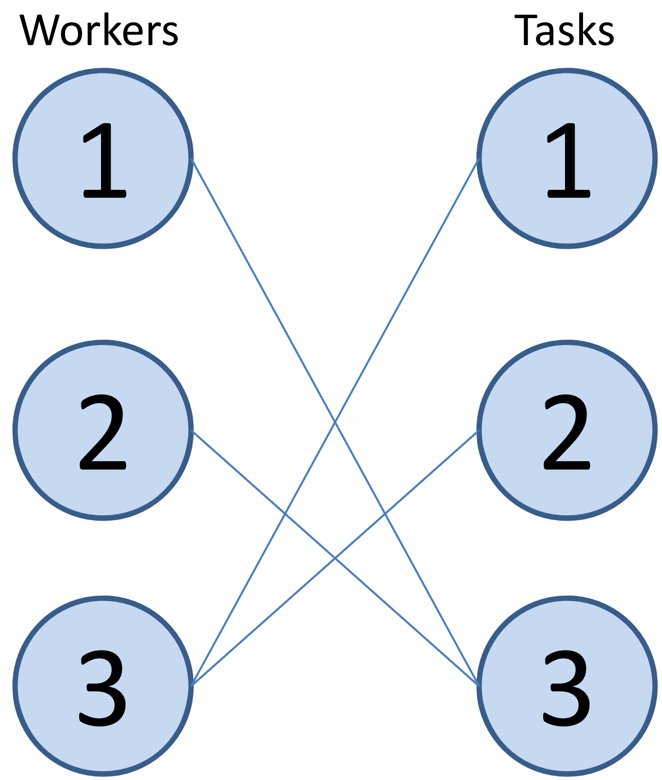
1	2	0
2	2	0
0	0	2



There is no all-zero-assignment.

The Assignment Problem: Example 2

1	2	0
2	2	0
0	0	2



There is no all-zero-assignment: No matching of size n in the graph containing just zero-cost edges