New Jersey Institute of Technology

Quick Minimization of Tardy Processing Time on a Single Machine

Baruch Schieber

Department of Computer Science, NJIT, NJ, USA

Joint work with

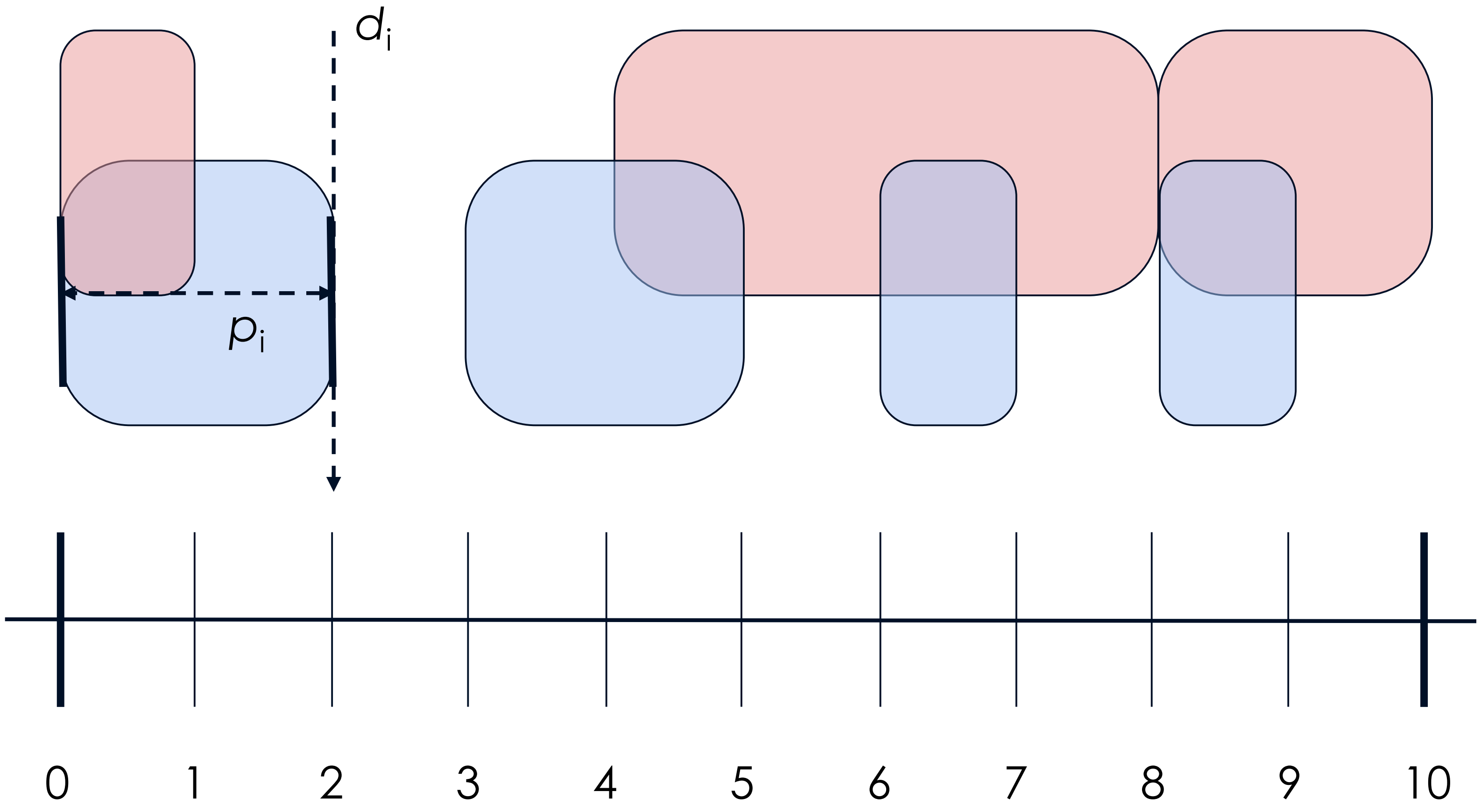
Pranav Sitaraman

Edison Academy Magnet School, Edison, NJ

Minimum Tardy Processing Time (MTPT) Problem

- n jobs $j_1, j_2, \ldots, j_n$
- Each job j_i is associated with
  - processing time p_i
  - due date d_i
- A job is tardy if it terminates after its due date.
- Goal: Find a feasible schedule of the jobs that minimizes the total processing time of tardy jobs.
- Assume that the jobs are ordered by due dates $d_1 < d_2 < \cdots < d_{D\#}$

Example



MTPT when all jobs have the same due date

- Let d be the due date and P be the sum of all processing times
- $P-d$ is a lower bound on the tardy processing time
- This lower bound can be achieved iff there is a subset of jobs whose total processing time is exactly d
- The Subset Sum problem (problem Weakly NP-Hard)

A pseudo polynomial time algorithm for MTPT

- [Lawler Moore 1969]
  - Any instance of the problem has an optimal Earliest Due Date (EDD) schedule.
  - In such a schedule:
    1. any early job precedes all late jobs
    2. any early job precedes all early jobs with later due dates
  - Dynamic programming: scan the jobs in order of due dates and at each stage maintain all the feasible "prefixes" of the EDD schedules
  - O(nP) time

Is there a faster algorithm for MTPT?

- [Bringmann et al. 2020]
  - Defined (max,min)-skewed-convolution
  - Showed an $\tilde{O}(P^{\alpha})$ time algorithm for MTPT, where $\tilde{O}(P^{\alpha})$ is the running time of a (max,min)-skewed-convolution of 2 vectors of size P
  - gave an $\tilde{O}(P^{7/4})$ time algorithm for a (max,min)-skewed convolution and thus also for MTPT
- [Klein et al. 2022]
  - gave an $\tilde{O}(P^{5/3})$ time algorithm for a generalized (max,min)-skewed convolution and thus for the MTPT problem

Our result

- An $\tilde{O}(P^{2-1/\alpha})$ time algorithm for MTPT, where $\tilde{O}(P^{\alpha})$ is the running time of a (max,min)-skewed-convolution of vectors of size P
- Results in an $\tilde{O}(P^{7/5})$ time algorithm for MTPT
- Breaks the $\tilde{O}(P^{3/2})$ time barrier of the previous approach
  - this is the running time of the best known and decades old algorithm of a (max,min)-convolution [Kosaraju 1989]
- Faster than [Lawler Moore 1969] when $n=\tilde{\omega}(P^{2/5})$

(max,min)-skewed-convolution

- Given two vectors with n+1 entries:
  - $A[0], \ldots, A[n]$ and $B[0], \ldots, B[n]$
- The (max,min)-skewed-convolution of A and B is a the 2n+1 vector $C[0], \ldots, A[2n]$ defined as
  - $C[k] = \max_{i+j=k} \min\{A[i], B[j] + k\}$
- We use a slight (equivalent) variation in which
  - $C[k] = \max_{i+j=k} \min\{A[i], B[j] - i\}$

Sumsets and set of subset sums

- Let A and B be two vectors of integers in the range $[0..P]$
- The sumset $A \oplus B = \{a + b | a \in A, b \in B\}$
  - Can be computed in $\tilde{O}(P)$ time via (+,x)-convolution of vector of size P
- The set of subset sums $\mathcal{S}(A) = \{\sum_{a \in Z} a \,|Z \subseteq A\}$
  - Can be computed in $\tilde{O}(\sum_{a \in A} a)$ time by successive sumset computations

An Õ($P \bullet D\#$) algorithm [Bringmann et al. 2020]

```
Let d_1 < ··· < d_{D#} denote the different due dates of jobs in 𝒥.
for i = 1, ..., D# do
    Compute X_i = {p_j : J_j ∈ 𝒥_i}
    Compute 𝒮(X_i)
Let S_0 = ∅.
for i = 1, ..., D# do          ▷ compute the sumsets and exclude infeasible sums
    Compute S_i = S_{i−1} ⊕ 𝒮(X_i).
    Remove any x ∈ S_i with x > d_i.
Return P − x, where x is the maximum value in S_{D#}.
```

Job bundles

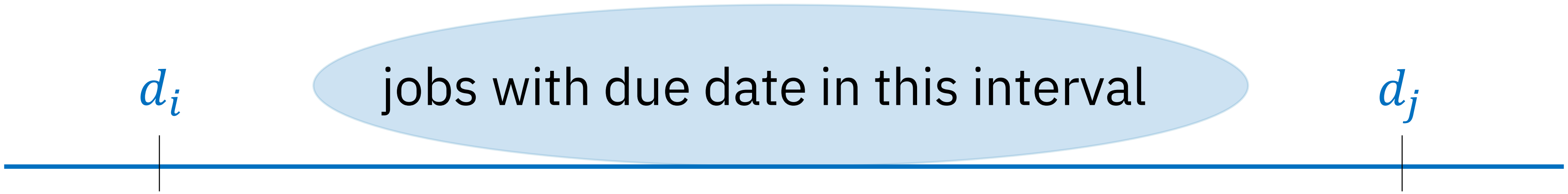
- Parameter $\delta \in (0,1)$
- Red due dates d_i all due dates such that the sum of processing time of jobs with this due date > $P^{1-\delta}$
- Group the jobs with the rest of the due dates into bundles
  - bundle: defined by maximal consecutive subsequences of due dates, none of which are red, such that the sum of processing time of jobs with these due dates ≤ $P^{1-\delta}$
- The number of red due dates and the number of bundles is $O(P^{\delta})$

The improved algorithm: outline

- Follows the structure of Algorithm 1 with additional processing of entire bundles that avoids processing each due date in a bundle individually
- The bundles are processed using a (max, min)-skewed-convolution
- Computing the bundles and red due dates takes $\tilde{O}(P)$ time
- Processing each bundle takes $\tilde{O}(P^{(1-\delta)\alpha}+P)$ time, where $\tilde{O}(P^{\alpha})$ is the running time of a (max,min)-skewed-convolution of vectors of size P
- Processing each red due date takes $\tilde{O}(P)$ time
- Total running time $\tilde{O}(P^{\delta} P^{(1-\delta)\alpha}+ P^{\delta} P)$
- Substituting $1-\delta=1/\alpha$ we get $\tilde{O}(P^{2-1/\alpha})$ time

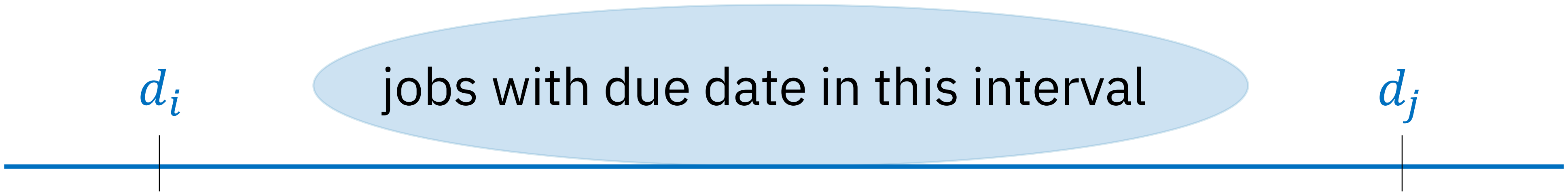
NJIT New Jersey Institute of Technology

Processing job bundles



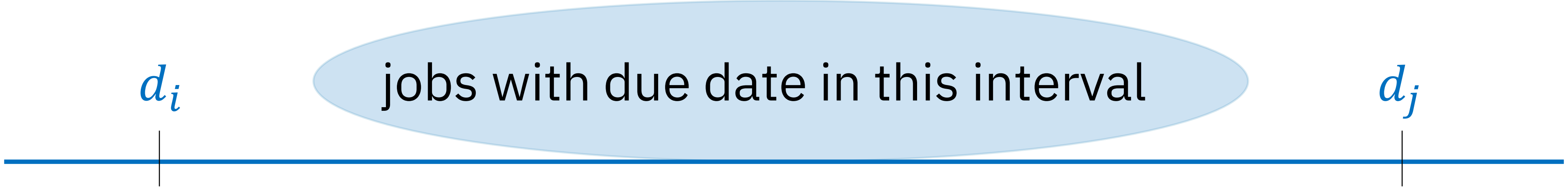
- J - the set of jobs in the bundle
- P_J - the total processing time of the jobs in J
- Compute the vector M
  - $M[x]$: the latest time t starting at which a subset of jobs in J with total processing time x can be scheduled feasibly , $-\infty$ otherwise
  - Can be done in $\tilde{O}(P_J)$ time via a (max,min)-skewed-convolution

Processing job bundles (cont.)



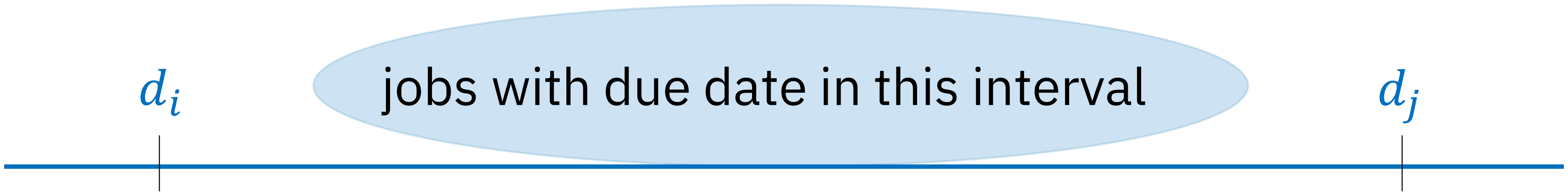
- **Input:** S_i - the processing times of all feasible schedules of jobs with due date up to (and including) d_i
- **Output:** S_j - the processing times of all feasible schedules of jobs with due date up to (and including) d_j
- Initially, $S_j \leftarrow S_i$

Computing S_i



- $T_2 = \{x | M[x] > -\infty\}$ and $T_1 = S_i \cap [0..d_i - P_J]$
- $S_j \leftarrow S_j \cup (T_1 \oplus T_2)$
- We are left with feasible schedules in which jobs with due date up to (and including) d_i end in $(d_i - P_J, d_i]$
- To find these schedules we compute a (max,min)-skewed-convolution

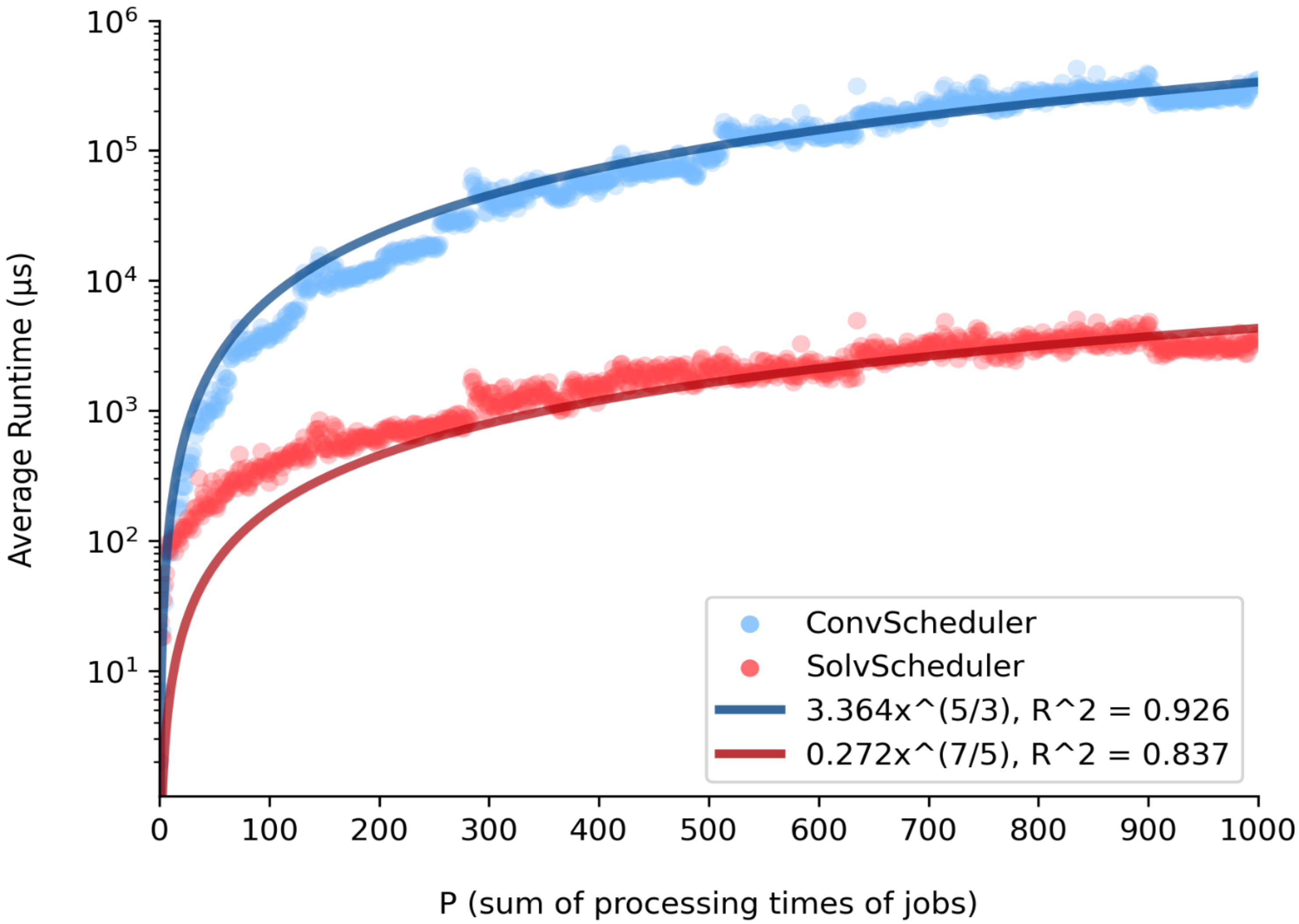
Computing the rest of the feasible schedules



- Two vectors of size P_J:
  - $A[x] = 0$ if $x + d_i - P_J + 1 \in S_i$, $-\infty$ otherwise
  - $B[x] = M[x] - d_i + P_J - 1$
- $C[k] = \max_{x+y=k} \min\{A[x], B[y] - x\}$
- $C[k] = 0 \Rightarrow k + d_i - P_J + 1 \in S_j$

New Jersey Institute of Technology

Runtime



NJIT New Jersey Institute of Technology

Open problems

- Since it is reasonable to assume that computing a (max, min)-skewed-convolution requires $\tilde{\omega}(P^{3/2})$ time our technique is unlikely to yield a $\tilde{o}(P^{4/3})$ running time
- It will be interesting to see whether this running time barrier can be broken, and whether the MTPT problem can be solved without computing a (max, min)-skewed-convolution
- Faster algorithms for (max, min)-skewed-convolution


Thanks!