Multi-Resource Allocation: Fairness-Efficiency Tradeoffs in a Unifying Framework

Fairness & Efficiency Problem in Multi-Resource Allocation

- The allocation of multi-resources: a simple example

  Total resources: 9 CPUs & 18GB of RAM

  Job type A: 1 CPU & 4GB of RAM per job

  Job type B: 3 CPUs & 1GB of RAM per job.

  Available allocations:

Type A:	Type B:	Leftover CPU:	Leftover RAM
4.5 jobs	0 jobs	4.5 CPUs	0
0 jobs	3 jobs	0 CPUs	15GB
3 jobs	2 jobs	0 CPUs	4GB
4.25 jobs	1 jobs	1.75 CPUs	0GB

Fairness & Efficiency Problem in Multi-Resource Allocation

- The multi-resource allocation satisfies resource constraints:

$$x_1 + 3x_2 \leq 9;$$
$$4x_1 + x_2 \leq 18.$$

(x_1, x_2) are the processed jobs for type A and type B respectively.

- Problems?
  1. What is the fairness measure of an allocation?
  2. What is the efficiency measure of an allocation?
  3. How do we tune the emphasis on fairness and efficiency in the allocation?

The case in single resource allocation

- Suppose job type A and job type B require only one resource, network bandwidth.

  Total resource: 100Mbps

  Job type A: 10Mbps per job; Job type B: 5Mbps.

  A fairness function is proposed for the single resource allocation:

$$f_{\beta,\lambda}(\vec{x}) = sign(1-\beta)[\sum_{i=1}^{n}\left(\frac{x_i}{\sum_{j=1}^{n} x_j}\right)^{1-\beta}]^{\frac{1}{\beta}}(\sum_{i=1}^{n} x_i)^{\lambda}$$

The allocation problem is:

$$\begin{aligned} \max \quad & f_{\beta,\lambda}(\vec{x}) \\ \text{subject to} \quad & x_1 + x_2 \leq 100. \end{aligned}$$

The case in single resource allocation

- The family of fairness functions for a single resource,

$$f_{\beta,\lambda}(\vec{x}) = sign(1-\beta)[\sum_{i=1}^{n}\left(\frac{x_i}{\sum_{j=1}^{n} x_j}\right)^{1-\beta}]^{\frac{1}{\beta}}(\sum_{i=1}^{n} x_i)^{\lambda}$$

x_i is the number of resources allocated to type-i jobs.

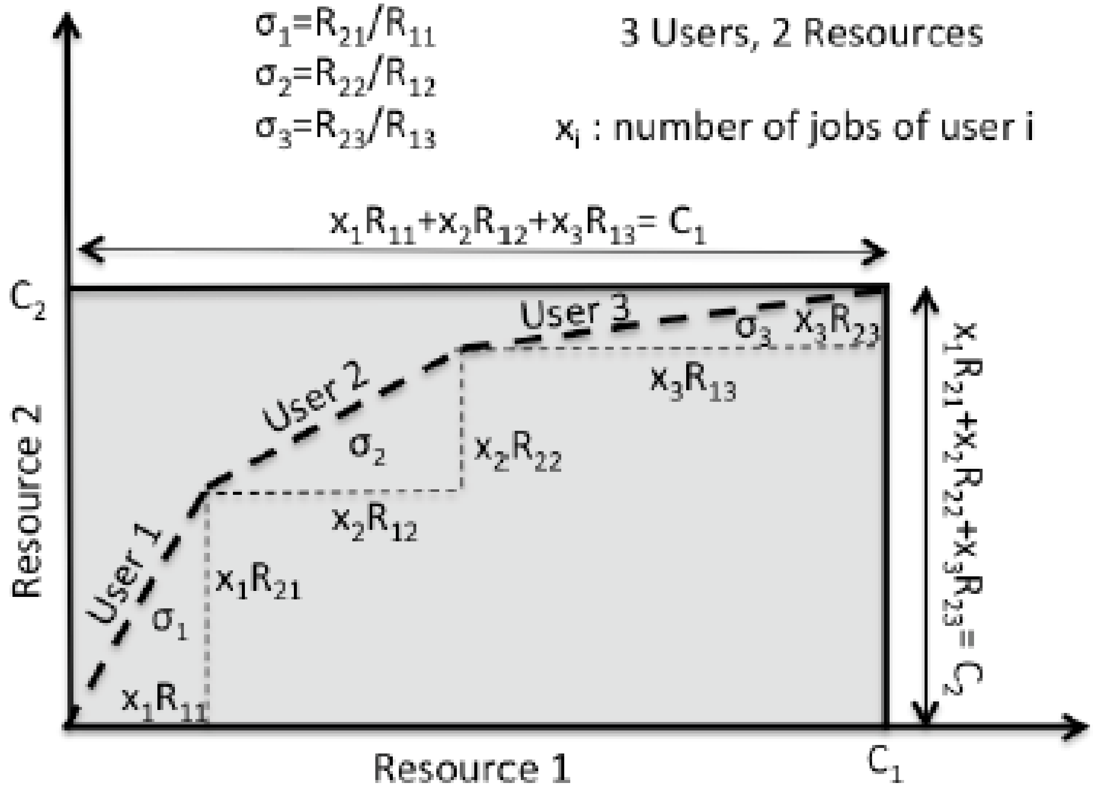
(a) Larger beta indicates more emphasis on fairness.

(b) Larger lambda indicates more emphasis on efficiency.

Multi-resource allocation vs. single resource allocation

Single resource allocation	Multi resource allocation
Resource allocated to one job type is a scalar.	Resource allocated to one job type is a vector.
The most efficient allocation use the entire resource.	Not all resources can be entirely used.

User Heterogeneity in multi-resource allocation



User heterogeneity (τ): variance of δ_i.

Define multi-resource fairness function

Two scalarization methods:

1. Fairness on Dominant Shares (FDS):

User j's dominant share: $max_i \left\{\frac{R_{ij}}{C_i}\right\} x_j = \mu_j x_j$

Use dominant share replacing allocations of single resource in single resource fairness function,

$$f_{\beta,\lambda}(\vec{x}) = sign(1-\beta)[\sum_{i=1}^{n}\left(\frac{\mu_i x_i}{\sum_{j=1}^{n}\mu_j x_j}\right)^{1-\beta}]^{\frac{1}{\beta}}(\sum_{i=1}^{n}\mu_i x_i)^{\lambda}$$

Define multi-resource fairness function

Two scalarization methods:

2. Generalized Fairness on Jobs (GFJ):

Use the number of type i's jobs processed replacing the number of allocated resources in the fairness function,

$$f_{\beta,\lambda}(\vec{x}) = sign(1-\beta)[\textstyle\sum_{i=1}^{n}\left(\frac{x_i}{\sum_{j=1}^{n} x_j}\right)^{1-\beta}]^{\frac{1}{\beta}}(\sum_{i=1}^{n} x_i)^{\lambda}$$

3. As $\beta \to \infty$ and $\lambda = \frac{1-\beta}{\beta}$, the fairness function of FDS approaches,

$$\min\ \{\mu_1 x_1, \mu_2 x_2, \dots, \mu_n x_n\}$$

The FDS becomes max-min fairness on the dominant share, which is called Dominant Resource Fairness (DRF).

An example of FDS & GFJ

Total resources: 9 CPUs & 18 GB of RAM per job

Job type A: 1 CPU & 4GB of RAM per job

Job type B: 3 CPUs & 1GB of RAM per job

Dominant share of job type A is $\frac{2}{9}x_1$, dominant share of job type B is $\frac{1}{3}x_2$.

The fairness function for FDS is:

$$f = sign(1-\beta)\,[\frac{(\frac{2}{9}x_1)^{1-\beta}+(\frac{1}{3}x_2)^{1-\beta}}{(\frac{2}{9}x_1+\frac{1}{3}x_2)^{1-\beta}}]^{\frac{1}{\beta}}(\frac{2}{9}x_1+\frac{1}{3}x_2)^{\lambda}$$

The fairness function for GFJ is:

$$f = sign(1-\beta)\,[\frac{{x_1}^{1-\beta}+{x_2}^{1-\beta}}{(x_1+x_2)^{1-\beta}}]^{\frac{1}{\beta}}(x_1+x_2)^{\lambda}$$

An example of FDS & GFJ

- The fairness function of DRF (Dominant Resource Fairness) is:

$$f = \min\{\frac{2}{9}x_1, \frac{1}{3}x_2\}$$

- FDS (including DRF) and GFJ then can be expressed as:

$$\begin{array}{rl} \text{max} & f_{\beta,\lambda}(\vec{x}) \\ \text{subject to} & x_1 + 3x_2 \leq 9; \\ & 4x_1 + x_2 \leq 18. \end{array}$$

Illustrative example & Outcomes

Total resources: 9 CPUs & 18 GB of RAM per job

Job type A: 1 CPU & 4GB of RAM per job

Job type B: 3 CPUs & 1GB of RAM per job

- Fairness measure:

Use DRF as the benchmark fairness:

(x_1, x_2): the optimal jobs processed obtained from DRF.

(x_1', x_2'): the optimal jobs obtained from FDS or GFJ.

Percent fairness of FDS or GFJ is:

$$\frac{\min\{\mu_1 x_1', \mu_2 x_2'\}}{\min\{\mu_1 x_1, \mu_2 x_2\}}$$

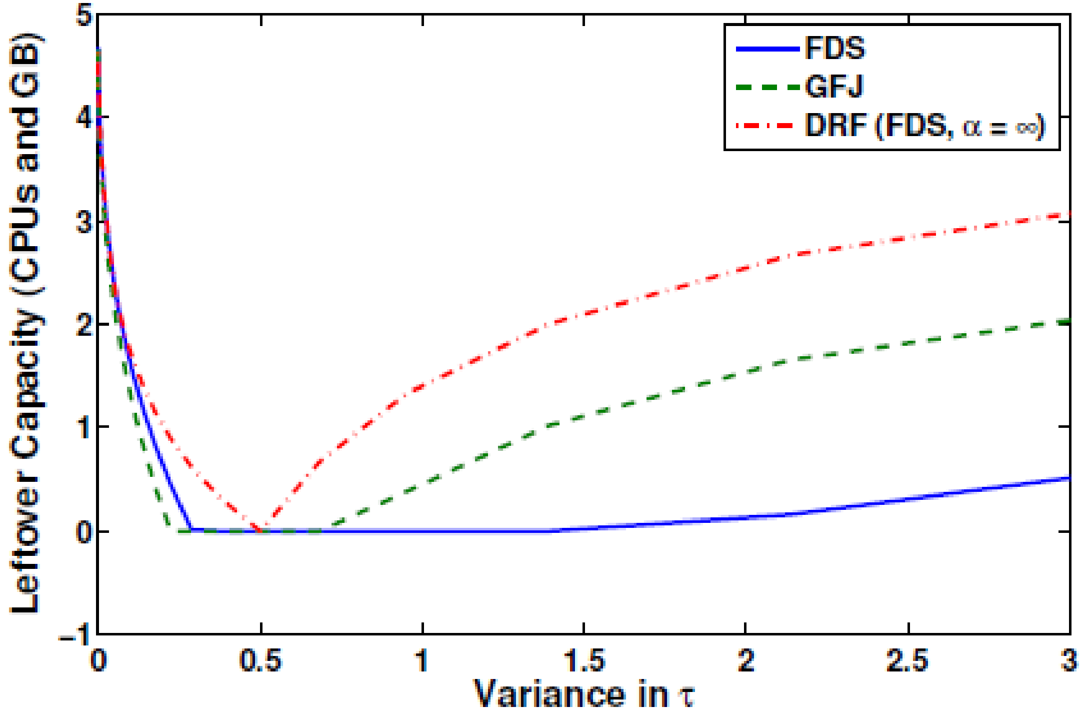
- Efficiency measure:

1. Percent efficiency: $\frac{Total\ jobs\ allocated}{Maximum\ No.of\ jobs\ that\ can\ be\ processed}$
2. The leftover capacity.

Illustrative example & Outcomes

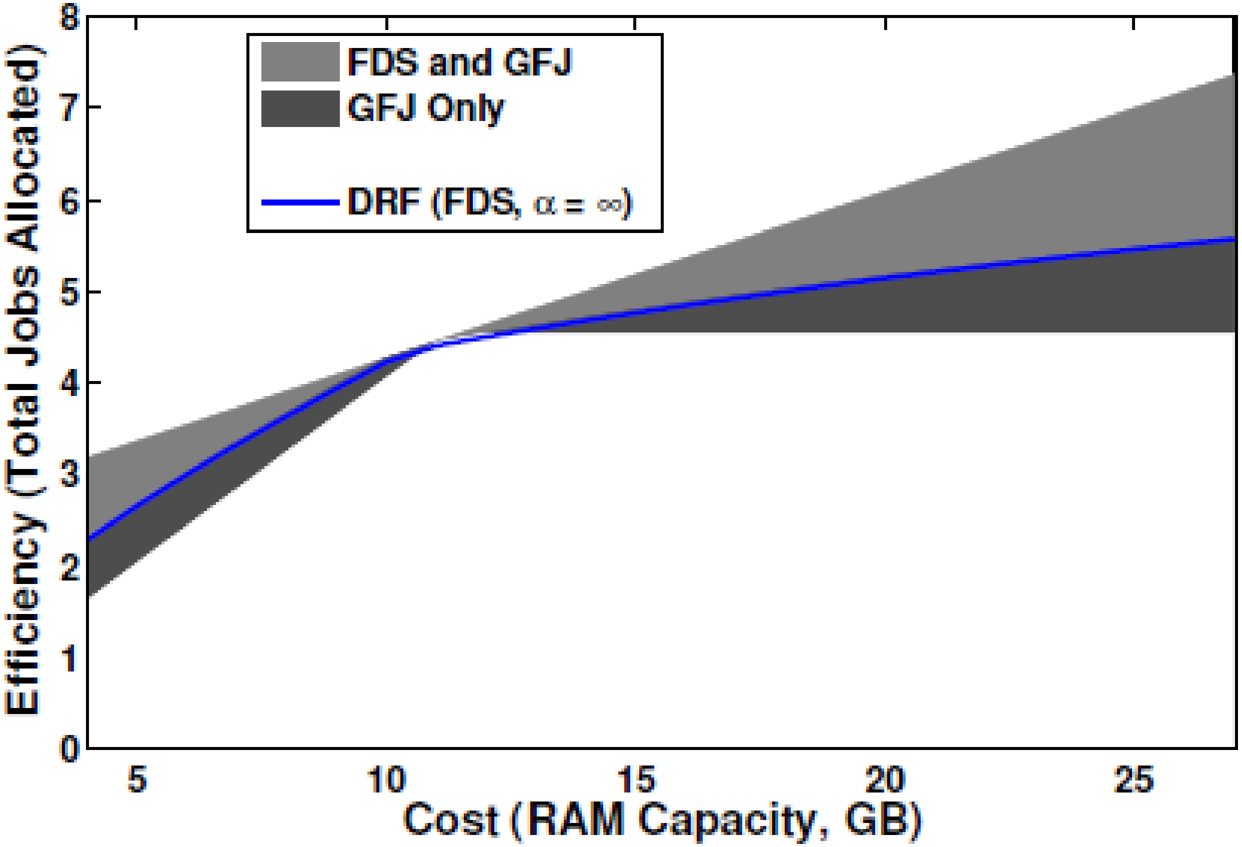
- Efficiency

1. User heterogeneity's effect on achieved efficiency: (Changing the RAM requirement of one type B job from 1GB to 13GB. $\beta = 2, \lambda = -0.5$.



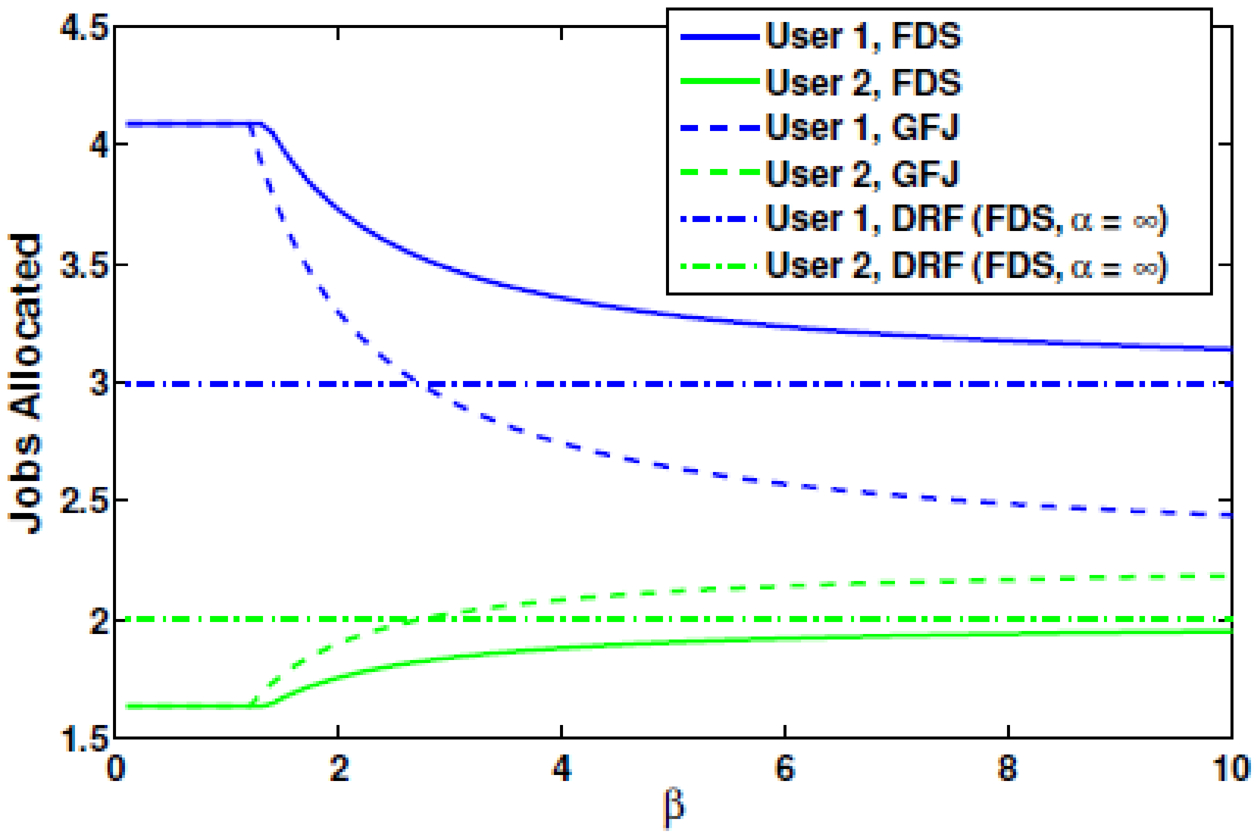
Illustrative example & Outcomes

2. Resource capacity's impact on the efficiency



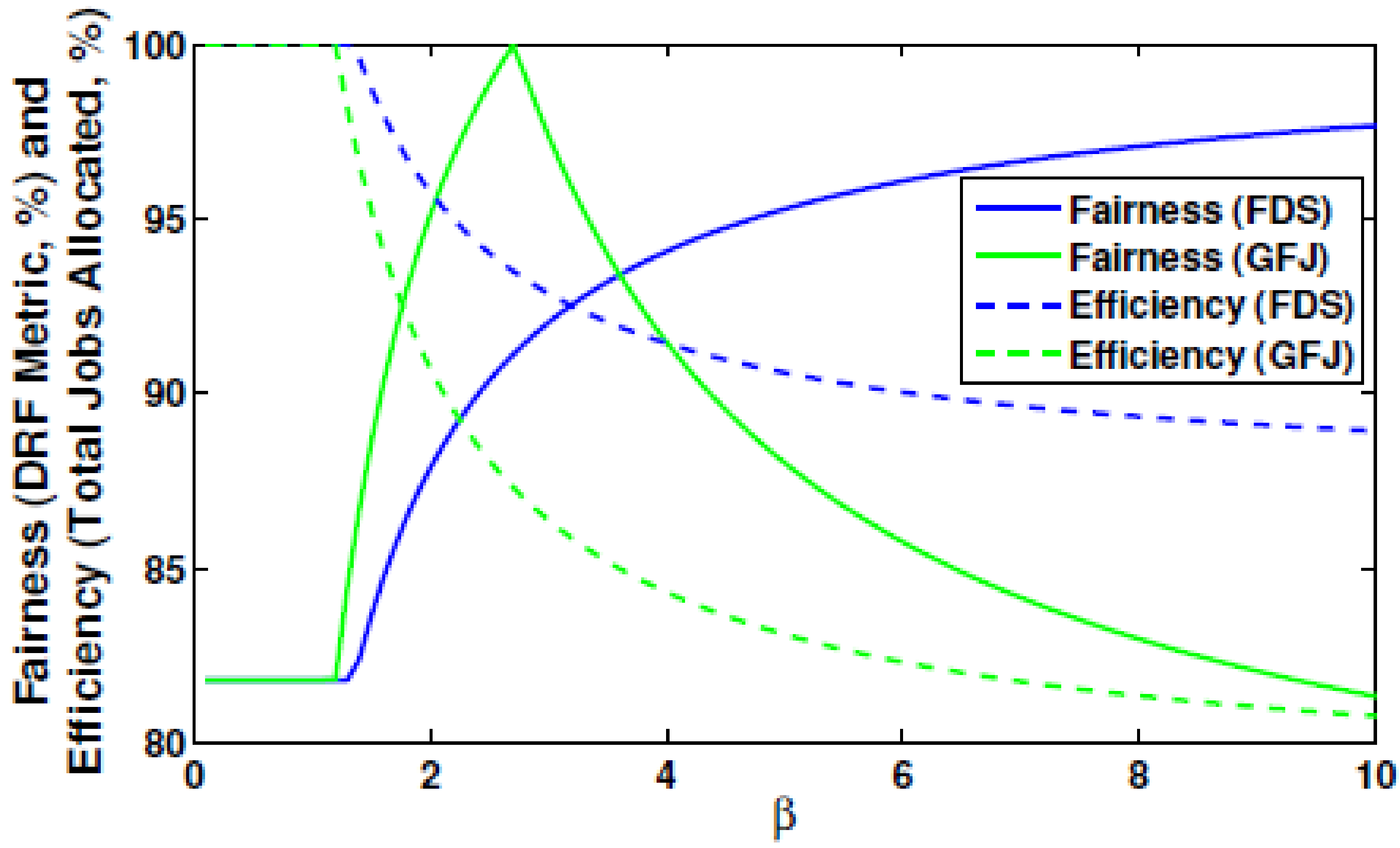
Illustrative example & Outcomes

- Fairness-efficiency tradeoffs: $(\lambda = \frac{1-\beta}{\beta})$



Illustrative example & Outcomes

- Fairness-efficiency tradeoffs: $(\lambda = \frac{1-\beta}{\beta})$



Comments

- This paper studies the concept of fairness and efficiency in the context of multi-resource allocations. This is new compared to single resource allocation.
- The paper extends the solutions in single resource to the multi-resource.