Termination

$\forall m,\ m \notin FreeM \longleftrightarrow \exists w$ s.t. $(m,w) \in Tent$
$\forall w,\ w \notin FreeW \longleftrightarrow \exists m$ s.t. $(m,w) \in Tent$] freeM / freeW keeps track of unmatched workers

$\forall m,w,\ (m,w) \in Tent \longrightarrow MatchM[m]=w \wedge MatchW[w]=m$
$\forall m,\ MatchM[m] \neq NIL \longrightarrow (m, MatchM[m]) \in Tent$
$\forall w,\ MatchW[w] \neq NIL \longrightarrow (MatchW[w], w) \in Tent$

$\forall m\ \ m \in FreeM \longleftrightarrow MatchM[m] = NIL$
$\forall w\ \ w \in FreeW \longleftrightarrow Match[w] = NIL$] MatchM/MatchW keeps track of current matching

✓ $\forall w,\ w \in FreeW \longleftrightarrow \sim\exists m$ s.t. $(m,w) \in Invites$] welders free iff no invitations yet

$Invites = \bigcup_{i=1}^{n} \bigcup_{j=1}^{Next[i]-1} \{(m_i, PrefM[i][j])\}$] Next keeps track of invitations

✓ Tent is a matching

$Tent \subseteq Invites$

$|Invites| = k$

$\forall w,\ j<k,\ MatchW^{j}[w] \neq NIL \rightarrow MatchW^{j+1}[w], \ldots, MatchW^{k}[w]\ != NIL$
once welder is matched, always matched (receives invitation)

$\forall w,\ j<k,\ MatchW^{j}[w] \neq NIL \rightarrow$ w prefers $MatchW^{j+1}[w]$ to $MatchW^{j}[w]$, or they are same
each welder's sequence of matches gets better

Termination: After n^2 iterations, $|Invites| = n^2$ and so $= M \times W$.
Then there is no m (let alone $m \in FreeM$) s.t. $\exists w$ s.t. $(m,w) \notin Invites$

Post-condition: Tent is a perfect matching
Proof: suppose not – then $\exists$ unmatched m and unmatched w
$m \in FreeM$, $w \in FreeW$
$(m,w) \notin Invites$
guard is T $\Rightarrow\Leftarrow$
$\therefore$ Tent is a perfect matching

Tent is stable

Proof: Suppose not – there is a (m,w') that is an instability wrt Tent
so m prefers w' to $MatchM[m] = w$
and w' prefers m to $MatchW[w'] = m'$ (def instability)
$(m,w') \notin Tent$

$(m,w) \in Tent$ (INV ; $MatchM[m]=w$)
$(m,w) \in Invites$ (INV : $Tent \subseteq Invites$) need to add this to INV; easy to prove

$(m,w') \in Invites$ (m makes invites in ↓ order, and $(m,w) \in Invites$ and m prefers w')

Consider iteration k when (m,w') added to Invites

Two cases: 1) accepted, so $Match^{k}[w'] = m$
but $Match^{end}[w'] = m' \neq m$ ($(m,w) \in Tent$)
so w' prefers m' to m (INV: welders' matches ↑)

2) rejected, so $Match^{k-1}[w'] = m''$ and w' prefers m'' to m (condition on if that leads to this case)
substitute $m'' = m'$
$Match^{k}[w'] = m''$
⋮
but $Match^{end}[w'] = m'$, where either $m'' = m'$ and ... prefers m' to m

that leads to this case)

substitute $m'' = m'$

$Match^k[w'] = m''$

Let $Match^{end}[w'] = m'$ where either $m'' = m'$ and so w prefers m' to m

or $m'' \neq m'$, so

w prefers m' to m'' (INV: w's matches ↑)

and m' to m (transitivity)

in all cases, w' prefers m' to m $\Rightarrow\Leftarrow$

∴ no instability

∴ stable

Machinist Optimality

w is a valid partner for m if there is <u>some</u> stable matching S s.t. $(m,w) \in S$

best(m) is m's best valid partner
(1st valid partner on m's preference list)

				stable matching #1	stable matching #2
A	X Y V W Z	V	A D C E B	(A, Y)	(A, V)
B	V X W Y Z	W	A B D C E	(B, W)	(B, W)
C	V Z W Y X	X	D E C A B	(C, Z)	(D, X)
D	W V X Z Y	Y	C B A E D	(D, V)	(C, Y)
E	X Y V W Z	Z	A B D E C	(E, X)	(E, Z)

V, Y both valid partners of A — is X a valid partner?
X, Z both valid partners of E; X is <u>best</u> valid partner

G-S always returns $\{ (m, best(m)) \mid m \in M \}$ so these are best valid partners

Proof: Suppose not. Then some execution $\mathcal{E}$ of G-S returns S where $(m^*, w^*) \in S$ but $w^* \neq best(m^*)$

m^* invited $best(m^*)$ before w^* in $\mathcal{E}$ (invitations in ↓ order of pref)

so there is a rejection by a best valid partner in $\mathcal{E}$ ($best(m^*)$ rejected m^*)
so there is a rejection by a valid partner in $\mathcal{E}$ (best valid → valid)

consider 1st rejection in $\mathcal{E}$ by a valid partner: <u>w rejects m</u> (well-ordering)
then $w = best(m)$

if $best(m) \neq w$ then
m prefers best(m) to w (def best)
m invited best(m) before w (invites ↓)
m rejected by best(m) before making invitation to w (otherwise no more invites)
that's the 1st rejection by a valid partner ⇒⇐

w rejects m in favor of m' where w prefers m' to m (code for rejection)

there is some <u>stable matching S'</u> s.t. $(m, w) \in S'$ (def. valid)

find m' s.t. $(m', w') \in S'$ (must exist b/c S' is perfect matching)
then $w \neq w'$ (S' is a matching; $m \neq m'$)

<u>w' is a valid partner of m'</u> ($(m', w') \in S'$, S' is stable; def. valid)

<u>m' prefers w to w'</u>

(m', w) is <u>instability</u> in S' ⇒⇐ (def. instability)

∴ G-S returns $\{ (m, best(m)) \mid m \in M \}$

back to $\mathcal{E}$: w rejected m for m'
2 cases: that rejection was immediate b/c w already matched with m' or w initially accepted m but broke tentative match on getting invite from m'

m' → w
⋮
1st rejection by valid → m → w
(w immediately rejects m)

m → w
⋮
m' → w ← 1st rejection by valid
(w switches away from m)

but if <u>m' prefers w' to w</u>, then invitation m' → w' must be before m' → w (invites in ↓ order)

and invitation must be rejected before m' invites w (only way to make m' free to make more invitations)
of m' by w'

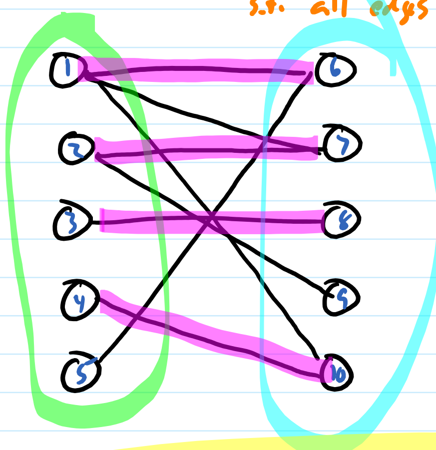
and invitation must be rejected before m' invites w (only way to make m' free to make
of m' by w' more invitations)

in either case, that rejection of m' by w' is by a valid partner and before the 1st rejection by a valid partner $\Rightarrow\Leftarrow$

Examples

Stable Matching : G – S

Bipartite Matching : Given bipartite G, find maximum matching
↳ can split verts into 2 sets s.t. all edges between sets

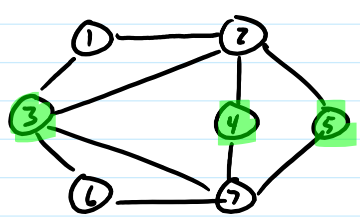


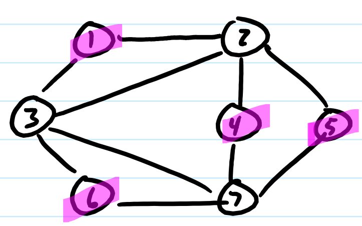
find maximum matching
↓
subset of edges s.t. each vert incident on ≤ 1 edge

Interval Scheduling : Given n requests with start s(i), finish f(i)
find largest set of non-overlapping requests
compatible

Weighted Interval Scheduling : add weight w_i to each request, find set of compatible requests that maximizes total weight

Independent Set : Given graph G, find largest set of vertices s.t. no edges between vertices in set

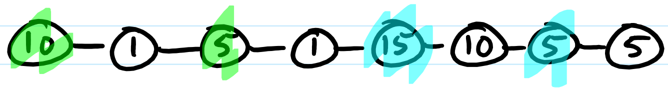




NP-complete

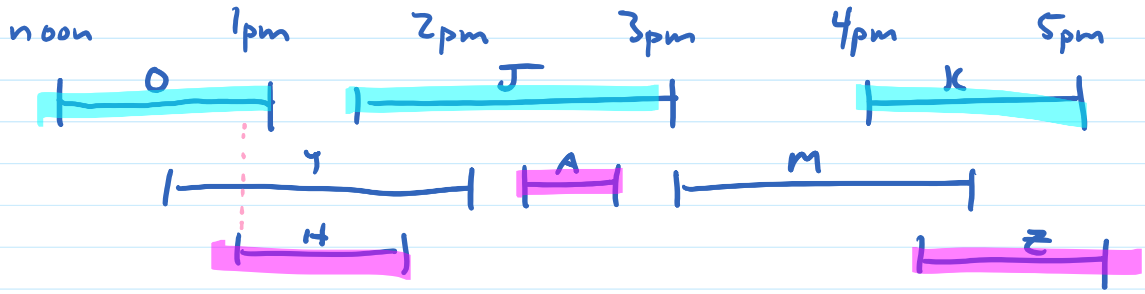
brute force: try all 2^n possible subsets

Competitive Facility Location : Given G with weighted verts, bound B
game between P1, P2 alternating choosing vert s.t. not adjacent to already chosen,
is there a strategy for P2 to guarantee a total ≥ B ?



PSPACE-complete

Continuing Education Credits



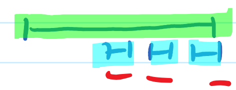
max is 3 compatible intervals

Interval Scheduling

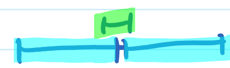
Given intervals labelled $1, \dots, n$ w/ interval i's start, finish $= s(i), f(i)$, find largest set of pairwise compatible intervals

interval i, j $(i \neq j)$ compatible if $s(i) \le f(i) \le s(j) \le f(j)$ or vice versa

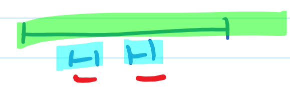
Greedy: choose 1st start? No:



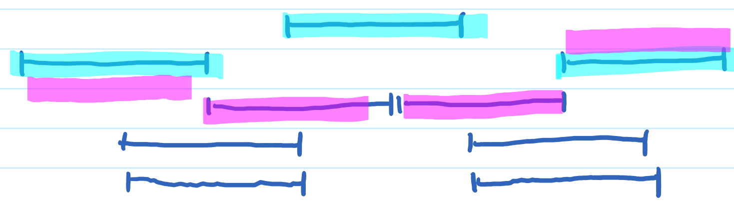
choose shortest? No



choose last finish? No



fewest overlaps? works for prev examples, but no:



from Jyh-haw Yeh, Boise State

choose 1st to finish...