Job Allocation with a Consideration of Fairness

Chi-Wei Liu

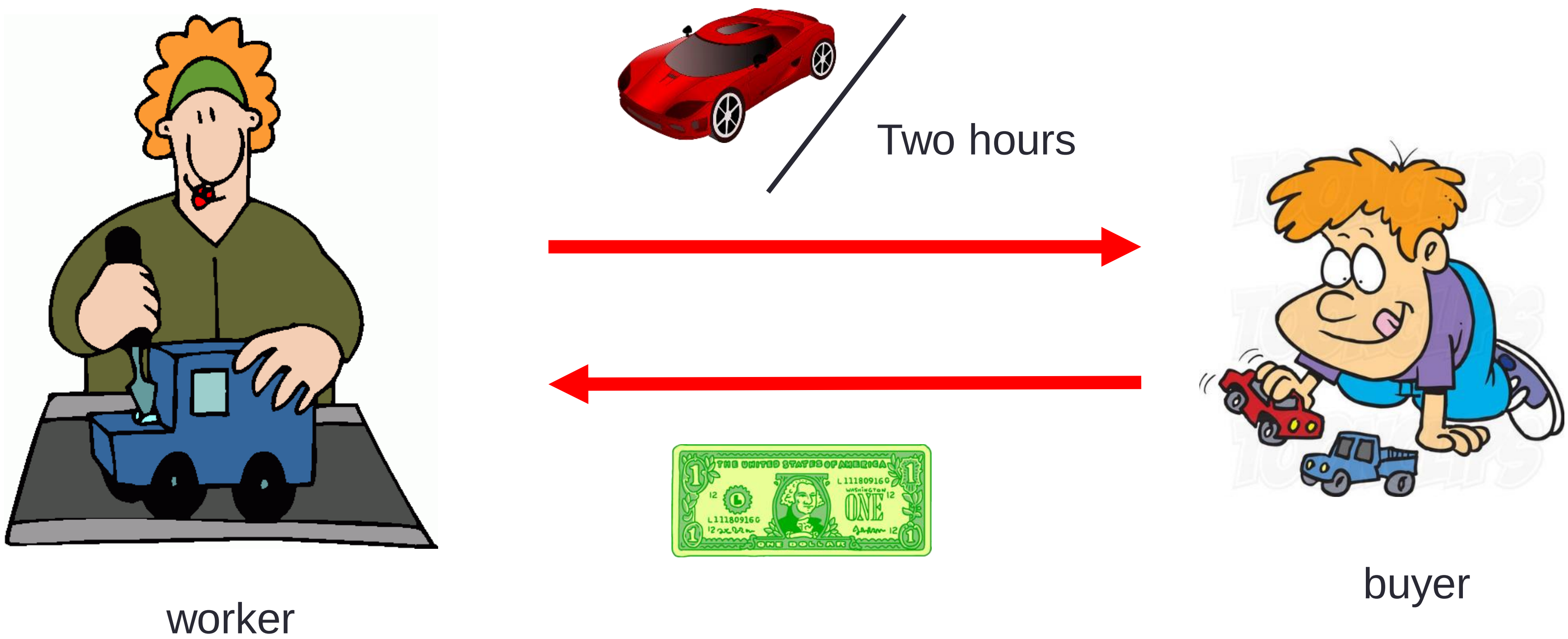
Department of Information Management
National Taiwan University

Adviser: Ling-Chieh Kung

2016/6/21

Introduction

- What will come to your mind when thinking of job allocation
- Consider a **job** of making toy cars



Introduction

- Factory owner may wonder how to assign jobs

Factory A

order 100 toy cars

time (workload)?
money (benefit)?

buyer

Company owner

?

?

?

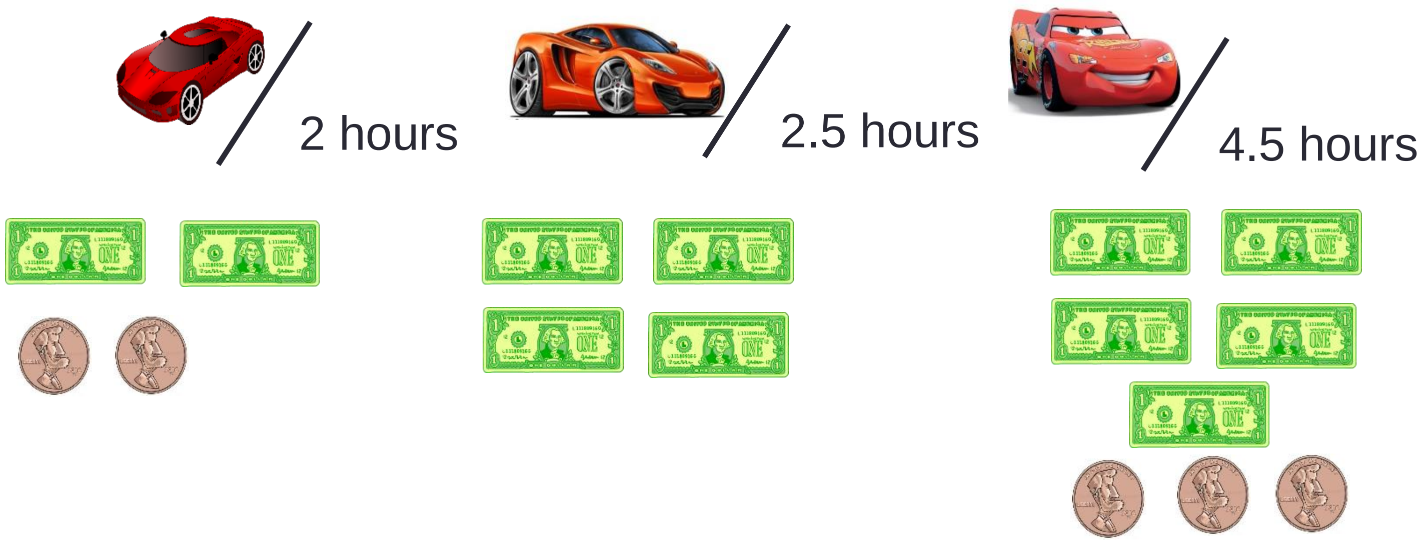
Factory B

Factory C

Everybody wants to earn more money!!!

Introduction

- More precisely, there may be different **workloads** and **benefits** of completing each job



- Also, a factory has limited capacity

Other examples

- “The Big Issue Taiwan” hires the homeless to sell magazines.
- “The Sock Mob Homeless Volunteer Network" in London hires and trains homeless people to become tour guides for London city.
- Local captions in the Kaohsiung harbor.

Introduction

- Jobs are valuable **resources**.
- We try to assign **jobs** to “**machines**.”
  - Not just consider the overall **profitability** but also **fairness** among machines.
  - Machines have **limited capacity** (it will prefer to be assigned more jobs as long as it has enough capacity).
  - All jobs **cannot be spilt** to be assigned.
- The objective function in our problem is set to **maximize the minimum benefit** generated by a machine.
  - And check whether that sacrifice efficiency too much.

Outline

- **Problem Description and Formulation**
- Analysis
- Numerical Study
- Conclusions

Model - Setting and Assumptions

- *I:* the set of machines
- *J:* the set of jobs
- *n*: the number of jobs (in set *J*)
- *m*: the number of machines (in set *I*)
- K: the capacity of machine *i* ($K > 0$)
- c_j: the finite workload of job *j* ($c_j > 0$)
- b_j: the finite benefit of job *j* ($b_j > 0$)

$$x_{ij} = \begin{cases} 1 & \text{if job } j \text{ is assigned to machine } i \\ 0 & \text{otherwise} \end{cases} \quad, i \in I, j \in J,$$

Model - Setting and Assumptions

- The fairness problem

$$\max \quad \min_{i \in I} \left\{ \sum_{j \in J} b_j x_{ij} \right\}$$

maximize the minimum total benefit among the machines

$$\text{s.t.} \quad \sum_{j \in J} c_j x_{ij} \leq K \quad \forall i \in I$$

capacity constraint

$$\sum_{i \in I} x_{ij} \leq 1 \quad \forall j \in J \qquad (3.1)$$

a job can only be done once

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J.$$

all jobs cannot be split (integrality)

Model - Setting and Assumptions

- Beside of the fairness, it is still important to ask what degree efficiency is sacrificed when we optimize fairness.
- Thus, we change the objective function to become a efficiency problem.

$$\max \sum_{i \in I} \sum_{j \in J} b_j x_{ij}$$

Model – NP-hardness

- **Theorem 1.** *The job allocation problem in (3.1) is NP-hard.*
  - Consider a special case with only two uncapacitated (i.e., $K \geq \sum_{j \in J} c_j$) machines.
  - The Partition problem reduces to this special case.
  - **Partition problem** is NP-hard.

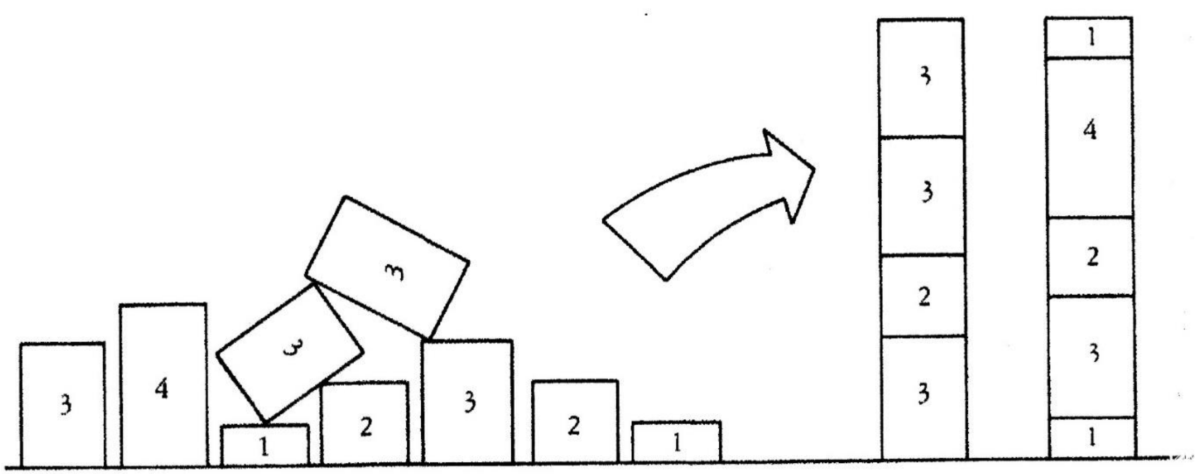
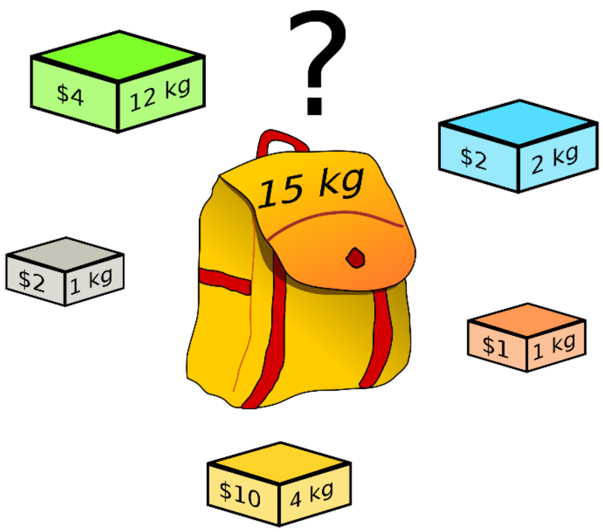


Figure 30.1 An example of the partition problem

Model – NP-hardness

- **Theorem 2.** *The job allocation problem in (3.1) with* $m = 1$ *is NP-hard.*
  - Our problem with one capacitated agent is exactly a **knapsack problem.**



A heuristic algorithm

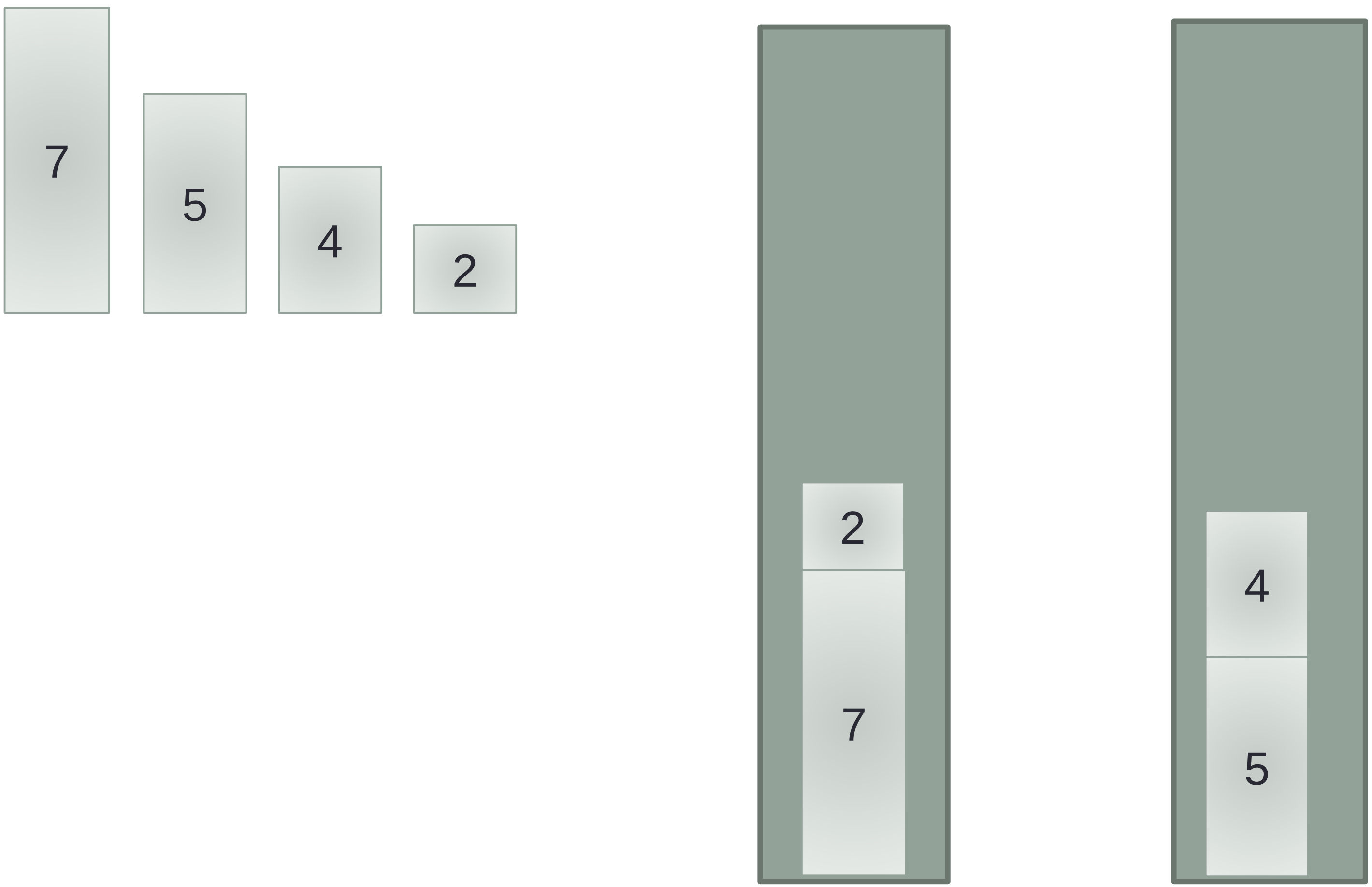
- We do not want to find an optimal solution.
- We will design a **heuristic algorithm** that:
  - Can find a near-optimal feasible solution.
  - Is fast.
- A heuristic algorithm is really applicable for large-scale problems in practice.
- What heuristic algorithm will you design?

Literature Review

- Graham (1966, 1969)
  - Minimum makespan problem for multiple identical machines
- Deuermeyer (1982)
  - Maximize the shortest completion time.

Literature Review

- Suppose there are four jobs and two machines

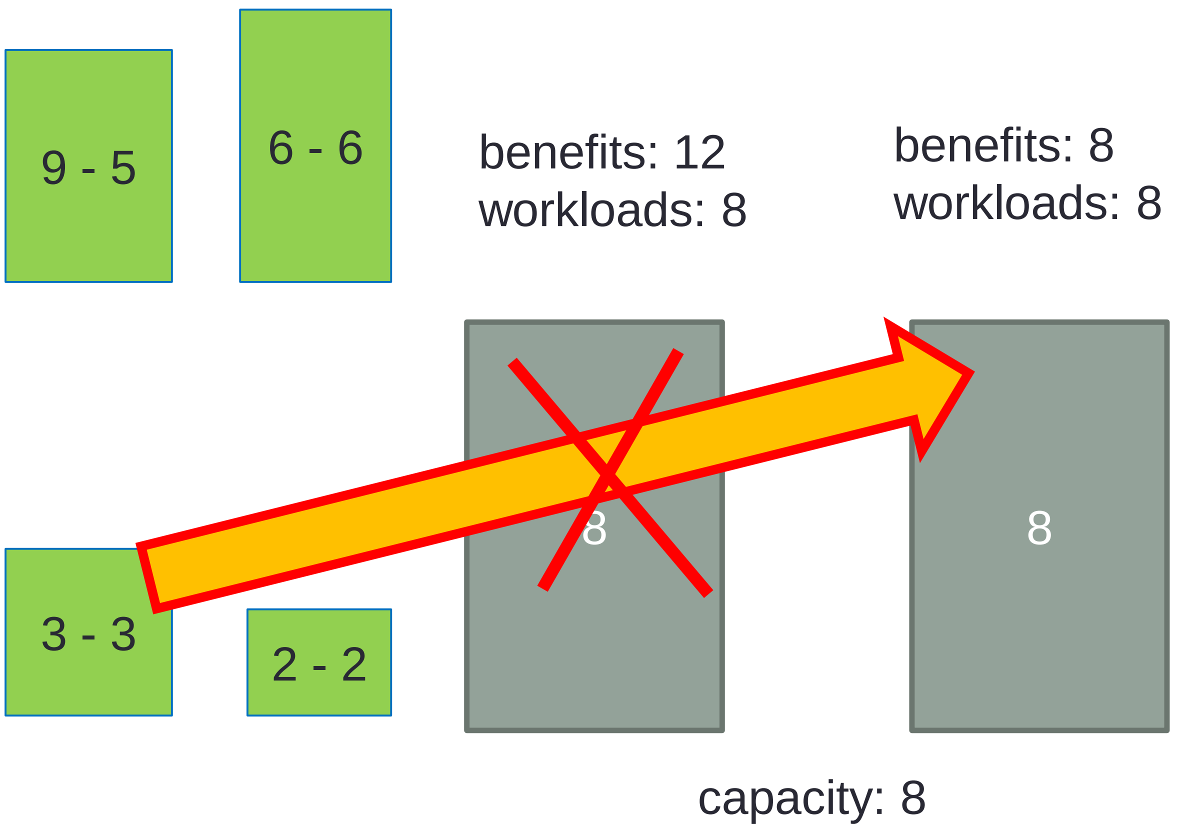


The CHBF algorithm

- We propose the algorithm based on LPT (longest time first) rule (denoted as capacitated highest-benefit job first, CHBF).
  1. Sort all jobs in descending order according to their benefit.
  2. Assign a job to the machine that currently has lowest cumulative benefit and ***enough capacity***. (If a job cannot be assigned, try the machine with the next lowest cumulative benefit.)
  3. Repeat step 2 until the last job has been tried to be assigned.
- The time complexity can be easily analyzed.

The CHBF algorithm

- Suppose there are four jobs and two machines



Research objectives

- Designing an algorithm is easy. Showing that it is “good” is hard.
- What is “good?”
  - Average-case performance.
  - Worst-case performance.
- What do we mean by “worst-case performance?”

Research objectives

- We hope our heuristic algorithm can have a **worst-case performance guarantee**.
  - For a given instance, our algorithm finds a solution.
  - We do not know how good it is compared to an optimal solution.
  - We want to show that “it may be bad, but it will not be too bad.”
  - We want to show this while we have no idea where an optimal solution is!
- If we can show it, we have an **approximation algorithm**.

Literature Review

- Graham (1966, 1969)
  - Minimum makespan problem for multiple identical machines
  - Develop a factor-$\frac{4}{3}$ approximation algorithm **(longest processing time first algorithm)**.
- Deuermeyer (1982)
  - Maximize the shortest completion time.
  - The same algorithm is a factor-$\frac{3}{4}$ approximation algorithm.
- Csirik et al. (1992)
  - Go further from Deuermeyer et al. (1982) and use the same method to show that the performance guarantee can be improved to $\frac{3m-1}{4m-2}$

Outline

- Problem Description and Formulation
- **Analysis**
- Numerical Study
- Conclusions

Analysis – CHBF algorithm

- We show the that CHBF has three different worst-case performance guarantees for three different scenarios of **benefit-workload relationship** $\boldsymbol{b_j = hc_j^t}$ for some $h > 0$.
    - Linear: $t = 1$
    - Convex: $t > 1$
    - Concave: $t < 1$
- We denote $\boldsymbol{z}^*$ as the objective value of an optimal solution and $\boldsymbol{z}'$ as that of the CHBF solution.

Analysis – CHBF algorithm (linear)

- **Theorem 3.** *If* $b_j = hc_j$ *for all* $j \in J$ *for some* $h > 0$, *we have* $z' \geq \frac{1}{2} z^*$.
  - The largest value of z^* is hK.
  - We assume that $c_j \leq K$ for all $j \in J$.
  - We only need to prove the case if there is any job failing to be assigned due to limited capacity (otherwise, it has been proved to have a performance guarantee $\frac{3}{4}$ in Deuermeyer et al. (1982))

Analysis – CHBF algorithm (linear)

- Obviously, $z^* \leq hK$ (why?).
- We use contradiction to prove that

$$z' \geq \frac{1}{2}hK.$$

- Suppose that there is a counterexample in which

$$z' < \frac{1}{2}hK.$$

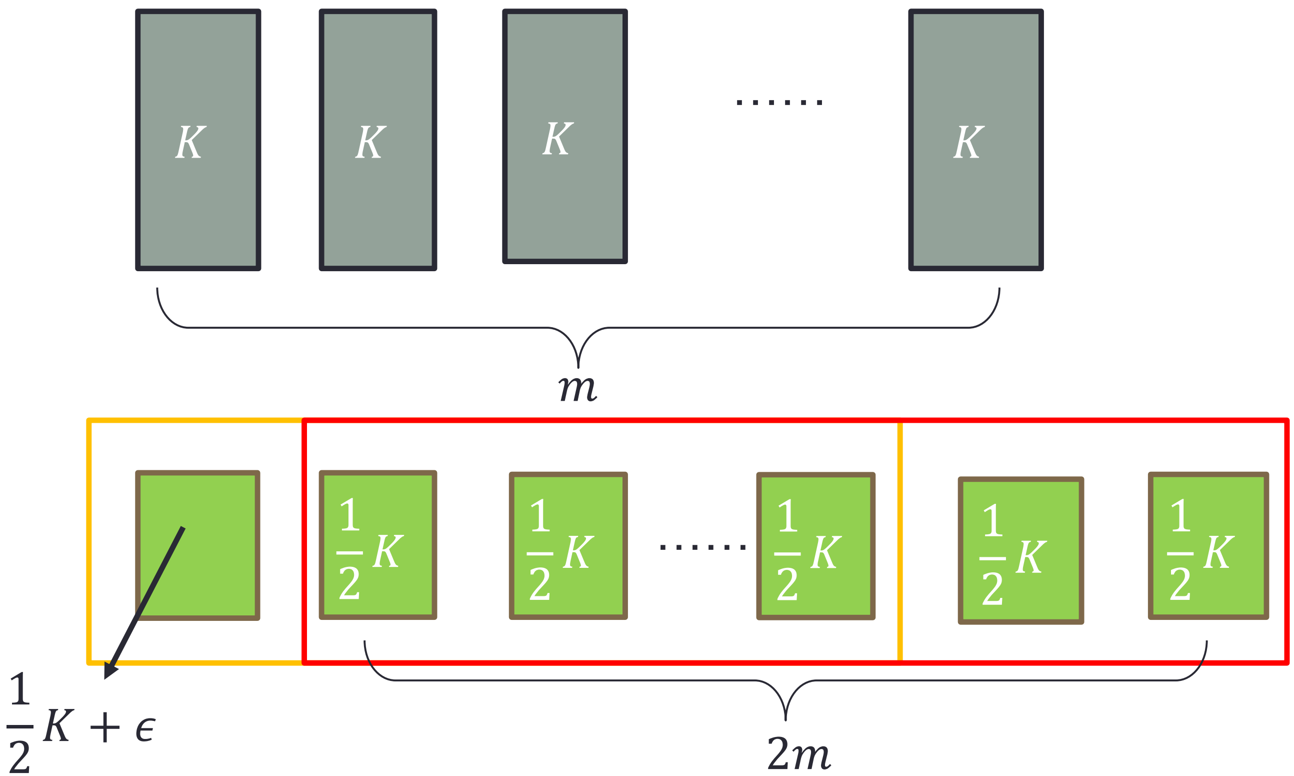
  - Let job l is the first job not assigned during the CHBF process.
  - Let machine i be the one having the lowest cumulative benefit when CHBF tries (but fails) to assign job l.

Analysis – CHBF algorithm (linear)

1. There must be at least one job assigned to machine i.
2. $z_i < \frac{1}{2}hK$, because $z' < \frac{1}{2}hK$.
3. For job l, CHBF implies $b_l < \frac{1}{2}hK$, so we know $c_l < \frac{1}{2}K$.
4. But because $z_i < \frac{1}{2}hK$, the cumulated benefit on machine l is also less than $\frac{1}{2}K$. If $c_l \leq \frac{1}{2}K$, job l can actually be assigned to machine i. Contradiction!
5. We obtain $z' \geq \frac{1}{2}hK \geq \frac{1}{2}z^*$.

Analysis – CHBF algorithm (linear)

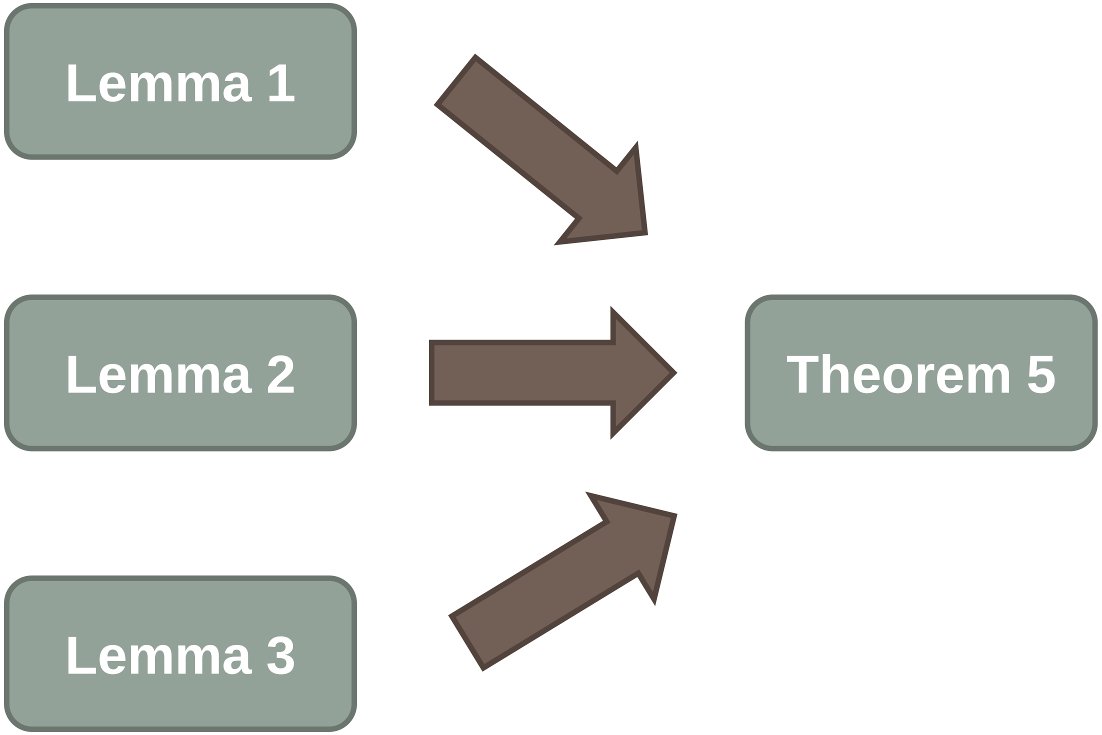
- **Theorem 4.** *The performance guarantee $\frac{1}{2}$ of CHBF in Theorem 3 is tight.*
- Let $h = 1$ and ϵ be a small positive number.



$$\frac{z'}{z^*} = \frac{\frac{1}{2}K + \epsilon}{K} \cong \frac{1}{2}$$

$$n = 2m + 1$$

Analysis – CHBF algorithm (convex)



Analysis – CHBF algorithm (convex)

- Let $c_m = \beta K$ for some $\beta \in (0,1]$.
- **Lemma 1.** *If* $b_j = hc_j^t$ *for all* $j \in J$ *for some given* $h > 0$ *and* $t > 1$, *we have*

$$hK^t\beta^{t-1} \geq z^*,$$

where $\beta = \frac{c_m}{K}$.

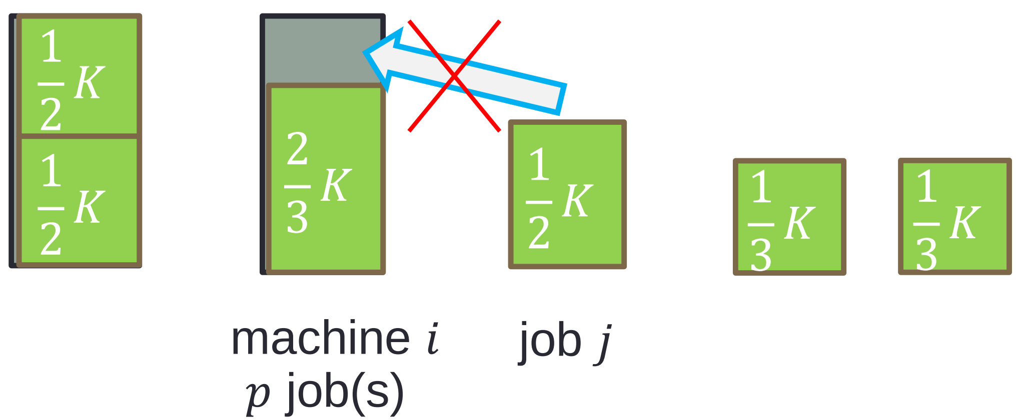
- In each machine contains exactly one job of workload $c_j = K$, we will have hK^t as the objective value.
- To further improve this bound:
  - We could only assign $\left\lfloor \frac{1}{\beta} \right\rfloor$ jobs to the machines.
  - The cumulative benefit is thus no greater than $\left\lfloor \frac{1}{\beta} \right\rfloor h(\beta K)^t < \frac{1}{\beta} h(\beta K)^t = hK^t\beta^{t-1}$.

Analysis – CHBF algorithm (convex)

- **Lemma 2.** *If* $b_j = hc_j^t$ *for all* $j \in J$ *for some given* $h > 0$ *and* $t > 1$*, we have*

$$z' \geq \min\left\{\frac{1}{2^t}, \frac{n-m}{(n-m+1)^t}\right\} hK^t.$$

- We prove this result by considering the moment of our first failure.
- The failure happens when we cannot assign job j to the least cumulative machine i at that moment.
- Let p be the number of jobs that have been assigned to machine i at that moment.

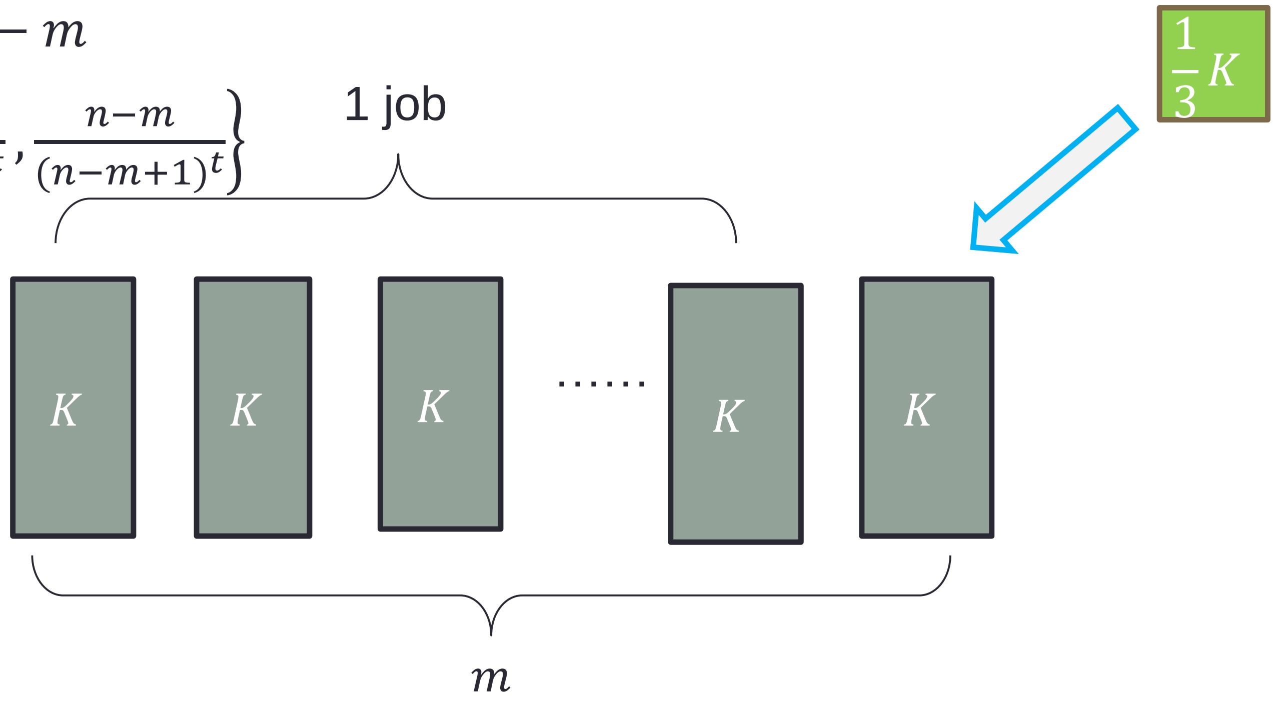


Analysis – CHBF algorithm (convex)

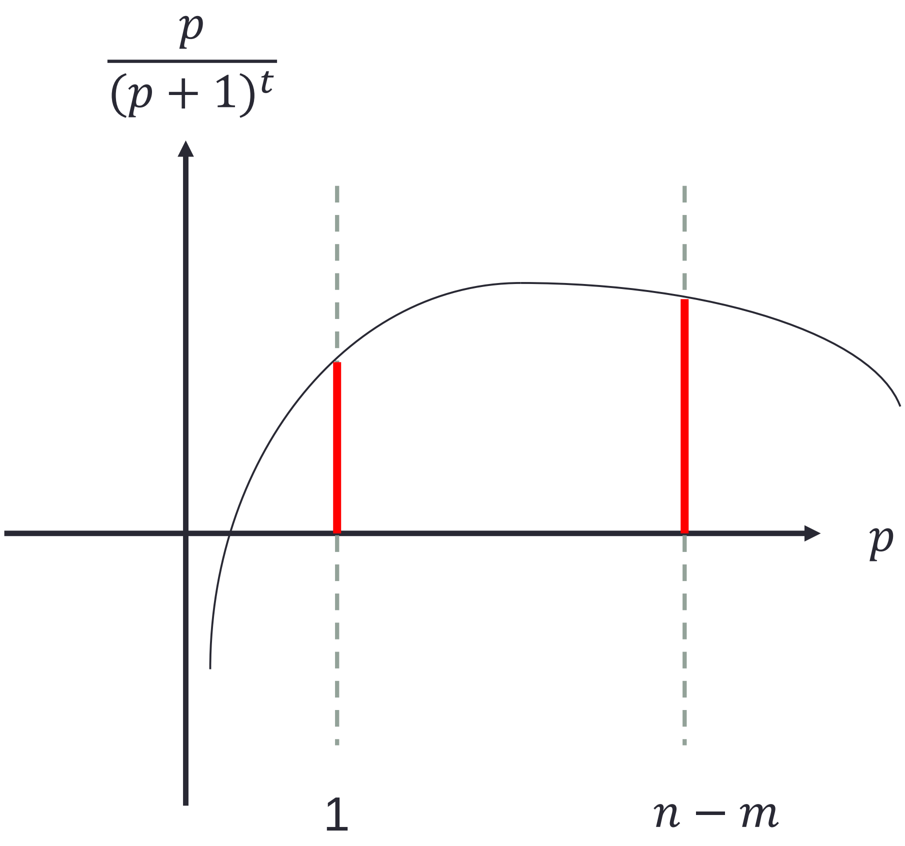
- Let $\alpha = \min\left\{\frac{1}{2^t}, \frac{n-m}{(n-m+1)^t}\right\}$ be the bound to be proved.
- We prove by contradiction by assuming that that $z' < \alpha h K^t$.
  - $z_i < \alpha h K^t$
  - ➢ $b_j \leq \frac{\alpha h K^t}{p}$ (CHBF algorithm)
  - ➢ $c_j \leq \sqrt[t]{\frac{\alpha}{p}} K$ $(b_j = h c_j^t)$
  - ➢ $K_i \geq K - \sqrt[t]{\frac{\alpha}{p}} K$
  - ➢ the cumulative benefit: $ph\left(\frac{K - \sqrt[t]{\frac{\alpha}{p}} K}{p}\right)^t \geq \alpha h K^t$ if and only if $\alpha \leq \frac{p}{(p+1)^t}$

Analysis – CHBF algorithm (convex)

- We would like to obtain an *a priori*, not an *a posteriori* bound.
- The smallest and largest possible numbers of jobs on machine i is 1 and $n-m$.
  - $1 \leq p \leq n-m$
  - $\alpha = \min\left\{\frac{1}{2^t}, \frac{n-m}{(n-m+1)^t}\right\}$

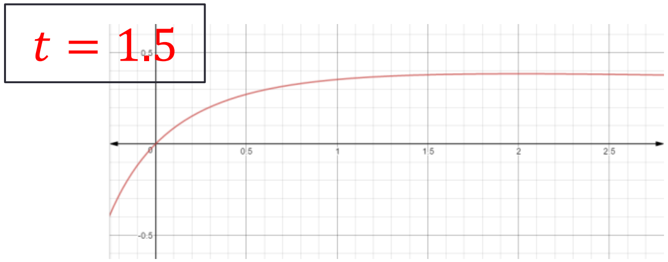


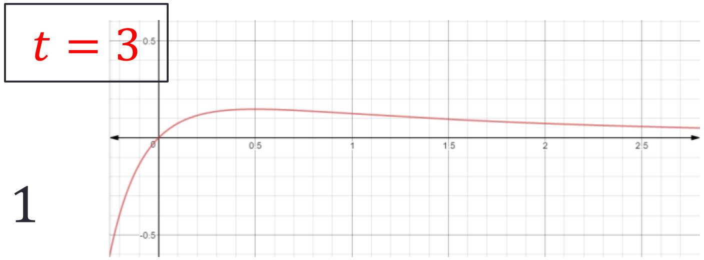
Analysis – CHBF algorithm (convex)



Analysis – CHBF algorithm (convex)

- Let $a = \frac{p}{(p+1)^t}$
- $\frac{da}{dp} = \frac{1}{(p+1)^{t+1}}\ [1 - (t-1)p]$
  - If $t \geq 2$, $\frac{p}{(p+1)^t}$ is decreasing as $p \geq 1$
  - If $t < 2$, $\frac{p}{(p+1)^t}$ is increasing when $p = 1$





- When t approaches 1, eventually $\frac{1}{2^t}$ will be the smaller one (as long as $n - m > 1$) , and this bound converges to $\frac{1}{2}$ as Theorem 3 suggests for the linear benefit-workload relationship (where $t = 1$).

Analysis – CHBF algorithm (convex)

- **Lemma 3.** *If $b_j = hc_j^t$ for all $j \in J$ for some given $h > 0$ and $t > 1$, we have*

$$z' \geq h\beta^t K^t,$$

where $\beta = \frac{c_m}{K}$.

- According to CHBF rule, we assign the first mth jobs one at a time to the m machines. After that, each machines' benefit will be at least $h(\beta K)^t$, which implies that $z' \geq h\beta^t K^t$.

Analysis – CHBF algorithm (convex)

- From **Lemma 1**, we know $hK^t \geq \frac{z^*}{\beta^{t-1}}$.
- From **Lemma 2**, we know $z' \geq \min\left\{\frac{1}{2^t}, \frac{n-m}{(n-m+1)^t}\right\} hK^t$.
- From **Lemma 3**, we know $z' \geq h\beta^t K^t$.

$$\max\left\{\min\left\{\frac{1}{2^t}, \frac{n-m}{(n-m+1)^t}\right\}, \beta^t\right\} hK^t$$

- **Theorem 5.** *If $b_j = hc_j^t$ for all $j \in J$ for some given $h > 0$ and $t > 1$, we have*

$$z' \geq \frac{\max\left\{\min\left\{\frac{1}{2^t}, \frac{n-m}{(n-m+1)^t}\right\}, \beta^t\right\}}{\beta^{t-1}} z^*.$$

- When β is large, β^t would dominate $\min\left\{\frac{1}{2^t}, \frac{n-m}{(n-m+1)^t}\right\}$.
- Note that it is possible for the worst-case performance guarantee to be above $\frac{1}{2}$.

Analysis – CHBF algorithm (concave)

- Let $c_m = \beta K$ for some $\beta \in (0,1]$.
- **Theorem 6.** *If* $b_j = hc_j^t$ *for all* $j \in J$ *for some given* $h > 0$ *and* $t < 1$, *we have*
$$z' \geq \frac{K^{t-1}}{2^t} z^*,$$
where $\beta = \frac{c_m}{K}$.
- We prove by the similar way we did in convex version.
- Notice that the most beneficial way to consume all the capacity of one machine is to use K jobs with unit workload.
  - In each machine contains exactly K jobs of workload 1, we will have hK as the objective value.

Analysis – CHBF algorithm (concave)

- Let $\alpha = \frac{K^{t-1}}{2^t}$ be the bound to be proved.
- Note that $t < 1$
- We prove by contradiction by assuming that that $z' < \alpha hK$
  - $z_i < \alpha hK$
  - ➢ $b_j \leq \alpha hK$ (CHBF algorithm)
  - ➢ $c_j \leq (\alpha K)^{\frac{1}{t}}$ $(b_j = hc_j^t)$
  - ➢ $K_i \geq K - (\alpha K)^{\frac{1}{t}}$
  - ➢ the cumulative benefit: $h(K - (\alpha K)^{\frac{1}{t}})^t$
    - ➢ If it is at least equal to αhK, then $\alpha \leq \frac{K^{t-1}}{2^t}$
    - ➢ Then we obtain $z' \geq \frac{K^{t-1}}{2^t} hK \geq \frac{K^{t-1}}{2^t} z^*$

Outline

- Problem Description and Formulation
- Analysis
- **Numerical Study**
- Conclusions

Numerical study

- To understand how CHBF and MCHBF perform in the problem, we consider the following factors for $c_j \sim U(0{,}50)$:
  - The relationship of job benefits and workloads ($b_j = c_j^t$)
    1. linear: $t = 1$ - scenario L
    2. convex: $t = 2$ - scenario X
    3. concave: $t = \frac{1}{2}$ - scenario A
    4. unrelated: $b_j \sim U(0{,}50)$ - scenario R
  - Machine capacity
    1. unlimitation: $K = \infty$ - scenario N
    2. loose capacity: $K = \left(\frac{\sum_{j \in J} c_j}{m}\right)$ - scenario L
    3. tight capacity: $K = \frac{3}{4}\left(\frac{\sum_{j \in J} c_j}{m}\right)$ - scenario T

Numerical study

- For each of the twelve combinations of scenarios, we generated 100 experiments with several combination of m and n.

benefit-workload
R
L
X
A

capacity
N
L
T

m	n
5	20
5	50
5	500
15	50
15	500
50	500

- For $m = 5$, $n = 20$, we use branch-and-bound algorithm IP solution.
- Due to memory limitation, we find a LP solution instead in other case.

Numerical study

- We denote solution of
  - “fairness” version as w^{IP}, w^{LP}, w^{CHBF}
  - “efficiency” version as z^{IP}, z^{LP}, z^{CHBF}

Numerical study

- CHBF performs better if the capacity is looser

capacity	$\frac{z(x^{CHBF})}{z^{IP}}$	$\frac{w^{CHBF}}{w^{IP}}$
N	1	0.988
L	0.987	0.964
T	0.946	0.912

Table 5.1: Numerical results of number of capacity tightness

Numerical study

- CHBF performs better when $\frac{n}{m}$ becomes bigger.

$\frac{n}{m}$	$\frac{z(x^{CHBF})}{z^{IP}}$	$\frac{w^{CHBF}}{w^{IP}}$
100	0.983	0.980
33.333	0.983	0.977
10	0.981	0.965
4	0.971	0.942
3.333	0.961	0.899

Table 5.2: Numerical results of number of machines and jobs

Numerical study

- For “fairness” version:
  - CHBF performs the best when benefits are linear in workloads.
  - CHBF performs well when benefits are convex in workloads while performing the worst when benefits are concave in workloads.

benefit-workload	$\frac{z(x^{CHBF})}{z^{IP}}$	$\frac{w^{CHBF}}{w^{IP}}$
R	0.992	0.947
L	0.990	0.979
X	0.988	0.976
A	0.936	0.918

Table 5.3: Numerical results of the benefit-workload relationship

Numerical study - summary

- The convexity or concavity of the benefit-workload relationship has an important managerial implication.
  - Convex → production environment is of significant economy of scale
  - Concave → the product is of diminishing marginal benefit for consumers

Outline

- Problem Description and Formulation
- Analysis
- Numerical Study
- **Conclusions**

Conclusions

- We consider a job allocation problem with fairness.
  - We modify a classic algorithm to develop our own algorithm for our problem.
  - We prove the performance guarantees of our algorithm when the relationship between benefits and workloads is linear, convex, and concave.
  - The CHBF algorithm is more appropriate when production environment exhibits **significant economy of scale.**
- Further investigation
  - Prove worst-case performance guarantee of our algorithm under general problem or under some conditions.
  - Modify our algorithm by the ideas coming up with when we prove the bounds.

THANK YOU