CSE515 Advanced Algorithms
Lecture 18
Local Search and Scheduling Jobs on Identical Parallel Machines

Antoine Vigneron

antoine@unist.ac.kr

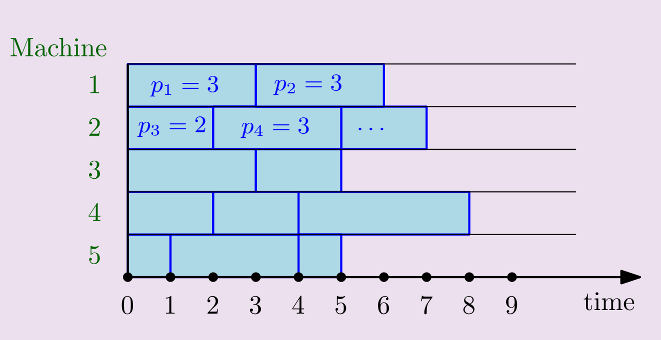
Ulsan National Institute of Science and Technology

April 29, 2021

Introduction

- Assignment 3 is due tomorrow.
- Reference: Section 2.3 in The design of approximation algorithms by David P. Williamson and David B. Shmoys.

Problem Statement



Here $m = 5$.

Here $C_3 = 5$, $C_{\max} = 8$.

INPUT:

- Number of machines m.
- Processing times $p_1, \dots, p_n$.

OUTPUT:

- A schedule that minimizes the *makespan*, which is the maximum completion time $C_{\max} = \max_{j=1,\dots,n} C_j$.

Local Search

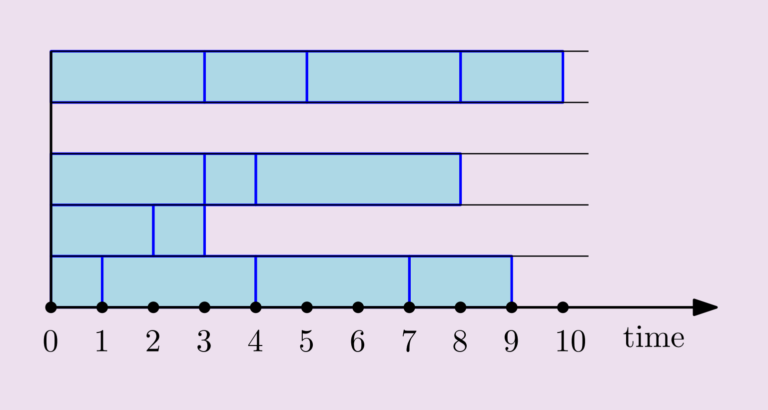
Simple local search:

- Start with a *feasible* solution.
- Apply a local change that improves the solution.
- Repeat until no local change gives an improvement.

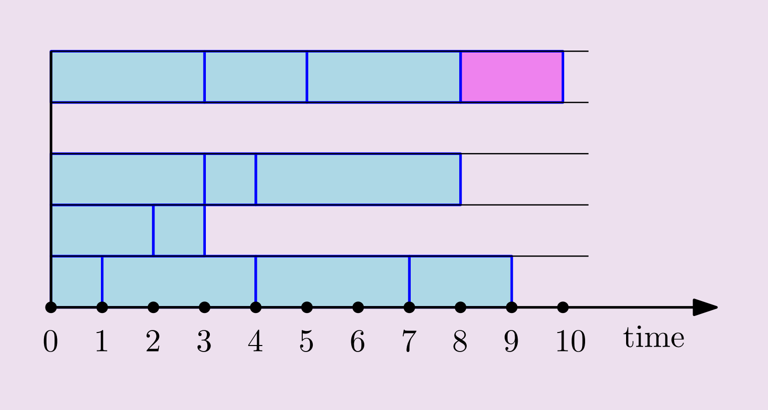
Difference with greedy algorithms:

- In a greedy algorithm, the initial solution may not be feasible.

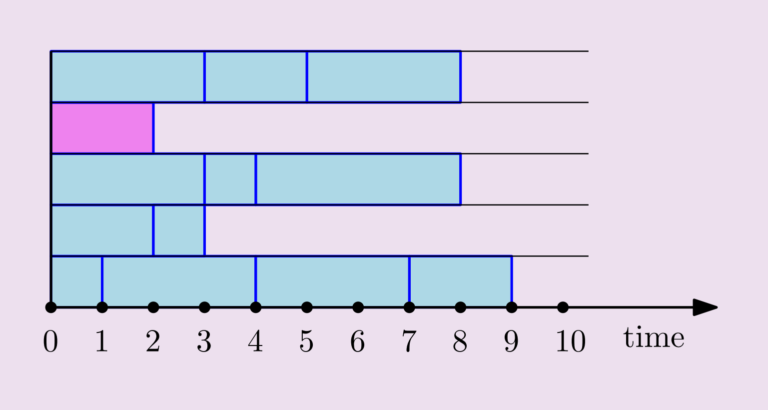
Example of Local Search



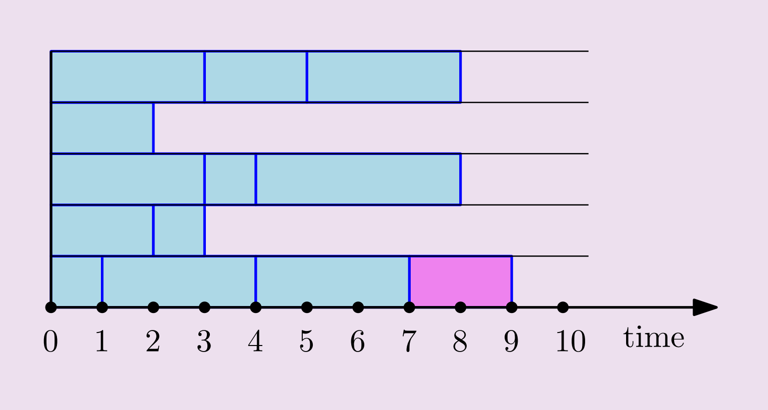
Example of Local Search



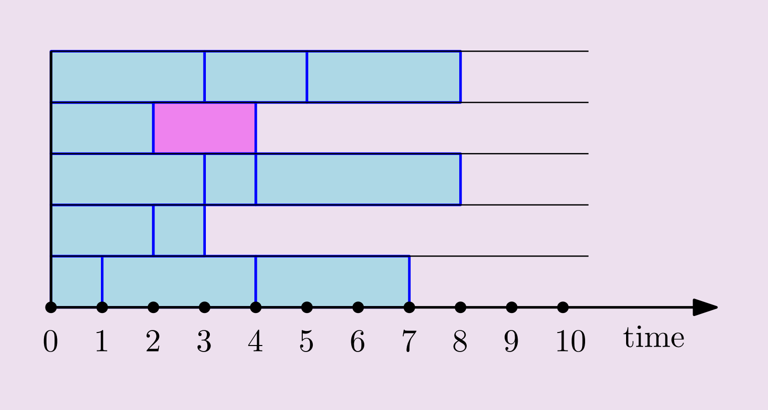
Example of Local Search



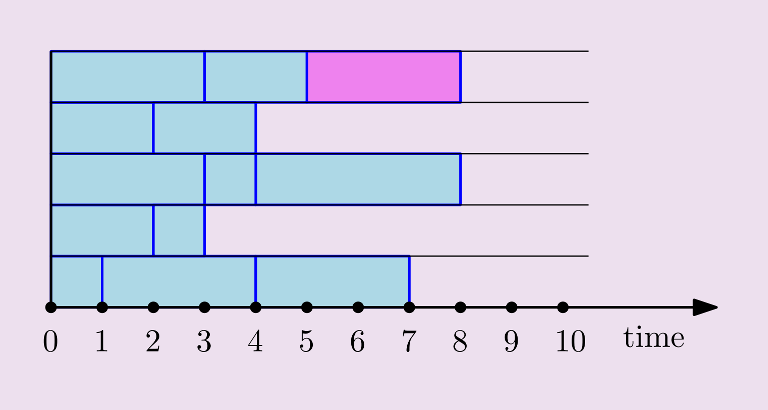
Example of Local Search



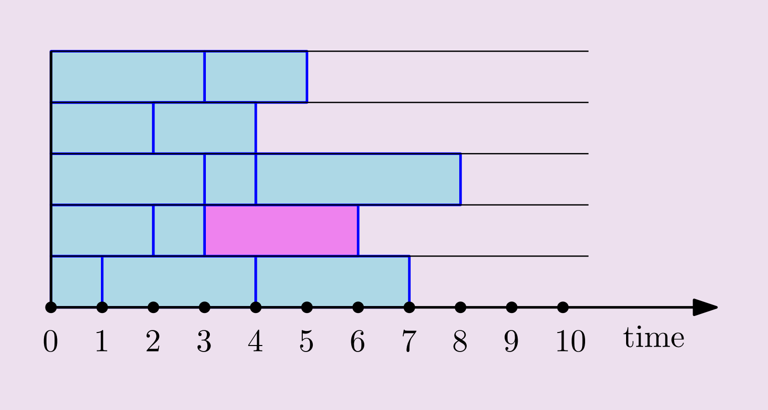
Example of Local Search



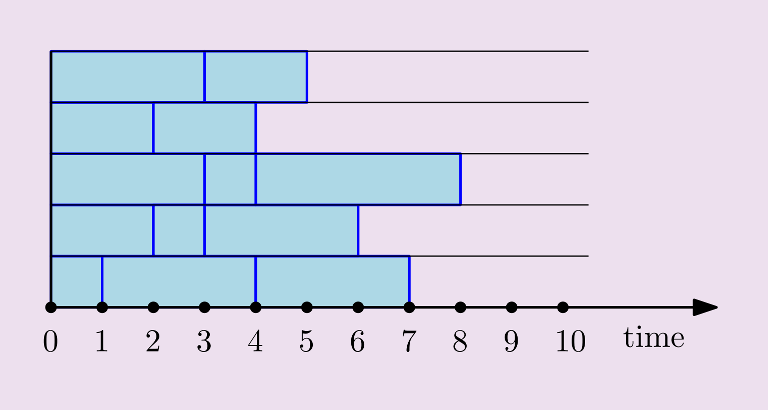
Example of Local Search



Example of Local Search



Example of Local Search



Pseudocode

Let $C_{\min}$ be the completion time of a machine that completes all its jobs earliest.

Local search algorithm for scheduling on parallel identical machines

1: start with an arbitrary feasible solution.
2: **while** there is a job j with $C_j = C_{\max} > C_{\min} + p_j$ **do**
3:     transfer job j to a machine that completes at time $C_{\min}$.
4: **return** current solution

Theorem

This algorithm is a 2-factor approximation algorithm.

Proof

- We need to prove:
  - The local search algorithms runs in polynomial time.
  - Its approximation factor is 2.
- *Done in class. See textbook.*

List Scheduling Algorithm

List scheduling algorithm

1. Build a list of all the jobs.
2. Whenever a machine becomes idle, process the next job on the list with this machine.

Theorem

The list scheduling algorithm is a 2-approximation algorithm for scheduling jobs on identical parallel machines.

Proof.

The local search algorithm cannot improve the solution returned by this algorithm. □

Longest Processing Time Rule

The *Longest Processing Time rule* (LPT) is to sort the jobs in the list by decreasing processing time.

Theorem

The list scheduling algorithm with LPT is a 4/3-approximation algorithm.

Proof: Partly done in class. See textbook and exercise set.