Single machine models: Maximum Lateness

Problem $1||L_{max}$:

- Earliest due date first (EDD) is optimal for $1||L_{max}$ (Jackson's EDD rule)
- Proof: special case of Lawler's algorithm

Problem $1|r_j|C_{max}$:

Single machine models: Maximum Lateness

Problem $1||L_{max}$:

- Earliest due date first (EDD) is optimal for $1||L_{max}$ (Jackson's EDD rule)
- Proof: special case of Lawler's algorithm

Problem $1|r_j|C_{max}$:

- $1|r_j|C_{max} \propto 1||L_{max}$
  - define $d_j := K - r_j$, with constant $K > \max r_j$
  - reversing the optimal schedule of this $1||L_{max}$-problem gives an optimal schedule for the $1|r_j|C_{max}$-problem

Single machine models: Maximum Lateness

Problem $1|prec|L_{max}$:

- if $d_j < d_k$ whenever $j \rightarrow k$, any EDD schedule respects the precedence constraints, i.e. in this case EDD is optimal
- defining $d_j := \min\{d_j, d_k - p_k\}$ if $j \rightarrow k$ does not increase L_{max} in any feasible schedule

Single machine models: Maximum Lateness

Problem $1|prec|L_{max}$:

- if $d_j < d_k$ whenever $j \rightarrow k$, any EDD schedule respects the precedence constraints, i.e. in this case EDD is optimal
- defining $d_j := \min\{d_j, d_k - p_k\}$ if $j \rightarrow k$ does not increase L_{max} in any feasible schedule

Algorithm $1|prec|L_{max}$

1. make due dates consistent: set $d_j = \min\{d_j, \min_{k|j \rightarrow k} d_k - p_k\}$
2. apply EDD rule with modified due dates

Single machine models: Maximum Lateness

Remarks on Algorithm $1|prec|L_{max}$

- leads to an optimal solution
- Step 1 can be realized in $O(n^2)$
- problem $1|prec|L_{max}$ can be solved without knowledge of the processing times, whereas Lawler's Algorithm (which also solves this problem) in general needs this knowledge (Exercise),
- Problem $1|r_j, prec|C_{max} \propto 1|prec|L_{max}$

Single machine models: Maximum Lateness

Problem $1|r_j|L_{max}$:

- problem $1|r_j|L_{max}$ is NP-hard
- Proof: by reduction from 3-PARTITION (on the board)

Single machine models: Maximum Lateness

Problem $1|pmtn, r_j|L_{max}$:

- preemptive EDD-rule: at each point in time, schedule an available job (job, which release date has passed) with earliest due date.
- preemptive EDD-rule leads to at most k preemptions (k = number of distinct release dates)

Single machine models: Maximum Lateness

Problem $1|pmtn, r_j|L_{max}$:

- preemptive EDD-rule: at each point in time, schedule an available job (job, which release date has passed) with earliest due date.
- preemptive EDD-rule leads to at most k preemptions (k = number of distinct release dates)
- preemptive EDD solves problem $1|pmtn, r_j|L_{max}$
- Proof (on the board) uses following results:
  - $L_{max} \geq r(S) + p(S) - d(S)$ for any $S \subset \{1, \ldots, n\}$, where $r(S) = \min_{j \in S} r_j$, $p(S) = \sum_{j \in S} p_j$, $d(S) = \max_{j \in S} d_j$
  - preemptive EDD leads to a schedule with $L_{max} = \max_{S \subset \{1,\ldots,n\}} r(S) + p(S) - d(S)$

Single machine models: Maximum Lateness

Remarks on preemptive EDD-rule for $1|pmtn, r_j|L_{max}$:

- can be implemented in $O(n \log(n))$
- is an 'on-line' algorithm
- after modification of release and due-dates, preemptive EDD solves also $1|prec, pmtn, r_j|L_{max}$

Single machine models: Maximum Lateness

Approximation algorithms for problem $1|r_j|L_{max}$:

- a polynomial algorithm A is called an α-approximation for problem P if for every instance I of P algorithm A yields an objective value $f_A(I)$ which is bounded by a factor α of the optimal value $f^*(I)$; i.e. $f_A(I) \leq \alpha f^*(I)$

Single machine models: Maximum Lateness

Approximation algorithms for problem $1|r_j|L_{max}$:

- a polynomial algorithm A is called an α-approximation for problem P if for every instance I of P algorithm A yields an objective value $f_A(I)$ which is bounded by a factor α of the optimal value $f^*(I)$; i.e. $f_A(I) \leq \alpha f^*(I)$
- for the objective L_{max}, α-approximation does not make sense since L_{max} may get negative
- for the objective T_{max}, an α-approximation with a constant α implies $\mathcal{P} = \mathcal{NP}$ (if $T_{max} = 0$ an α-approximation is optimal)

Single machine models: Maximum Lateness

The head-body-tail problem $(1|r_j, d_j < 0|L_{max})$

- n jobs
- job j: release date r_j (head), processing time p_j (body), delivery time q_j (tail)
- starting time $S_j \geq r_j$;
- completion time $C_j = S_j + p_j$
- delivered at $C_j + q_j$
- goal: minimize $\max_{j=1}^{n} C_j + q_j$

Single machine models: Maximum Lateness

The head-body-tail problem ($1|r_j, d_j < 0|L_{max}$), (cont.)

- define $d_j = -q_j$, i.e. the due dates get negative!
- result: $\min_{j=1}^{n} C_j + q_j = \min_{j=1}^{n} C_j - d_j = \min_{j=1}^{n} L_j = L_{max}$
- head-body-tail problem equivalent with $1|r_j|L_{max}$-problem with negative due dates
  Notation: $1|r_j, d_j < 0|L_{max}$
- an instance of the head-body-tail problem defined by n triples (r_j, p_j, q_j) is equivalent to an inverse instance defined by n triples (q_j, p_j, r_j)
- for the head-body-tail problem considering approximation algorithms makes sense

Single machine models: Maximum Lateness

The head-body-tail problem $(1|r_j, d_j < 0|L_{max})$, (cont.)

- $L_{max} \geq r(S) + p(S) + q(S)$ for any $S \subset \{1, \ldots, n\}$, where

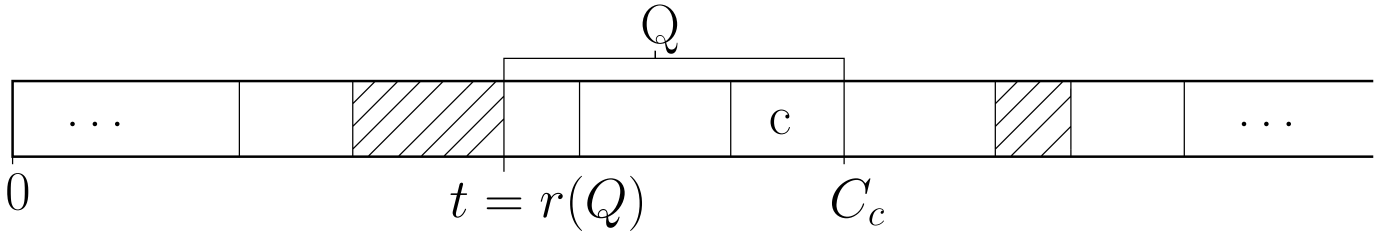
$$r(S) = \min_{j \in S} r_j,\ p(S) = \sum_{j \in S} p_j,\ q(S) = \min_{j \in S} q_j$$

(follows from $L_{max} \geq r(S) + p(S) - d(S)$ - slide 5)

Single machine models: Maximum Lateness

Approximation ratio for EDD for problem $1|r_j, d_j < 0|L_{max}$

- structure of an schedule



- critical job c of a schedule: job with $L_c = \max L_j$
- critical sequence Q: jobs processed in the interval $[t, C_c]$, where t is the earliest time that the machine is not idle in $[t, C_c]$
- if $q_c = \min_{j \in Q} q_j$ the schedule is optimal since then

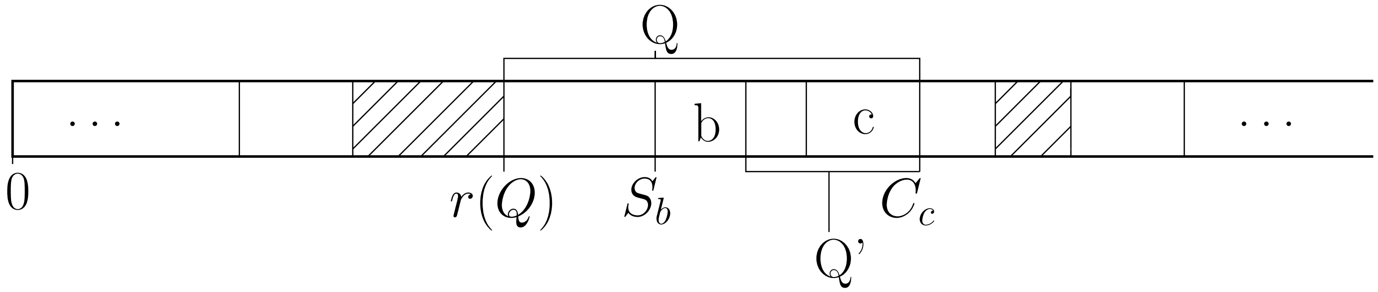
$$L_{max}(S) = L_c = C_c - d_c = r(Q) + p(Q) + q(Q) \leq L^*_{max}$$

- Notation: L^*_{max} denotes the optimal value

Single machine models: Maximum Lateness

Approximation ratio for EDD for problem $1|r_j, d_j < 0|L_{max}$

- structure of an schedule



- interference job b: last scheduled job from Q with $q_b < q_c$
- Lemma: For the objective value $L_{max}(EDD)$ of an EDD schedule we have
  1. $L_{max}(EDD) - L^*_{max} < q_c$
  2. $L_{max}(EDD) - L^*_{max} < p_b$
- Theorem: EDD is 2-approximation algorithm for $1|r_j, d_j < 0|L_{max}$

Single machine models: Maximum Lateness

Approximation ratio for EDD for problem $1|r_j, d_j < 0|L_{max}$

- Remarks:
  - EDD is also a 2-approximation for $1|prec, r_j, d_j < 0|L_{max}$ (uses modified release and due dates)
  - by an iteration technique the approximation factor can be reduced to 3/2

Single machine models: Maximum Lateness

Enumerative methods for problem $1|r_j|L_{max}$

- we again will use head-body-tail notation
- Simple branch and bound method:
  - branch on level i of the search tree by selecting a job to be scheduled on position i
  - if in a node of the search tree on level i the set of already scheduled jobs is denoted by S and the finishing time of the jobs from S by t, for position i we only have to consider jobs k with

$$r_k < \min_{j \notin S}(\max\{t, r_j\} + p_j)$$

  - lower bound: solve for remaining jobs $1|r_j, pmtn|L_{max}$
  - search strategy: depth first search + selecting next job via lower bound

Single machine models: Maximum Lateness

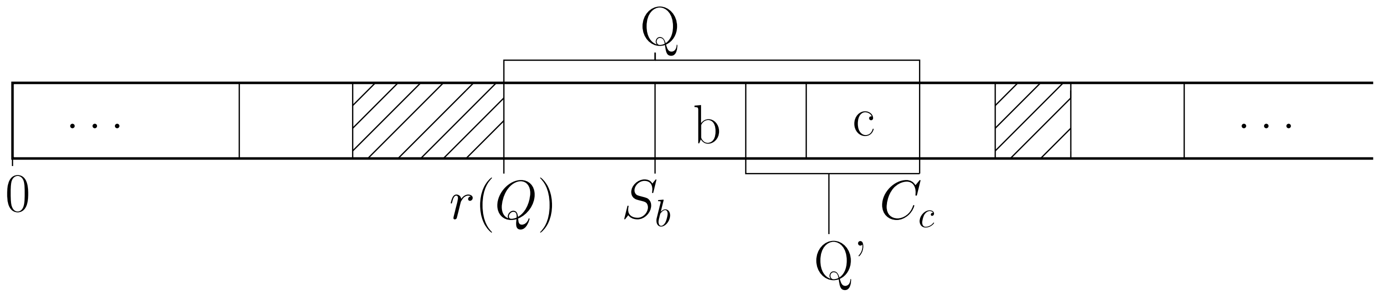
Advanced b&b-methods for problem $1|r_j|L_{max}$

- node of search tree = restricted instance
- restrictions = set of precedence constraints
- branching = adding precedence constraints between certain pairs of jobs
- after adding precedence constraints, modify release and due dates
- apply EDD to instance given in a node
  - critical sequence has no interference job: EDD solves instance optimal
    $\longrightarrow$ backtrack
  - critical sequence has an interference job: branch

Single machine models: Maximum Lateness

Advanced b&b-methods for problem $1|r_j|L_{max}$ (cont.)

branching, given sequence Q, critical job c, interference job b, and set Q' of jobs from Q following b

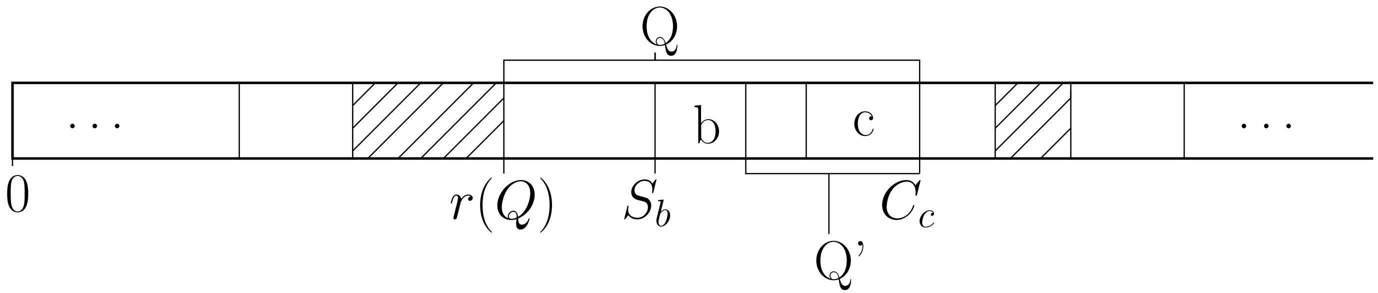


- $L_{max} = S_b + p_b + p(Q') + q(Q') < r(Q') + p_b + p(Q') + q(Q')$
- if b is scheduled between jobs of Q' the value is at least $r(Q') + p_b + p(Q') + q(Q')$; i.e. worse than the current schedule

Single machine models: Maximum Lateness

Advanced b&b-methods for problem $1|r_j|L_{max}$ (cont.)

branching, given sequence Q, critical job c, interference job b, and set Q' of jobs from Q following b



- $L_{max} = S_b + p_b + p(Q') + q(Q') < r(Q') + p_b + p(Q') + q(Q')$
- if b is scheduled between jobs of Q' the value is at least $r(Q') + p_b + p(Q') + q(Q')$; i.e. worse than the current schedule
- branch by adding either $b \to Q'$ or $Q' \to b$

Single machine models: Maximum Lateness

Advanced b&b-methods for problem $1|r_j|L_{max}$ (cont.)

- lower bounds in a node: maximum of
  - lower bound of parent node
  - $r(Q') + p(Q') + q(Q')$
  - $r(Q' \cup \{b\}) + p(Q' \cup \{b\}) + q(Q' \cup \{b\})$

  using the modified release and due dates
- upper bound UB: best value of the EDD schedules
- discard a node if lower bound $\geq UB$
- search strategy: select node with minimum lower bound

Single machine models: Maximum Lateness

Advanced b&b-methods for problem $1|r_j|L_{max}$ (cont.)

- speed up possibility:
  - let $k \notin Q' \cup \{b\}$ with $r(Q') + p_k + p(Q') + q(Q') \geq UB$
  - if $r(Q') + p(Q') + p_k + q_k \geq UB$ then add $k \rightarrow Q'$
  - if $r_k + p_k + p(Q') + q(Q') \geq UB$ then add $Q' \rightarrow k$