上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

SJTU

Task Scheduling among Geographically Distributed Datacenters with Max-min Fairness

Wendi Chen Wenhao Chen Haoyi You

Department of Computer Science, Shanghai Jiao Tong University, Shanghai, China

May 30, 2021

饮 水 思 源 • 爱 国 荣 校

目录

01 Definition and Assumption

- Problem Introduction and Challenges
- Problem Definition and Assumption
- Problem Formulation

Problem Introduction

Nowadays, there are **countless** bytes of data generated every second. We need to design a strategy to schedule data-analytic jobs to minimize the overall run time with max-min fairness.

Problem Introduction

Job Layer

- Multiple parallel jobs
- Shared datacenters

Stage Layer

- Precedence constraints
- Data reliance

Task Layer

- Dependency-free

Challenges

Multi-stage

- Previous one influences later one

Network Structure

- not necessarily fully-connected
- limited bandwidth

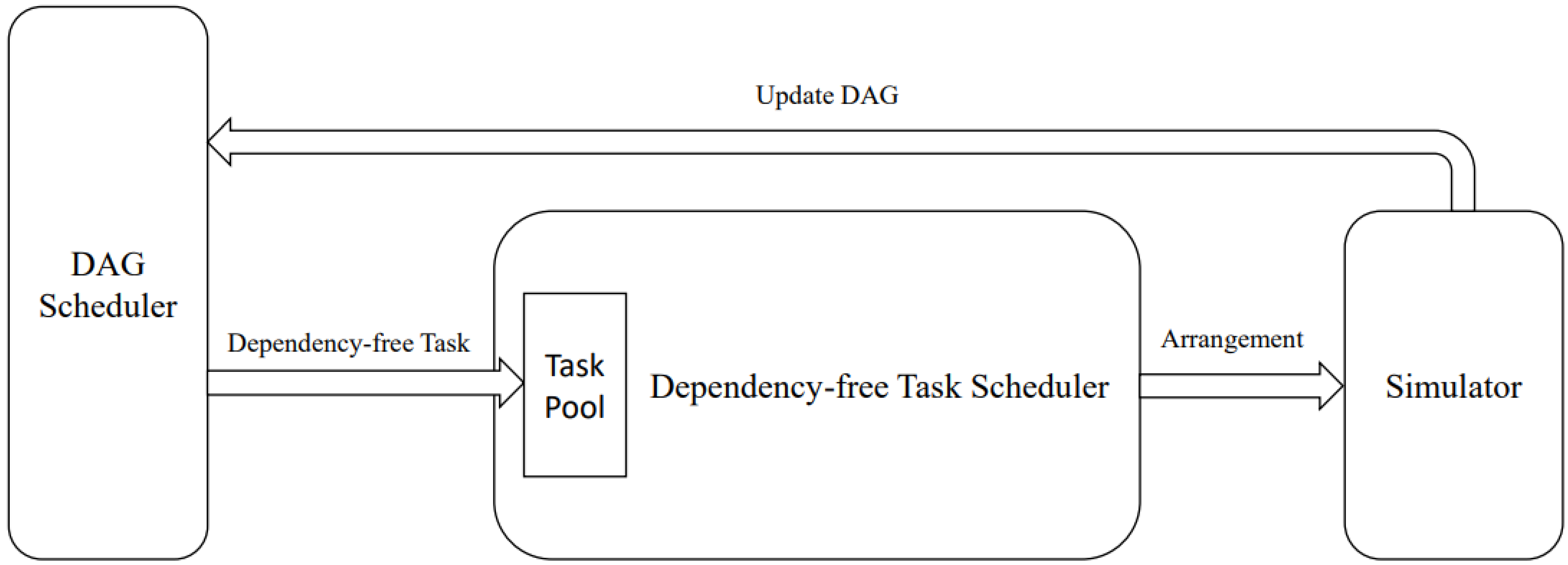
Max-min fairness

- Contradictory optimization objectives

Limited Slots

- some tasks may have to wait
- Some slots may be idle

Model Flowchart (Solutions to Challenge 1)



Assumptions (Solutions to Challenges 2-3)

- The bandwidth between different DC should not differ too much.
- Use path to connect unconnected DCs. The bandwidth is the lowest bandwidth along the path. (**Floyd** here)

$$b_{i,j} = \max_{k \in V}\{\min\{b_{i,k}, b_{k,j}\}\}$$

- Bandwidths are independent to tasks.

Definition of Max-min (Solutions to Challenge 4)

- Find an arrangement strategy to minimize the completion time of the slowest task.
- Do the relaxation.
- Find an arrangement strategy to minimize the completion time of the second slowest task.
- Repeat the above process until all tasks are minimized or relaxed.

Formulation of The Original Problem

$$
\begin{aligned}
\mathbf{lexmin}_x \quad & \mathbf{f} = (\tau_1, \tau_2, \cdots, \tau_k) \\
s.t. \quad & \tau_k = \max_{i \in \mathcal{T}_k} f_i^k, \forall k \in \mathcal{K} \\
& f_i^k = s_i^k + \sum_{j \in \mathcal{D}} x_{i,j}^k (c_{i,j}^k + e_{i,j}^k), \forall i \in \mathcal{T}_k, \forall k \in \mathcal{K} \\
& \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{T}_k} x_{i,j}^k [s_i^k \leq t < f_i^k] \leq a_j, \forall j \in \mathcal{D}, \forall t \geq 0 \\
& f_q^k \leq s_i^k, \forall q \in R_i^k, \forall i \in \mathcal{T}_k, \forall k \in \mathcal{K} \\
& \sum_{j \in \mathcal{D}} x_{i,j}^k = 1, \forall i \in \mathcal{T}_k, \forall k \in \mathcal{K} \\
& x_{i,j}^k \in \{0, 1\}, \forall i \in \mathcal{T}_k, \forall j \in \mathcal{D}, \forall k \in \mathcal{K}
\end{aligned}
$$

Formulation of The Sub-problem

$$\begin{aligned}
\textbf{lexmin}_x \quad & \mathbf{f} = (\tau_1, \tau_2, \cdots, \tau_k) \\
s.t. \quad & \tau_k = \max_{i \in \mathcal{T}_k, j \in \mathcal{D}} x_{i,j}^k (c_{i,j}^k + e_{i,j}^k), \forall k \in \mathcal{K} \\
& \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{T}_k} x_{i,j}^k \leq a_j, \forall j \in \mathcal{D} \\
& \sum_{j \in \mathcal{D}} x_{i,j}^k = 1, \forall i \in \mathcal{T}_k, \forall k \in \mathcal{K} \\
& x_{i,j}^k \in \{0, 1\}, \forall i \in \mathcal{T}_k, \forall j \in \mathcal{D}, \forall k \in \mathcal{K}
\end{aligned}$$

Assumptions for Data Generator

- 80% short jobs and 20% long jobs
- Distribution is similar to toy data
- Links are the same as these in toy data

02

NP-Completeness and Proof

New problem definition

Problem 1

Single Execution Time Scheduling：

- Tasks n
- Processors k
- Tasks partial order $\prec$
- Time limitation t
- Each task executes a unit time

Problem 2

Single Execution Time Scheduling with variable number of processors:

- Tasks n
- Processors c_i at time i
- Tasks partial order $\prec$
- Time limitation t
- Each task executes a unit time
- $\sum_{i=0}^{t-1} c_i = n$

Problem 3

$3 - SAT$ problem

is known as an NP-Complete problem

Problem Relationship

Problem 2$\preccurlyeq_p$Problem 1

Problem 1

- Tasks n
- Processors k
- Tasks partial order $\prec$
- Time limitation t
- Each task executes a unit time

Problem 2

- Tasks n
- Processors c_i at time i
- Tasks partial order $\prec$
- Time limitation t
- Each task executes a unit time
- $\sum_{i=0}^{t-1} c_i = n$

- Let $k = \sum_{i=0}^{t-1} c_i = n$
- Add $n - c_i$ tasks J_{ij} at time i
- Add partial order to limit J_i at time i

$$J_{i,l} \leq J_{i+1,s}$$

Problem Relationship

Problem $3 \preccurlyeq_p$ Problem 2

Problem 2

- Tasks n
- Processors c_i at time i
- Tasks partial order $\prec$
- Time limitation t
- Each task executes a unit time
- $\sum_{i=0}^{t-1} c_i = n$

Problem 3

$3 - SAT$

- Assume m liberates and n clauses
- Construct tasks

$x_{ij}, \overline{x_{ij}}$ $\quad 1 \le i, j \le m$
$y, \overline{y_i}$ $\quad 1 \le i \le m$
D_{ij} $\quad 1 \le i \le m, 1 \le j \le 7$

- Construct partial order

$x_{ij} \prec x_{i+1,j}$ $\quad \overline{x_{ij}} \prec \overline{x_{i+1,j}}$
$x_{i,i-1} \prec y_i$ $\quad \overline{x_{i,i-1}} \prec \overline{y_i}$
For D_{ij}, consider $j = (a_1 a_2 a_3)_2$ and let corresponding
x_{im} or $\overline{x_{im}} \prec D_{ij}$

- Add c_i and time limitation

$t = m + 3$
$c_i = 2m + 2$ $\quad 2 \le i \le m$
$c_0 = m$ $\quad c_1 = 2m + 1$
$c_{m+1} = n + m + 1$ $\quad c_{m+2} = 6n$

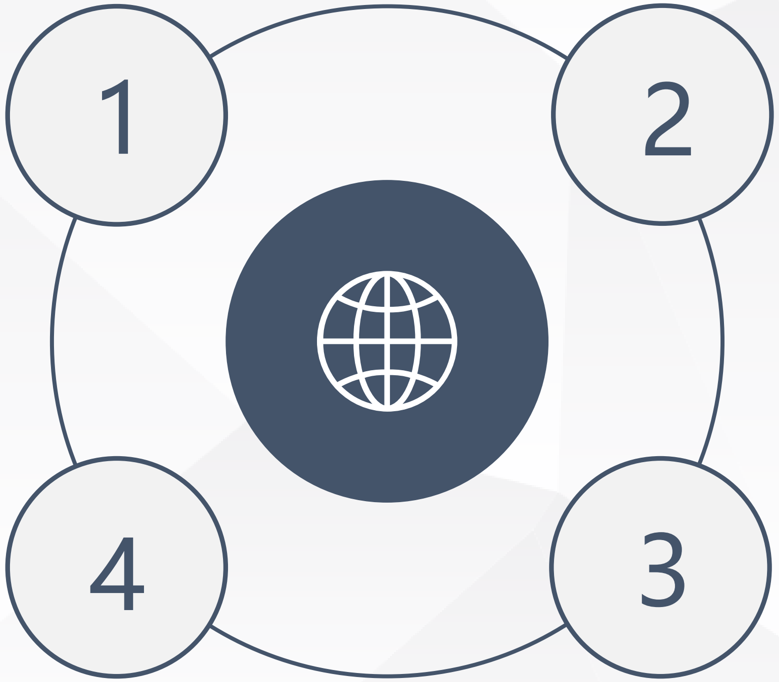
03

Models

Models

Greedy

K-Greedy



Network-Based Fair

Network-Based Greedy

Overview

Greedy Approach:

Naïve and intuitive approach

K-Greedy Approach:

Make concessions

Network-Flow-Based Greedy:

Think at high level

Network-Flow-Based Fair:

Ensure max-min fairness

Greedy

- Intuitive approach
- Assign task with shortest transferring time
- $O\big(m\ log\ (|J|m)\big)$

K-Greedy

- Compromise approach
- For task i, skip first k[i] assignments.
- $O(Km\ log\ (|J|m))$
- Example

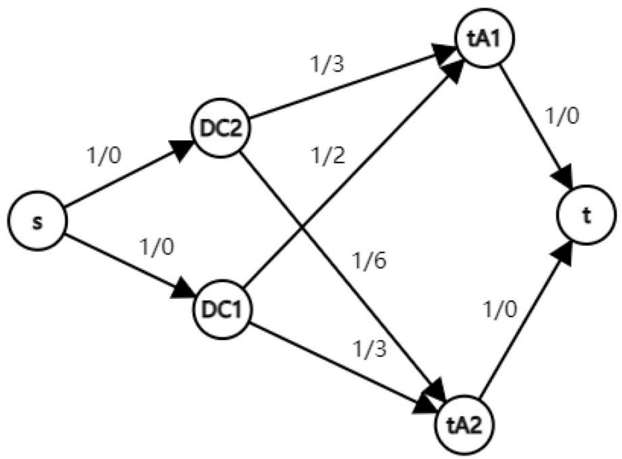
(a) (b)

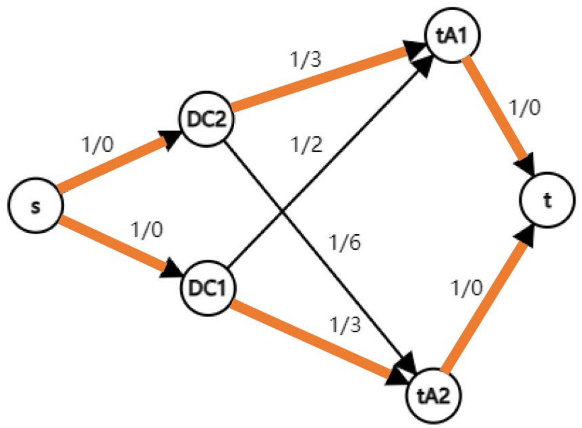
Network-Flow-Based Greedy Approach

- Consider k tasks simultaneously.
- Build a network!
- Compute maximum flow minimum cost

Network Example

- $|J|m$ $edges$ and $|J| + m$ $vertices$
- $O(|J|m^2 + |J|^2m)$
- Tighter bound: $O(|J|m^2)$

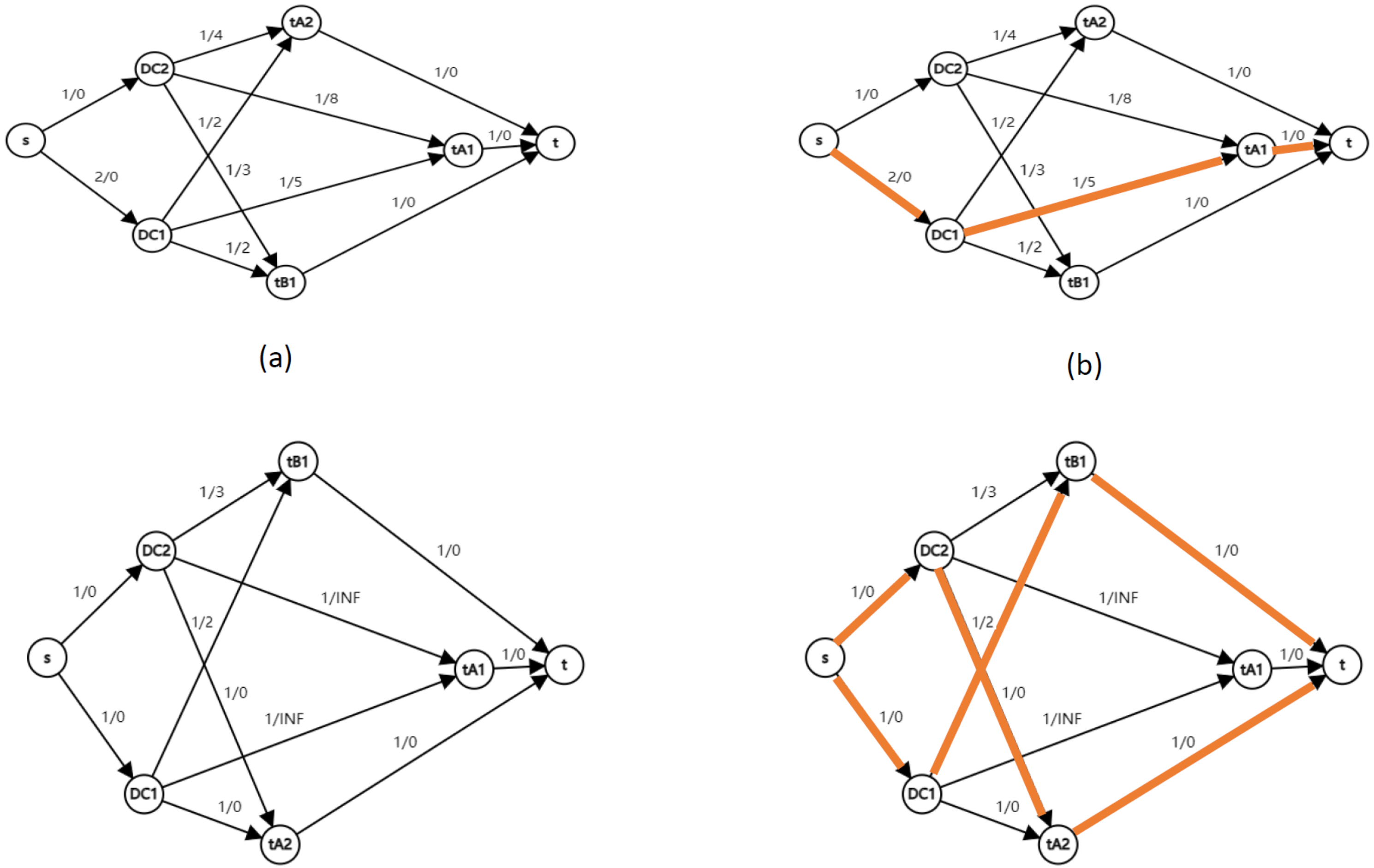




Network-Flow-Based Fair Approach

- Ensure max-min fairness.
- Maximum flow with minimum $\max\{cost\}$
- Certifier: Let $G(\Delta)$ be the of edges with $cost \leq \Delta$
- Use binary search to find bottleneck(Δ).
- Find bottleneck, update job group and repeat.
- $O\big((|J|m^2 + |J|^2 m)\log(cost_{max})\,|K|\big)$

Network Example



Fair: (5,4,2)

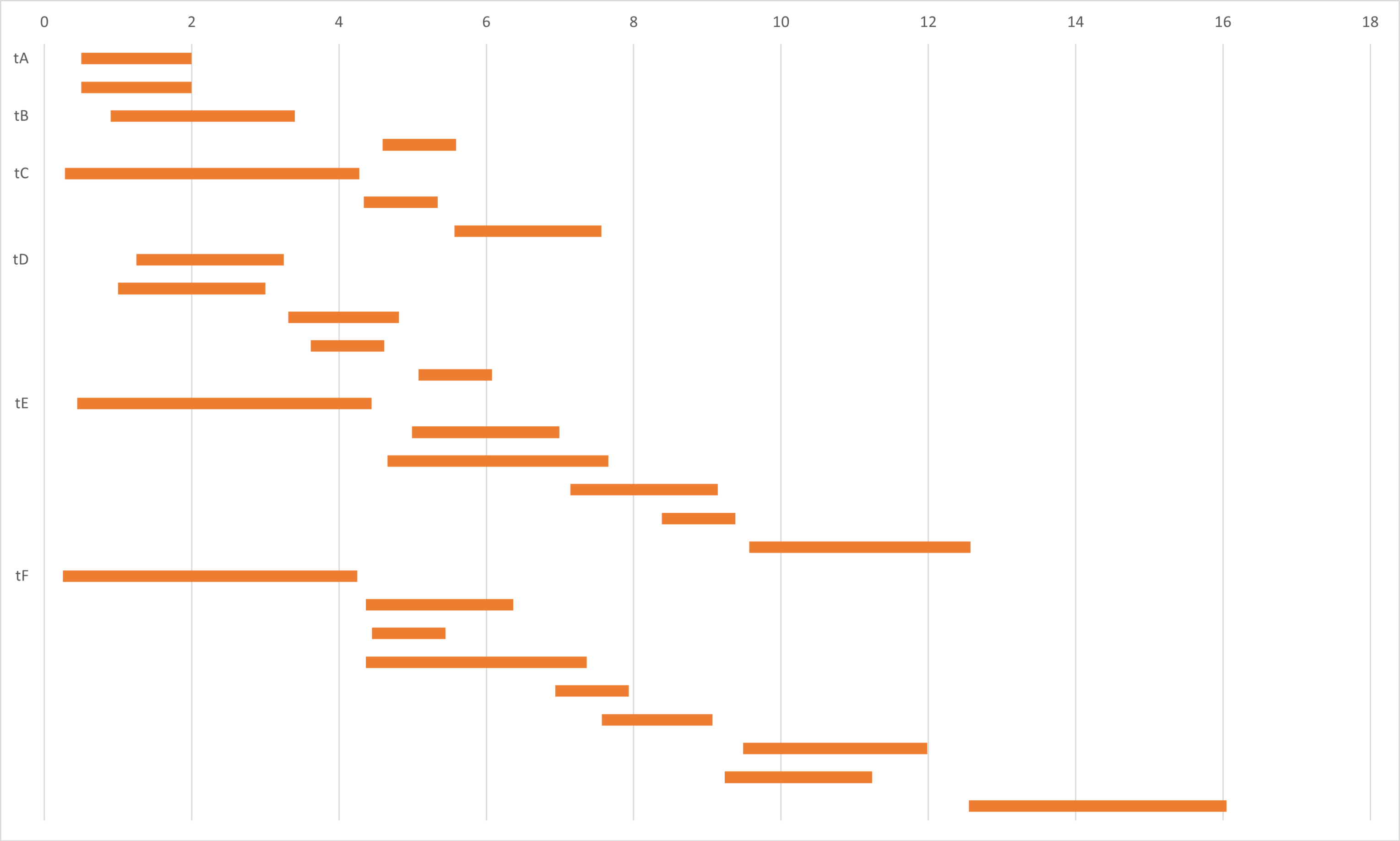
Greedy: (5,2,3)

04

Performance Analysis

Optimal Solution of Toy Data

- **Property**: Greedy yields optimal solution with ∞ slots.
- Toy data has relatively large slots in DC.



Performances Overview

- Different job amount and different task duration

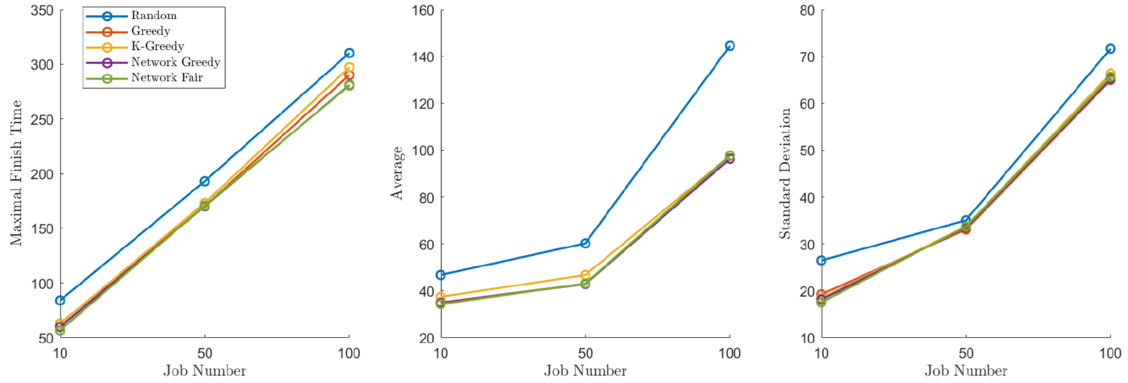


Fig. 6. 60% small jobs

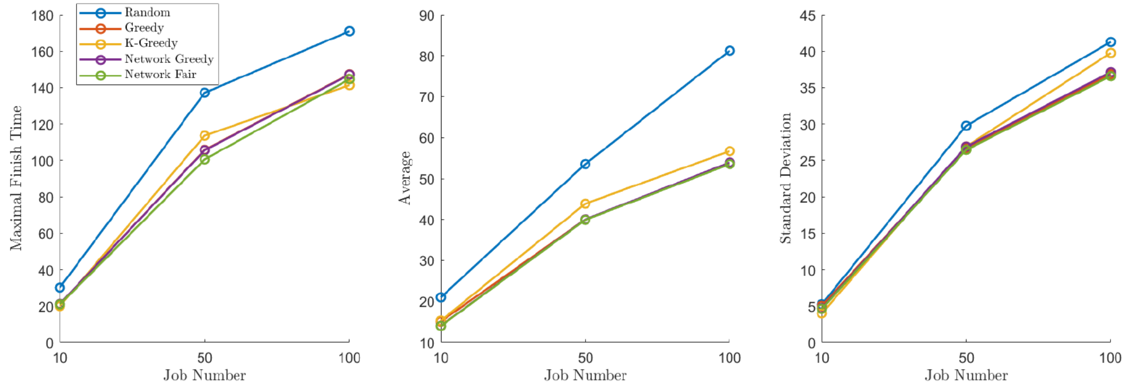
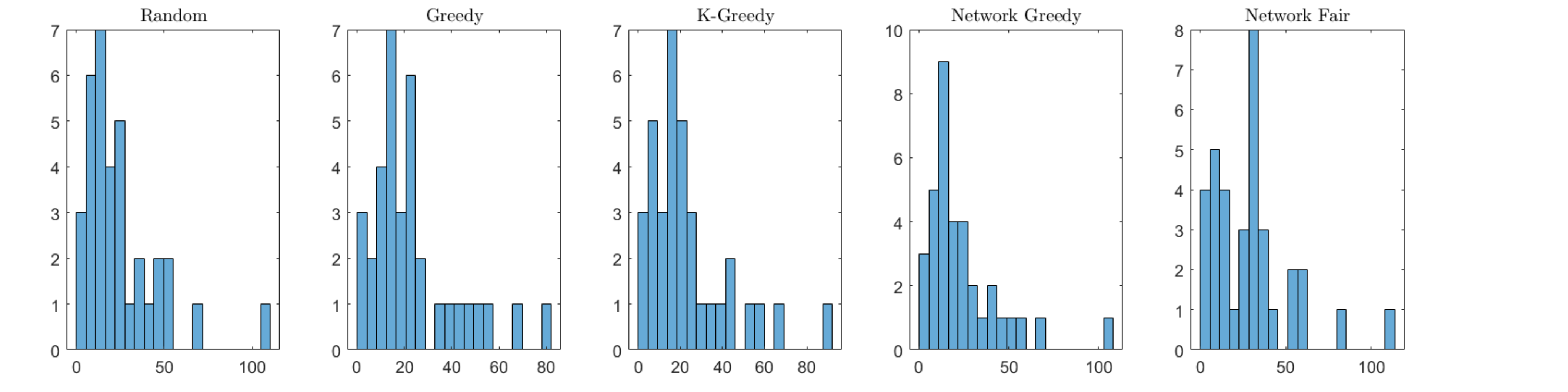


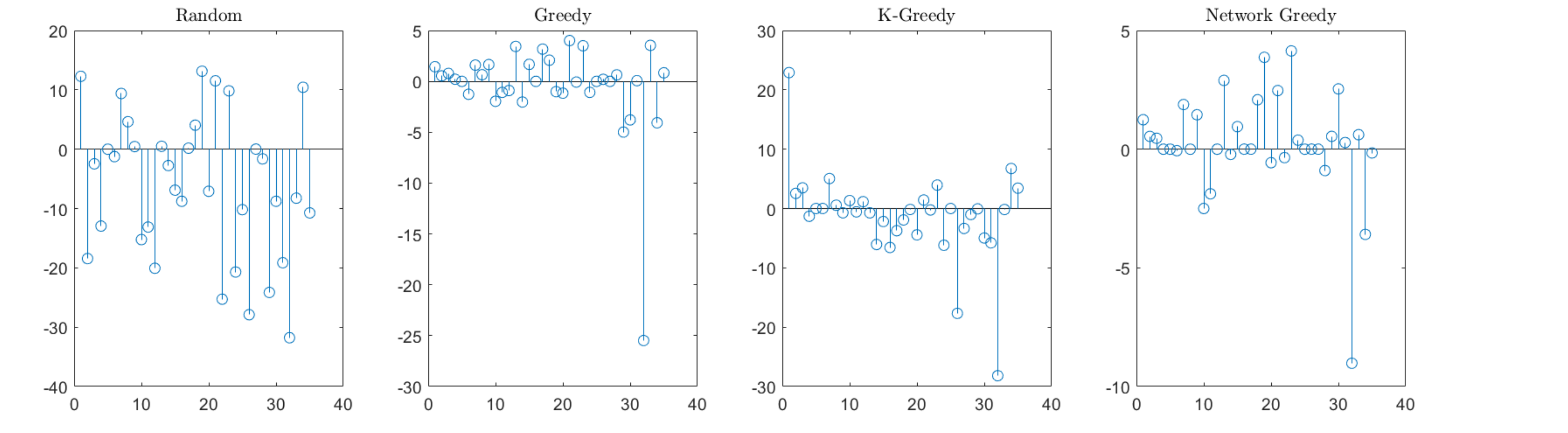
Fig. 7. 80% small jobs

Performances Details

Basic Mathematic Characteristics of Performance			
Methods	Average Time	Finish Time	Standard Deviation
Random Approach	30.823	107.331	22.6757
Greedy Approach	24.203	106.76	20.8664
k-Greedy Approach	24.8509	114.503	22.8843
Network-Flow-Based Greedy Approach	23.4635	90.35	19.2471
Network-Flow-Based Fair Approach	23.6664	81.33	18.4392

Performances Details





上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

Thanks

饮 水 思 源　爱 国 荣 校