Generalized Assignment Problems

n jobs and m machines

s_{ij} : size of job/item j on machine/bin i

Assignment: A function f: jobs $\rightarrow$ machines. $f(j)$ is the machine to which job j is assigned

Minimum makespan: find f to minimize maximum load: $\max_i \sum_{j:\ f(j) = i} s_{ij}$

Note: problem also called *unrelated machine scheduling*

Generalized Assignment Problems

n jobs and m machines

s_{ij} : size of job/item j on machine/bin i

b_i : capacity of machine/bin i

c_{ij} : cost of assigning item i to bin j

Assignment f is *feasible* if forall i, $\sum_{j:\ f(j) = i} s_{ij} \leq b_i$

Minimum cost assignment: find a feasible f to minimize

$\text{cost}(f) = \sum_j c_{f(j),j}$

Generalized Assignment Problems

n jobs and m machines

s_{ij} : size of job/item j on machine/bin i

b_i : capacity of machine/bin i

p_{ij} : profit for assigning item i to bin j

Assignment f is *feasible* if forall i, $\sum_{j:\ f(j) = i} s_{ij} \leq b_i$

Maximum profit assignment: find a feasible f to maximize $\text{profit}(f) = \sum_j p_{f(j),j}$

Note: can assume dummy machine so that all jobs are assigned

Assignment polytope

x_{ij} : variable to indicate if j is assiged to i

Assignment polytope:

$\sum_i x_{ij} \leq 1$ for each j

$\sum_j s_{ij} x_{ij} \leq b_i$ for each i

Easy case: when $s_{ij} \in \{0, 1, \infty\}$, assignment polytope has integral vertices (adjacency matrix of a bipartite graph which is TUM). All variants can be solved exactly

Solvability of GAP

- For general values of s_{ij}, it is NP-Complete to check if there exists a feasible assignment satsifying capacity constraints
- implies cost minimization is inapproximable
- Also NP-hard to minimize makespan to better than a factor of 3/2
- APX-hard to maximize profit

Above results hold for $s_{ij} \in \{0, 1, 2, \infty\}$

Approximation for GAP

We will show

- 2-approx for minimizing makespan
- a bi-criteria approx for cost minimization, that is we find an assignment f such that $cost(f) \leq cost(f^*)$ and forall i, $\sum_{j:\ f(j) = i} s_{ij} \leq 2\ b_i$
- 1/2-approx for maximizing profit

Minimizing makespan/unrelated machine scheduling

Natural relaxation:

min z

$\sum_j s_{ij} x_{ij} \leq z$ for each machine i

$\sum_i x_{ij} = 1$ for each job j

$x_{ij} \geq 0$

Has unbounded integrality gap!

Minimizing makespan

Integrality gap example

1 job with $s_{i1} = 1$ for each i

Feasible solution to LP:

$x_{i1} = 1/m$

$z = 1/m$

OPT = 1

so gap = m

Improving gap

Guess value of OPT, say T

set $x_{ij} = 0$ if $s_{ij} > T$

LP relaxation with parameter T

LP(T):

$\sum_j s_{ij} x_{ij} \leq T$ for each machine i

$\sum_i x_{ij} = 1$ for each job j

$x_{ij} = 0$ if $s_{ij} > T$

$x_{ij} \geq 0$ if $s_{ij} \leq T$

Improving gap

$T^* = \inf_T$ (LP(T) is feasible)

Claim: $OPT \geq T^*$

T^* can be found by binary search

Theorem: $OPT \leq 2T^*$

Rounding of LP(T)

$\sum_j s_{ij} x_{ij} \leq T$ for each machine i

$\sum_i x_{ij} = 1$ for each job j

$x_{ij} = 0$ if $s_{ij} > T$

$x_{ij} \geq 0$ if $s_{ij} \leq T$

$A_i = \{ j \mid s_{ij} \leq T \}$: allowed items for i

Let x^* be a *basic feasible solution* (extreme point/vertex) to LP(T)

Properties of basic solutions

a basic solution is defined by k constraints where k is total number of variable (here $k = nm$)

# of non-trivial constraints in system is $n+m$ (one for each job and one for each machine)

therefore in x^*, the number of strictly positive variables ($x^*_{ij} > 0$) is at most $n+m$

$F = \{ j \mid x^*_{ij} \neq 1$ for any $i \}$

$j \in F$ implies j is assigned fractionally to more than one machine

Claim: $|F| \leq m$

Properties of basic solutions

of non-trivial constraints in system is $n+m$ (one for each job and one for each machine)

therefore in x^*, the number of strictly positive variables ($x^*_{ij} > 0$) is at most $n+m$

Claim: $|F| \leq m$

Counting: if $j \notin F$ then exactly one i s.t $x^*_{ij} > 0$
if $j \in F$ then at least two i s.t $x^*\{ij\} > 0$

therefore $(n - |F|) + 2|F| \leq n + m$

implies $|F| \leq m$

Properties of basic solutions

Can say something stronger

Create a bipartite graph $G = (M, F, E)$ where
F is fractionally assigned jobs, M is machines and $(i,j) \in E$

iff $x^*_{ij} > 0$

Let $G_1, G_2, \ldots, G_k$ be *connected components* of G

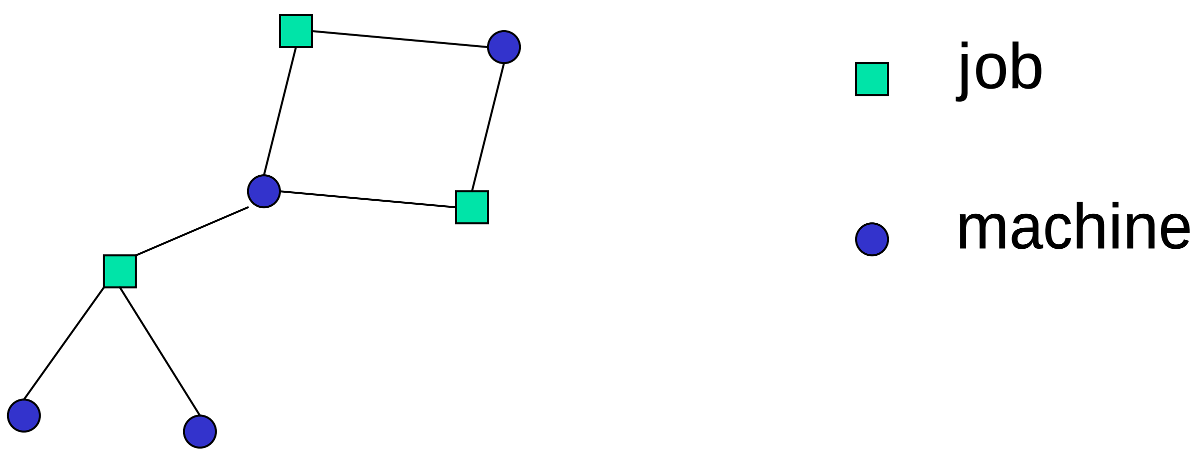
with $G_l = (M_l, F_l, E_l)$
Claim: $|F_l| \leq |M_l|$

Exercise using same argument as previous claim

Rounding x^*

hence graph G is a *pseudo-forest:* collection of *pseudo-trees*

pseudo-tree is a tree + one edge
Note: each job in G has deg ≥ 2



Rounding x^*

Process G as follows:

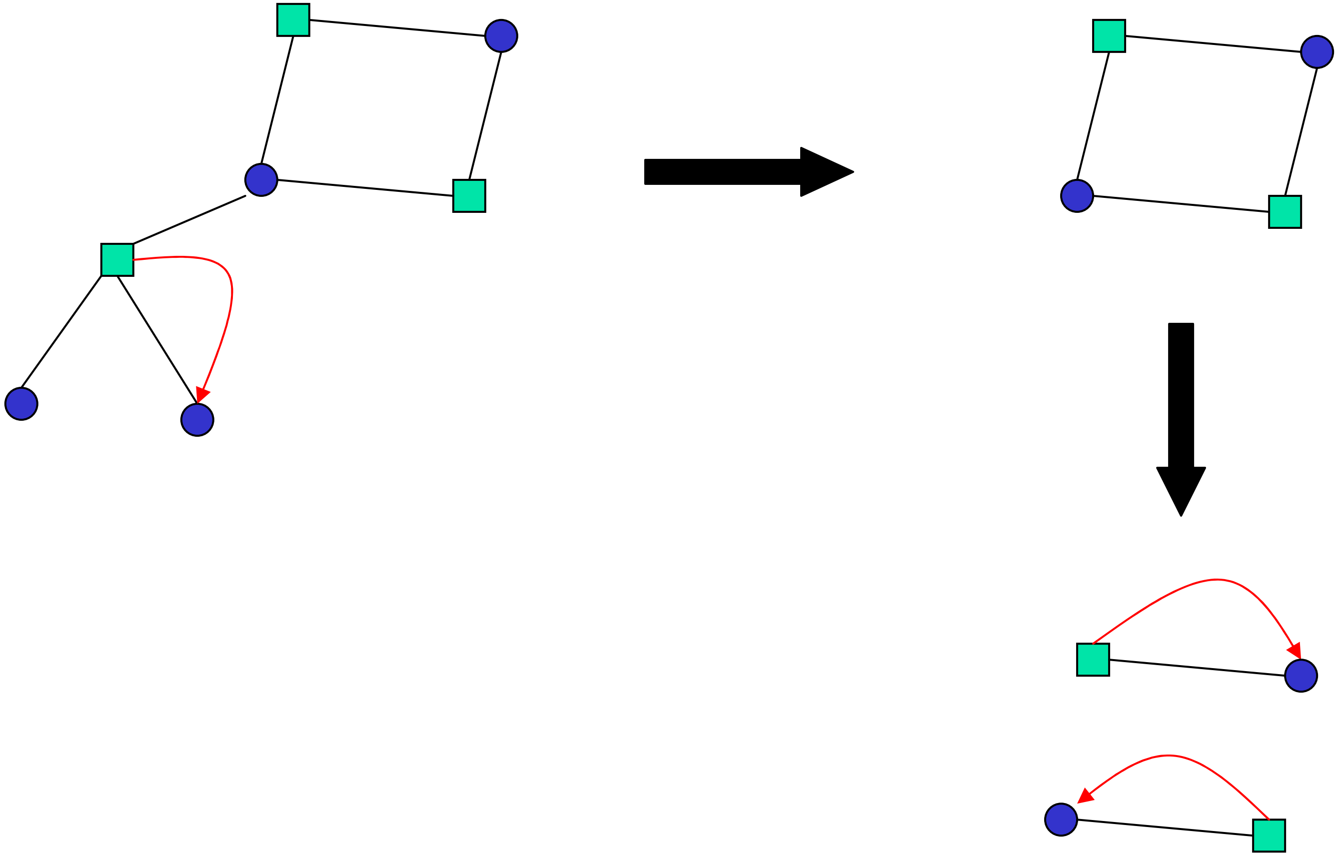
if there is a machine i s.t deg(i) = 1 in G then assign the unique job j with $x^*_{ij} > 0$ to i and remove i, j from G

if no machine i with deg(i) = 1 then G is a collection of cycles

find an aribitrary maximum matching in each cycle and assign jobs according to matching

since cycle is even all jobs in cycle assigned

Example



Analysis

Easy to see that each job in F assigned

Further a machine i is assigned at most one job j from F s.t $x^*_{ij} > 0$ (which implies $s_{ij} \leq T$)

Final load on machine i?

$\leq T + \sum_{j:\, x^*_{ij} = 1} s_{ij}$

$\leq T + \sum_j s_{ij} x^*_{ij}$

$\leq T + T$

$\leq 2\, T$

Analysis

Analysis shows that if x^* is a feasible soln to

$$\sum_i x_{ij} = 1$$

$$\sum_j s_{ij} x_{ij} \leq b_i$$

$$x_{ij} = 0 \text{ if } s_{ij} > b_i$$

$$x_{ij} \geq 0 \text{ if } s_{ij} \leq b_i$$

Then there is a 0,1 soln x' s.t

$$\sum_i x_{ij} = 1$$

$$\sum_j s_{ij} x'_{ij} \leq 2b_i$$

and $x'_{ij} > 0$ implies $x^*_{ij} > 0$

Cost minimization

$\min \sum_{ij} c_{ij} x_{ij}$

s.t

$\sum_i x_{ij} = 1$

$\sum_j s_{ij} x_{ij} \leq b_i$

$x_{ij} = 0$ if $s_{ij} > b_i$

$x_{ij} \geq 0$ if $s_{ij} \leq b_i$

Given a feasible soln x^* to above LP we can round x^* to a 0,1 soln x' s.t

Cost minimization

Given a feasible soln x^* to above LP we can round x^* to a 0,1 soln x' s.t

$\sum_j s_{ij} x'_{ij} \leq 2 b_i$

and

$\sum_j c_{ij} x'_{ij} \leq \sum_j c_{ij} x^*_{ij}$

and

$x'_{ij} > 0$ implies $x^*_{ij} > 0$

Rounding

Rounding is more involved than for makespan minimization. Need to preserve cost. Easy to check that previous rounding might violate cost by large amount.

We set $x'_{ij} = x^*_{ij}$ if $x^*_{ij} \in \{0, 1\}$

As before F is set of fractionally assigned items, that is $F = \{ j \mid x^*_{ij} \neq 1 \text{ for any } i \}$

We can focus on F

Rounding

Reduce problem to the case where $s_{ij} \in \{0,1,\infty\}$, the simple assignment problem and use the integrality properties of the assignment polytope

For each bin i, let $n_i = \lceil \sum_j x^*_{ij} \rceil$, the ceiling of number of items assigned to i

We create a new instance in which we replace machine i by n_i new bins

Rounding

New instance. Same items as before, but $\sum_i n_i$ bins. An item j that could have been assigned to an original bin i can be assigned to any copy i' of bin i . The cost of assigning to any copy of i is the same as assigning to i

We no longer have sizes - only have assignment constraints

Construct a feasible solution y to the new assignment problem carefully as follows

Setting up y

Consider bin i with $k=n_i$ copies $i_1, i_2, \ldots, i_k$

Let $A_i = \{j \mid s_{ij} \leq b_i\}$

Note that $x^*_{ij} > 0$ implies $j \in A_i$

Assume wlog that items are numbered s.t
$s_{i1} \geq s_{i2} \geq \ldots s_{ih}$ where $h = |A_i|$

By defn $\lceil \sum_{j=1}^{h} x^*_{ij} \rceil = n_i$

Setting up y

Consider bin i with $k=n_i$ copies $i_1, i_2, \ldots, i_k$
Let $A_i = \{j \mid s_{ij} \leq b_i\}$

Assume wlog that items are numbered s.t
$s_{i1} \geq s_{i2} \geq \ldots s_{ih}$ where $h = |A_i|$

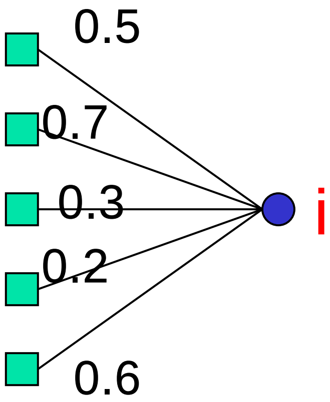
We explain how to set y using an example.

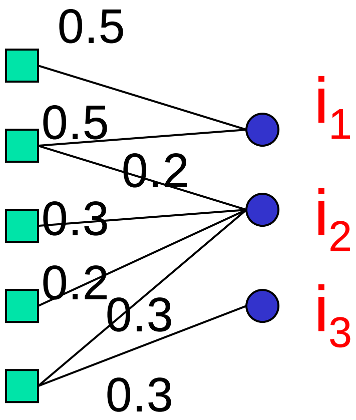
$h = 5$ with $x^*_{ij} = 0.5, 0.7, 0.3, 0.2, 0.6$, $j = 1$ to 5

Setting up y

We explain how to set y using an example.

$h = 5$ with $x^*_{ij} = 0.5, 0.7, 0.3, 0.2, 0.6$, $j = 1$ to 5





Setting up y

We ensure that y assigns to each copy of i a fractional value of at most 1 item

More formally the assignment is computed as follows (easier to understand informally from example)

For $j = 1$ to h

let $z_j = \sum_{l < j} x^*_{ij}$ and let $a = \lfloor z_j \rfloor$

Set $y_{i_{a+1}\, j} = \min (x^*_{ij}, a+1 - z_j)$

if $x^*_{ij} > y_{i_{a+1}\, j}$ then set $y_{i_{a+2}\, j} = x^*_{ij} - y_{i_{a+1}\, j}$

Properties of y

- By construction each new bin is fractionally assigned at most 1 item in y
- For item j, x^*_{ij} is distributed to copies of bin i (at most two copies)
- Important property: each job fractionally assigned to i_l in y has size (in i) no larger than any job fractionally assigned to i_{l-1}

- The cost of y is same as cost of x^*
- Therefore y defines a feasible solution to the simple assignment problem

Rounding y to y'

y is a feasible solution to an assignment problem

By integrality of the assignment polytope there *exists* a integral solution y' s.t

$\text{cost}(y') \leq \text{cost}(y)$

$y'_{i_a j} = 1$ implies $y_{i_a j} > 0$

at most one item is assigned to each bin

Such a y' can also be obtained in polynomial time from y using simple operations.

Using y' to obtain x'

If an item j is assigned to a copy of bin i in y' we set $x'_{ij} = 1$

We claim that $cost(x') \leq cost(x^*)$

and
$\sum_j s_{ij} x'_{ij} \leq 2 b_i$ for each i

The first part is easy since $cost(x') = cost(y') \leq cost(y) = cost(x^*)$

Load property of x'

Consider any bin i with copies $i_1, i_2, \ldots, i_k$

Let j_l be the item assigned to copy i_l by y'

(assume for simplicity that i_k is also assigned an item thought it is not always the case)

for $l = 2$ to k we claim that $s_{ij_l} \leq \sum_j s_{ij}\, y_{i_{l-1},j}$

This follows from the ordering of jobs according to sizes in construction y from x^*

Therefore $\sum_{l=2}^{k} s_{ij_l} \leq \sum_{l=2}^{k} \sum_j s_{ij}\, y_{i_{l-1},j}$

Load property of x'

Therefore

$\sum_{l=2}^{k} s_{ij_l} \leq \sum_{l=2}^{k} \sum_j s_{ij}\, y_{i_{l-1},j} \leq \sum_j s_{ij}\, x^*_{ij} \leq b_i$

The second inequality follows from the fact x^*_{ij} is spread amongst copies of i by y

Thus the load of items $j_2, j_3, \ldots, j_k$ is at most b_i. The load of item j_1 can be at most b_i . Therefore the total load on i is at most $2b_i$

Load property of x'

We charge j_k to fractional load on i_{k-1}, charge j_{k-1} to fractional load on i_{k-1} and so on. The last item j_1 cannot be charged.

Profit maximization

$\min \sum_{ij} p_{ij} x_{ij}$

s.t

$\sum_i x_{ij} \leq 1$

$\sum_j s_{ij} x_{ij} \leq b_i$

$x_{ij} = 0$ if $s_{ij} > b_i$
$x_{ij} \geq 0$ if $s_{ij} \leq b_i$

Given a feasible soln x^* to above LP we can round x^* to a feasible 0,1 soln x' s.t $\text{profit}(x') \geq ½ \text{ profit}(x^*)$

Note that no bin capacity constraints are violated in x'

Reduction to cost minimization

First by adding an extra dummy bin $m+1$ with infinite capacity and $p_{(m+1),j} = 0$ for all j we can assume that all items should be assigned. That is

$\min \sum_{ij} p_{ij}\, x_{ij}$

s.t

$\sum_i x_{ij} = 1$

$\sum_j s_{ij}\, x_{ij} \leq b_i$

$x_{ij} = 0 \text{ if } s_{ij} > b_i$

$x_{ij} \geq 0 \text{ if } s_{ij} \leq b_i$

Reduction to cost minimization

Since all items are assigned we can reduce profit maximization to cost minmization by setting $c_{ij} = -p_{ij}$

Exercise (hw): show how the algorithm for cost minimization can be used to obtain a *feasible* solution to profit maximization with ½ the optimum profit

Hint: each bin gets only one extra item that violates its capacity