Equitable Multi-Objective Optimization

Krzysztof Rzadca
Polish-Japanese Institute of Information Technology

based on joint work with
Denis Trystram,
Adam Wierzbicki

In many situations, fairness is not equal division

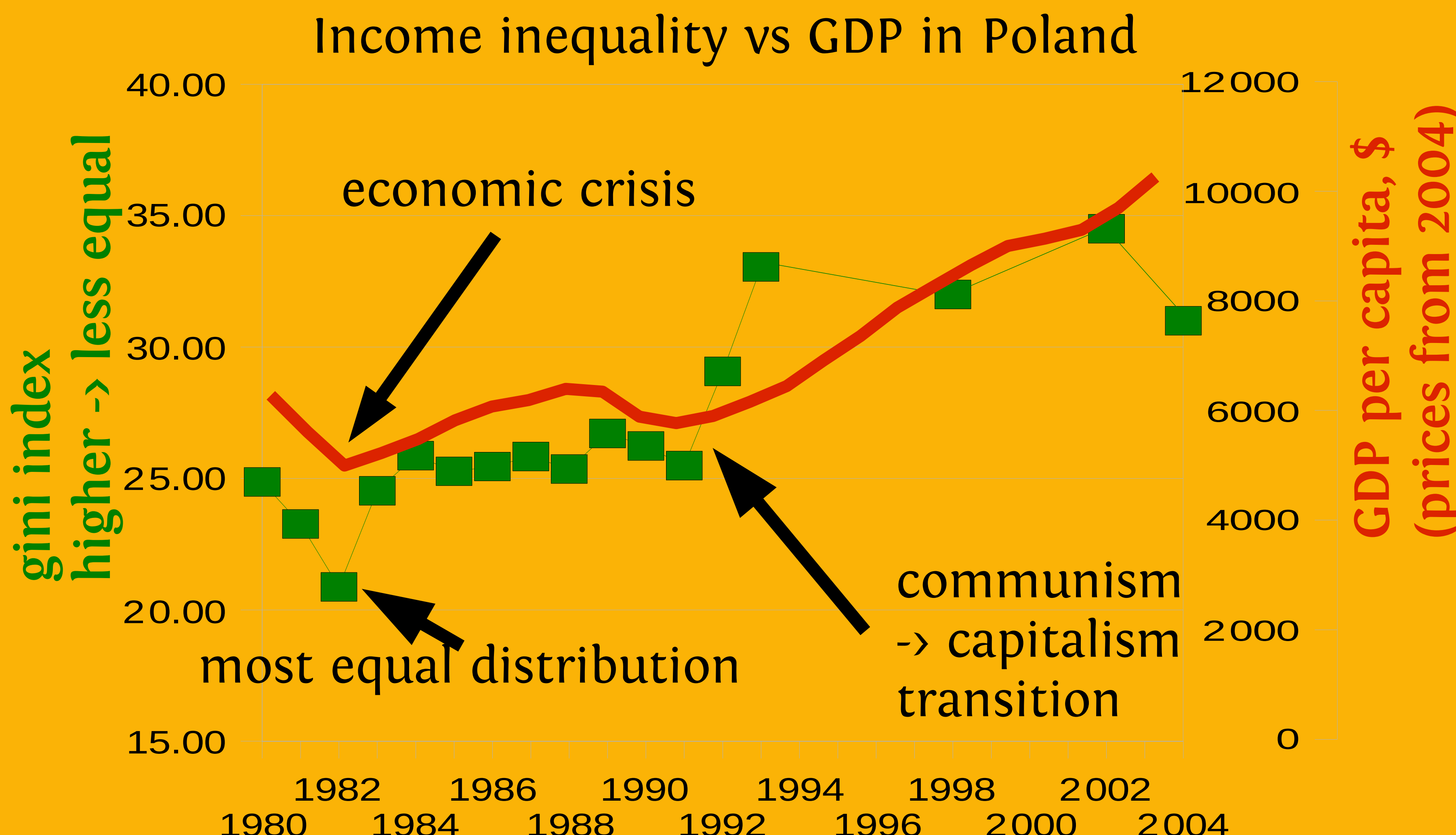
"Cake" gets bigger

oranges [Fisher, 1981]

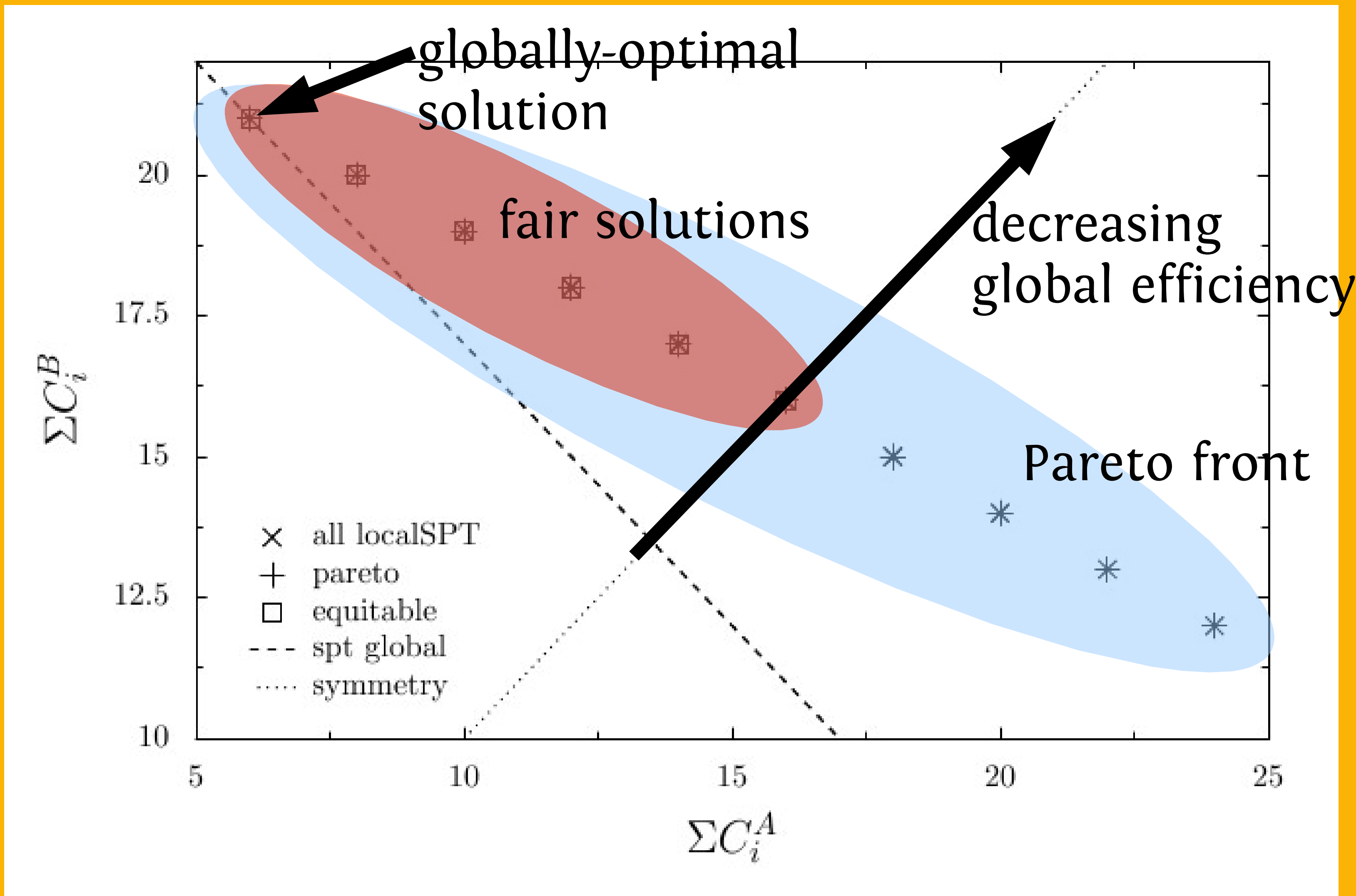
two halfs

peel and juice

However, sometimes after an "equal" division the "cake" gets smaller.



Idea: extend multi-objective optimization to include fairness.



2 criteria minimalization

Table of Contents

Multi-objective optimization problem with multiple agents that must be treated fairly.

Centralized decision maker
Agents' utilities depend on each other (shared system)
Payoffs not transferable

$$\min\ (f_1(x), f_2(x), \ldots., f_n(x))$$

$f_1(x)$ – first agent's objective

$f_2(x)$ – second agent's objective

$f_n(x)$ – last agent's objective

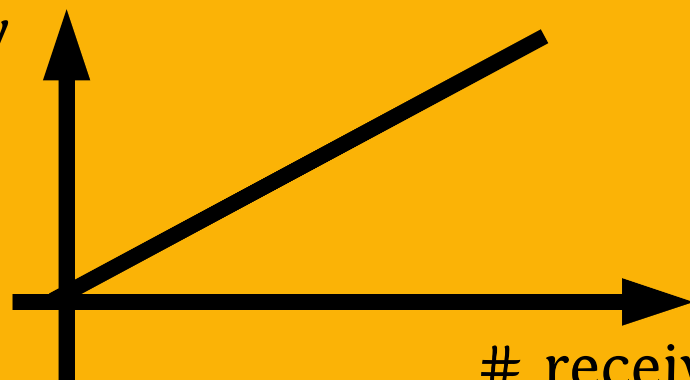
Examples:

- multi-agent scheduling [Agnetis et al, 2004]
- robust shortest path [Perny et al, 2006]
- budget redistribution [Kostreva et al, 2004]
- facility location [Marsch, 1994]

(Values of) agents' utilities have to be directly comparable.

fair division

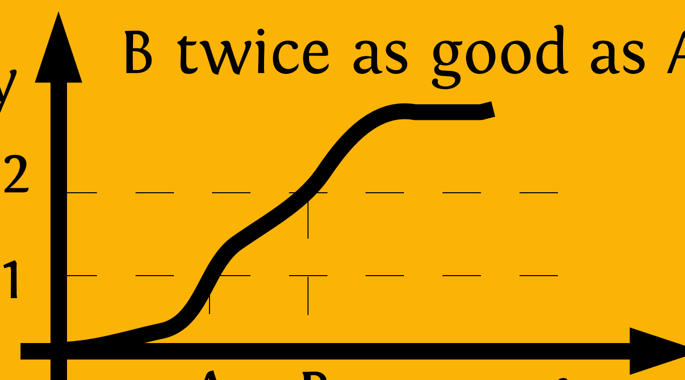
finite goods to distribute
each agent: max her share



•facility location [Marsch, 1994]
•budget redistribution [Kostreva et al, 2004]

cardinal utilities

utility functions express strength of preferences

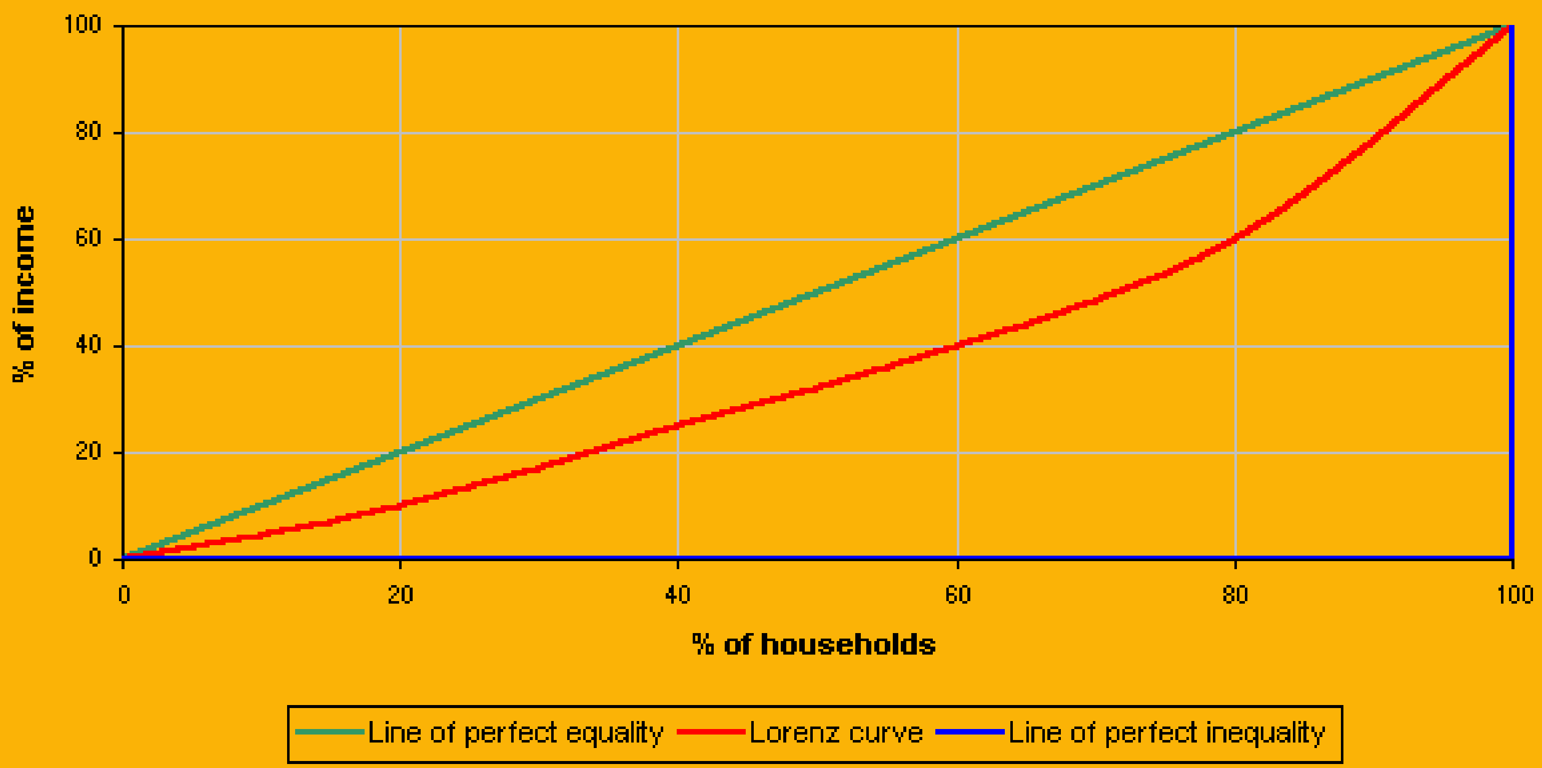


•robust shortest path [Perny et al, 2006]
•multi-agent scheduling [Agnetis et al, 2004]

Economics and Social Science have similar problem: How to measure fairness of income distribution

Sort outcomes from worst to best;
measure cumulative outcome of n worst outcomes

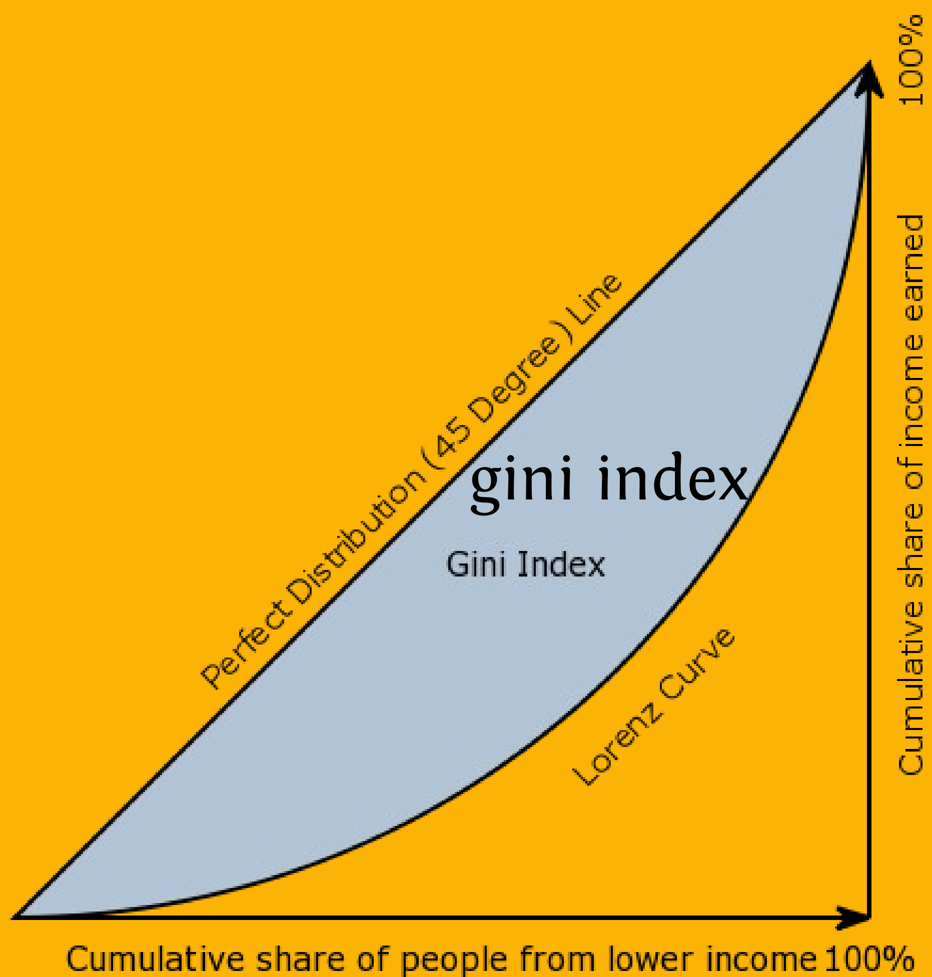
Lorentz Curve [Lorentz 1905]



[wikipedia]

Gini index expresses the unfairness in Lorentz curve.

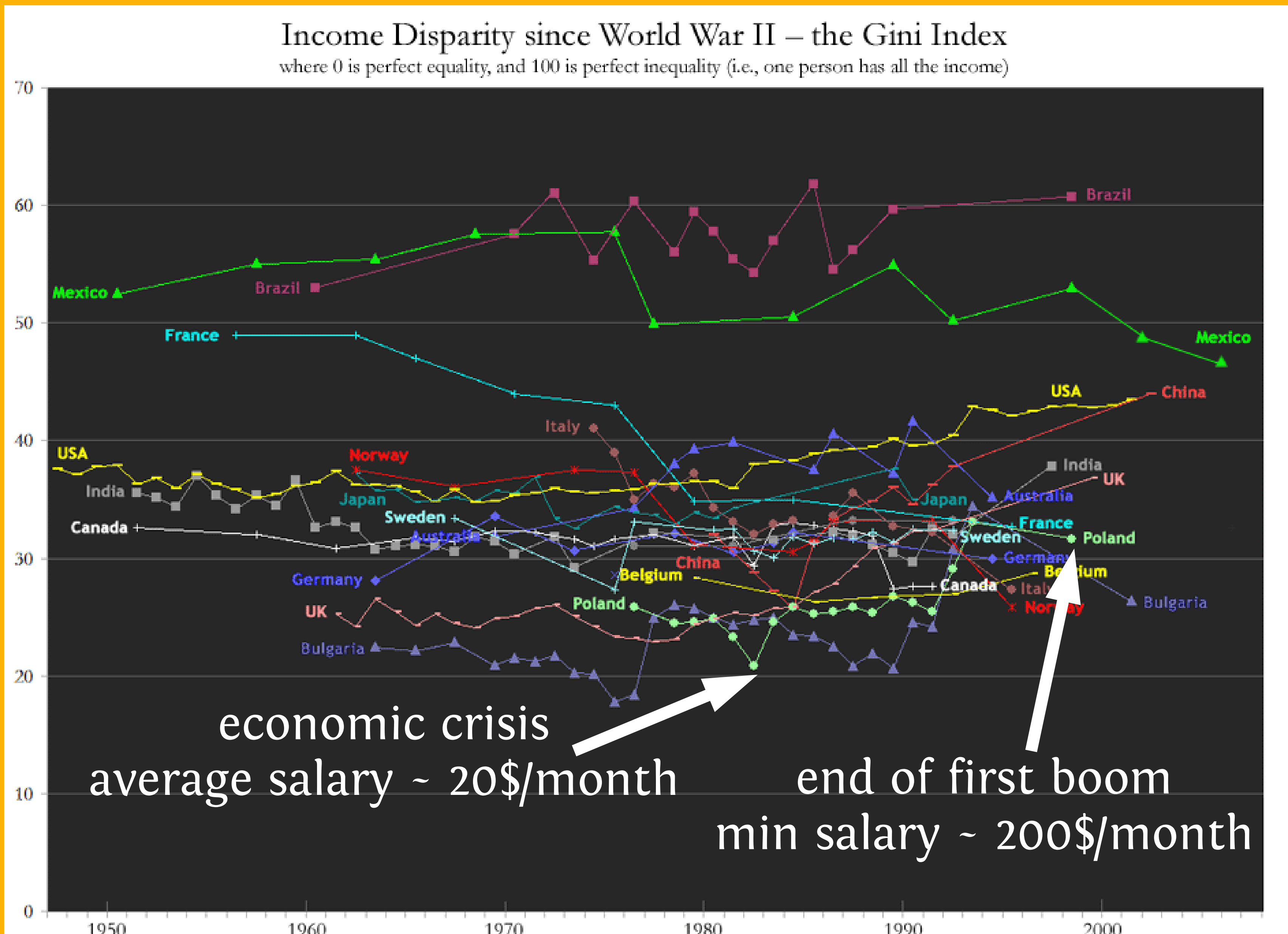
[Gini, 1912]



[wikipedia]

Gini Index can favor inefficient solutions

gini index: lower -> more equal



[wikipedia]

Idea: compare equity of solutions that have the same global efficiency

economics: Pigou-Dalton principle of transfers
mathematics: majorization

multi-agent minimization

y = [y1, y2, ..., y',...., y'',..., yn]
y' > y'' (y' is worse)

$$\mathbf{y} + \boldsymbol{\varepsilon}'' - \boldsymbol{\varepsilon}' \text{ preferred to } \mathbf{y}$$

$\boldsymbol{\varepsilon}'$ = [0,…,0,…,ε,…,0,…0]
$\boldsymbol{\varepsilon}''$= [0,…,0,…,0,…,ε,…0]

e.g. [9,6] is preferred to [10, 5]

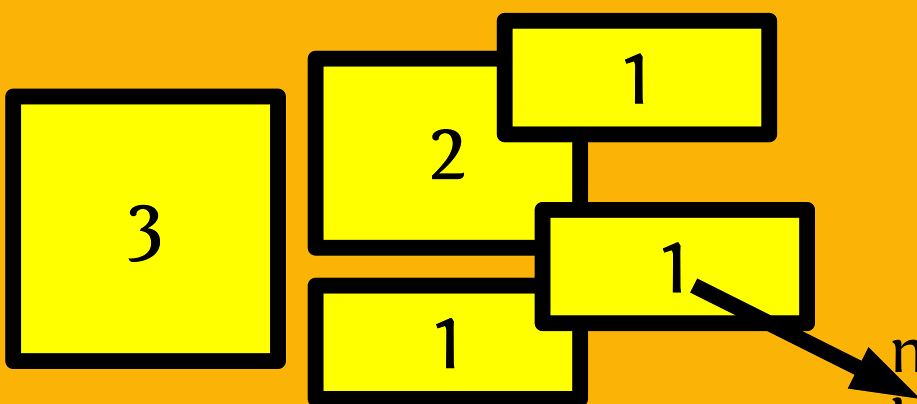
both [9,6] and [10,5] have the same global efficiency **15**

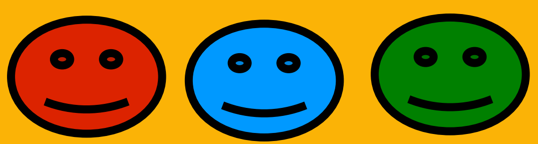
Axiomatic theory of fairness: properties of a fair division

work to be done

lazy workers

(minimizing their work)

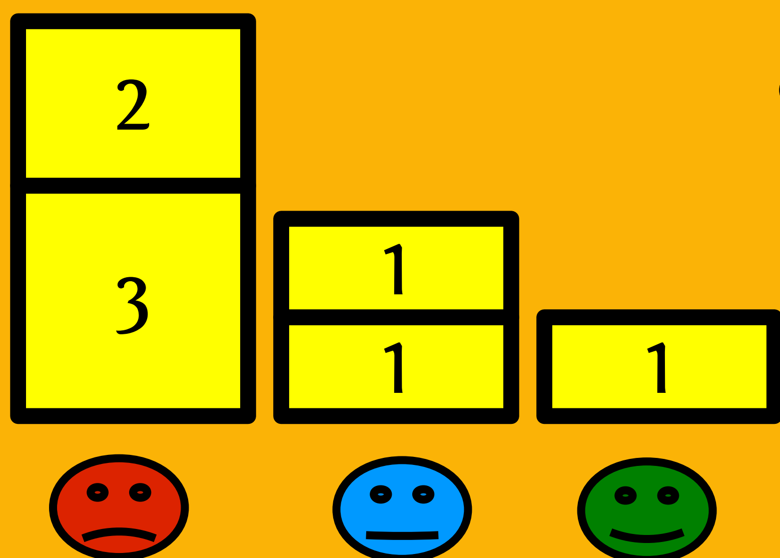




axioms:

- symmetry
- optimality
- principle of transfers

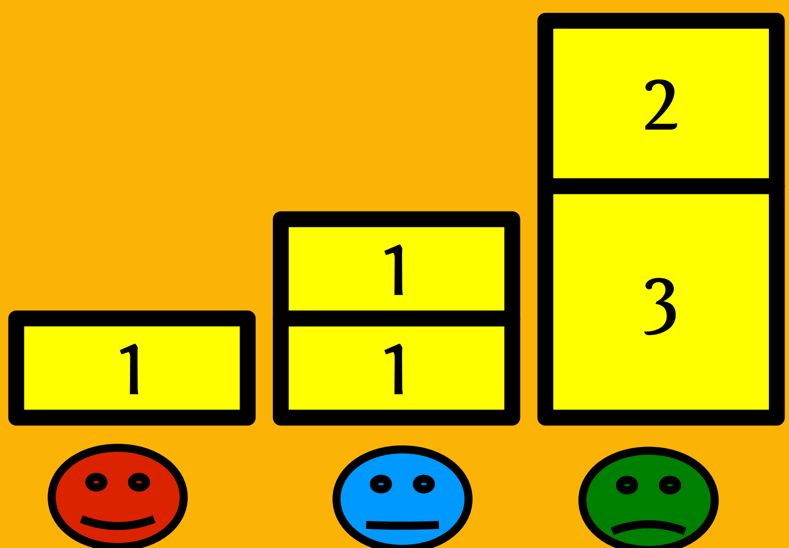
A fair solution is **symmetric**: no one is favored



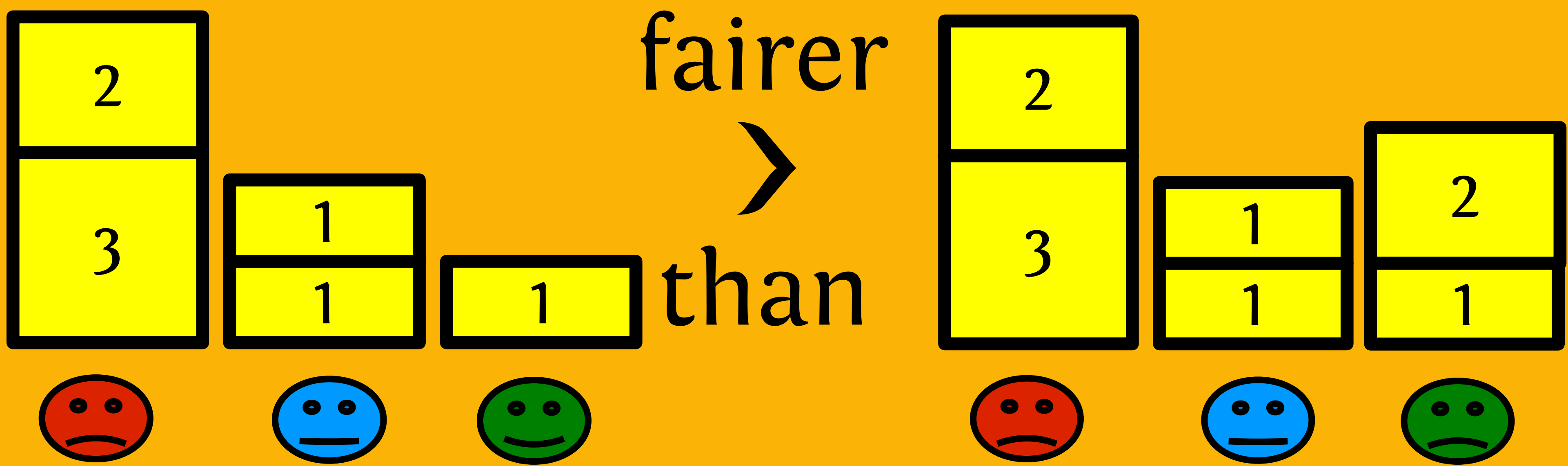
as fair

=

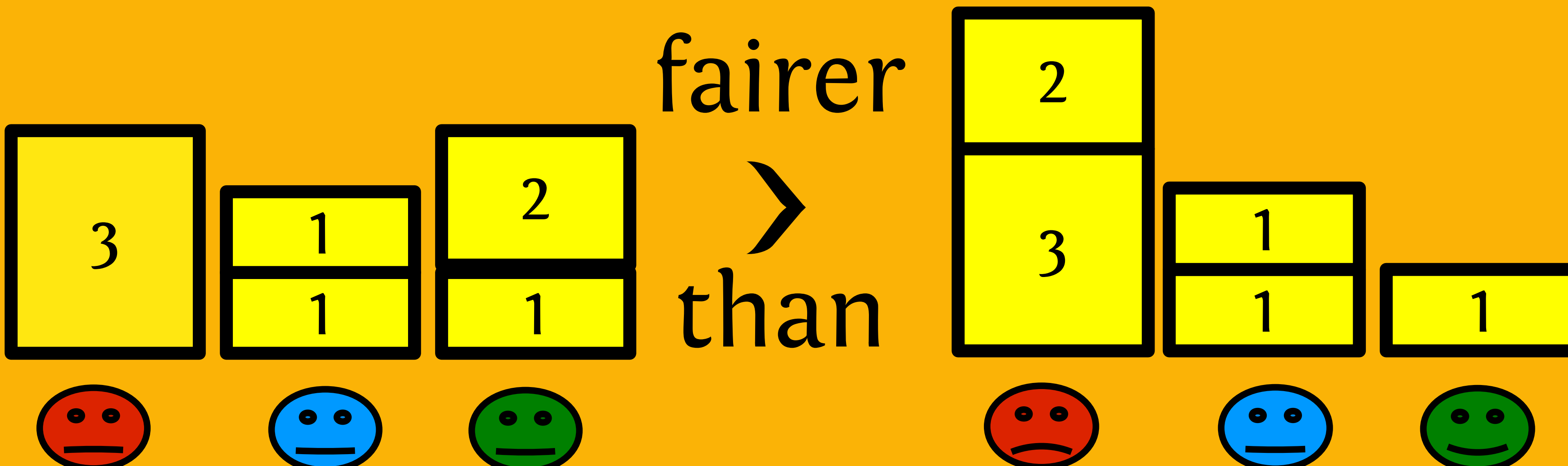
as



A fair solution is **optimal**: it can't be improved unilaterally



Principle of transfers:
Rob the riches, feed the poor



We define a relation of equitable dominance that fulfills these axioms

Relation $\leq_L$: partial order on R^N

fulfils axioms of equity

Which is more equitable: [4,6] or [10,2] ?

- [10, 2] $=_L$ [2, 10] (symmetry)
- [2, 10] $<_L$ [4,8] (transfers)
- [4, 8] $<_L$ [4, 6] (Pareto-optimality)
- So, [10, 2] less equitable than [4, 6]

These criteria can be encapsulated in a relation of Generalized Lorentz Dominance

[Chong, 1976]

(3 agents minimizing outcomes)

- sort outcomes from the worst to the best

  [1, 5, 3] -> [5, 3, 1]   [2, 5, 2] -> [5, 2, 2]

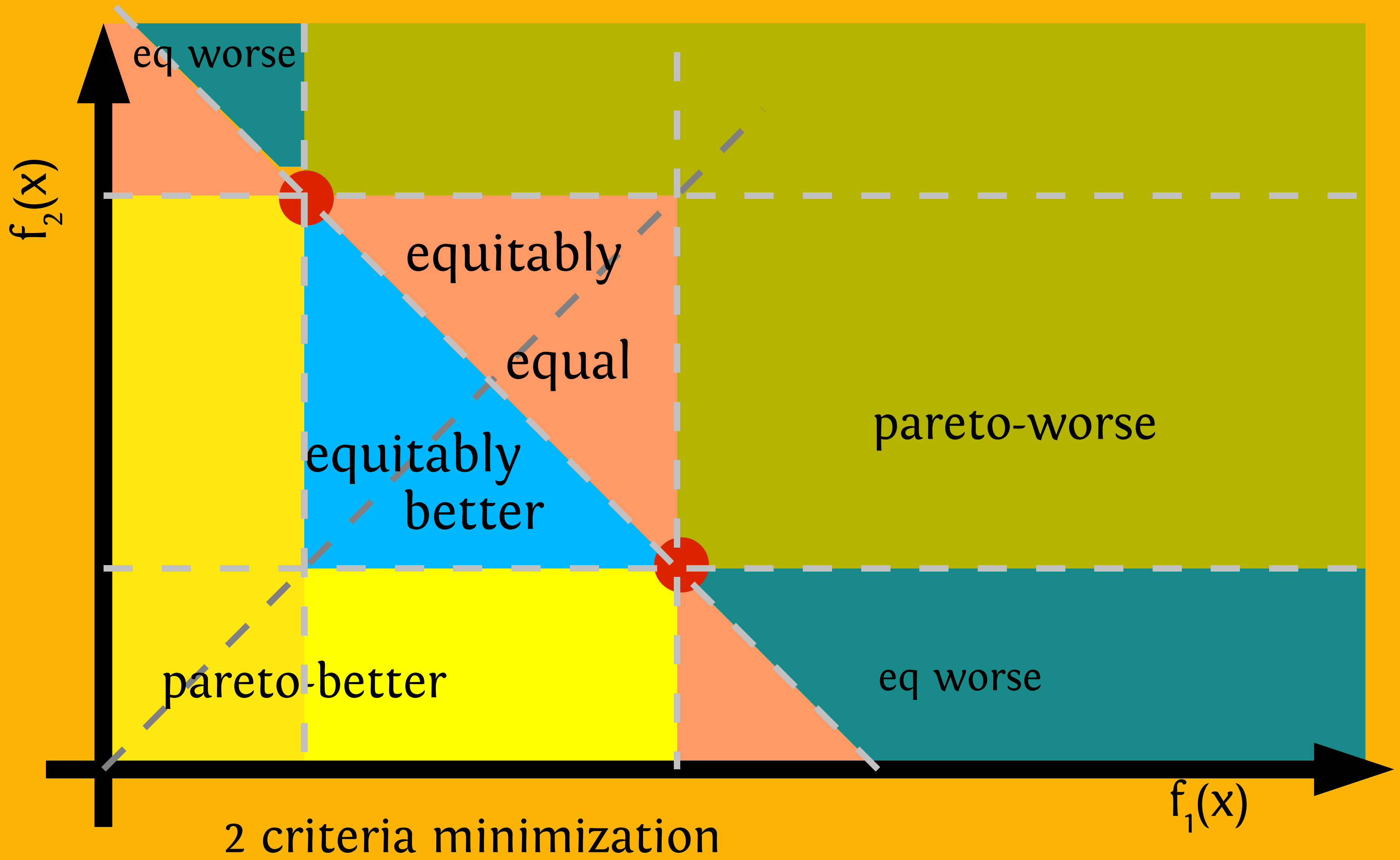
- construct cumulative outcome vector: [worst outcome, sum of two worst outcomes, ..., some of n worst outcomes]

  [5, 3, 1] -> [5, 8, 9]   [5, 2, 2] -> [5, 7, 9]

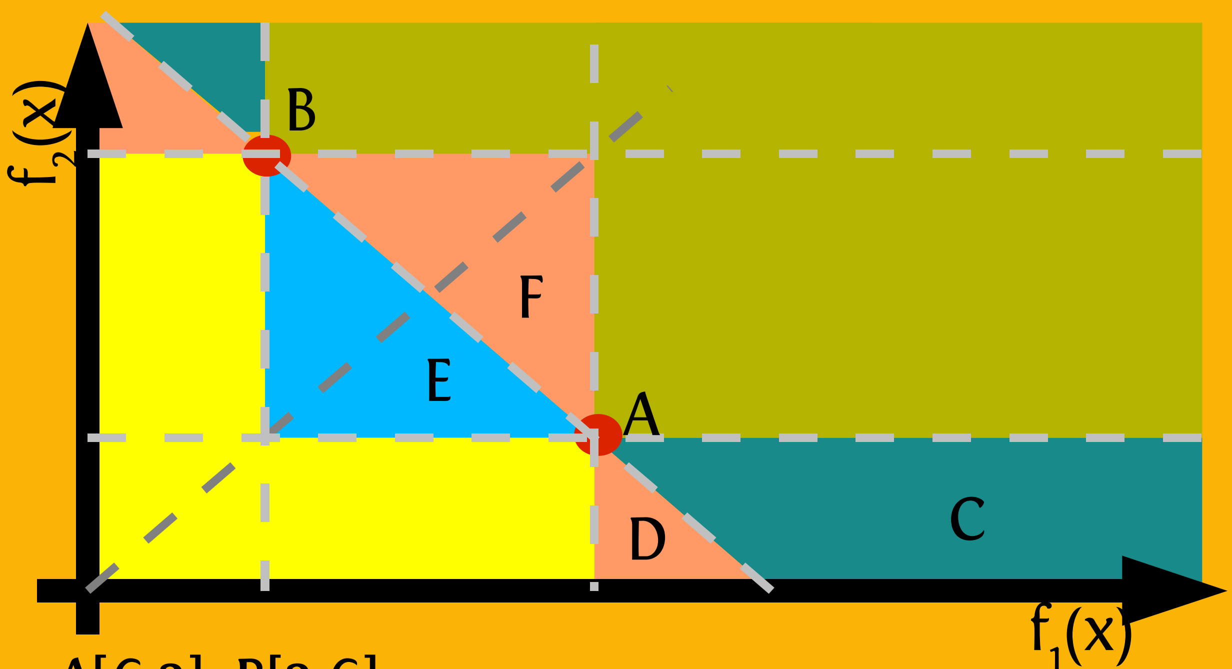
- compare cumulative vectors by Pareto dominance

  [5, 7, 9] dominates [5, 8, 9]

Lorentz dominance gives us straightforward interpretation



The same results can be obtained by using axioms directly



types of dominance

- ‹ pareto
- ‹ transfers
- ‹ equitable

A[6,2]=B[2,6]

C[100,1]‹[99,2]‹A[6,2] → C ‹ A

D[7,0]‹[6,1]‹[5,2]‹[4,3] (none is pareto-dominated by [6,2])

A[6,2]‹[4,4]‹E[4,3] → A ‹ E

F[5,4]›[6,3] and A[6,2]‹[5,3]‹[4,4] (no pareto-dominance)

Finding equitably-optimal solutions

Example application: Scheduling problem with multiple agents

[Agnetis et al., 2004]

One processor
Two sets of jobs

$$1 || \Sigma C^G_i, \Sigma C^R_i$$

green optimization goal: $\sum C^G_i = 1 + 6 + 18 = 25$

red optimization goal: $\sum C^R_i = 3 + 13 = 26$

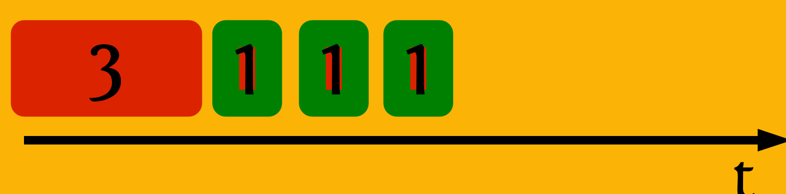
budget based approach: NP-hard

Pareto-set can have exponential size

Multicriteria optimization can model fairness between mutliple users.

(best total completion time)

$\sum C^g_i = 1 + 2 + 3$ $\quad \sum C^g_i = 6$ $\quad \sum C^g_i = 1 + 2 + 3 + 6 = 12$



(better for red agent)

$\sum C^g_i = 4 + 5 + 6$ $\quad \sum C^g_i = 3$ $\quad \sum C^g_i = 3 + 4 + 5 + 6 = 18$

One Fair Solution

Ordered Weighted Average (OWA) with positive, strictly decreasing weights gives an equitable solution.

[Ogryczak, 2000]

- OWA [Yager, 1988]: sort outcomes by increasing performance, then apply weighted average

vector: [1, 10, 5]
weights: [0.5, 0.2, 0.1]

OWA=0.5*10 + 0.2 * 5 + 0.1*1

OWA(**y**') < OWA(**y**'') <=> **y**' equitably-dominates **y**''

sketch of proof for '<='

- symmetry holds: OWA sorts outcomes
- optimality holds: one outcome decreased -> OWA decreased
- transfers holds: larger outcomes have larger weights

Other equitable aggregation functions are also possible.

[Kostreva et al, 2004]

- Aggregator function g(**y**) maps outcome vector to R
  min(g(**f**(**x**))

  Requirements on g(**y**):
- strictly increasing (-> on each outcome)
- impartial (-> not sensitive to permutations of **y**)
- strictly Schur-convexity:

$$g(y_1, \ldots, y_{i'} - \varepsilon, \ldots, y_{i''} + \varepsilon, \ldots, y_m) \leqslant g(y_1, y_2, \ldots, y_m) \quad \text{for } 0 < \varepsilon < y_{i'} - y_{i''}$$

examples: (s : R -> R, strictly convex, strictly monotonic)

$g(\mathbf{y}) = \Sigma s(y_i)$

so, e.g. $g(\mathbf{y}) = \Sigma\, y_i^k \quad ,k > 1$

Classic fairness criteria give only one "Best" solution

- Σf_i
- min max f_i
- lexmin -> generalizes min max;
  -> OWA with infinite weight difference
- Σ-log f_i -> ok, as strictly Schur convex
- Nash Bargaining Solution -> not compatible

Sum of criteria is too weak to produce an equitable solution.

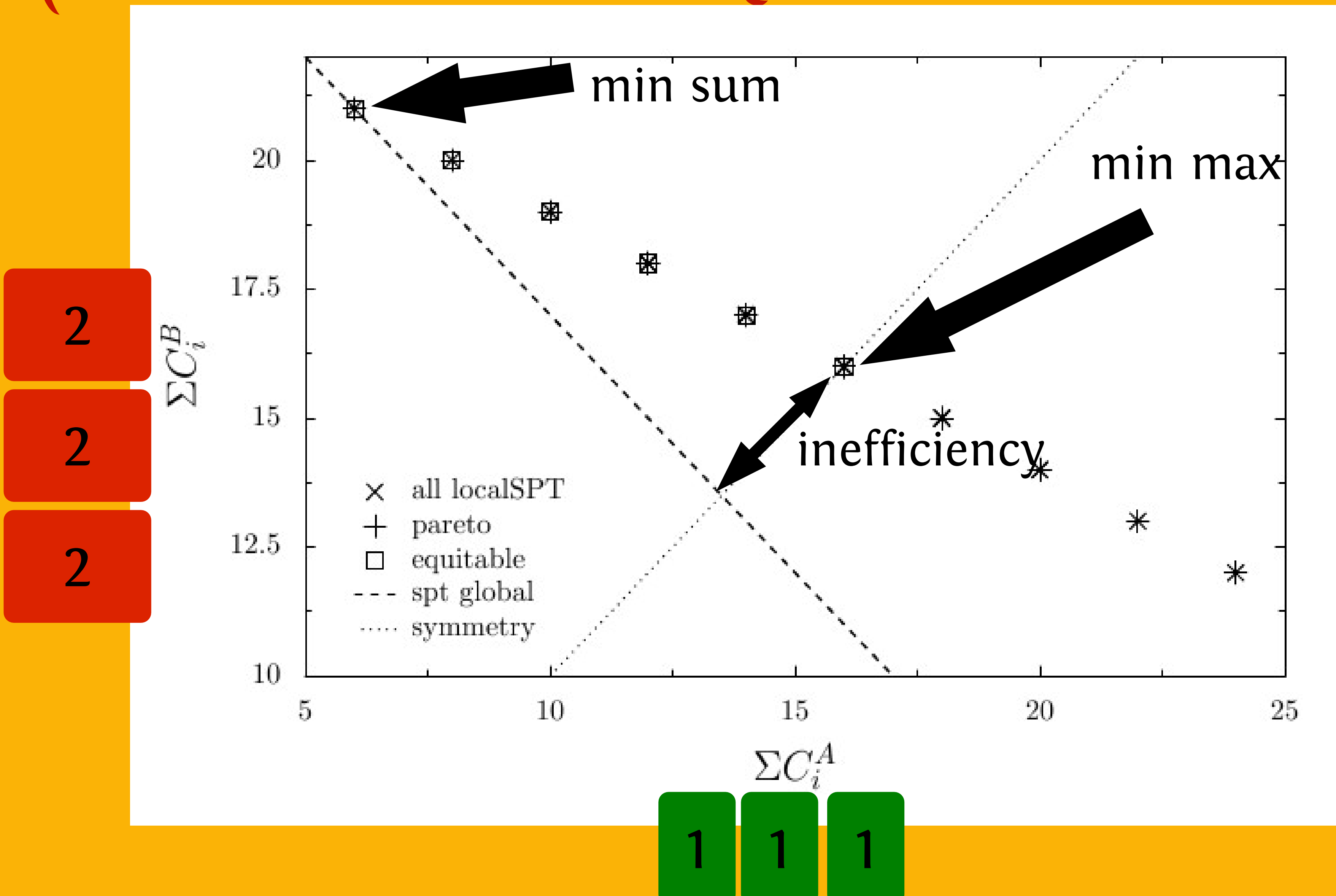
$$\min\left(\sum C^G_i + \sum C^R_i\right) = \min \sum C_i$$

Minimized by Shortest Processing Time (SPT) schedule:

This is unfair to red agent... Pairs of jobs have to be permutated

But should it be the only solution?

MIN MAX CAN RESULT IN A VERY INEFFICIENT SOLUTION (...BUT LEXMIN IS EQUITABLY OPTIMAL)

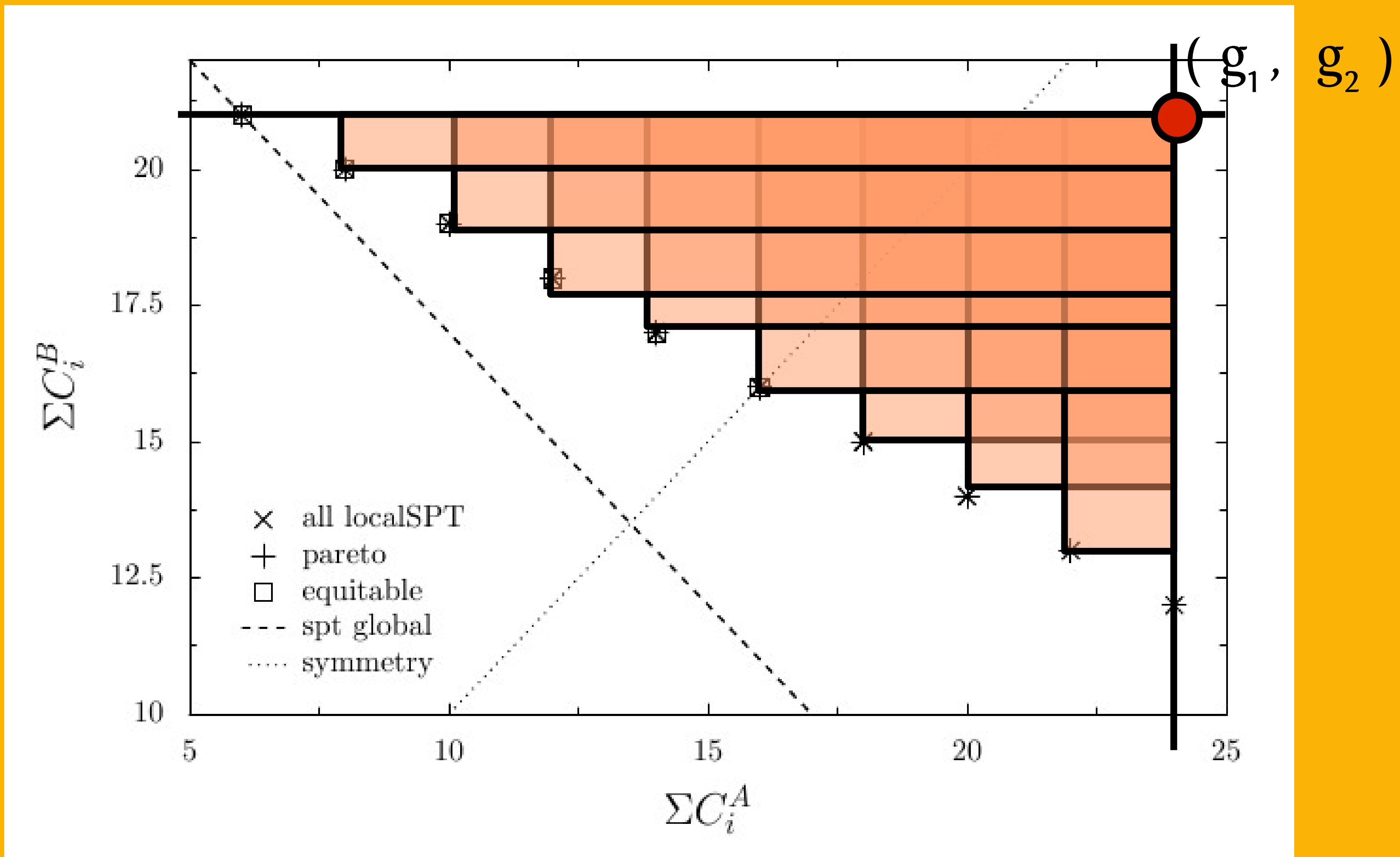


Nash Bargaining Solution takes into account best alternative to agreement.

[Agnetis, 2008]

min (g_1 - f_1) (g_2 - f_2)

How to chose g ? What is the best alternative?

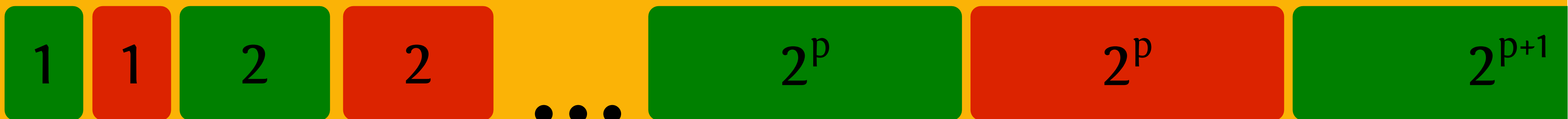


All Equitable Solutions

Pareto-prune

k best sum

In the worst case, there are exponentially many equitably-optimal solutions...



Red jobs delayed by 2^{p+1}-1

but last green jobs adds to green agents' goal: $2(2^{p+1} - 1) + 2^{p+1}$

Thus, this schedule favours red by: $2^{p+2} - 3$

After switching order of two jobs (i <= p):

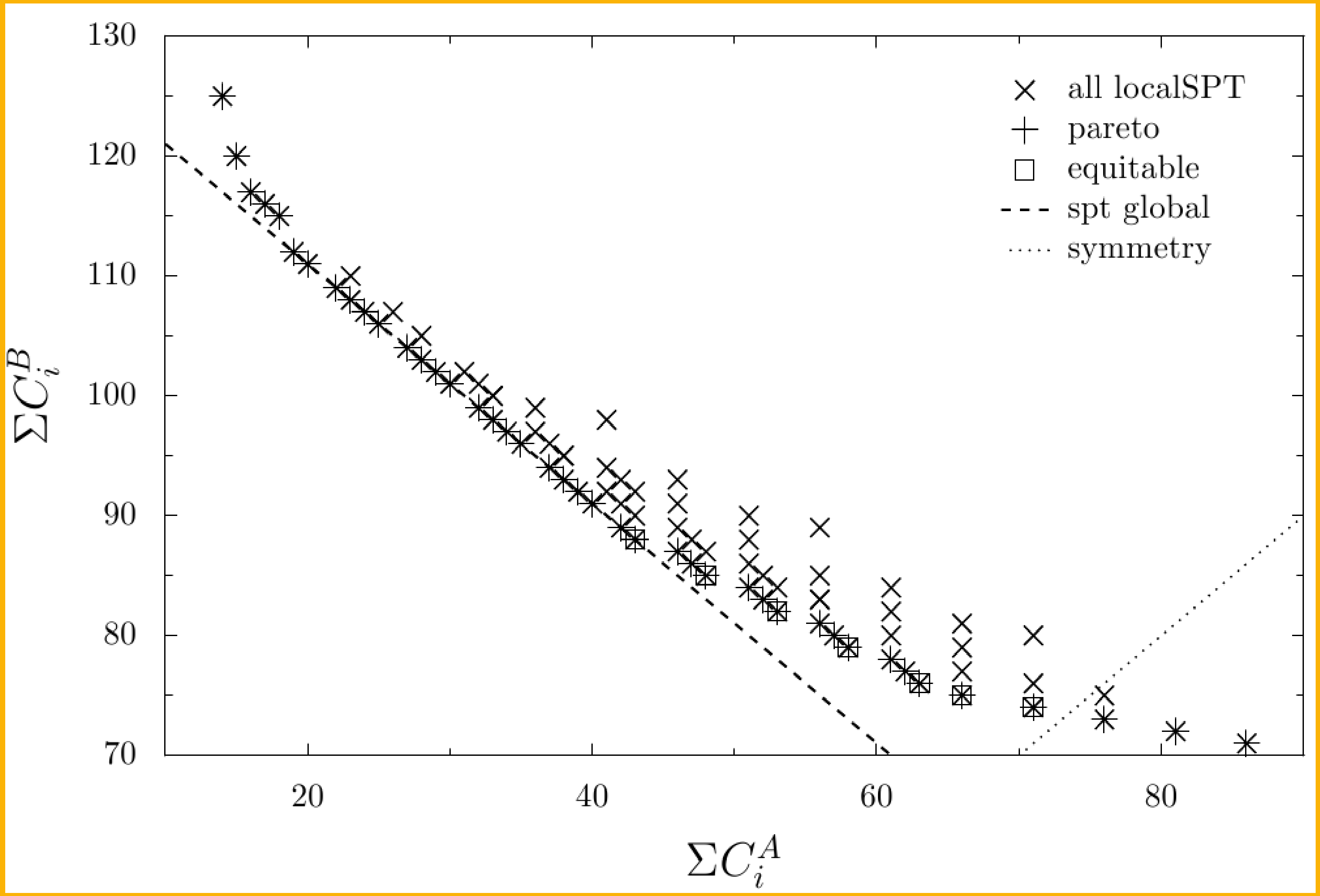
red: $+2^i$ green: -2^{i-1} sum: $+2^{i-1}$

Generalized Lorentz Dominance: (xi == 1 if there was a switch)

$[C_1 - \Sigma x_i 2^i ; C_2 + \Sigma x_i 2^i]$ ($C_1 < C_2$, both are constants)

Pareto-independent; thus all possible switches equitably-independent

Pareto-prune: Find the complete Pareto-optimal set, then remove equitably dominated solutions.



possible problem: too many Pareto-optimal solutions

Applied to multi-agent scheduling: dynamic programming for Pareto-opt schedules

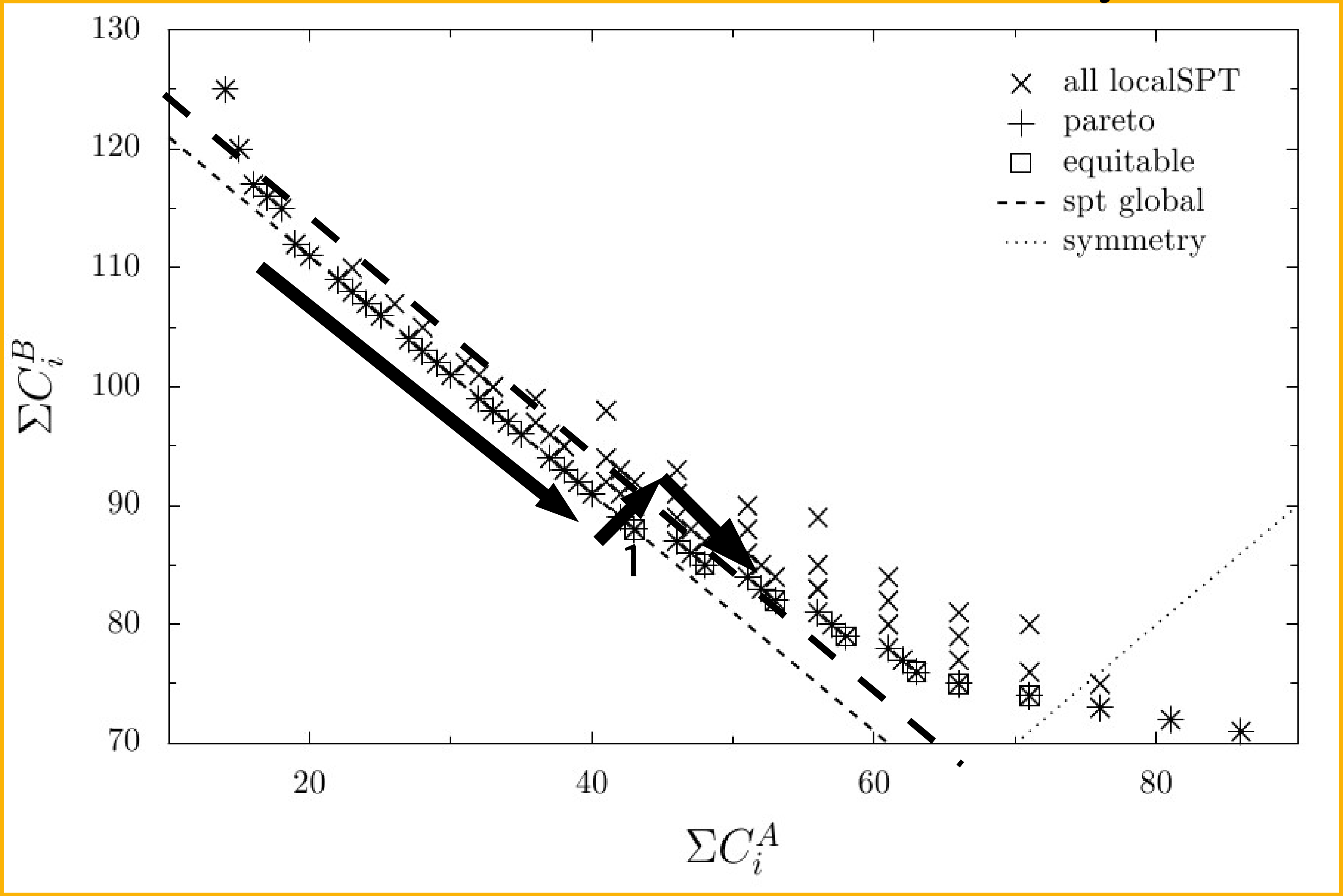
Pareto-opt schedules with n jobs:

extend Pareto-opt schedules with n-1 jobs
remove Pareto-dominated combinations

order between jobs of the same owner: SPT
thus schedule specifies owners of the first n jobs

K-BEST SUM: DETERIORATING MIN SUM, FIND EQUITABLY-OPTIMAL SOLUTIONS

[Perny et al, 2006]



Finding equitable schedule for the same total score

- find chains of jobs of the same length
- starting from the chain of longest job, greedy reduce the difference of payoffs

Why it works?

- after each chain, the difference of payoffs is decreased
- by starting from the longest jobs, we are able to reduce “errors”

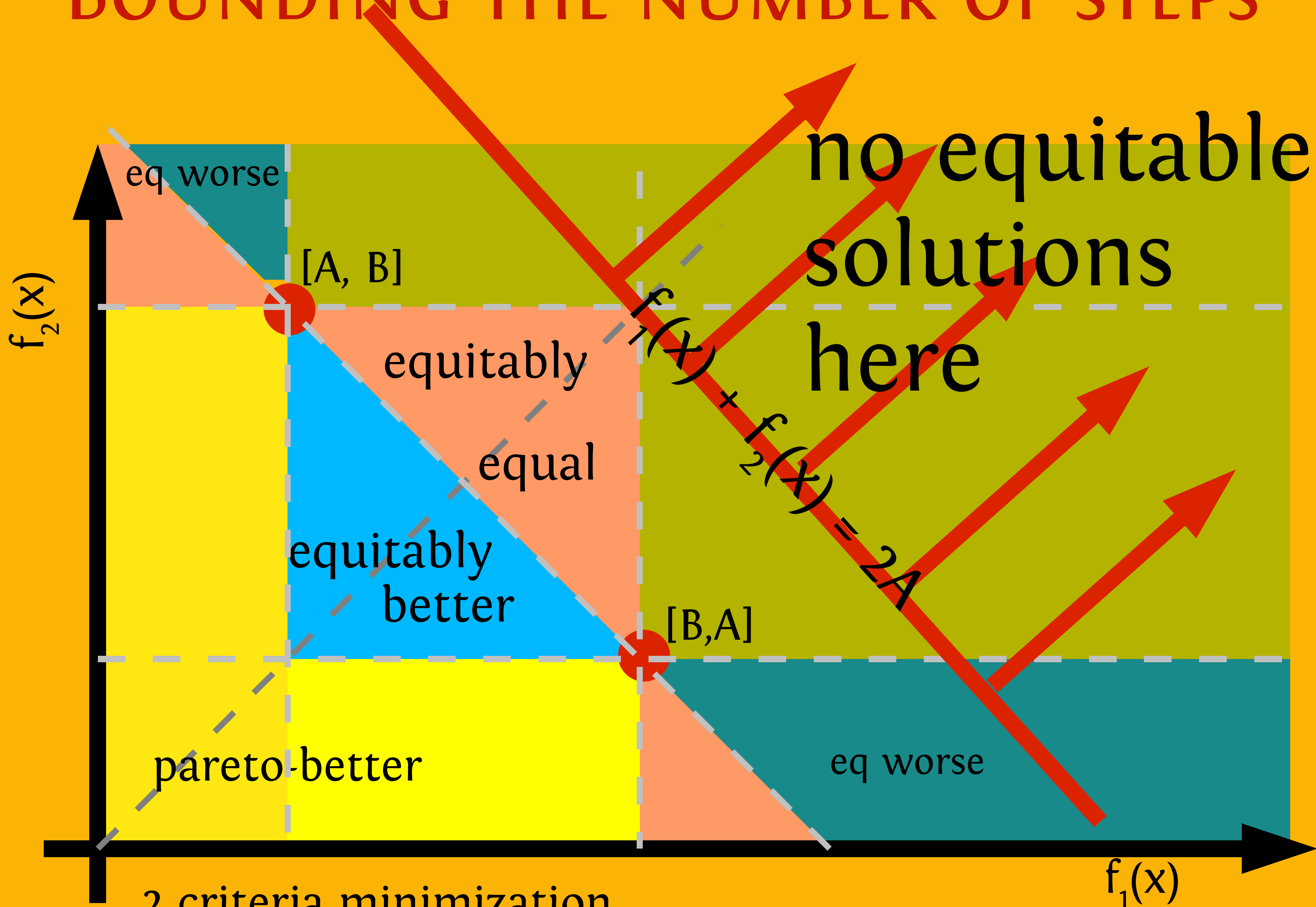
Finding next degradation

- in current schedule,
  for each pair of jobs: (longer after shorter)

- switch such pair 

- compute total score: $\Sigma C_i \mathrel{+}= (m-l)$
- add to not visited schedules (heap by ΣC_i)
- take the smallest element from heap

Bounding the number of steps



Equitable multi-agent optimization: conclusions

- multi-agent optimization becomes important (shared system, centralized control)
- “true” multicriteria approach to fair optimization (many solutions to chose from)
- single equitable solution: generalization of some known concepts
- all equitable solutions: can be intractable... (->approximation algorithms?)

Equitable Multi-Objective Optimization

Krzysztof Rzadca, krz@pjwstk.edu.pl
Polish-Japanese Institute of Information Technology

based on joint work with
Denis Trystram,
Adam Wierzbicki

Thank you!