APPROXIMATION ALGORITHMS

GREEDY ALGORITHMS & LOCAL SEARCH

RASMUS PAGH
UNIVERSITY OF COPENHAGEN

TODAY

FOUR CASE STUDIES:

- SCHEDULING OVERDUE JOBS ← GREEDY
- k-CENTER PROBLEM ← GREEDY
- SCHEDULING JOBS IN PARALLEL ← LOCAL SEARCH
- METRIC TRAVELLING SALESPERSON ← GREEDY

SCHEDULING OVERDUE JOBS

INPUT: FOR $i = 1, \ldots, n$:

$p_i \geq 0$: TIME NEEDED TO COMPLETE JOB i

$r_i \geq 0$: THE EARLIEST TIME JOB i CAN START

$d_i \leq 0$: AMOUNT OF TIME LEFT FOR JOB i

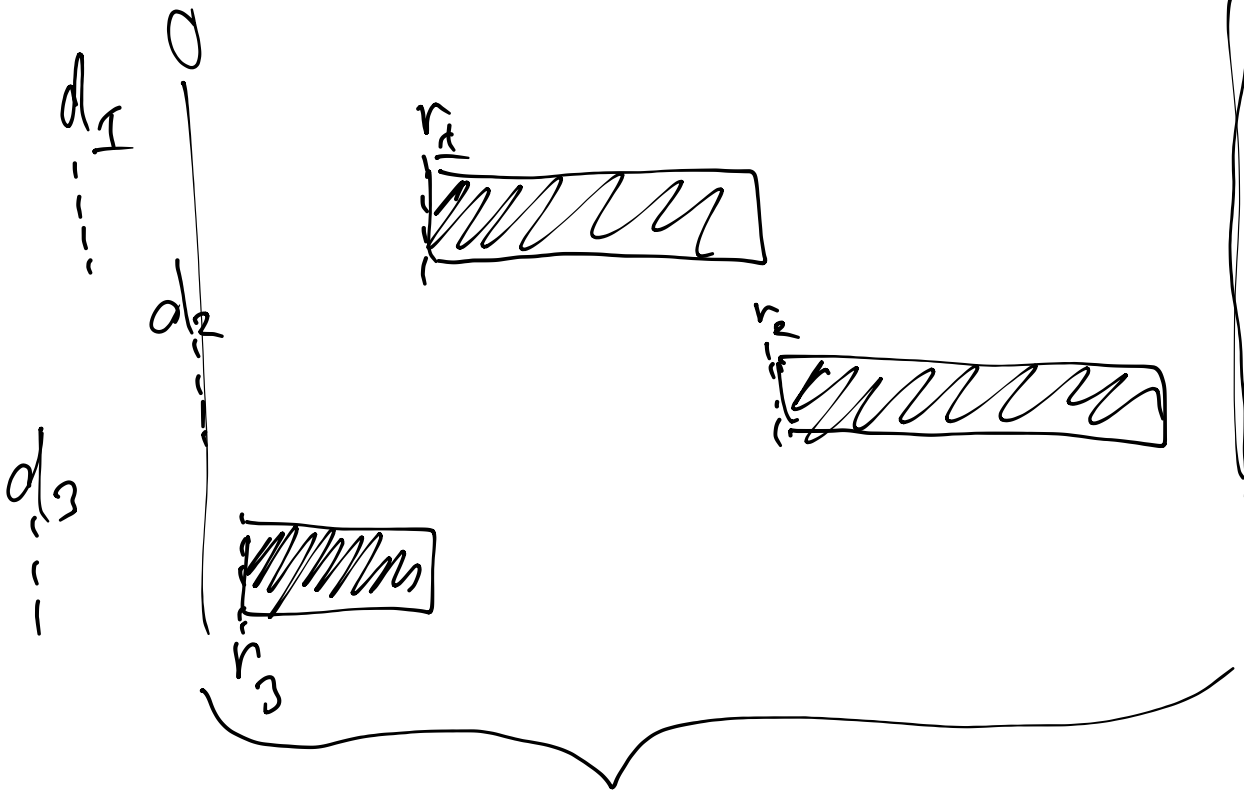
OBJECTIVE: CREATE SCHEDULE WITH NO OVERLAPS THAT MINIMIZES THE MAXIMUM LATENESS

$$\max_i c_i - d_i$$

WHERE c_i IS THE COMPLETION TIME OF JOB i.

GREEDY HEURISTIC

AT TIME t, CONSIDER UNSCHEDULED JOBS WITH $r_i \geq t$ AND SCHEDULE THE ONE WITH THE EARLIEST DUE TIME d_i.



PAUSE TO THINK:
FIND AN INPUT W. $n=2$ JOBS WHERE GREEDY'S SCHEDULE HAS COST ≥ 1.99 OPT

OPTIMAL COST

COST: $r_3 + p_3 + p_1 + p_2 - d_2$

ANALYSIS OF GREEDY

LAST JOB SCHEDULED STOPS AT TIME C_j.

COST OF GREEDY

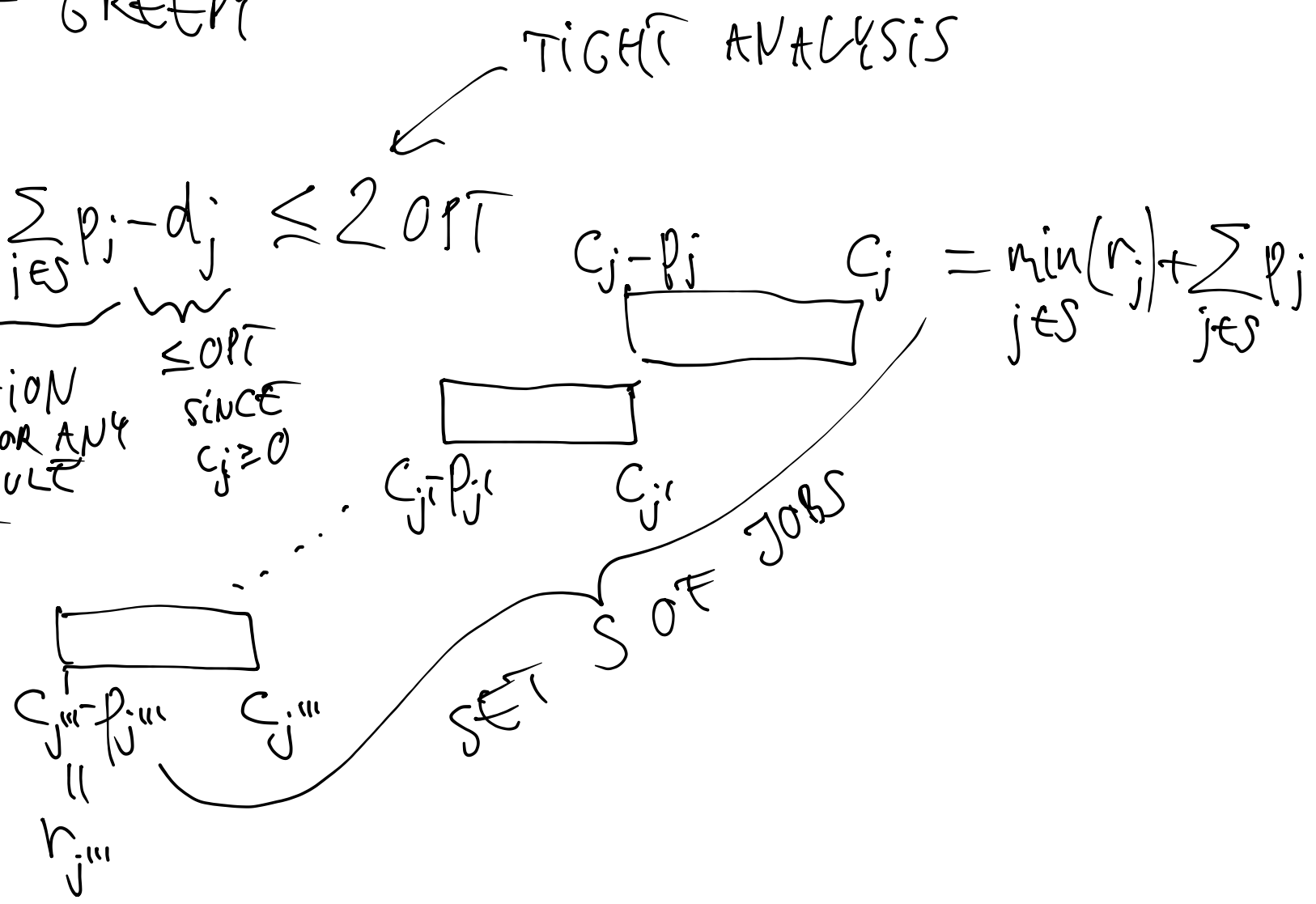
$= C_j - d_j$

$= \min_{j \in S}(r_j) + \sum_{j \in S} p_j - d_j \leq 2\,OPT$

TIGHT ANALYSIS

$\leq$ COMPLETION TIME FOR ANY SCHEDULE $\leq OPT$

$\leq OPT$ SINCE $c_j \geq 0$

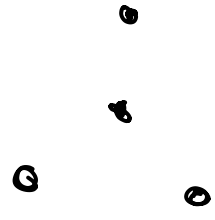


k-center problem

GIVEN POINT SET P, COVER IT WITH k DISKS OF RADIUS r WHERE r IS AS SMALL AS POSSIBLE.

distance to point set:
$d(x,S) = \min_{y \in S} d(x,y)$

OBJECTIVE: CHOOSE $S \subseteq P$ SUCH THAT $|S| = k$ AND $\max_{x \in P} d(x,S)$ is minimized

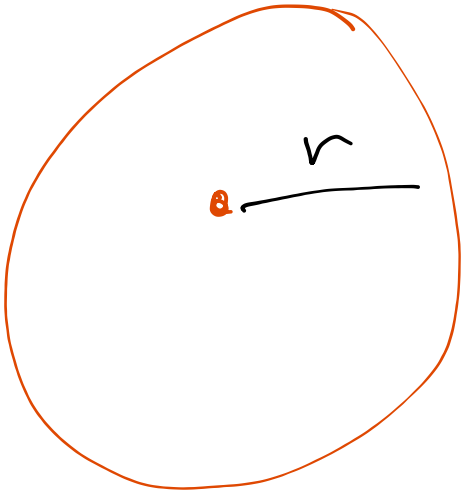


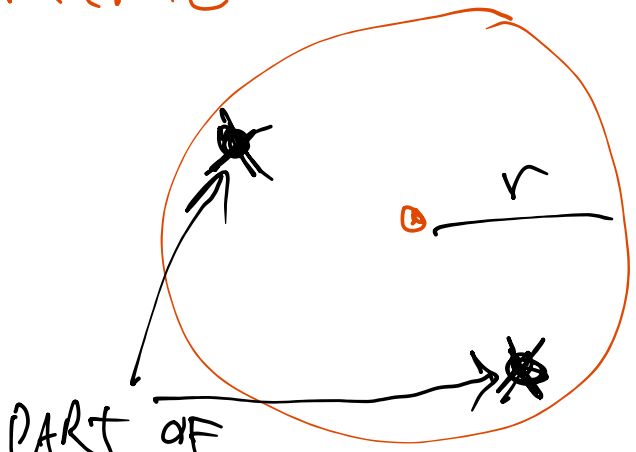
PAUSE AND THINK:
GREEDY CAN RETURN A k-CENTER THAT IS OF RADIUS 2·OPT.

GREEDY:
- START WITH $S = \emptyset$.
- ADD POINT $x \in P$ TO S THAT MAXIMIZES $d(x,S)$
- REPEAT UNTIL $|S| = k$

ANALISIS OF GREEDY

CONSIDER AN OPTIMAL SOLUTION





PART OF GREEDY SOLUTION

MAXIMUM DISTANCE $\leq 2r$

CASE 1:

GREEDY SELECTS A POINT FROM EACH CLUSTER ✓

CASE 2:

GREEDY MISS THE POINTS OF SOME CLUSTER.

$\Rightarrow$ PICKS TWO POINTS IN SOME CLUSTER

$\Rightarrow$ ALL DISTANCES TO S ARE $\leq 2r$. ✓

SCHEDULING JOBS IN PARALLEL

INPUT: INTEGER m: NUMBER OF MACHINES

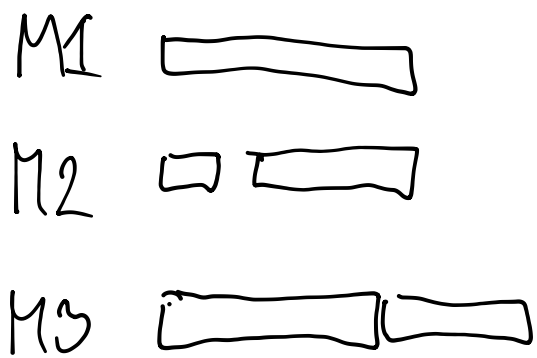
$p_1, \ldots, p_n \geq 0$: PROCESSING TIMES

OBJECTIVE: ALLOCATE JOBS TO MACHINES TO MINIMIZE $\max_i C_i$

PAUSE TO THINK:
SHOW THAT LOCAL SEARCH MAY NOT FIND THE OPTIMAL SOLUTION.
HOW LARGE A GAP CAN YOU GET?

LOCAL SEARCH:

- START WITH ANY FEASIBLE SOLUTION
- REPEAT:
  - MAKE LOCAL CHANGE: MOVE ONE JOB TO IMPROVE MAKESPAN
  - UNTIL NO LONGER POSSIBLE

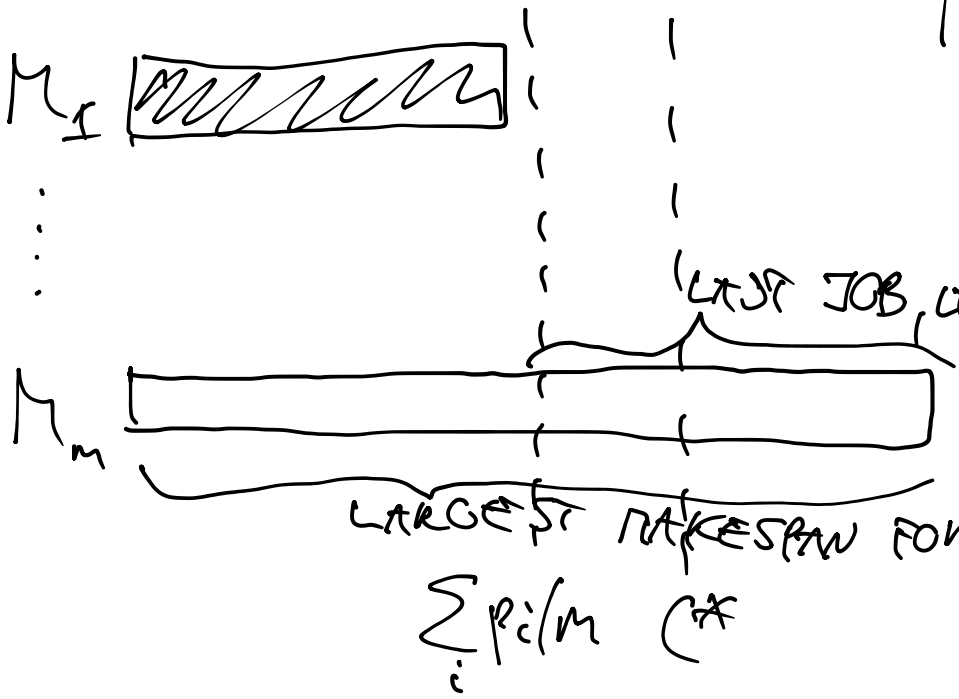


ANALYSIS OF LOCAL SEARCH

LET C^* BE THE COST OF AN OPTIMAL SCHEDULE

LEMMA: $C^* \geq \sum_i p_i/m$ (WORST COST CANNOT BE SMALLER THAN AVG.)

SUPPOSE LOCAL SEARCH RETURNS SOLUTION OF COST $> 2C^*$.



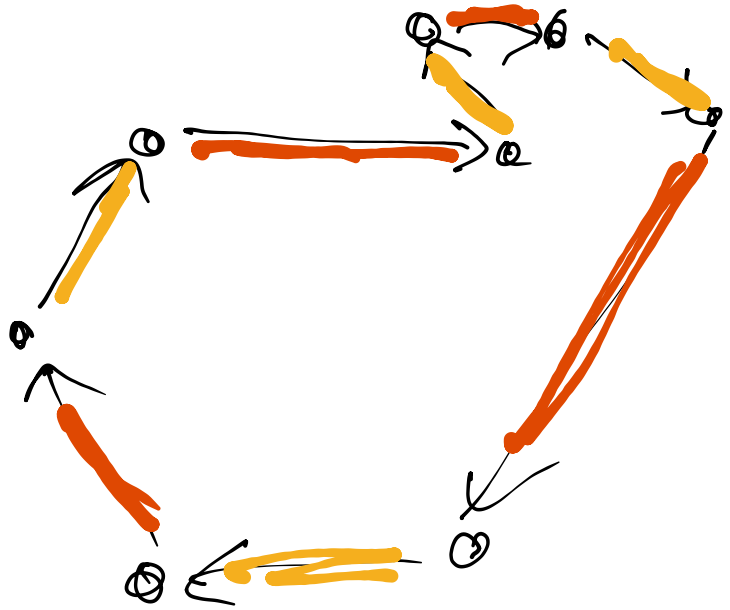
MAKESPAN = START TIME JOB j + p_j

$\leq \left(\sum_i p_i/m\right) + p_j$

$\leq C^* + C^* = 2C^*$

METRIC TRAVELING SALESPERSON PROBLEM

GIVEN POINT SET P AND A METRIC OF DISTANCES, $c_{ij} = d(x_i, x_j)$ FOR $x_i, x_j \in P$, FIND SHORTEST TOUR T OF ALL POINTS. MINIMIZE $\sum_{(x_i,x_j)\in T} c_{ij}$.



OBSERVATION:

a) A TOUR INCLUDES A SPANNING TREE

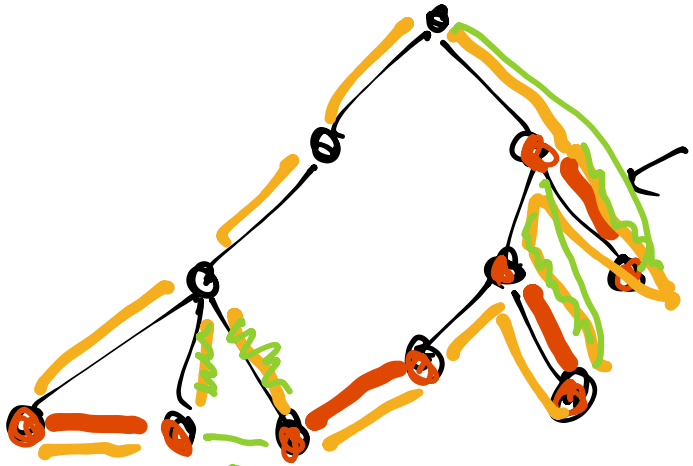
b) IF $|P|$ IS EVEN, A TOUR CONSISTS OF TWO MATCHINGS

OPT $\geq$ 2×MIN WEIGHT MATCHING $\longleftrightarrow$ OPT $\geq$ MIN WEIGHT SPANNING TREE

ADP?

CHRISTOFIDES ALGORITHM (1976)

1) COMPUTE MINIMUM SPANNING TREE T
2) ON THE ODD-DEGREE NODES IN T, COMPUTE A MINIMUM WEIGH MATCHING M
3) CONSTRUCT TOUR FROM EDGES OF T AND M WITH SHORTCUTS



COST OF TOUR
$\leq$ WEIGHT OF T + WEIGHT OF M
$\leq$ OPT + $\frac{1}{2}$ OPT

KARLIN, KLEIN & GHARAN (2020): APPROX FACTOR $\frac{3}{2} - \varepsilon$ FOR $\varepsilon > 10^{-36}$