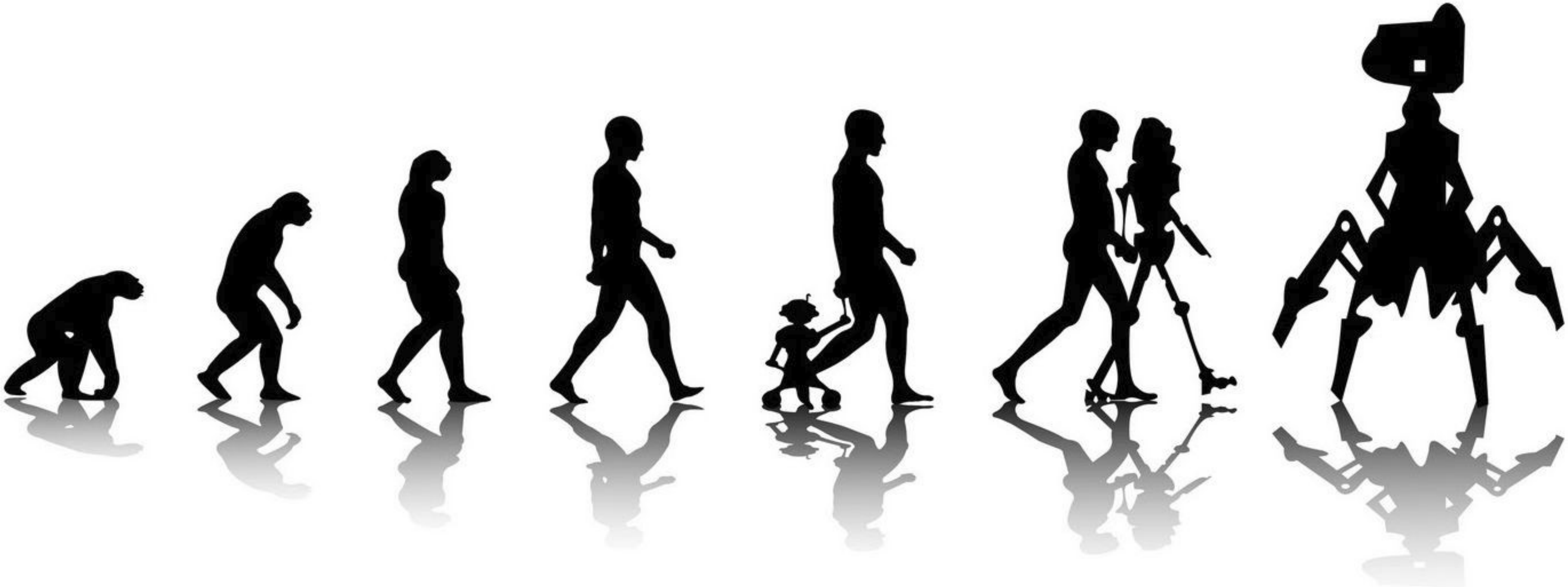


Modern Robotics: Evolutionary Robotics

COSC 4560 / COSC 5560

Professor Cheney

2/5/18

When and why would evolutionary algorithms be preferable to other types of optimization?

For instance, when the function we are trying to optimize is not a straightforward cost (e.g. a trade-off between competing costs)

Multi-Objective Optimization

For instance, when the function we are trying to optimize is not a straightforward cost (e.g. a trade-off between competing costs)

What might be examples of this?

Strength-to-Weight of a mechanical design

Speed-vs-Fuel Efficiency of a vehicle/robot

Riskiness-vs-Return of an investment strategy

Production-vs-Marketing budget allocation strategy

Classification Performance-vs-Size of a neural net

These types of trade-off problems are often very difficult to design by hand

Even if you knew the solutions to both problems individually, designing a solution that optimally balances both is often unintuitive

However, gradient descent algorithms struggle with multiple-objective optimization

Given that they need to find the derivative of a cost function, the best way to do this is to combine all objectives into a single cost function

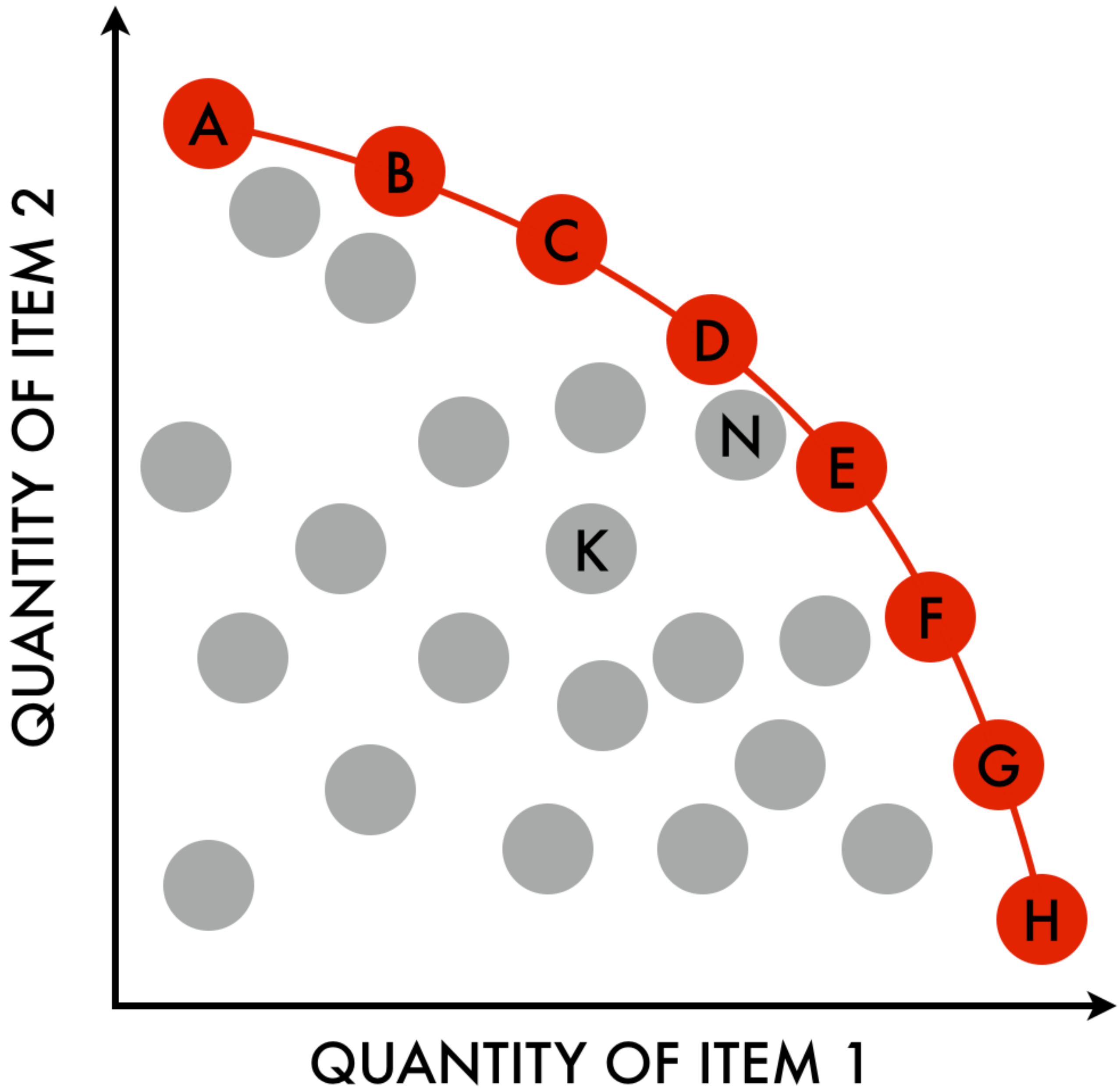
$$\min_{w} \sum_{i=1}^{n} V(\hat{x}_i \cdot w, \hat{y}_i) + \lambda \|w\|_2^2$$

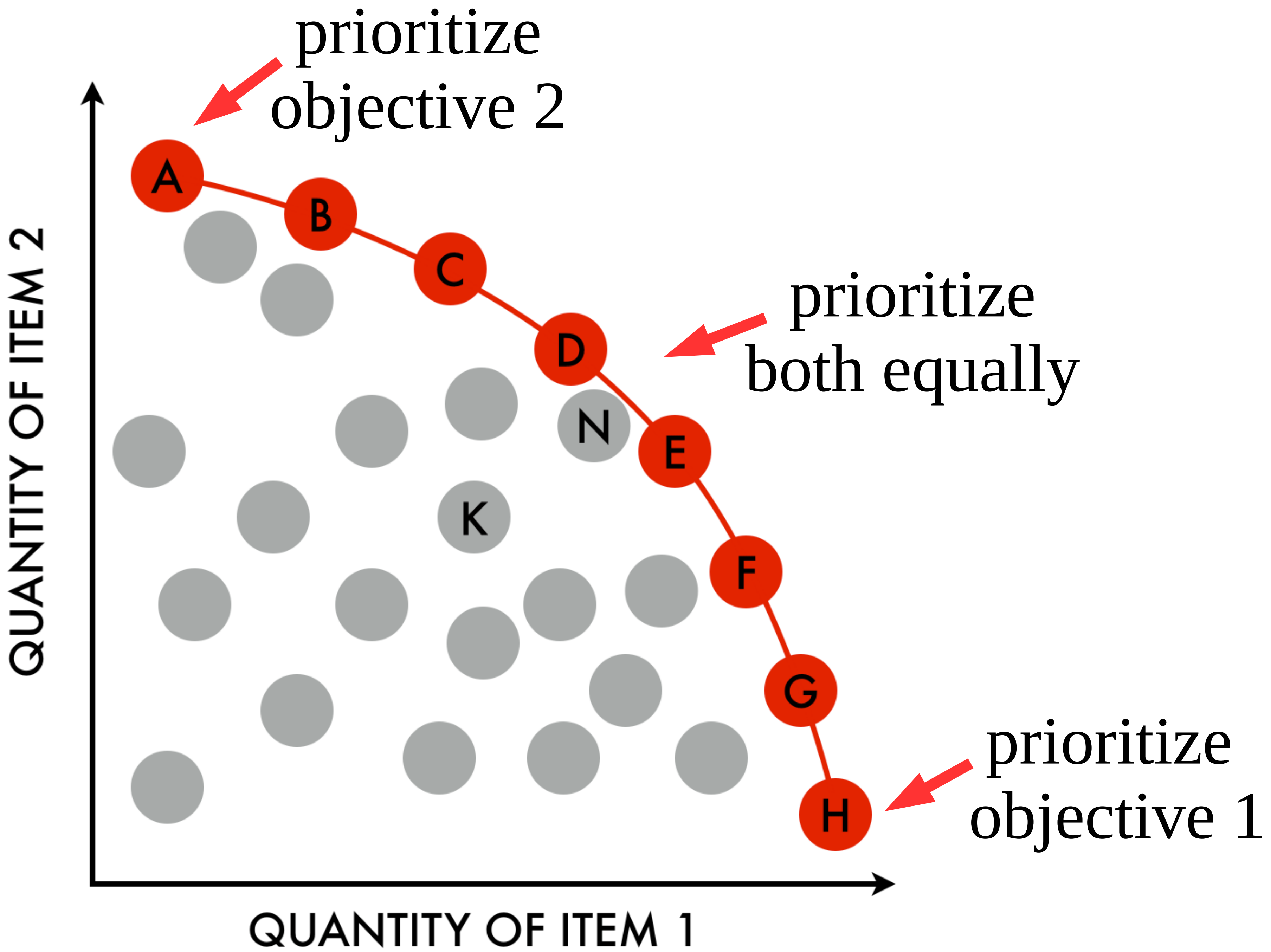
Why is this not ideal?

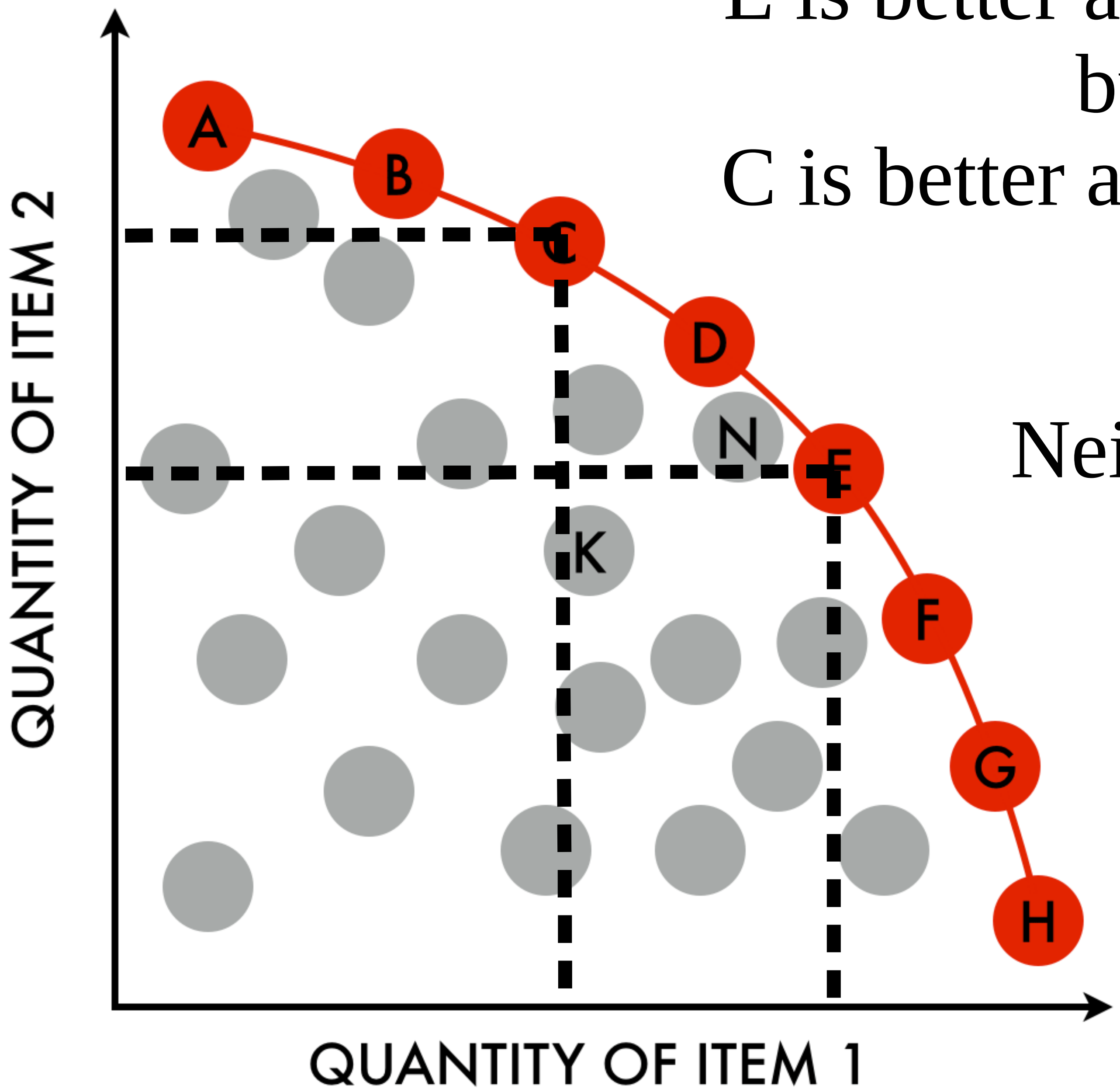
$$\min_{w} \sum_{i=1}^{n} V(\hat{x}_i \cdot w, \hat{y}_i) + \lambda \|w\|_2^2$$

We need to explicity specify the trade-off (weighting of the two cost terms) before we can solve the problem!

Since we do not know the optimal trade-off between these two dimensions, ideally we would like to explore many possible compromises between them

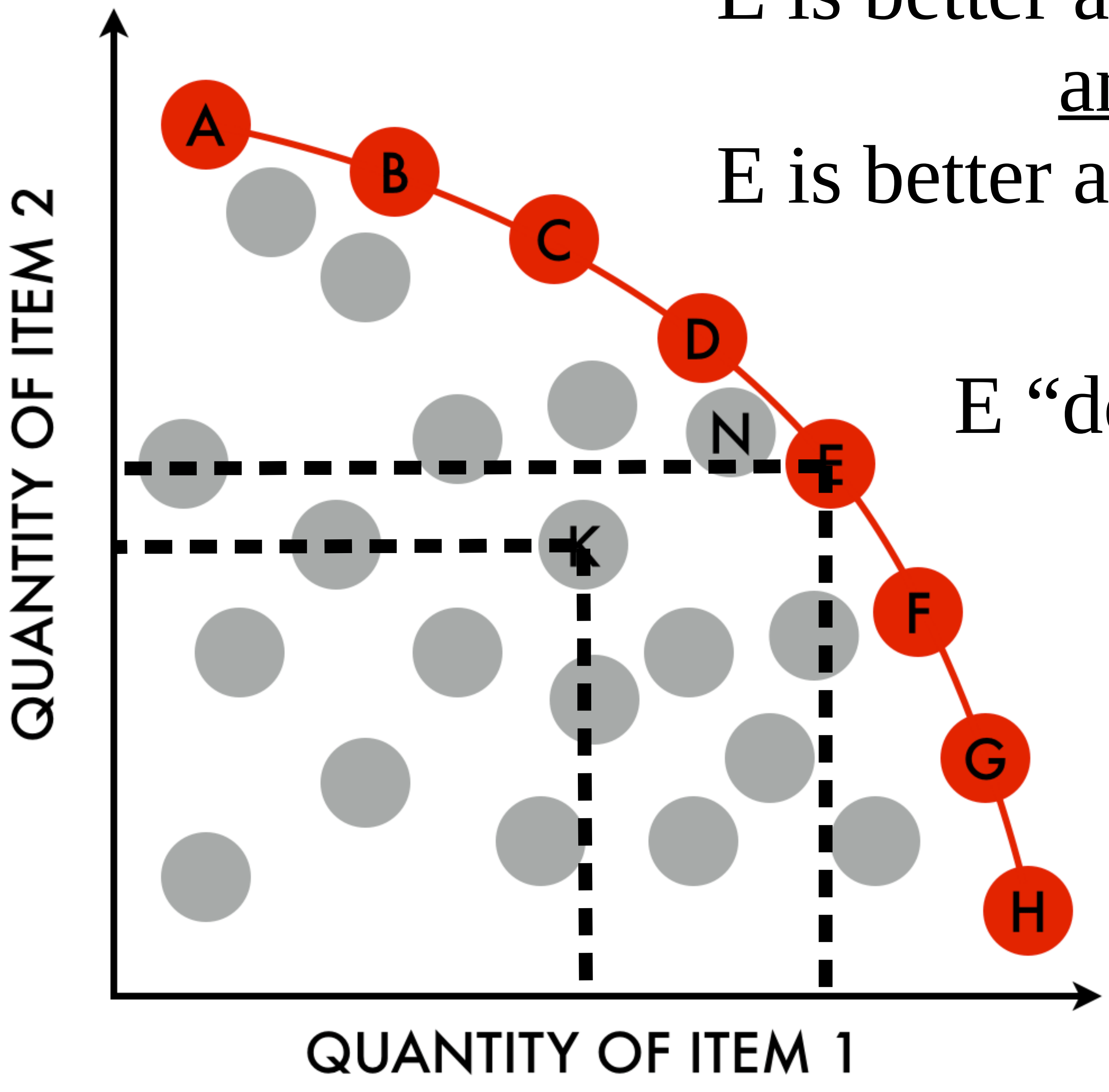

A
B
C
D
E
F
G
H
N
K
QUANTITY OF ITEM 2
QUANTITY OF ITEM 1


prioritize objective 2
prioritize both equally
prioritize objective 1
A
B
C
D
E
F
G
H
N
K
QUANTITY OF ITEM 2
QUANTITY OF ITEM 1



E is better at objective 1
but
C is better at objective 2

Neither C nor E
“dominate”
the other

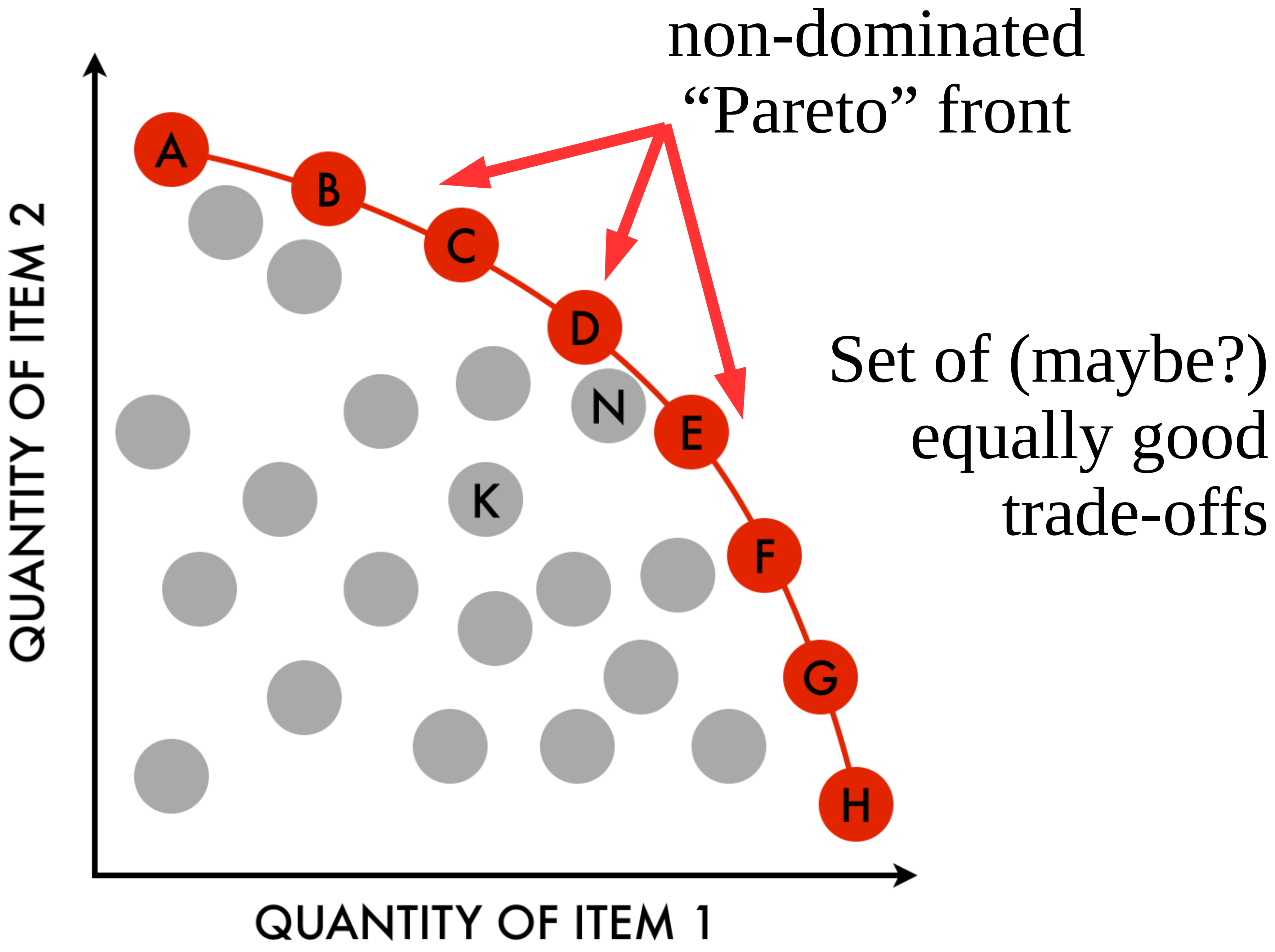


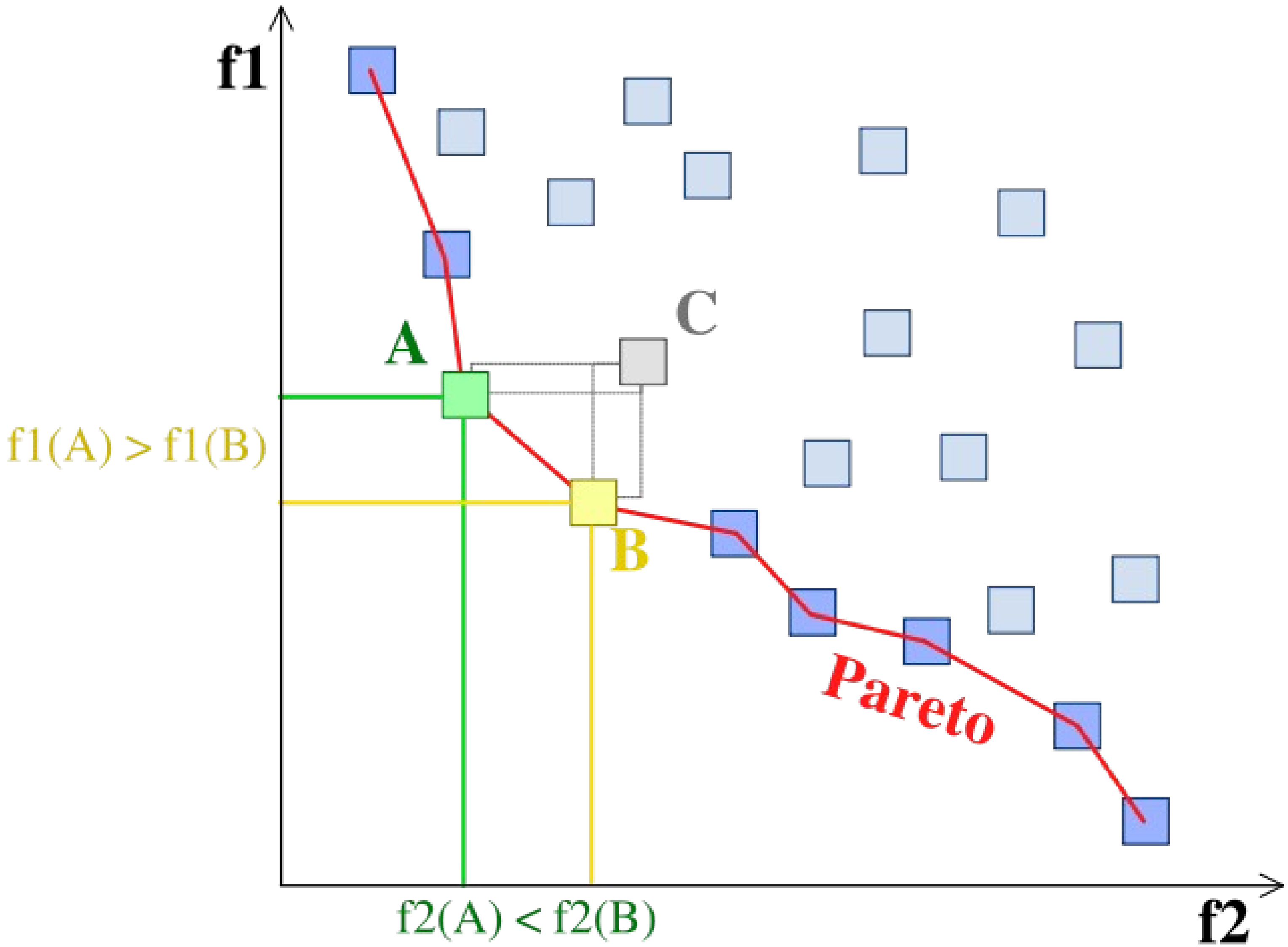
E is better at objective 1

<u>and</u>

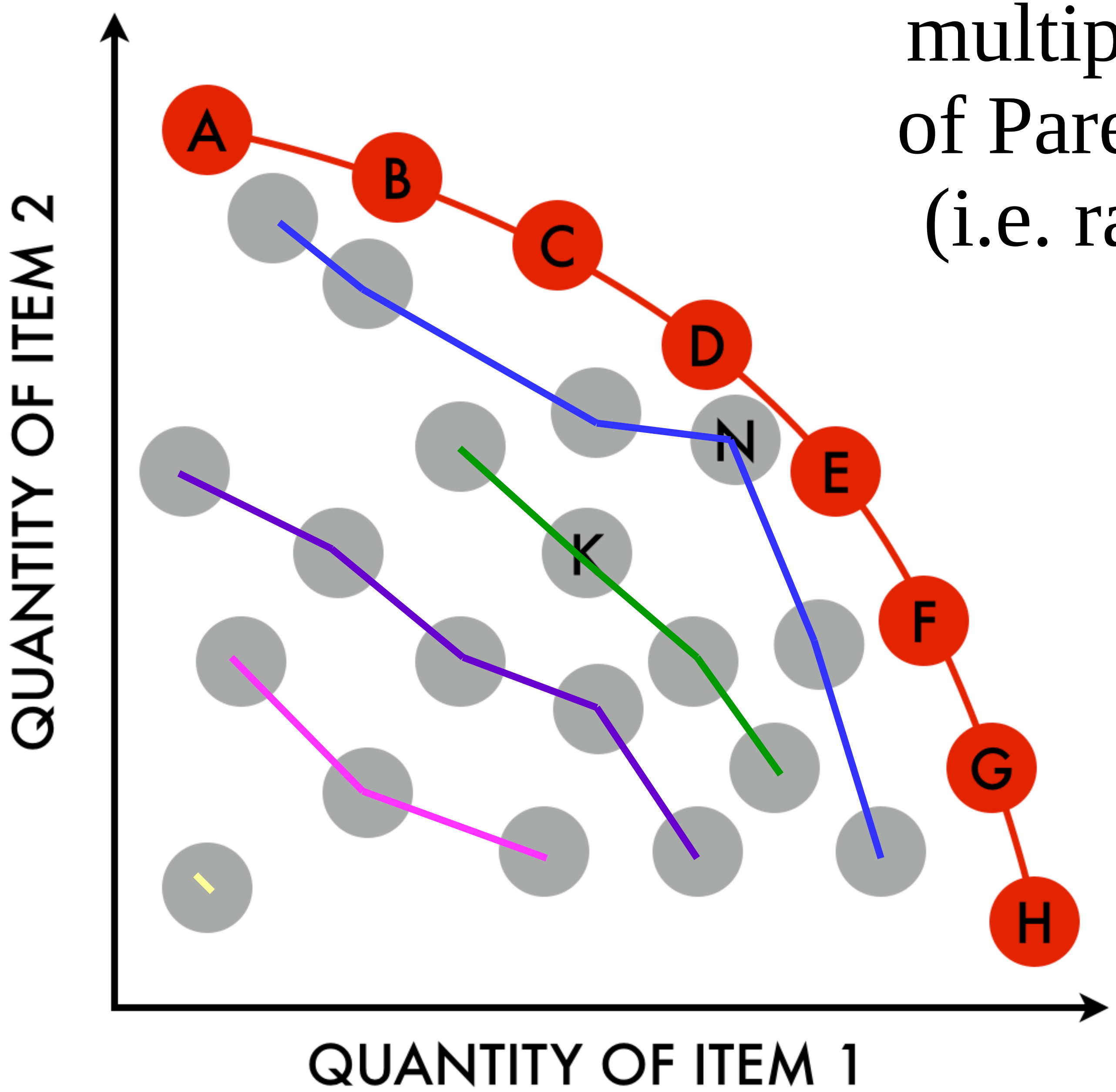
E is better at objective 2

E “dominates” K


non-dominated
"Pareto" front
Set of (maybe?)
equally good
trade-offs
A
B
C
D
E
F
G
H
N
K
QUANTITY OF ITEM 2
QUANTITY OF ITEM 1

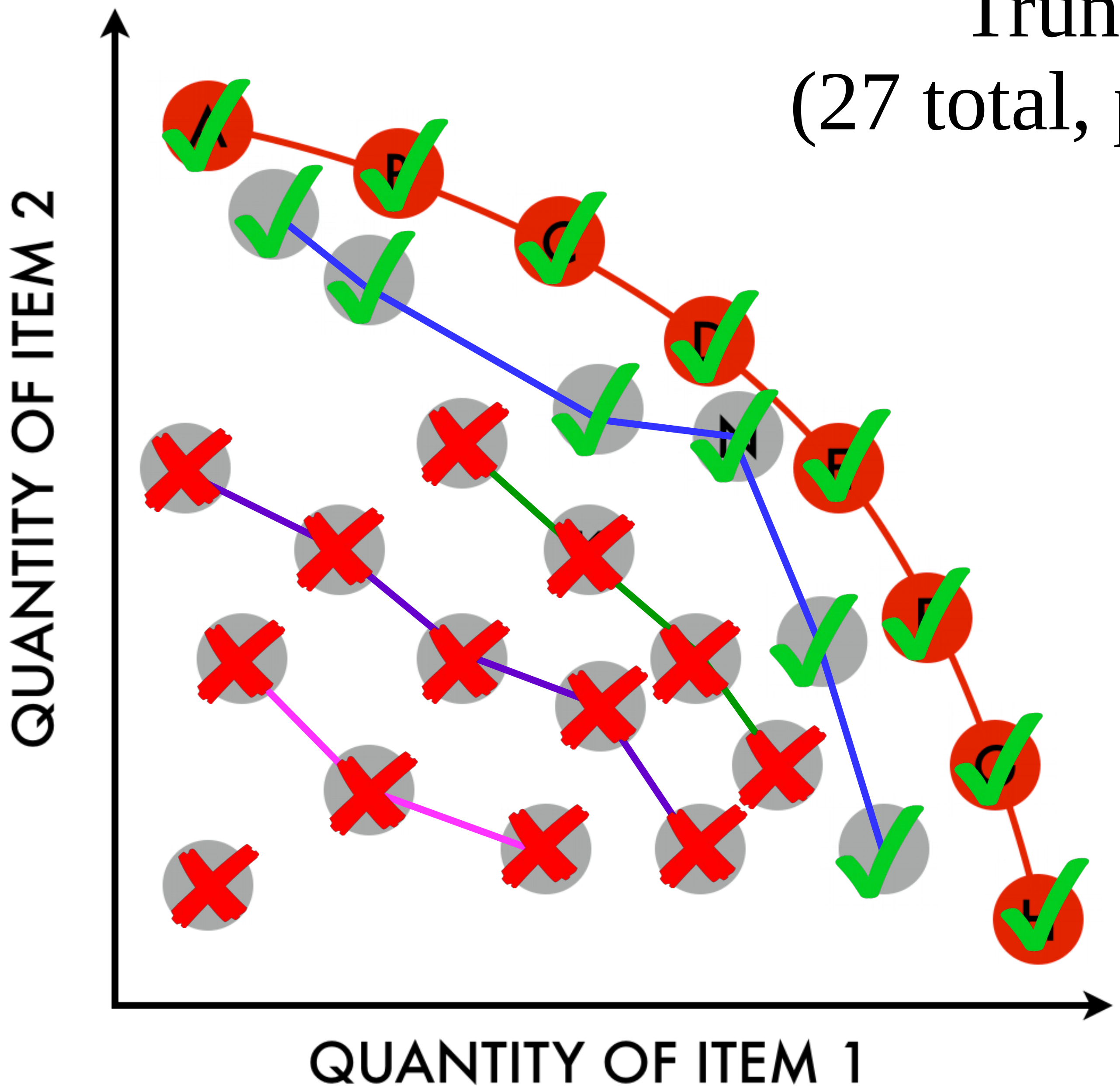

f1
A
C
f1(A) > f1(B)
B
Pareto
f2(A) < f2(B)
f2

multiple-levels
of Pareto fronts
(i.e. rankings)



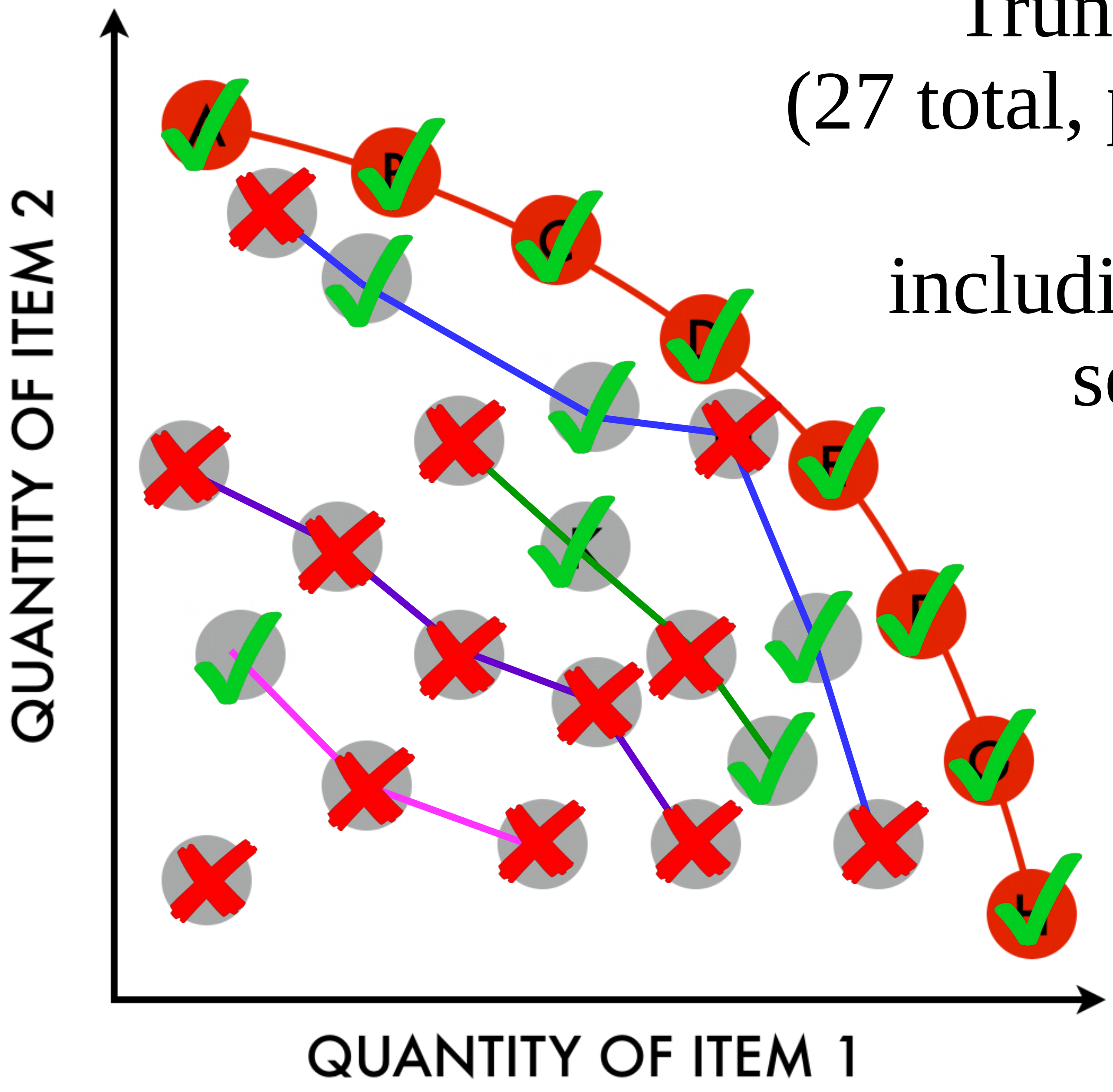
Since we have a systematic way of ranking all potential solutions, we can apply an evolutionary algorithm to the multi-dimensional optimization problem!

Truncation:
(27 total, pick top 14)

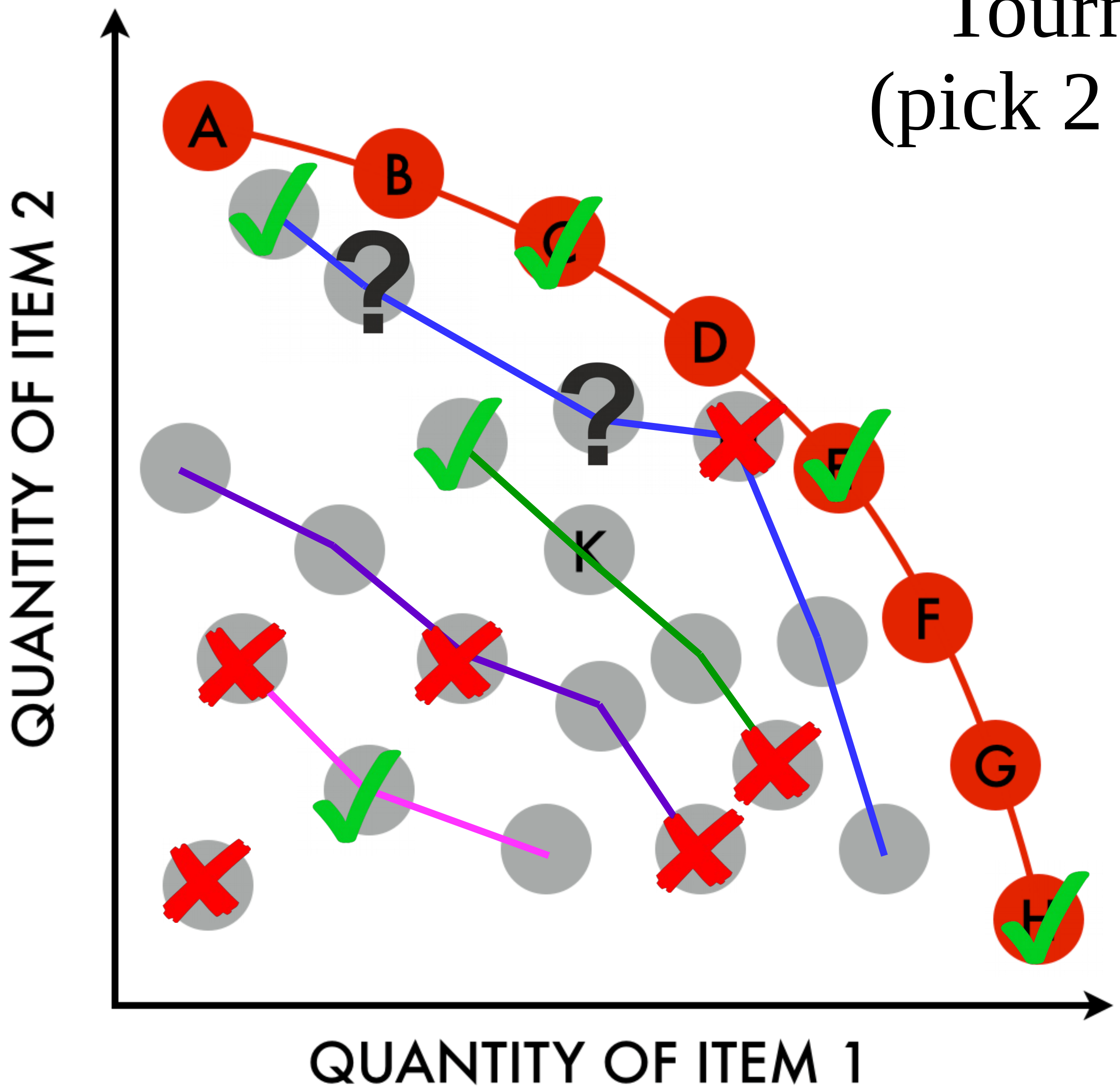


Truncation:
(27 total, pick top 14)

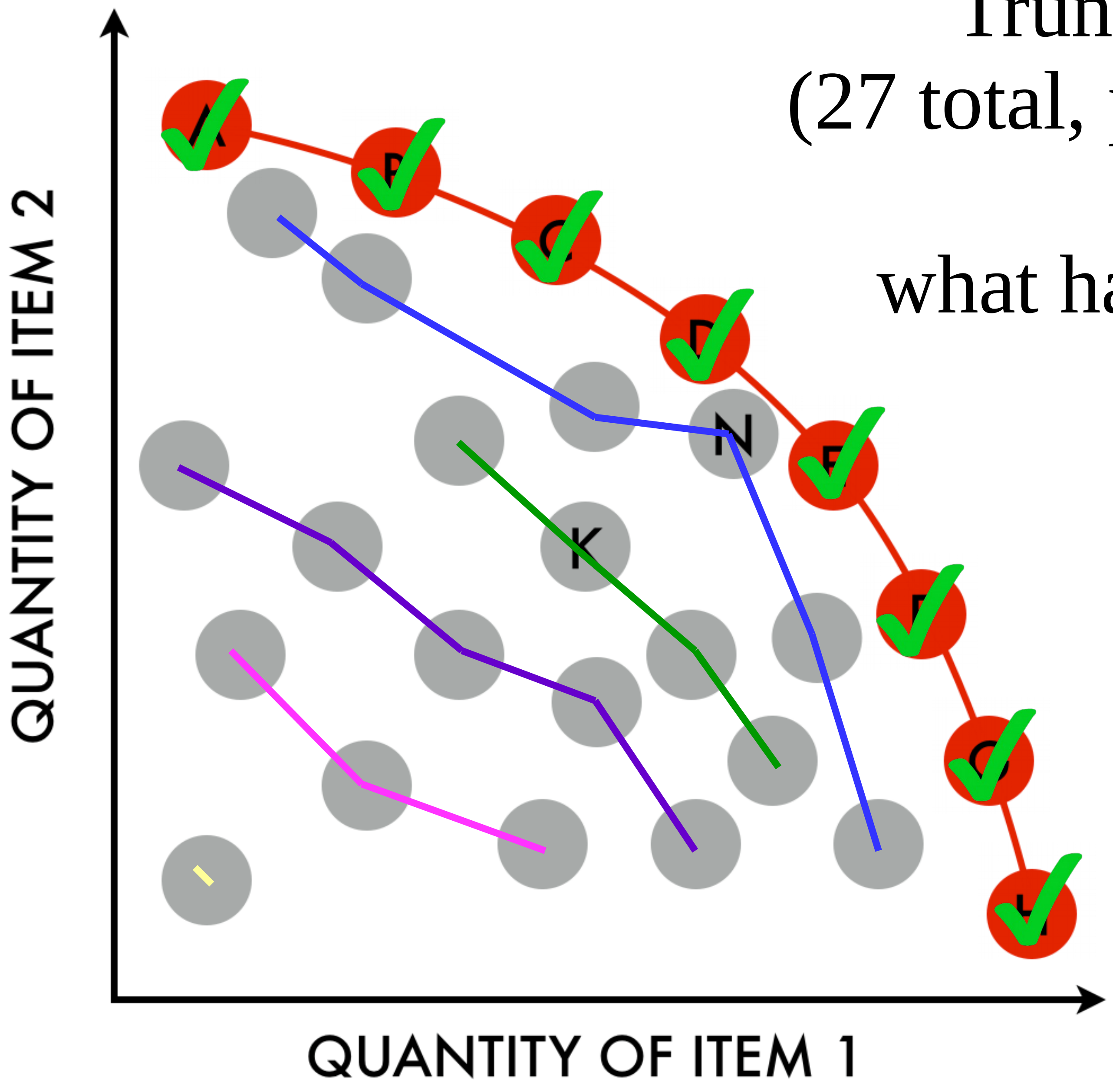
including stochastic selection



Tournament:
(pick 2 at a time)

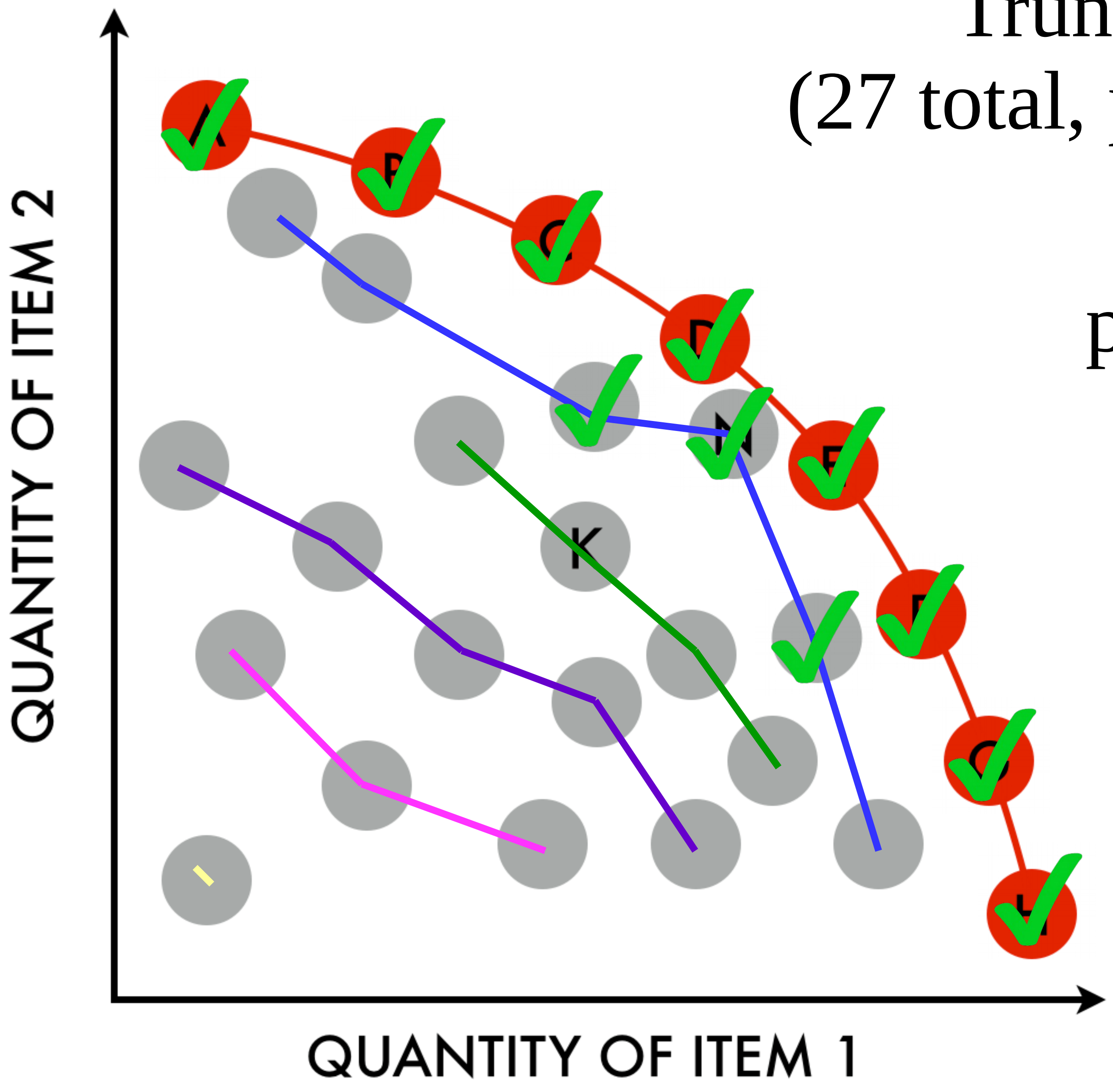


Truncation:
(27 total, pick top 11)



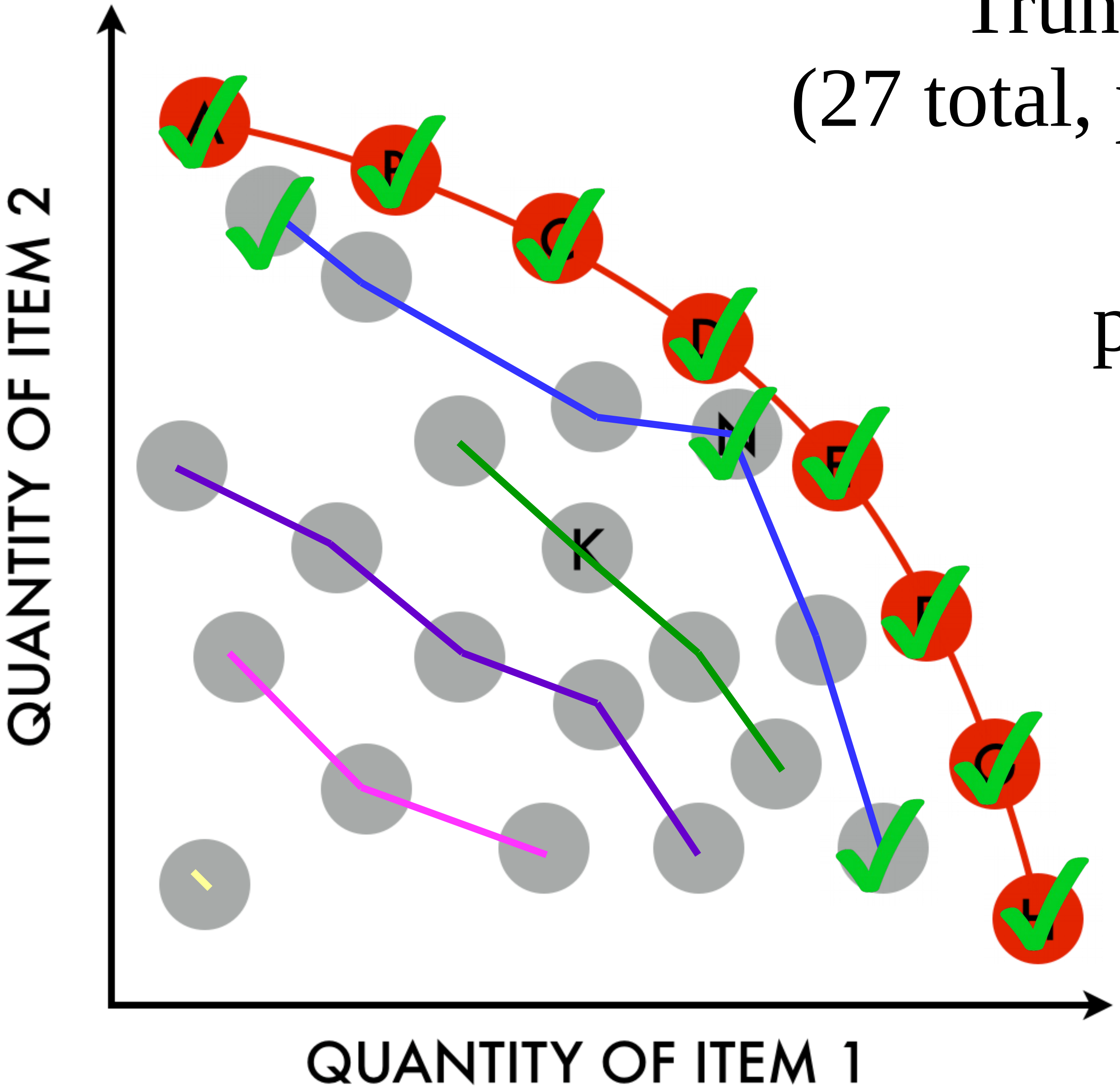
what happens when you have to select within rows?
(3 left)

Truncation:
(27 total, pick top 11)



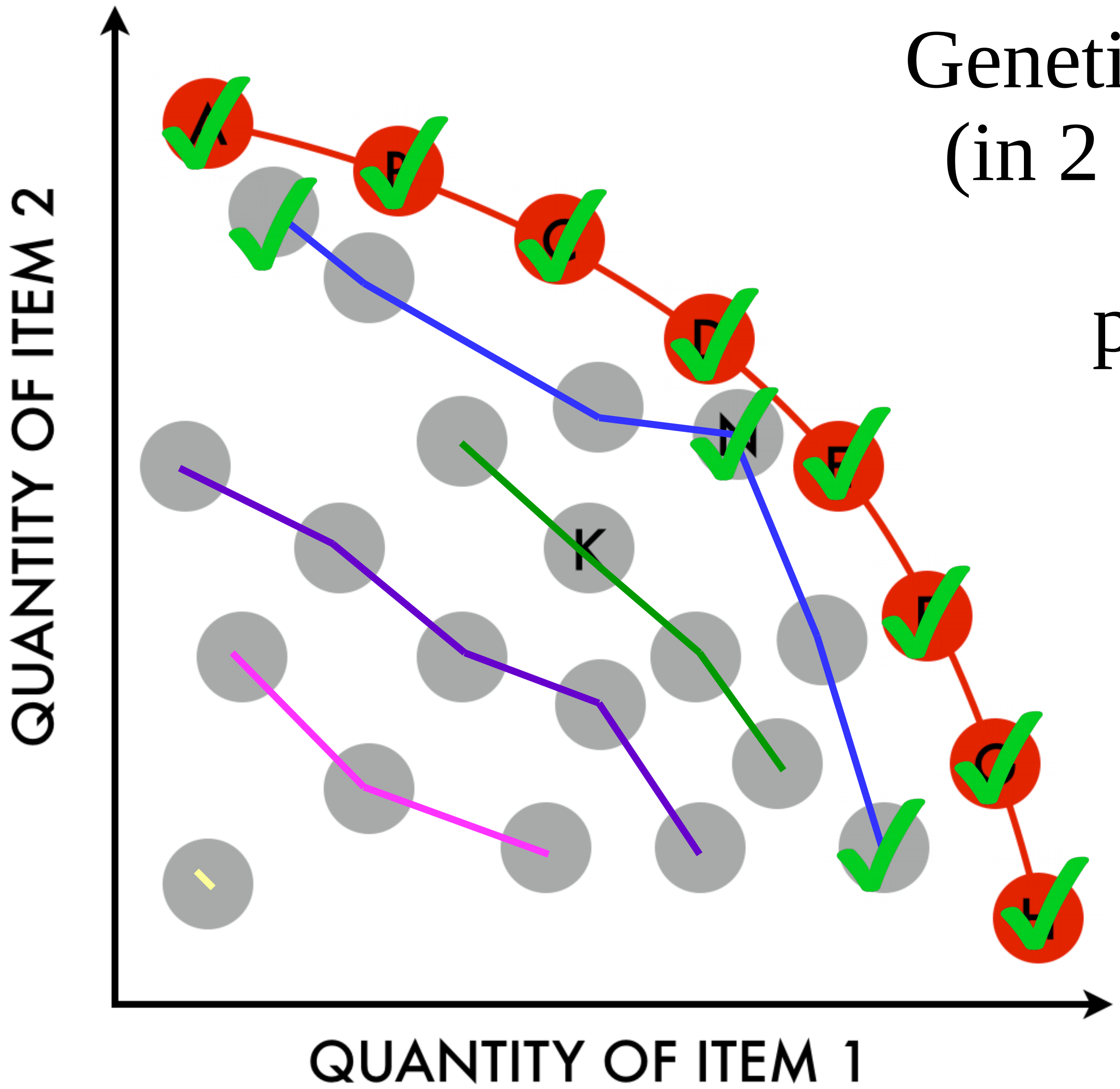
pick the most balanced trade-off?

Truncation:
(27 total, pick top 11)



pick the most <u>diverse</u> trade-off?

NSGA-II: Non-dominated Sorting Genetic Algorithm (in 2 dimensions)

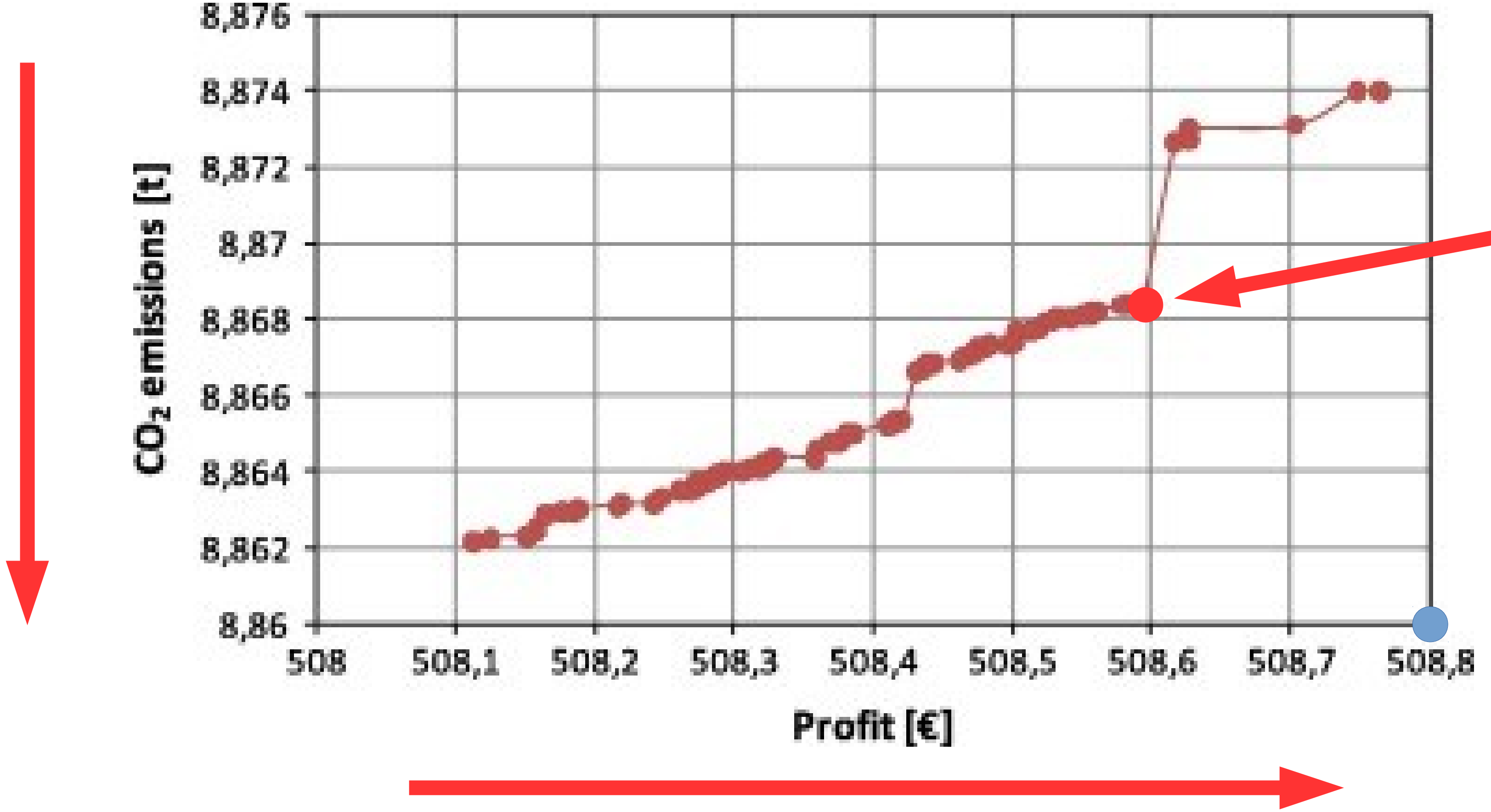


pick the most <u>diverse</u> trade-off?

We'll revisit the idea of selecting for both effectiveness and diversity throughout the semester!

In real problem, the Pareto fronts won't be smooth

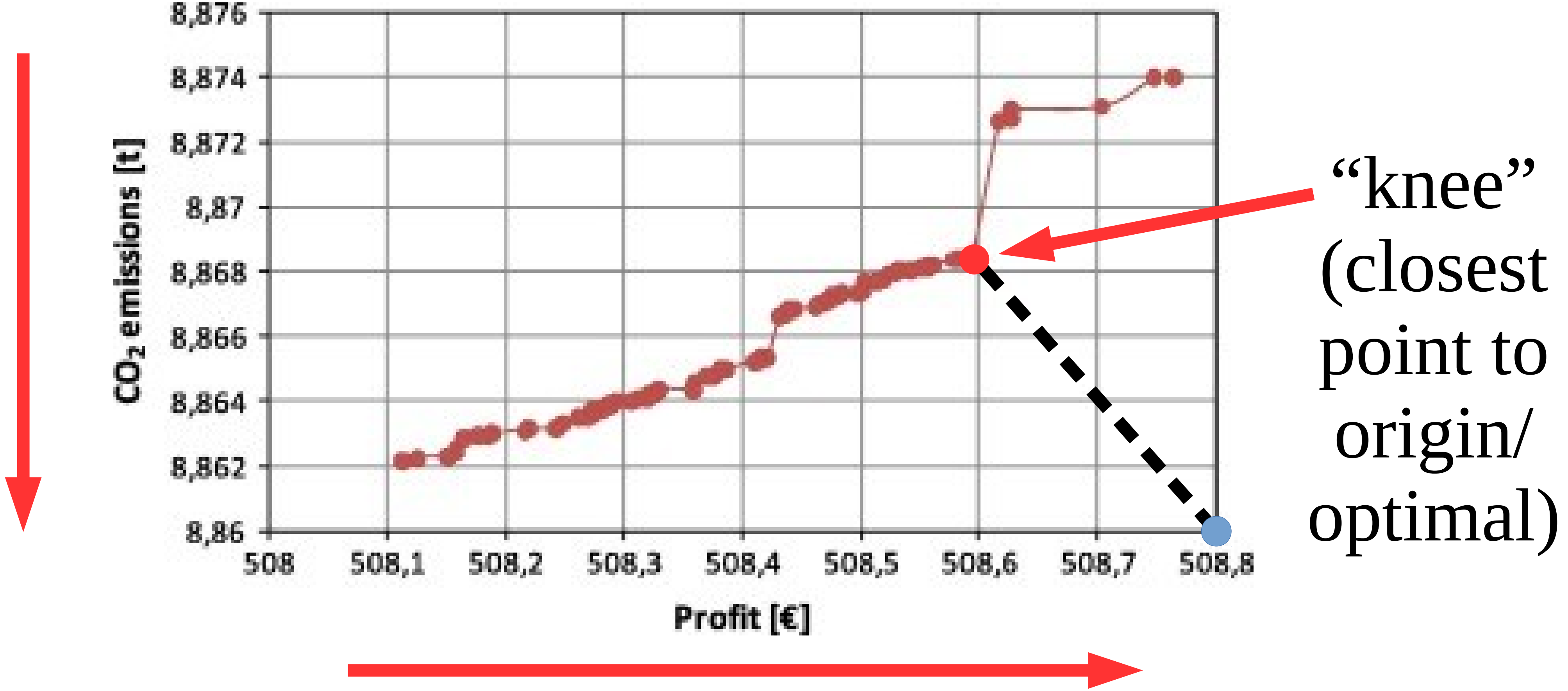
If we had to choose just one point on the front, what would it be?



“knee” (point of a sharp trade-off)

In real problem, the Pareto fronts won't be smooth

If we had to choose just one point on the front, what would it be?



We can use multi-objective selection
to optimize entire populations
(and eventually choose a single best solution)!

You'll be doing this later on in your homeworks