HAL
open science

Tutorial on Fitness landscape analysis: Understanding and predicting algorithm performance for single- and multi-objective optimization

Sébastien Verel, Arnaud Liefooghe

To cite this version:

Sébastien Verel, Arnaud Liefooghe. Tutorial on Fitness landscape analysis: Understanding and predicting algorithm performance for single- and multi-objective optimization. International Conference on Parallel Problem Solving from Nature (PPSN 2020), 2020. hal-04029576

HAL Id: hal-04029576

https://hal.science/hal-04029576v1

Submitted on 15 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fitness Landscape Analysis Understanding and Predicting Algorithm Performance for Single- and Multi-objective Optimization

Bilel Derbel
Univ. Lille – CRIStAL / Inria Lille-Nord Europe, France
bilel.derbel@univ-lille.fr

Arnaud Liefooghe
Univ. Lille – CRIStAL / Inria Lille-Nord Europe, France
arnaud.liefooghe@univ-lille.fr

Sébastien Verel
Univ. Littoral Côte d'Opale – LISIC Calais, France
verel@univ-littoral.fr

PPSN conference, Leiden, Sept. 7th, 2020.

Sources

Final version can be found :

http://fitness-landscape.com

or,

https://www-lisic.univ-littoral.fr/~verel/

Slides :

http://fitness-landscape.com/talks/ppsn2020-tuto-fitness-landscapes.pdf

and code :

http://fitness-landscape.com/RESEARCH/code-tutorial-ppsn2020.zip

Program for today

1. The Basics of Fitness Landscapes
2. Geometries of Fitness Landscapes
3. Local Optima Network
4. Multi-objective Fitness Landscapes

1. The Basics of Fitness Landscapes

Fitness Landscape Analysis
Understanding and Predicting Algorithm Performance
for Single- and Multi-objective Optimization

Outline

1. The Basics of Fitness Landscapes
   - Introductory example
   - Brief history and background
2. Geometries of Fitness Landscapes
3. Local Optima Network
4. Multi-objective Fitness Landscapes

Single-objective optimization

- Search space : set of candidate solutions

$$X$$

- Objective fonction : quality criteria (or non-quality)

$$f : X \to \mathbb{R}$$

X discrete : combinatorial optimization
$X \subset \mathbb{R}^n$: numerical optimization

Solve an optimization problem (maximization)

$$X^\star = \mathrm{argmax}_X \ f$$

or find an approximation of $X^\star$.

Context : black-box optimization

$x \longrightarrow$ $\longrightarrow f(x)$

No information on the objective function definition f

Objective fonction :

- can be irregular, non continuous, non differentiable ...
- given by a computation or a simulation

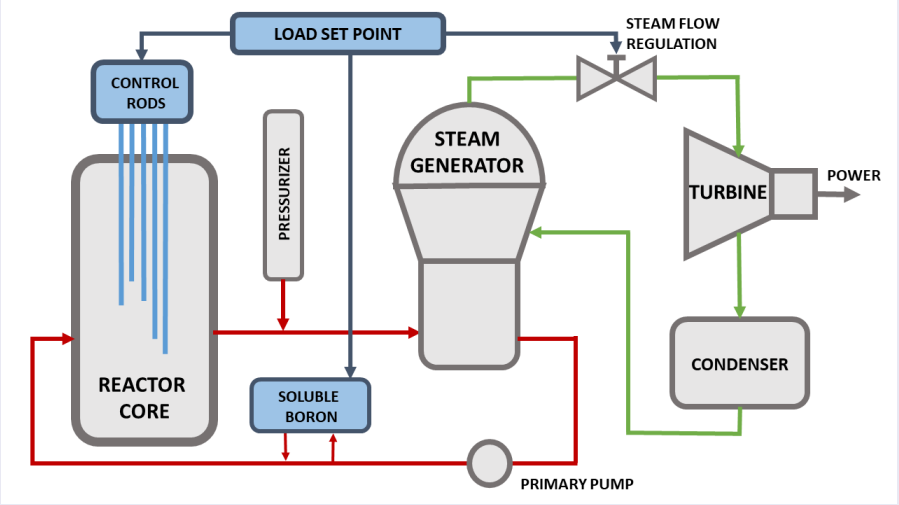
Real-world black-box optimization : an example

PhD of Valentin Drouet, Saclay Nuclear Research Centre (CEA), Paris

$x \longrightarrow$

$\longrightarrow f(x)$

$(73, \ldots, 8) \longrightarrow$



Multi-physic simulator

$\longrightarrow \Delta_z P$

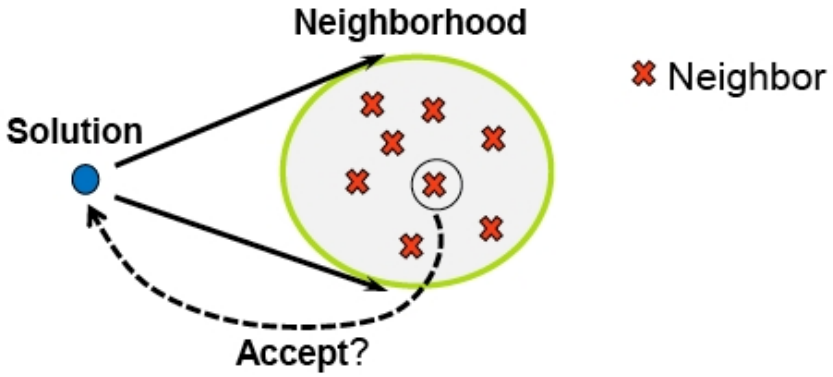
Search algorithms

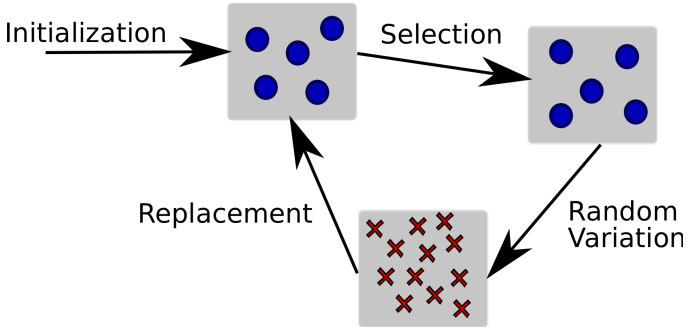
Principle

(implicite) **enumeration of a subset of the search space**

- Many ways to enumerate the search space
  - Exact methods : $A^\star$, Branch&Bound ...
  - Random sampling : Monte Carlo, approximation with guarantee, bayesian optimization, ...

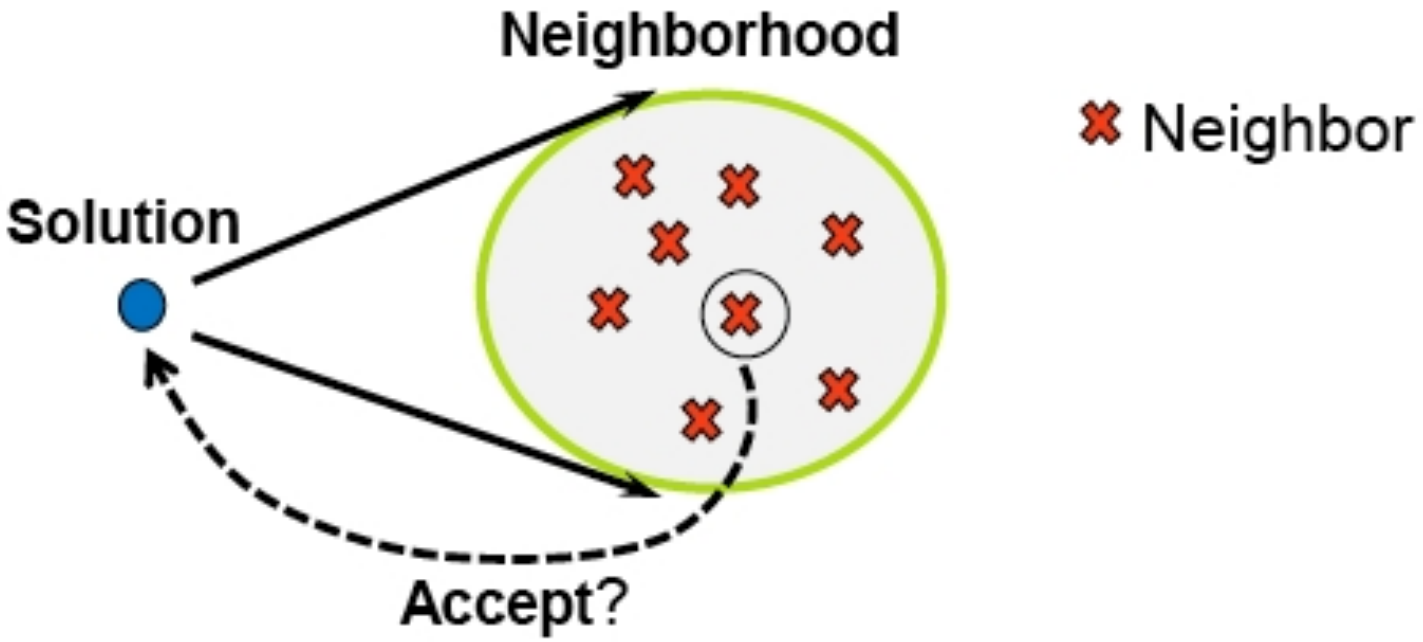
Local search / Evolutionary algorithms





Stochastic algorithms with a single solution (Local Search)

- X set of candidate solutions (the search space)
- $f : X \to \mathbb{R}$ objective function
- $\mathcal{N}(x)$ set of neighboring solutions from x



So, we need a tool to study this...

Motivations on fitness landscape analysis

For the search to be efficient, the sequence of local optimization problems must be related to the global problem

Main motivation : "Why using local search"

- Study the search space from the point of view of local search
  $\Rightarrow$ Fitness Landscape Analysis
- To understand and design effective local search algorithms

Fitness landscape : original plots from S. Wright [Wri32]

S. Wright. "The roles of mutation, inbreeding, crossbreeding, and selection in evolution.", 1932.

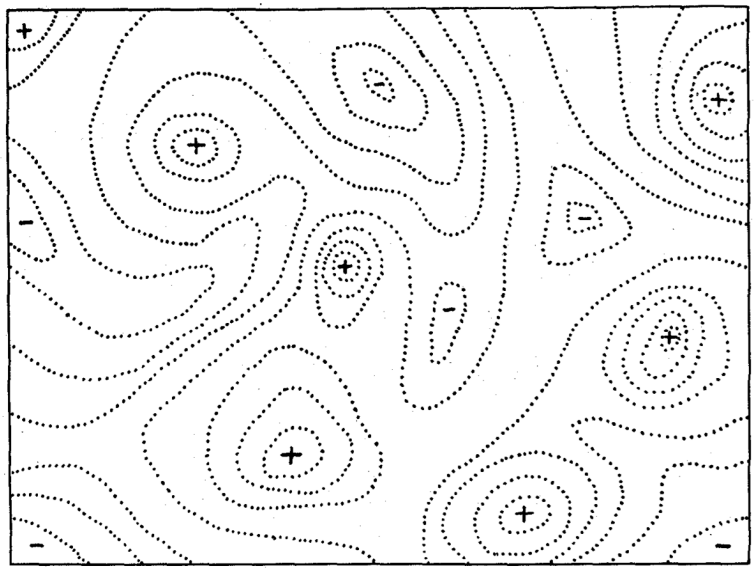


FIGURE 2.—Diagrammatic representation of the field of gene combinations in two dimensions instead of many thousands. Dotted lines represent contours with respect to adaptiveness.

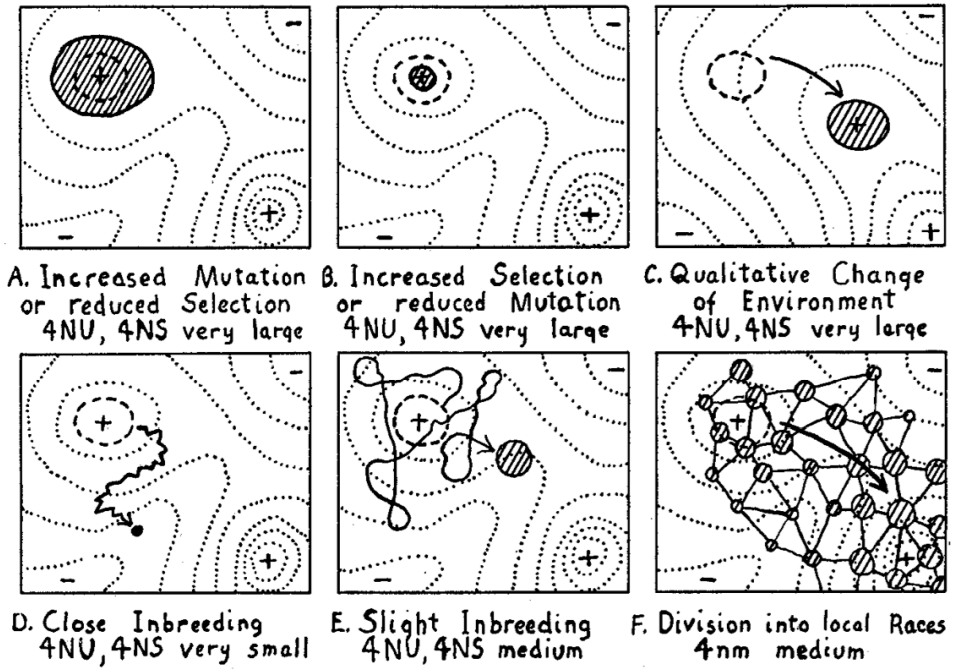
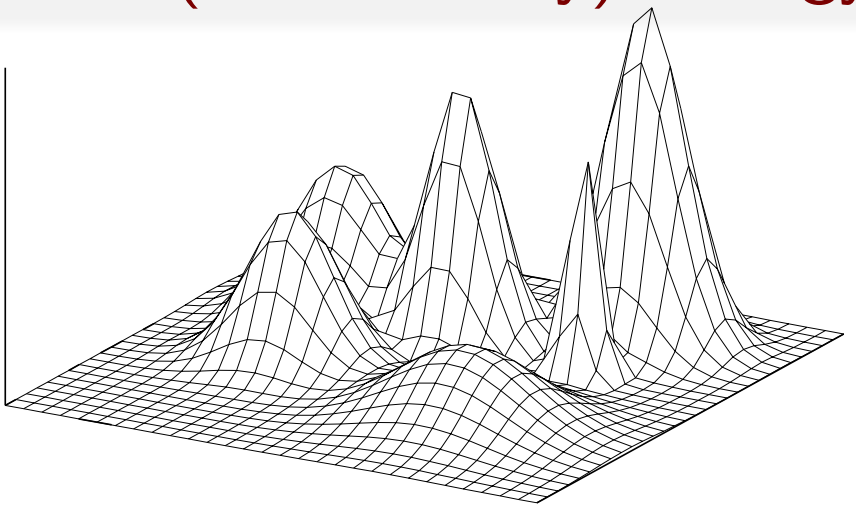


FIGURE 4.—Field of gene combinations occupied by a population within the general field of possible combinations. Type of history under specified conditions indicated by relation to initial field (heavy broken contour) and arrow.

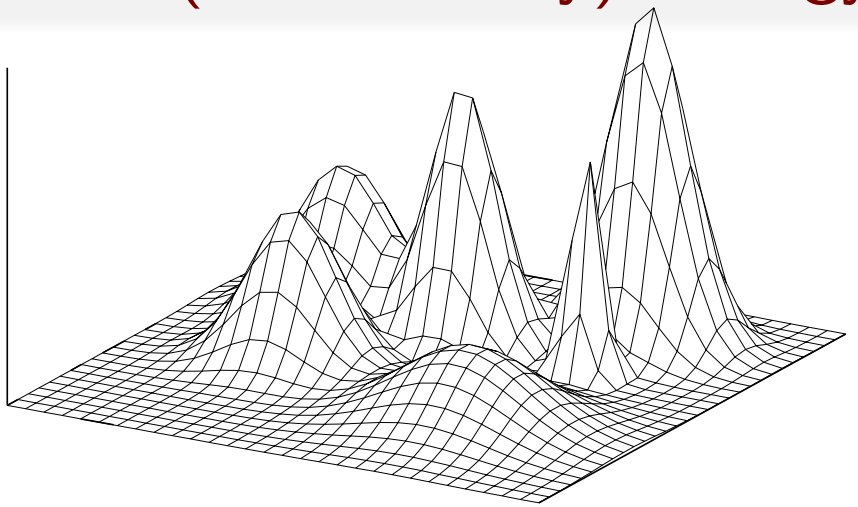
source : Encyclopaedia Britannica Online

Fitness landscapes in (evolutionary) biology



- Metaphorical uphill struggle across a "fitness landscape"
  - mountain **peaks** represent high "fitness"
    (ability to survive/reproduce)
  - **valleys** represent low fitness
- Evolution proceeds :
  population of organisms
  performs an "**adaptive walk**"

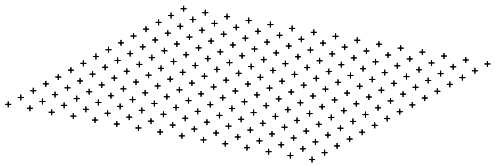
Fitness landscapes in (evolutionary) biology

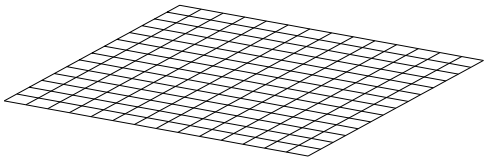


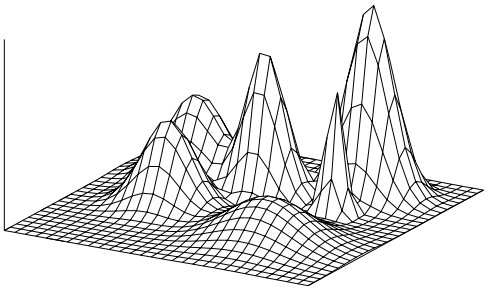
- Metaphorical uphill struggle across a "fitness landscape"
  - mountain **peaks** represent high "fitness" (ability to survive/reproduce)
  - **valleys** represent low fitness
- Evolution proceeds :
  population of organisms
  performs an "**adaptive walk**"

be careful : "2 dimensions instead of many thousands"

Fitness landscapes in biology and others sciences







In biology :

- model of species evolution

Extended to model dynamical systems :

- statistical physic
- molecular evolution
- ecology
- ...

Fitness landscapes in biology

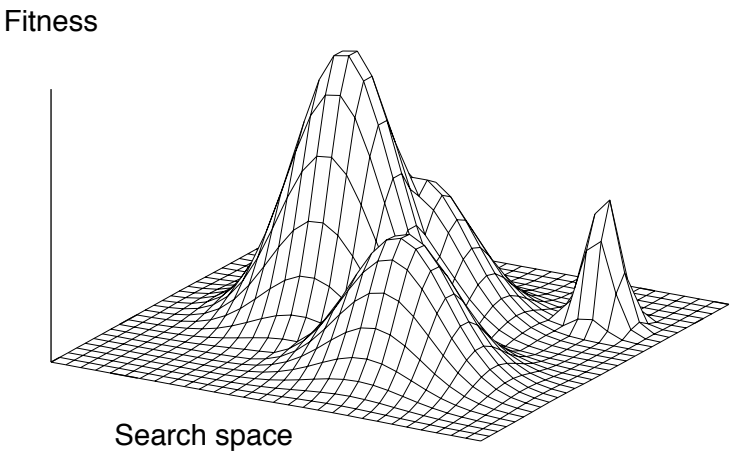
2 sides of Fitness Landscapes

- **Metaphor** : most profound concept in evolutionary dynamics
  - give pictures of evolutionary process
  - be careful of misleading pictures :
    "smooth low-dimensional landscape without noise"
- **Quantitative** concept : predict the evolutionary paths

$$X \longrightarrow X$$

  - Quasispecies equation : mean field analysis
    x_t
  - Stochastic process : Markov chain
    $\Pr(x_{t+1} \mid x_t)$
  - Individual scale : network analysis

Fitness landscape for combinatorial optimization [Sta02]

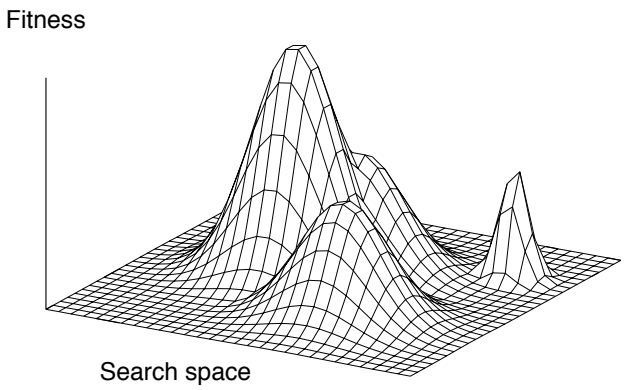


Definition

Fitness landscape $(X, \mathcal{N}, f)$:

- **search space** :
$$X$$
- **neighborhood relation** :
$$\mathcal{N} : X \to 2^X$$
- **objective function** :
$$f : X \to \mathbb{R}$$

What is a neighborhood ?



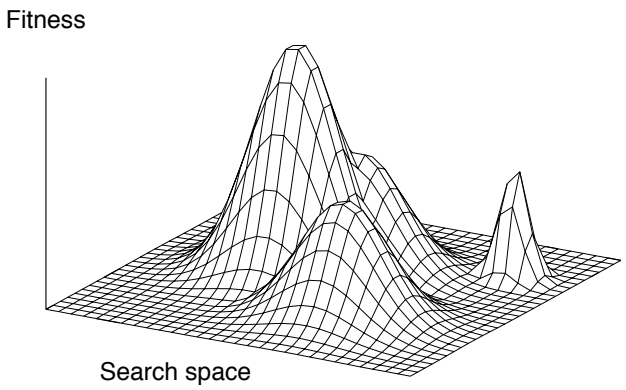
Neighborhood function :

$$\mathcal{N} : X \rightarrow 2^X$$

Set of "**neighbor**" solutions associated to each solution

$$\mathcal{N}(x) = \{y \in X \mid y = op(x)\}$$

What is a neighborhood ?



Neighborhood function :

$$\mathcal{N} : X \to 2^X$$

Set of "**neighbor**" solutions associated to each solution

$$\mathcal{N}(x) = \{y \in X \mid y = op(x)\}$$

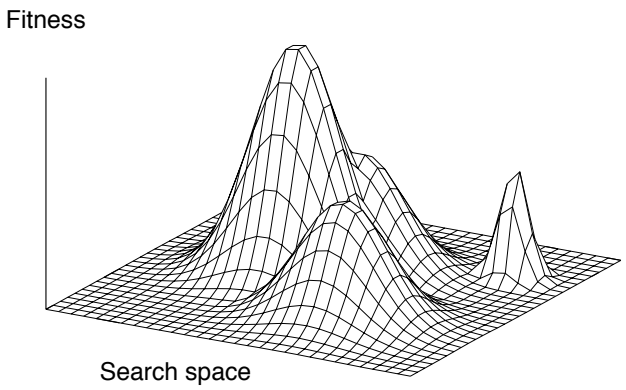
or

$$\mathcal{N}(x) = \{y \in X \mid \Pr(y = op(x)) > 0\}$$

or

$$\mathcal{N}(x) = \{y \in X \mid \Pr(y = op(x)) > \varepsilon\}$$

What is a neighborhood ?



Neighborhood function :

$$\mathcal{N} : X \to 2^X$$

Set of "**neighbor**" solutions associated to each solution

$$\mathcal{N}(x) = \{y \in X \mid y = op(x)\}$$

or

$$\mathcal{N}(x) = \{y \in X \mid \Pr(y = op(x)) > 0\}$$

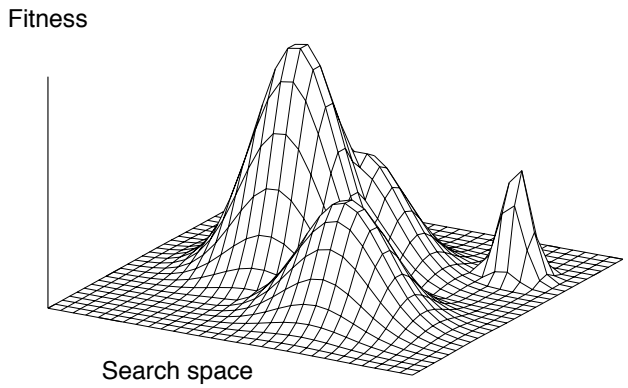
or

$$\mathcal{N}(x) = \{y \in X \mid \Pr(y = op(x)) > \varepsilon\}$$

or

$$\mathcal{N}(x) = \{y \in X \mid \text{distance}(x, y) = 1\}$$

What is a neighborhood ?



Neighborhood function :

$$\mathcal{N} : X \rightarrow 2^X$$

Set of "**neighbor**" solutions associated to each solution

Important !

Neighborhood must be based on the operator(s) used by the algorithm

Neighborhood $\Leftrightarrow$ Operator

$$\mathcal{N}(x) = \{y \in X \mid y = op(x)\}$$

or

$$\mathcal{N}(x) = \{y \in X \mid \Pr(y = op(x)) > 0\}$$

or

$$\mathcal{N}(x) = \{y \in X \mid \Pr(y = op(x)) > \varepsilon\}$$

or

$$\mathcal{N}(x) = \{y \in X \mid \text{distance}(x, y) = 1\}$$

Typical example : bit strings

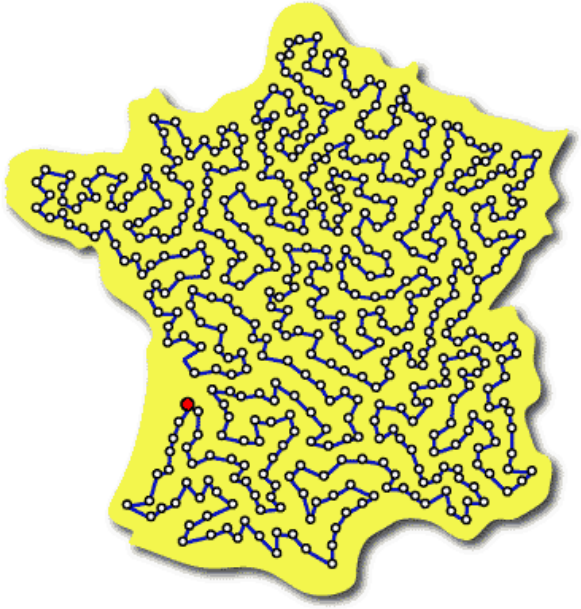
Search space : $X = \{0,1\}^N$

$$\mathcal{N}(x) = \{y \in X \mid d_{Hamming}(x,y) = 1\}$$

Example :
$\mathcal{N}(01101) = \{11101, 00101, 01001, 01111, 01100\}$

Typical example : permutations

Traveling Salesman Problem :
find the shortest tour which cross one time every town



Search space : $X = \{\sigma \mid \sigma \text{ permutations } \}$

$$\mathcal{N}(x) = \{y \in X \mid y = op_{2opt}(x)\}$$

More than 1 operator. . . ?

What can we do with 2 operators (ex : memetic algorithm) ?

$\mathcal{N}_1(x) = \{y \in X \mid y = op_1(x)\}$ $\mathcal{N}_2(x) = \{y \in X \mid y = op_2(x)\}$

More than 1 operator. . . ?

What can we do with 2 operators (ex : memetic algorithm) ?

$\mathcal{N}_1(x) = \{y \in X \mid y = op_1(x)\}$ $\mathcal{N}_2(x) = \{y \in X \mid y = op_2(x)\}$

Severals possibilities according to the goal :

- Study 2 landscapes : $(X, \mathcal{N}_1, f)$ and $(X, \mathcal{N}_2, f)$
- Study the landscape of "union" : $(X, \mathcal{N}, f)$

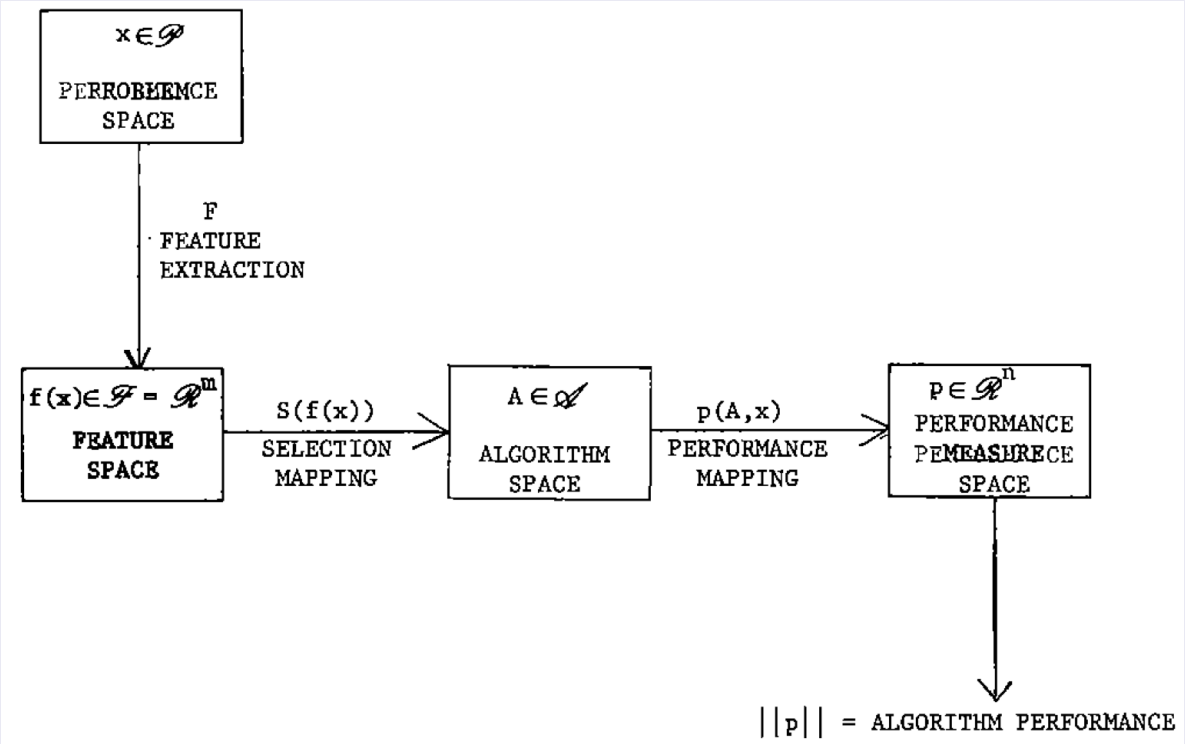
$$\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2 = \{y \in X \mid y = op_1(x) \text{ or } y = op_2(x)\}$$

- Study the landscape of "composition" : $(X, \mathcal{N}, f)$

$$\mathcal{N} = \{y \in X \mid y = op \circ op^{'}(x) \text{ with } op, op^{'} \in \{id, op_1, op_2\}\}$$

Rice's framework for algorithm selection [Ric76]

Algorithm selection



Rice, J. R. (1976). The algorithm selection problem. Advances in computers, 15, 65-118.

Fitness landscape analysis

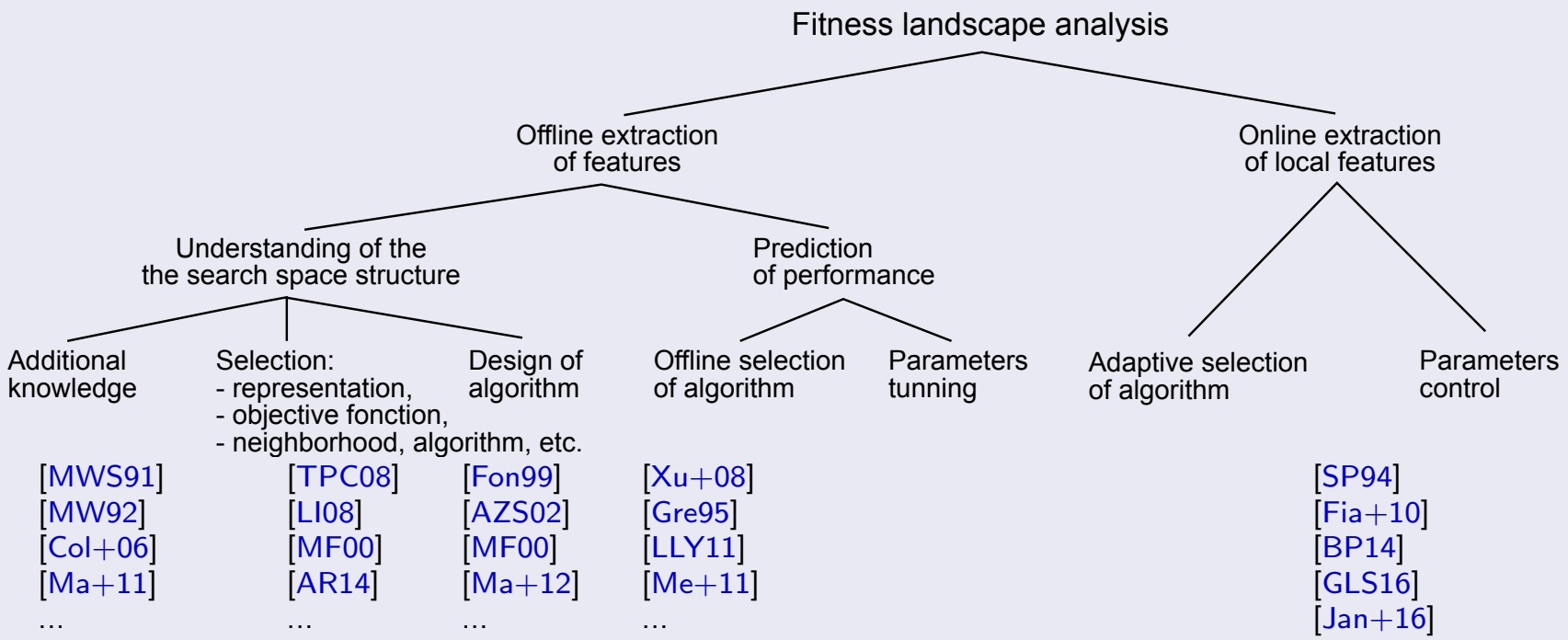
Algebraic approach, grey-box :

$$\Delta f = \lambda.(f - \bar{f})$$

Statistical approach, black-box :

Problems $\rightsquigarrow$ Features
$\rightsquigarrow$ Algorithm $\rightsquigarrow$ Performances

Goals



J. J. Grefenstette, in FOGA 3, 1995.[Gre95]

"Predictive Models Using Fitness Distributions of Genetic Operators"

"An important goal of the theory of genetic algorithms is to build **predictive models** of how well genetic algorithms are expected to perform, given a representation, a **fitness landscape**, and a set of genetic operators. (...)"

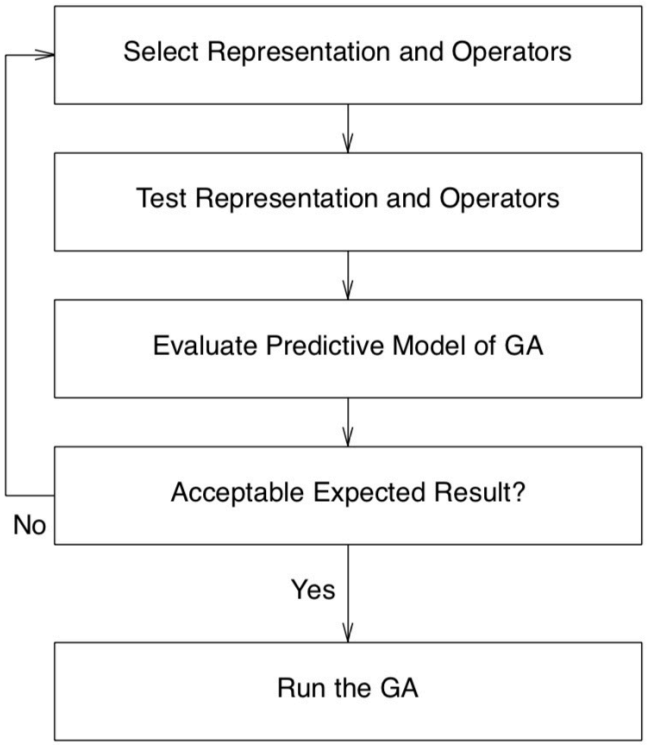
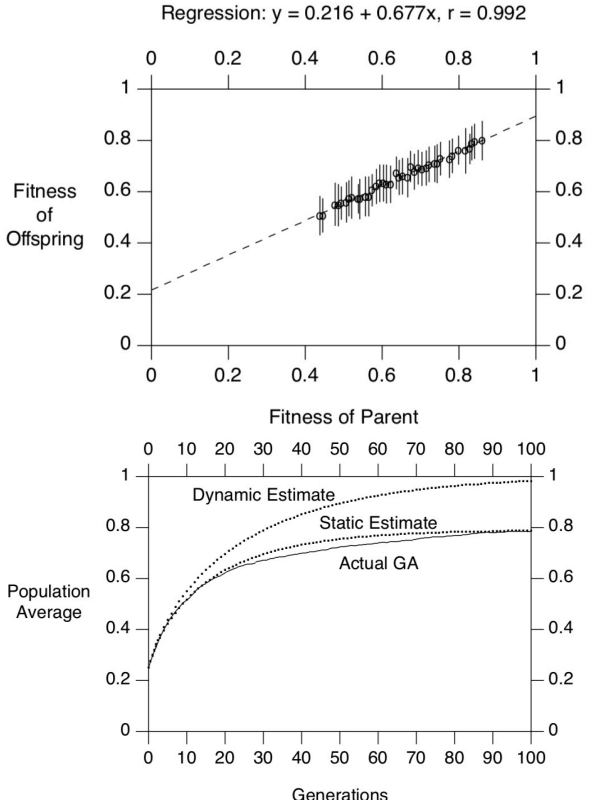


Figure 1: Predicting GA Performance

$FD_{op}(F_p) = Prob\,(F_c = \text{fitness of offspring} \mid \text{parents have mean fitness } F_p)$



Typical use cases of fitness landscapes analysis

1. **Comparing the difficulty of two landscapes** :
   - one problem, different encodings : $(X_1, \mathcal{N}_1, f_1)$ vs. $(X_2, \mathcal{N}_2, f_2)$ different representations, variation operators, objectives ...

   Which landscape is easier to solve ?

2. **Choosing one algorithm** :
   - analyzing the global geometry of the landscape

   Which algorithm shall I use ?

3. **Tuning the algorithm's parameters** :
   - *off-line* analysis of the fitness landscape structure

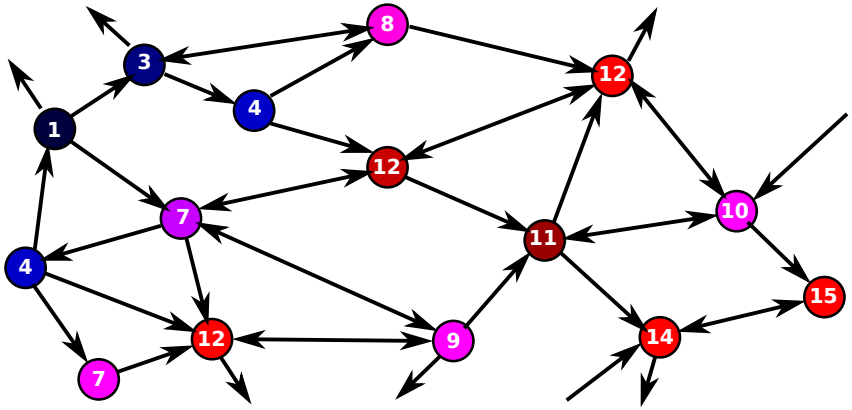
   What is the best mutation operator ? the size of the population ? the number of restarts ? ...

4. **Controlling the algorithm's parameters at runtime** :
   - *on-line* analysis of structure of fitness landscape

   What is the optimal mutation operator according to the current estimation of the structure ?

Back to the definition

Fitness landscape $(X, \mathcal{N}, f)$ is :
an oriented **graph** $(X, \mathcal{N})$ with weighted nodes given by f-values



Remarks :

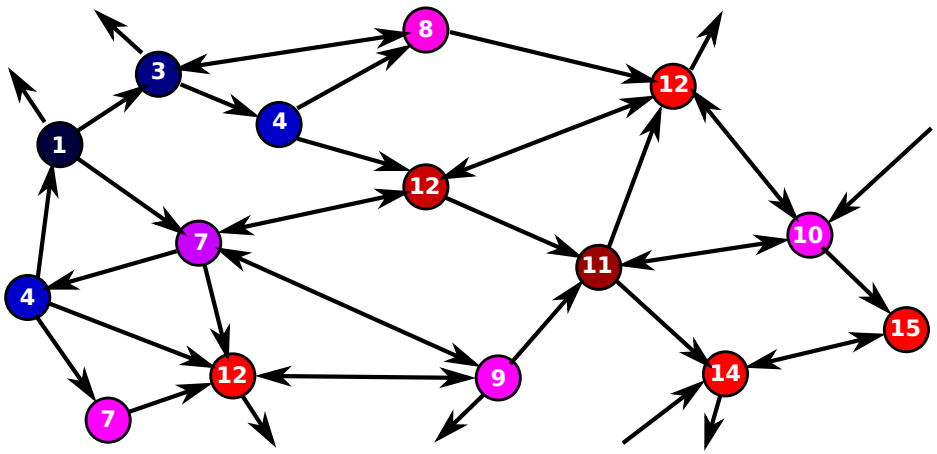
- Model of the search space
- Not specific to a particular local search

Fitness landscape and complex systems

Complex system : local *vs.* global properties

- Sample the neighborhood to have information on **local features** of the search space
- From this local information, deduce **global feature** such as general shape, difficulty, performance, best algorithm ...

$\Rightarrow$ Analysis using complex systems tools



Short summary for this part

Studying the **structure** of the fitness landscape
allows to understand the **difficulty**,
and to **design** good optimization algorithms

The fitness landscape is a **graph** $(X, \mathcal{N}, f)$:

- nodes are solutions and have a value (the fitness)
- edges are defined by the neighborhood relation

pictured as a real landscape

References I

John J Grefenstette.
Predictive models using fitness distributions of genetic operators.
In *Foundations of Genetic Algorithms*, volume 3, pages 139–161. Elsevier, 1995.

John R. Rice.
The algorithm selection problem.
Advances in Computers, 15 :65–118, 1976.

P. F. Stadler.
Fitness landscapes.
In M. Lässig and Valleriani, editors, *Biological Evolution and Statistical Physics*, volume 585 of *Lecture Notes Physics*, pages 187–207, Heidelberg, 2002. Springer-Verlag.

References II

S. Wright.
The roles of mutation, inbreeding, crossbreeding, and selection in evolution.
In *Proceedings of the Sixth International Congress of Genetics 1*, pages 356–366, 1932.

2. Geometries of Fitness Landscapes

Fitness Landscape Analysis
Understanding and Predicting Algorithm Performance
for Single- and Multi-objective Optimization

Outline

1. ~~The Basics of Fitness Landscapes~~
2. Geometries of Fitness Landscapes
   - Ruggedness and multimodality
   - Neutrality
3. Local Optima Network
4. Multi-objective Fitness Landscapes

Metrics, features of fitness landscape

Main idea

The "shape" of the neighborhood (local description)
is related to
the dynamics of the local search, and its performance

Main questions

- How to design relevant metrics ?
- What are the meaning of the metrics (benefits, and caveats) ?
- How to estimate the metrics ?

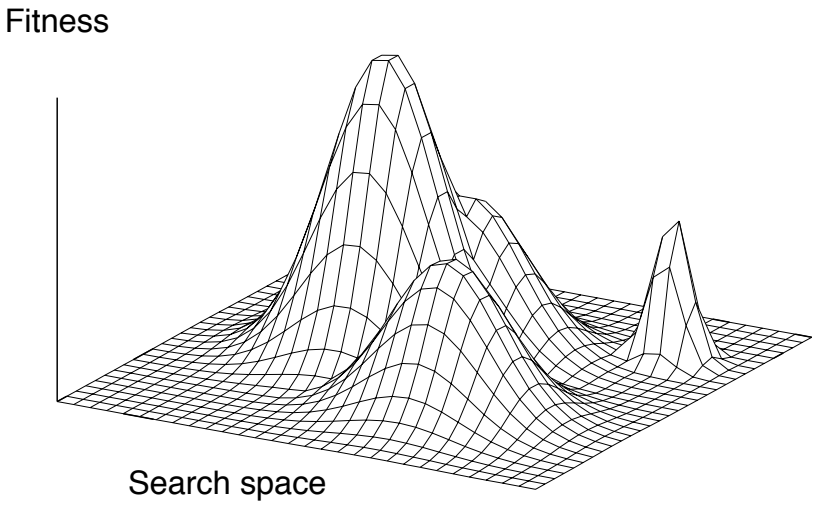
In the following, a (short) comprehensive list of metrics with their intuition, and their estimation

Multimodal fitness landscapes

Local optima $x^\star$

no neighboring solution with strictly better fitness value (maximization)

$$\forall x \in \mathcal{N}(x^\star), \quad f(x) \leqslant f(x^\star)$$



nota : If $\mathcal{N}$ is modified (distance, op), the local optima are modified

Typical example : bit strings

Search space : $X = \{0,1\}^N$

$$\mathcal{N}(x) = \{y \in X \mid d_{Hamming}(x,y) = 1\}$$

Example :
$x = 01101$ and $f_1(x) = f_2(x) = f_3(x) = 5$

	11101	00101	01001	01111	01100
f_1	4	2	3	0	3
f_2	2	3	6	2	3
f_3	1	5	2	2	4

Question

Is x is a local maximum for f_1, f_2, and/or f_3 ?

Sampling local optima

Basic estimator (Alyahya, K., & Rowe, J. E. 2016 [AR16])

Expected proportion of local optima :

Proportion of local optima in a sample of random solutions

- Complexity : $n \times |\mathcal{N}|$
- Pros :
  unbiased estimator
- Cons :
  poor estimation when expected proportion is lower than $1/n$

Sampling local optima by adaptive walks

Adaptive walk

$$(x_1, x_2, \ldots, x_\ell) \text{ such that } x_{i+1} \in \mathcal{N}(x_i) \text{ and } f(x_i) < f(x_{i+1})$$

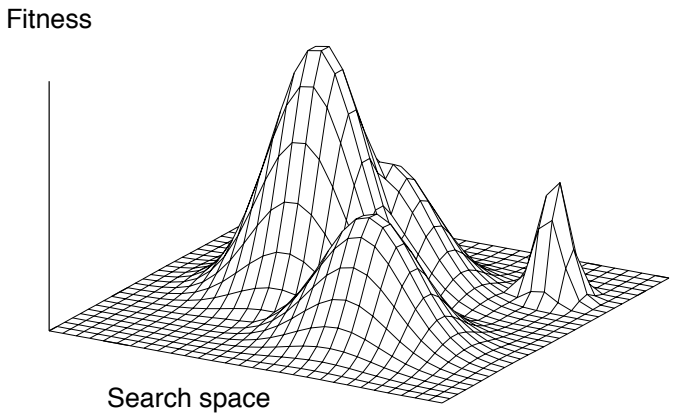
Hill-Climbing algorithm (first-improvement)

Choose initial solution $x \in X$
repeat
  choose $x' \in \{y \in \mathcal{N}(x) \mid f(y) > f(x)\}$
  if $f(x) < f(x')$ **then**
    $x \leftarrow x'$
  end if
until x is a Local Optimum

Basin of attraction of $x^\star$

$$\{x \in X \mid \mathit{HillClimbing}(x) = x^\star\}.$$

Multimodality and problem difficulty



The core idea :

- if the size of the basin of attraction of the global optimum is "small",
- then, the "time" to find the global optimum is "long"

Optimization difficulty :

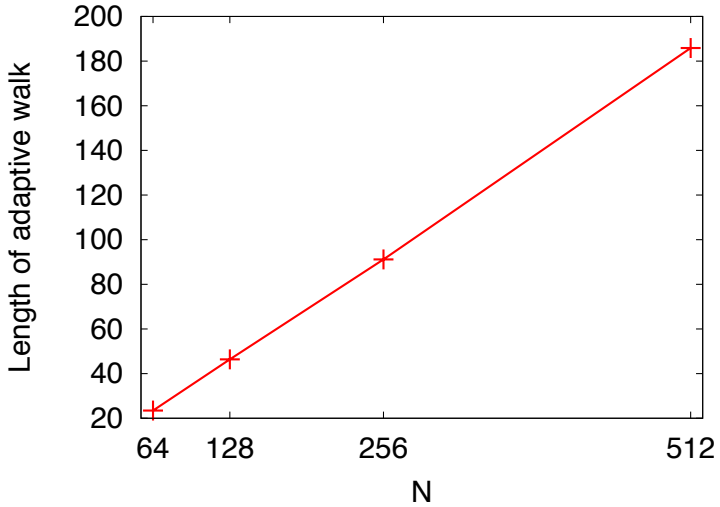
Number and size of the basins of attraction (Garnier *et al.* [GK02])

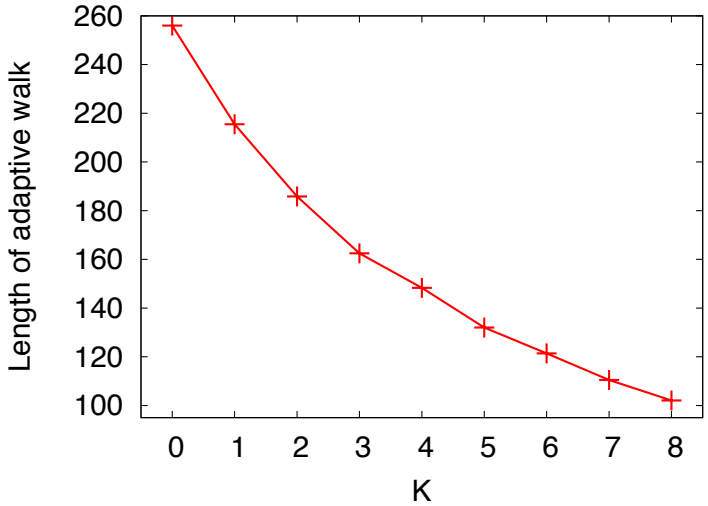
Feature to estimate the basins size :

- **Length of adaptive walks**

complexity : sample size $\times \ell \times |\mathcal{N}|$

Multimodality and problem difficulty





ex. nk-landscapes with $n = 512$

The core idea :

- if the size of the basin of attraction of the global optimum is "small",
- then, the "time" to find the global optimum is "long"

Optimization difficulty :

Number and size of the basins of attraction (Garnier *et al.* [GK02])

Feature to estimate the basins size :

- **Length of adaptive walks**

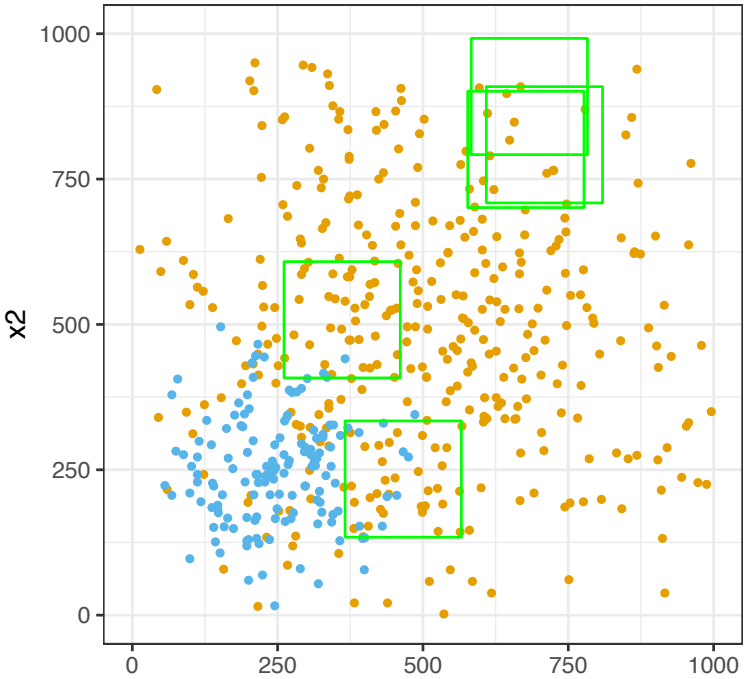
complexity : sample size $\times\, \ell \times |\mathcal{N}|$

Practice : the Squares Problem

a program design problem ?

Squares Problem (SP)

Find the position of 5 squares in order to maximize inside squares the number of brown points without blue points



Candidate solutions

$$X = ([0, 1000] \times [0, 1000])^5$$

	x_1	x_2
1	577	701
2	609	709
3	366	134
4	261	408
5	583	792

Fitness function

$f(x) =$ number of brown points $-$ number of blue points inside squares

Source code in R : ex01.R

Source code : `http://fitness-landscape.com/`

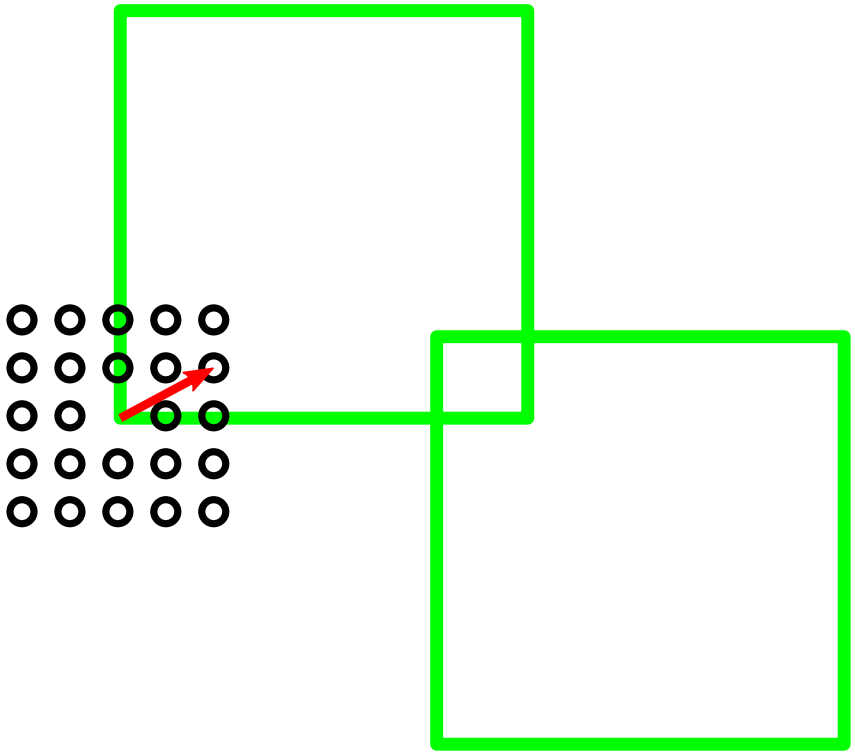
Different functions are already defined :

- `main` : example to execute the following functions
- `draw` and `draw_solution` :
  draw a problem and the squares of a solution
- `fitness_create` :
  create a fitness function from a data frame of points
- `pb1_create` and `pb2_create` :
  create two particular SP problems
- `init` :
  create a random solution with n squares
- `hc_ngh` :
  hill-climbing local search based on neighborhood

Neighborhood

Questions

- Execute line by line the `main` function
- Define the `neighborhood_create` which creates a neighborhood : a neighbor move one square



Adaptive walks to compare problem difficulty

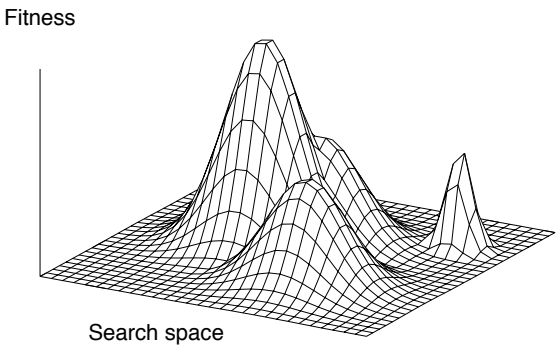
Pre-defined functions :

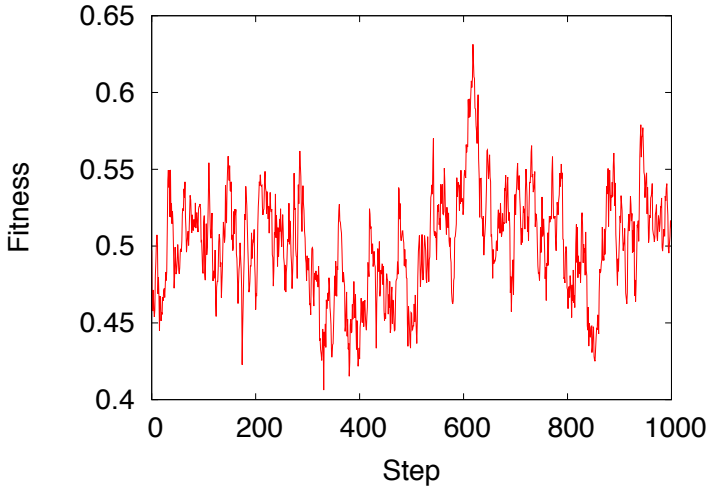
- `adaptive_length` :
  run the hill-climber and compute a data frame with the length of adaptive walks
- `main_adaptive_length_analysis` :
  Compute the adaptive length of two different SP problems

Questions

- Execute line by line the `main_adaptive_length_analysis` function to compute a sample of adaptive walk lengths
- Compare the lengths of adaptive walks for the two SP problems
- Which one is more multimodal ?

Random walk to estimate ruggedness





Random walk :

- $(x_1, x_2, \ldots)$ where $x_{i+1} \in \mathcal{N}(x_i)$ and equiprobability on $\mathcal{N}(x_i)$

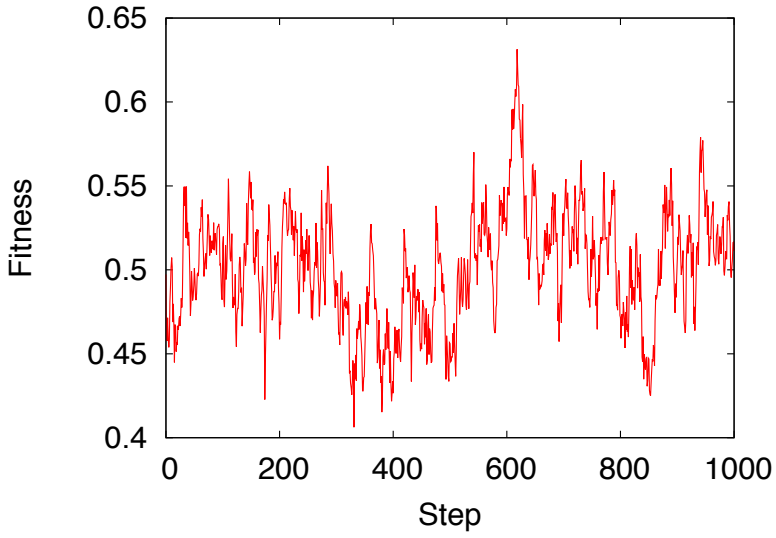
The idea :

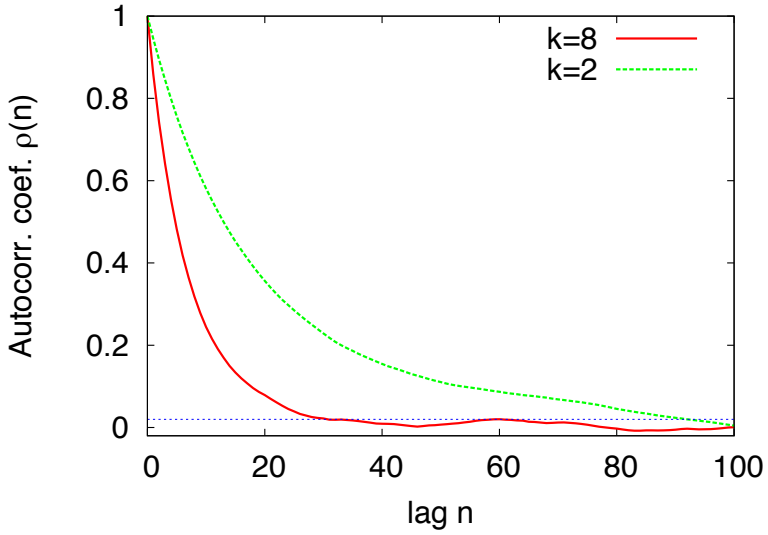
- if the profile of fitness is irregular,
- then the "information" between neighbors is low

Feature :

- Study the fitness profile like a signal

Rugged/smooth fitness landscapes





Autocorrelation function of the time series of fitness-values along a random walk [Wei90] :

$$\rho(n) = \frac{\mathbb{E}[(f(x_i) - \bar{f})(f(x_{i+n}) - \bar{f})]}{\mathrm{Var}(f(x_i))}$$

Autocorrelation length $\tau = \frac{1}{\rho(1)}$

"How many random steps such that correlation becomes insignificant"

- small τ : rugged landscape
- long τ : smooth landscape

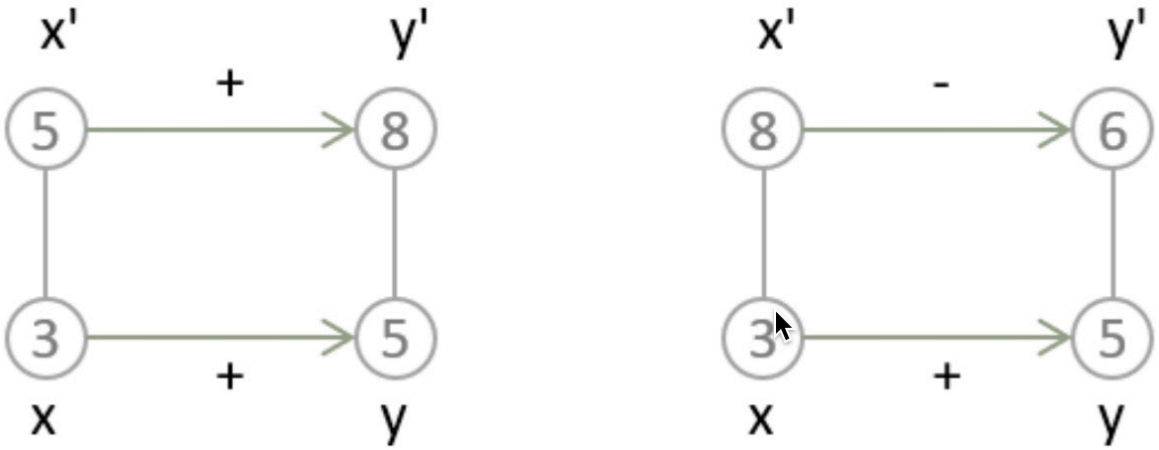
complexity : sample size $\approx 10^3$

Results on rugged fitness landscapes (Stadler 96 [Sta96])

Ruggedness decreases with the size of those problems

Problem	parameter	$\rho(1)$
symmetric TSP	n number of towns	$1 - \frac{4}{n}$
anti-symmetric TSP	n number of towns	$1 - \frac{4}{n-1}$
Graph Coloring Problem	n number of nodes α number of colors	$1 - \frac{2\alpha}{(\alpha-1)n}$
NK landscapes	N number of proteins K number of epistasis links	$1 - \frac{K+1}{N}$
random max-k-SAT	n number of variables k variables per clause	$1 - \frac{k}{n(1-2^{-k})}$

Rugged/smooth fitness landscapes : sign epistasis



Degree of epistasis :

Ratio of "negative" square

References :

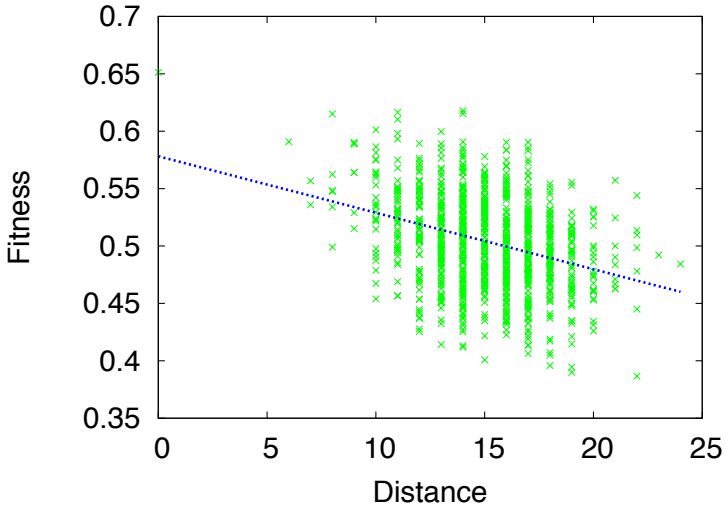
Biology : Poelwijk *et al.* [PKWT07]

EA : Basseur *et al.* [BG15]

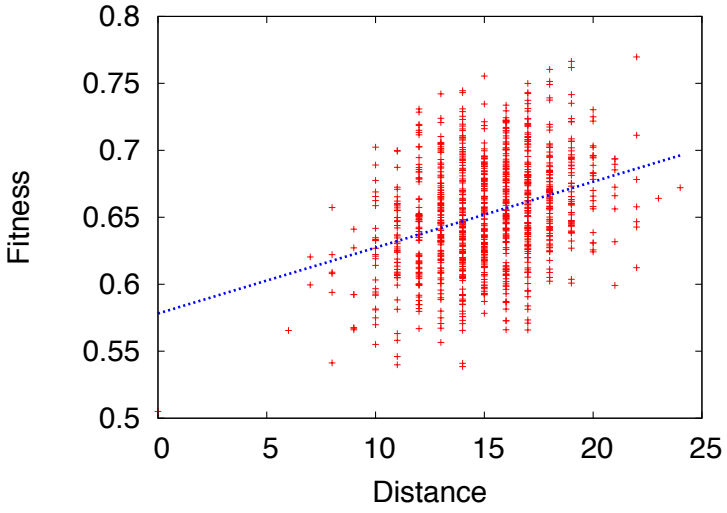
complexity : sample size $\approx 2.10^3$

Fitness distance correlation (FDC) (Jones 95 [Jon95])

Correlation between fitness and distance to global optimum



"easy"



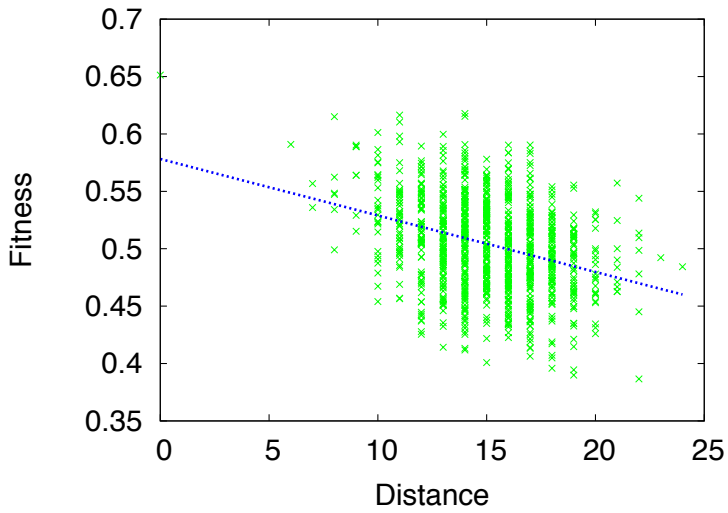
"hard"

Classification based on experimental studies

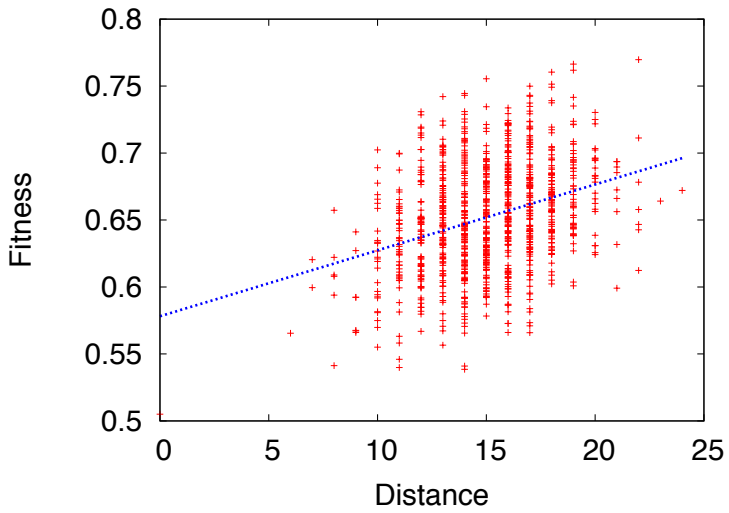
- $\rho < -0.15$: easy optimization
- $\rho > 0.15$: hard optimization
- $-0.15 < \rho < 0.15$: undecided zone

Fitness distance correlation (FDC) (Jones 95 [Jon95])

Correlation between fitness and distance to global optimum



"easy"



"hard"

- Important concept to understand search difficulty
- Not useful in "practice" (difficult to estimate)

Practice : computing the autocorrelation function

Source code `exo02.R` :

- `mutation_create` :
  Create a mutation operator,
  modify each square according to rate p,
  a new random value from $[(x-r, y-r), (x+r, y+r)]$.
- `main` :
  Code to obtain autocorrelation function

Questions

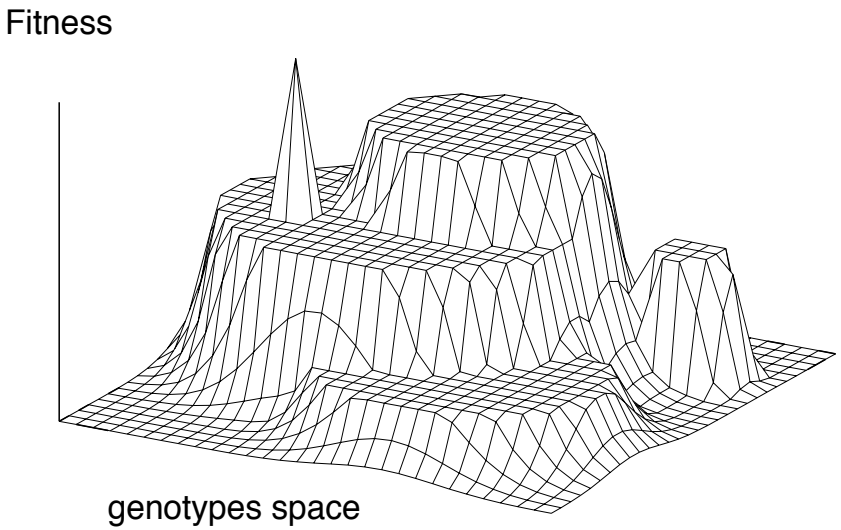
- Define the function `random_walk` to compute the fitness values during a random walk
- Execute line by line the `main` function to compute a sample of fitness value collected during a random walk
- Compare the first autocorrelation coefficient of the SP problems 1 and 2

Neutral fitness landscapes

Neutral theory (Kimura ≈ 1960 [Kim83])

Theory of mutation and random drift

Many mutations have no effects on fitness-values



- plateaus
- neutral degree
- neutral networks [Schuster 1994 [SFSH94], RNA folding]

Objects of neutral fitness landscapes

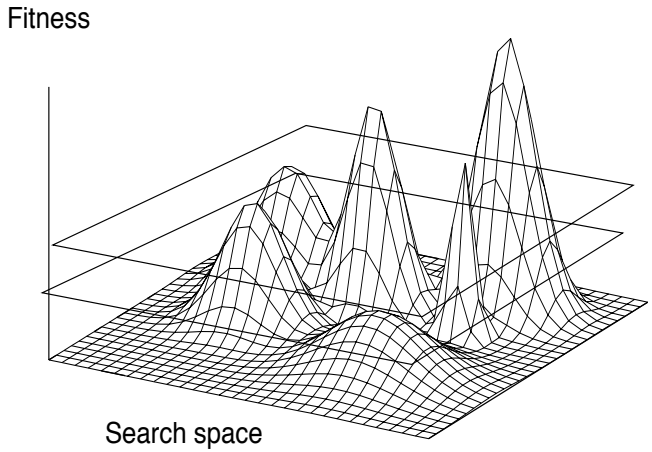
Description of multimodal fitness landscapes is based on :

- Local optima
- Basins of attraction

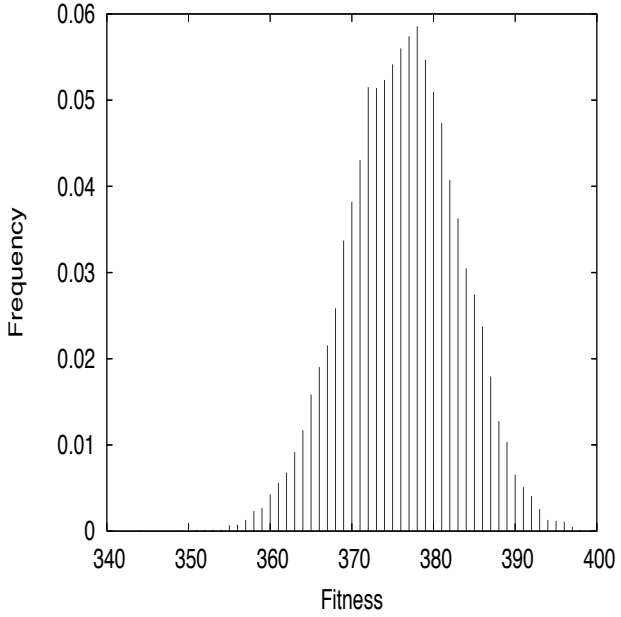
Description of neutral fitness landscapes is based on :

- **Neutral sets** :
  set of solutions with the same fitness
- **Neutral networks** :
  neutral sets with neighborhood relation

Neutral sets : density of states



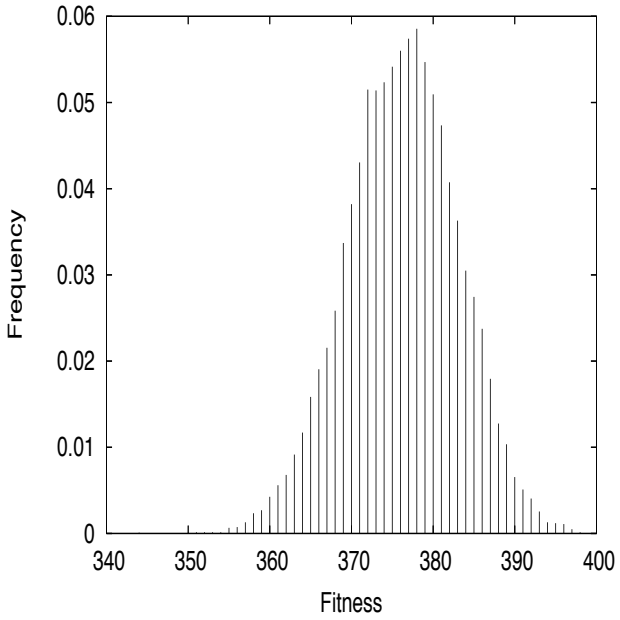
Set of solutions with same fitness



Density of states (D.O.S.)

- Introduced in physics (Rosé 1996 [REA96])
- Optimization (Belaidouni, Hao 00 [BH00])

Neutral sets : density of states



Informations given :

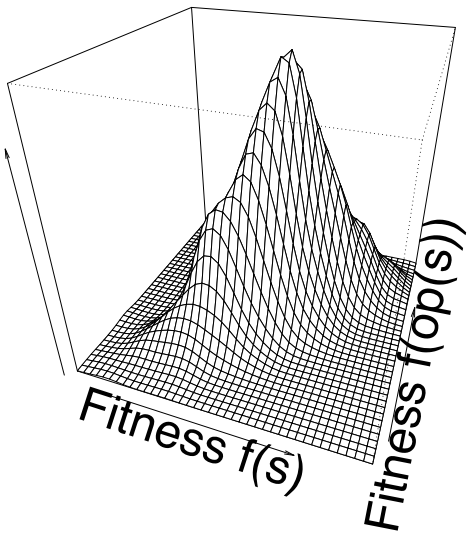
- Performance of random search
- Tail of the distribution is an indicator of difficulty :
  - the faster the decay, the harder the problem
- But do not care about the neighborhood relation

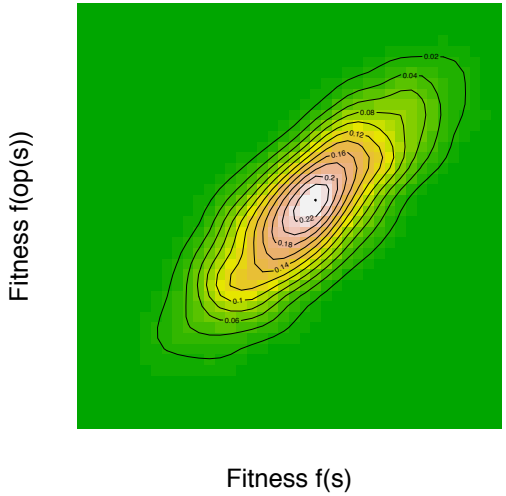
Features :

- **Average**, **sd**, **kurtosis** . . .

complexity : sample size

Neutral sets : fitness cloud [Verel et al. 2003]





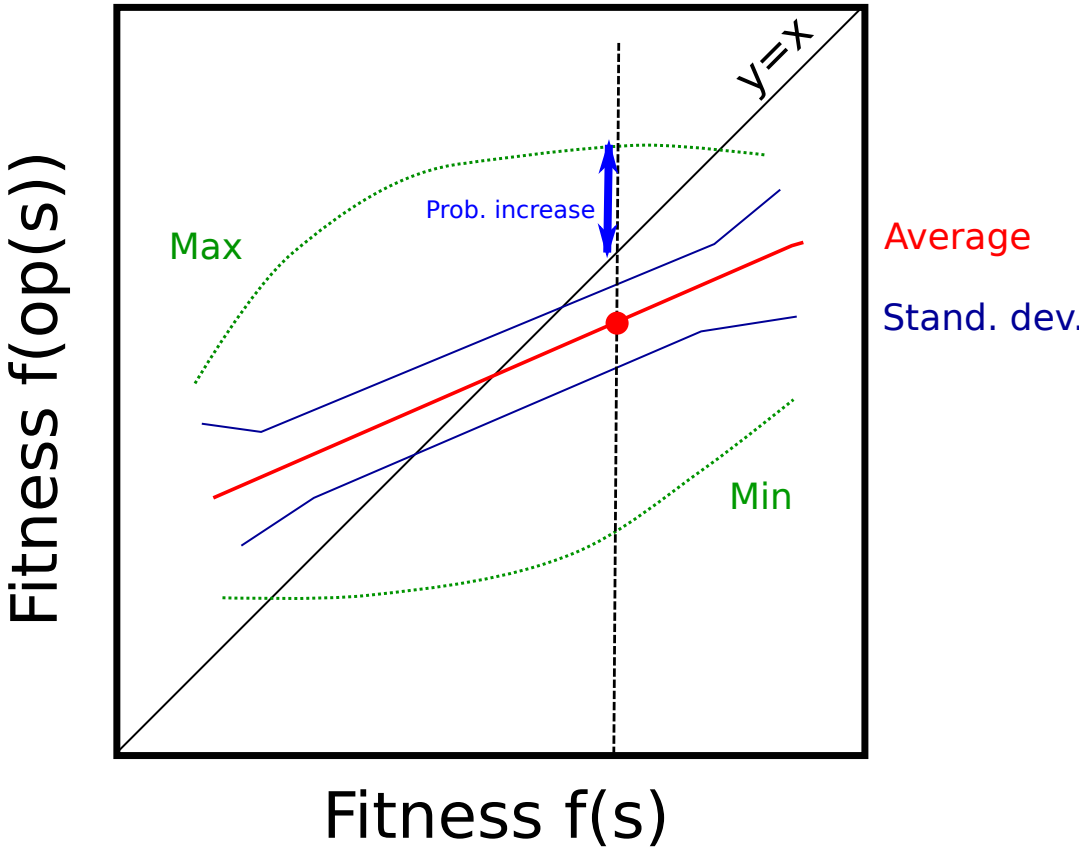
- $(X, \mathcal{F}, \Pr)$: probability space
- $op : X \to X$ stochastic operator of the local search

- $X(s) = f(s)$
- $Y(s) = f(op(s))$

Fitness Cloud of *op*

Conditional probability density function of Y given X

Fitness cloud : a measure of evolvability



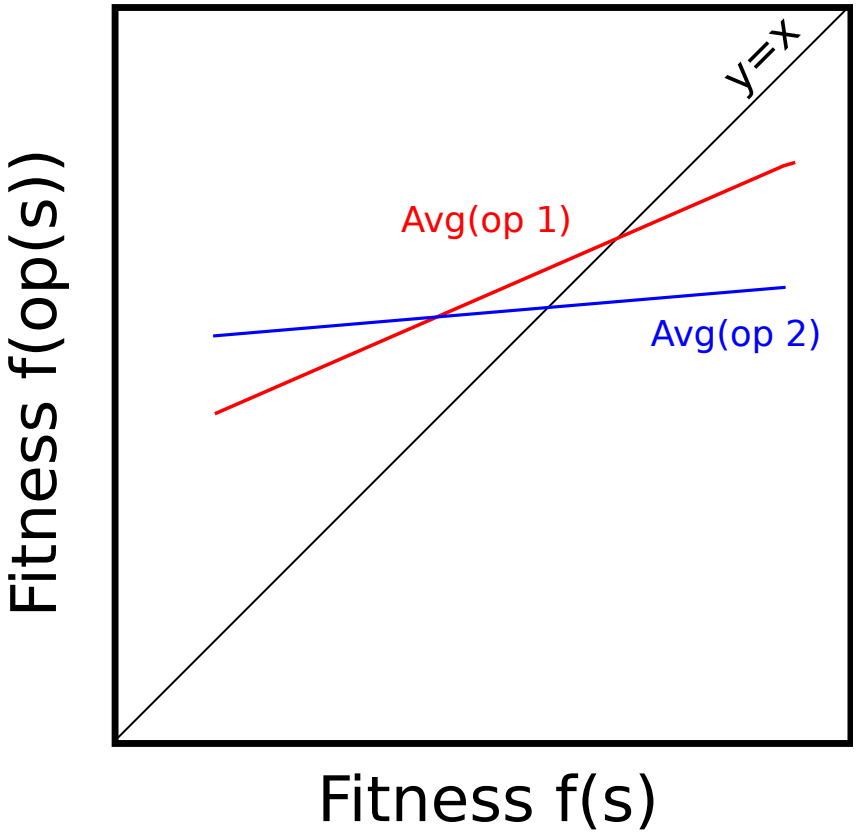
Evolvability

Ability to evolve : fitness in the neighborhood vs fitness of current solution

- Probability of finding better solutions
- Average fitness of better neighbors
- Average and standard dev. of fitness-values

Fitness cloud : comparing difficulty

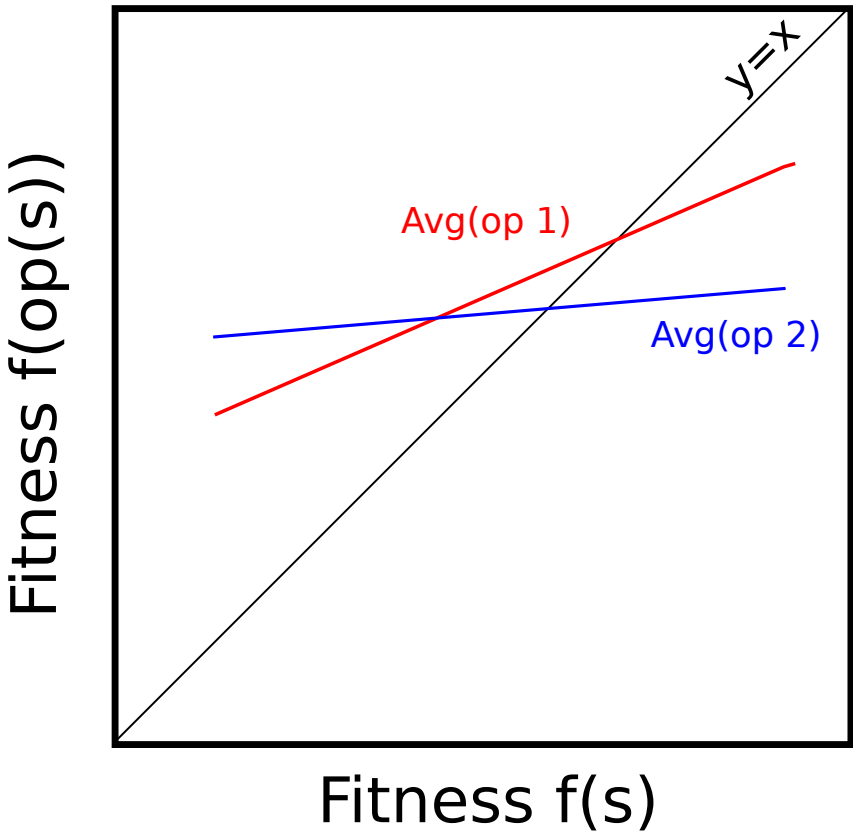
Average of evolvability



- Operator 1 ? ? Operator 2

Fitness cloud : comparing difficulty

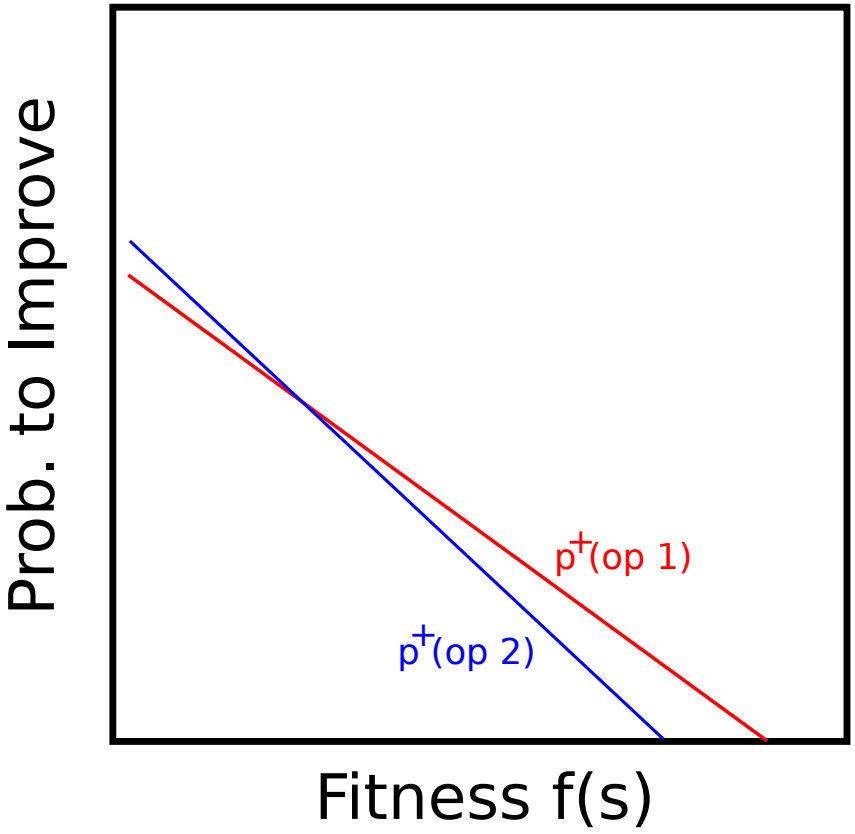
Average of evolvability



- Operator 1 > Operator 2
- Because Average 1 more correlated with fitness
- Linked to autocorrelation
- Average is often a line :
  - See works on Elementary Landscapes (Stadler, D. Wihtley, F. Chicano and others)
  - See the idea of Negative Slope Coefficient (NSC)

Fitness cloud : comparing difficulty

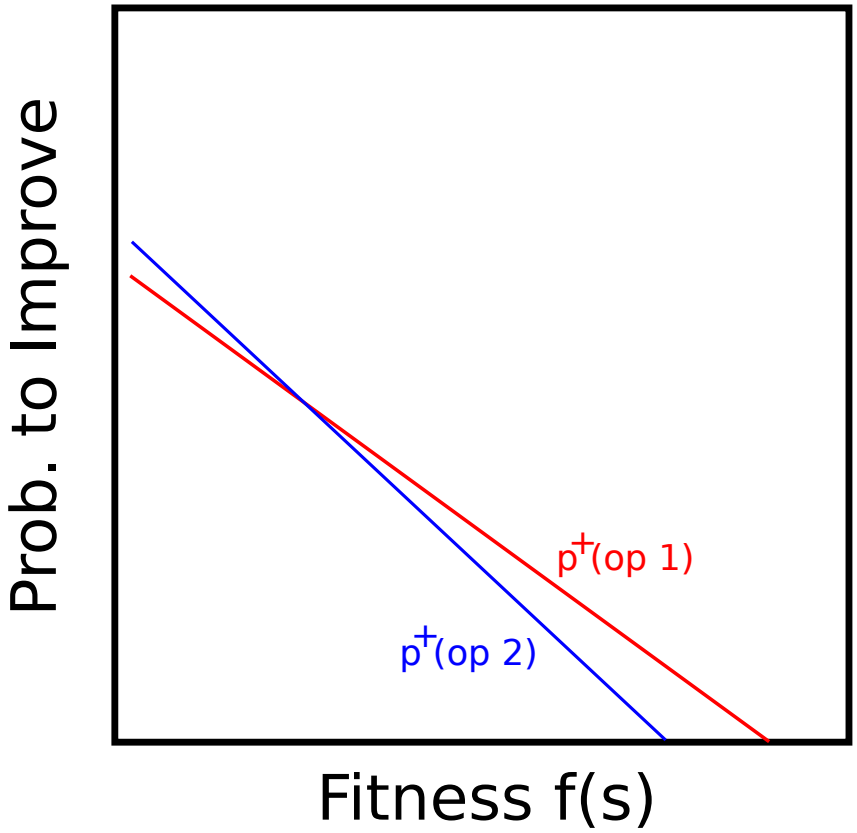
Probability to improve



- Operator 1 ? ? Operator 2

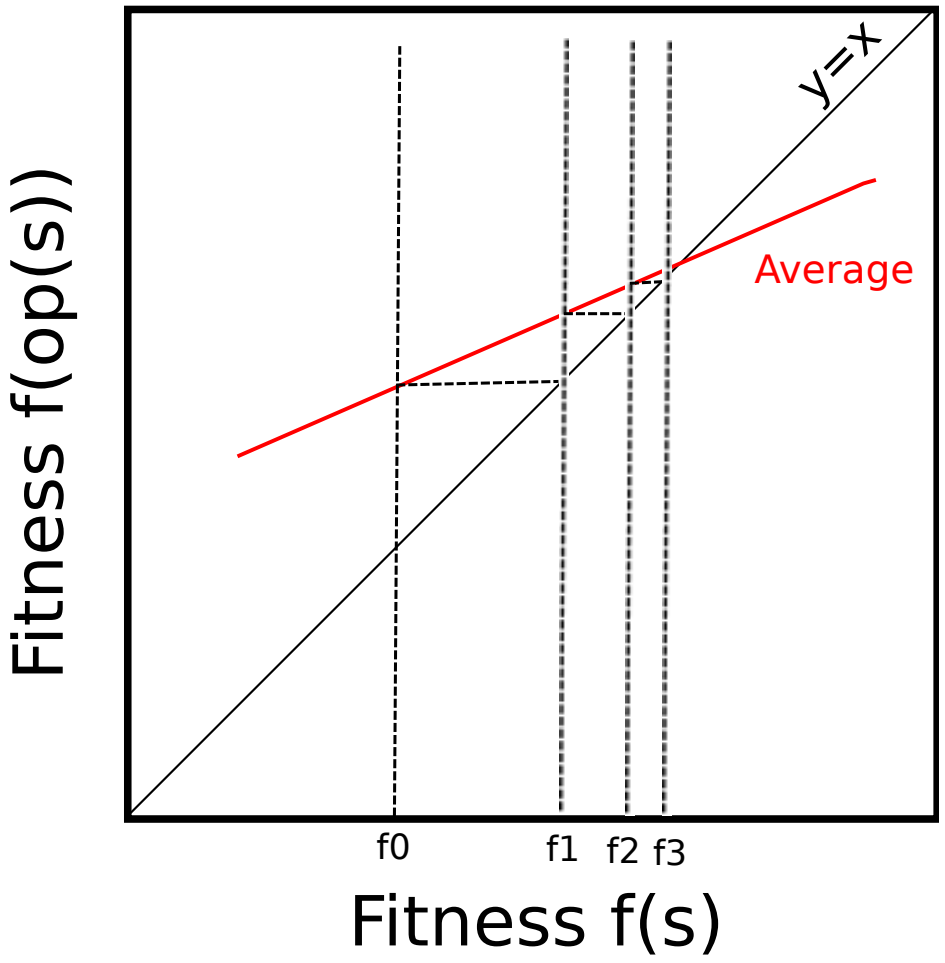
Fitness cloud : comparing difficulty

Probability to improve



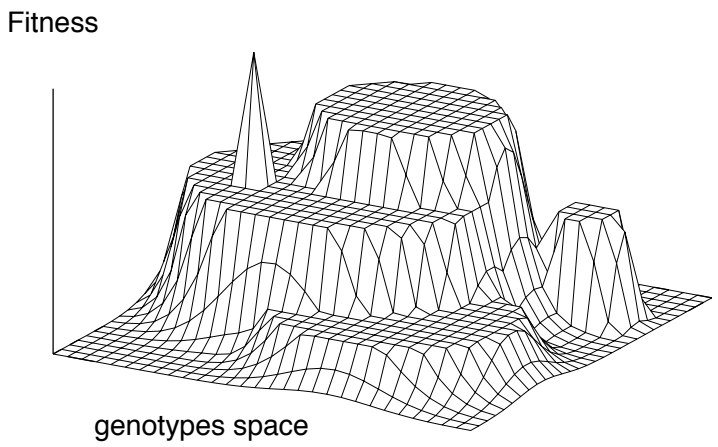
- Operator 1 > Operator 2
- Prob. to improve of Op 1 is often higher than Prob. to improve of Op 2
- Probability to improve is often a line
- See also works on fitness-probability cloud (G. Lu, J. Li, X. Yao [LLY11])
- See theory of EA and fitness level technics

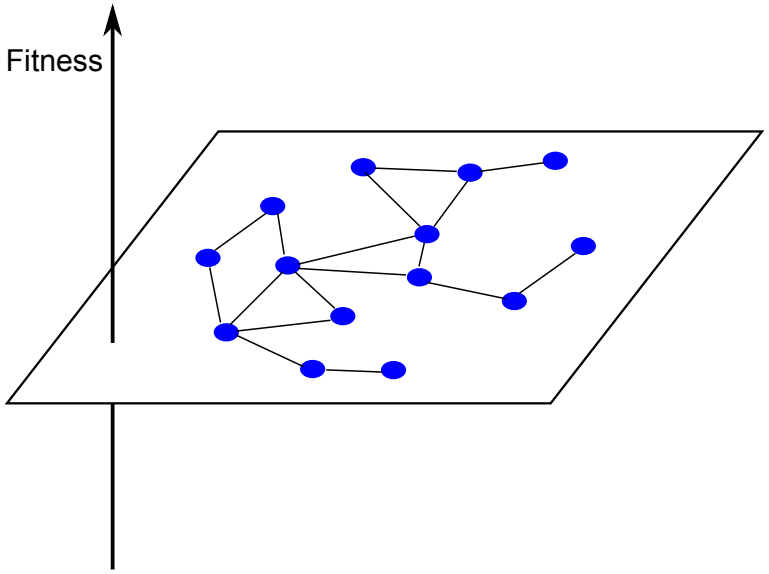
Fitness cloud : estimating the convergence point



- Approximation (only approximation) of the fitness value after few steps of local operator
- Indication on the quality of the operator
- See fitness level technic

Neutral networks (Schuster 1994 [SFSH94])





Basic definition of Neutral Network

- Node = solution with the same fitness-value
- Edge = neighborhood relation

Neutral degree

Neutral neighborhood

Set of neighbors which have the same fitness value

$$\mathcal{N}_{neutral}(x) = \{x' \in \mathcal{N}(x) \mid f(x') = f(x)\}$$

Nota : $f(x') = f(x)$ can be replaced by $|f(x') - f(x)| \leq \varepsilon$.

Neutral degree

Number of neutral neighbors : $\sharp\mathcal{N}_{neutral}(x)$

Neutral rate

Relative number of neutral neighbors : $\frac{\sharp\mathcal{N}_{neutral}(x)}{\sharp\mathcal{N}(x)}$

Estimation of the neutral rate with random walk

- The neutral rate can be estimated with a random walk :

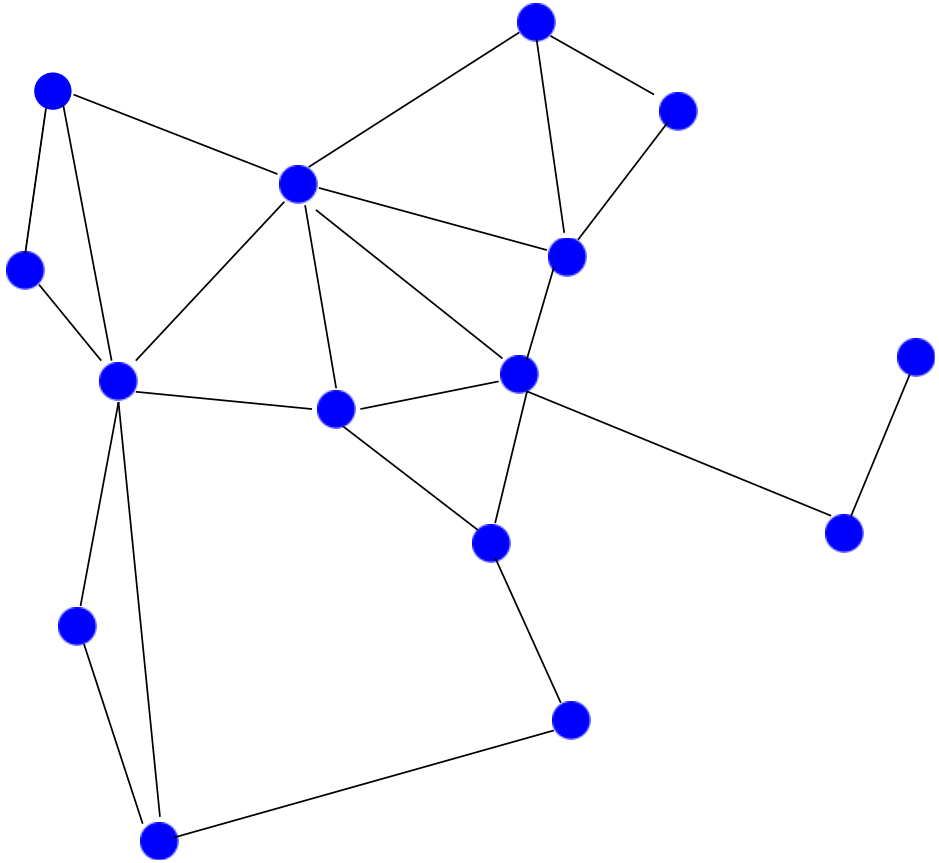
$$(x_1, x_2, \ldots, x_\ell) \text{ where } x_{t+1} \in \mathcal{N}(x_t)$$

Neutral rate estimation [evoCOP 17] [LDV+17]

$$\frac{\sharp\{(x_t, x_{t+1}) \ : \ f(x_t) = f(x_{t+1}), t \in \{1, \ell - 1\}\}}{\ell - 1}$$

Nota : With single random walk, fitness distribution, autocorrelation of fitness, probability of improvement, neutral rate can be estimated

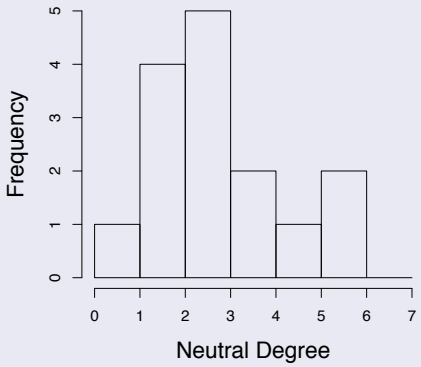
Neutral network : other metrics



1. **Size**

   avg, distribution ...

2. **Neutral degree distribution**



3. **Autocorrelation of the neutral degree**
   - neutral random walk
   - autocorr. of degrees

4. **Evolvability metrics**,

Practice : computing the neutral rate

Source code `exo03.R` :

- `main` :
  Code to compute the neutral rates

Questions

- Define the function `neutral_rate` to compute the neutral rate estimated with a random walk
- Execute the `main` function to compute the neutral rate
- Compare the neutrality of the SP problems 1 and 2

Practice : Performance *vs.* fitness landscape features

Explain the performance of ILS with fitness landscape features ?

- 20 random SP problems have been generated : `pb_xx.csv`
- The performance of Iterated Local Search has been computed in `perf_ils_xx.csv` (30 runs)
- Goal : regression of ILS performance with fitness landscape features

Practice : Performance *vs.* fitness landscape features

Source code `exo04.R` :

- `fitness_landscape_features` :
  Compute the basic fitness landscape features
- `random_walk_samplings` :
  Random walk sampling on each problem (save into file)
- `fitness_landscape_analysis` :
  Compute the features for each problems
- `ils_performance` :
  Add the performance of ILS into the data frame
- `main` :
  Execute the previous functions

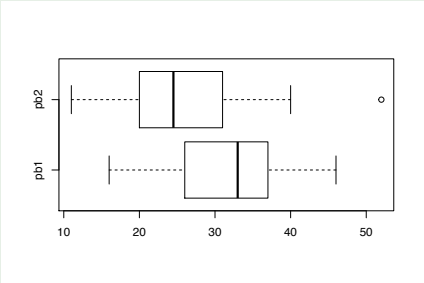
Practice : Performance *vs.* fitness landscape features

Questions

- What are the features computed by the function `fitness_landscape_features` ?
- Execute the `random_walk_samplings` function to compute the random walk samples
- Compute the correlation plots between features and ILS performance (use ggpairs)
- Compute the linear regression of performance with fitness landscape features

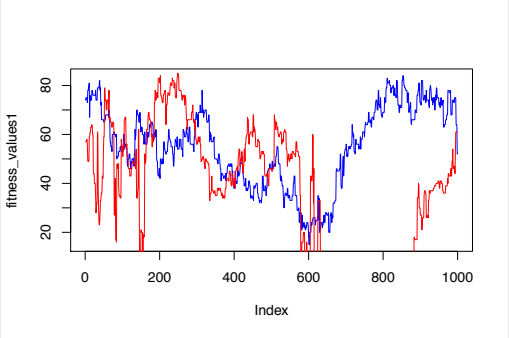
Practice : example of results

Adaptive walk lengths



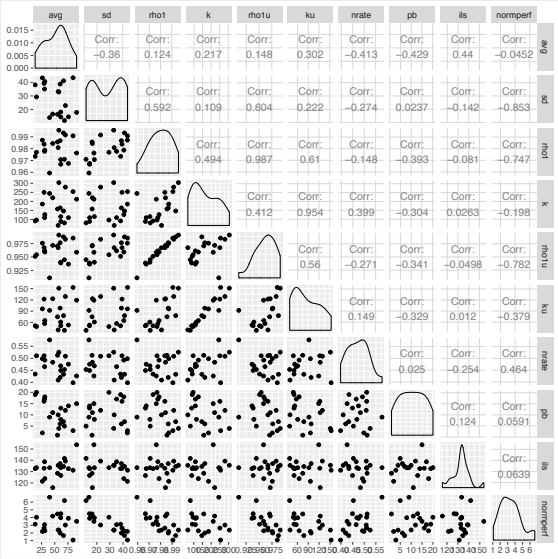
pb1 is "easier" than pb2

Random walks

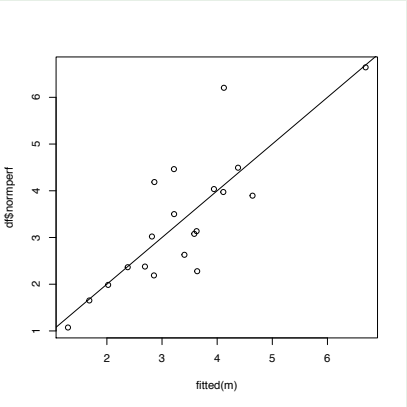


pb1 : $\rho(1) = 0.9856,\ nr = 0.513$
pb2 : $\rho(1) = 0.9872,\ nr = 0.498$

Correlation between features



ILS perf. prediction (lin. mod.)



$R^2 = 0.69$

Short summary

- Geometries :
  Multimodality, ruggedness, neutrality
- Metrics/features based on the neighborhood :
  probability to improve, fitness distribution, sign, etc.
- Covariance of the metrics across search space :
  autocorrelation, pearson/spearman/kendall correlation, entropy, etc.
- Estimation of metrics/features :
  random sampling, random walk, adaptive walk, etc.
  sample size, length, number : use sampling methodology

References I

Khulood Alyahya and Jonathan E Rowe.
Simple random sampling estimation of the number of local optima.
In International Conference on Parallel Problem Solving from Nature, pages 932–941. Springer, 2016.

Matthieu Basseur and Adrien Goëffon.
Climbing combinatorial fitness landscapes.
Applied Soft Computing, 30 :688–704, 2015.

References II

Meriema Belaidouni and Jin-Kao Hao.
An analysis of the configuration space of the maximal constraint satisfaction problem.
In PPSN VI : Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, pages 49–58, London, UK, 2000. Springer-Verlag.

U. Bastolla, M. Porto, H. E. Roman, and M. Vendruscolo.
Statiscal properties of neutral evolution.
Journal Molecular Evolution, 57(S) :103–119, August 2003.

Josselin Garnier and Leila Kallel.
Efficiency of local search with multiple local optima.
SIAM Journal on Discrete Mathematics, 15(1) :122–141, 2002.

References III

David E. Goldberg and Philip Segrest.
Finite markov chain analysis of genetic algorithms.
In ICGA, pages 1–8, 1987.

M. Huynen.
Exploring phenotype space through neutral evolution.
Journal Molecular Evolution, 43 :165–169, 1996.

E. Izquierdo-Torres.
The role of nearly neutral mutations in the evolution of dynamical neural networks.
In J. Pollack and al, editors, Ninth International Conference of the Simulation and Synthesis of Living Systems (Alife 9), pages 322–327. MIT Press, 2004.

References IV

T. Jones.
Evolutionary Algorithms, Fitness Landscapes and Search.
PhD thesis, University of New Mexico, Albuquerque, 1995.

M. Kimura.
The Neutral Theory of Molecular Evolution.
Cambridge University Press, Cambridge, UK, 1983.

Arnaud Liefooghe, Bilel Derbel, Sébastien Verel, Hernán Aguirre, and Kiyoshi Tanaka.
Towards Landscape-Aware Automatic Algorithm Configuration : Preliminary Experiments on Neutral and Rugged Landscapes, pages 215–232.
Springer International Publishing, Cham, 2017.

References V

Guanzhou Lu, Jinlong Li, and Xin Yao.
Fitness-probability cloud and a measure of problem hardness for evolutionary algorithms.
In European Conference on Evolutionary Computation in Combinatorial Optimization, pages 108–117. Springer, 2011.

Frank J Poelwijk, Daniel J Kiviet, Daniel M Weinreich, and Sander J Tans.
Empirical fitness landscapes reveal accessible evolutionary paths.
Nature, 445(7126) :383–386, 2007.

References VI

Helge Rosé, Werner Ebeling, and Torsten Asselmeyer.
The density of states - a measure of the difficulty of optimisation problems.
In Parallel Problem Solving from Nature, pages 208–217, 1996.

P. Schuster, W. Fontana, P. F. Stadler, and I. L. Hofacker.
From sequences to shapes and back : a case study in RNA secondary structures.
In Proc. R. Soc. London B., volume 255, pages 279–284, 1994.

Peter F. Stadler.
Landscapes and their correlation functions.
J. Math. Chem., 20 :1–45, 1996.

References VII

Sebastien Verel, Philippe Collard, and Manuel Clergue.
Measuring the evolvability landscape to study neutrality.
In M. Keijzer and et al., editors, Poster at Genetic and Evolutionary Computation – GECCO-2006, pages 613–614, Seatle, 8-12 July 2006. ACM Press.

E. D. Weinberger.
Correlated and uncorrelatated fitness landscapes and how to tell the difference.
In Biological Cybernetics, pages 63 :325–336, 1990.

3. Local Optima Network

Fitness Landscape Analysis
Understanding and Predicting Algorithm Performance
for Single- and Multi-objective Optimization

Outline

1. ~~The Basics of Fitness Landscapes~~
2. ~~Geometries of Fitness Landscapes~~
3. Local Optima Network
   - Features from the network, algorithm design and performance
   - Performance prediction and algorithm portfolio
4. Multi-objective Fitness Landscapes

Joint initial work with

- Gabriela Ochoa, University of StirlingUK
- Marco Tomassini, University of Lausanne, Switzerland
- Fabio Daolio, University of Stirling, UK

Key idea : complex system tools

Principle of variable aggregation

A model for dynamical systems with two scales (time/space)

- Split the state space according to the different scales
- Study the system at the large scale

Key idea : complex system tools

Principle of variable aggregation

A model for dynamical systems with two scales (time/space)

- Split the state space according to the different scales
- Study the system at the large scale

$$X \xrightarrow{op} X$$

Variable aggregation for fitness landscape

- At solutions level (small scale) :
  - Stochastic local search operator
  - Exponential number of solutions
  - Exponential size of the stochastic matrix of the process (Markov chain)
- Projection on a relevant space :
  - Reduce the size of state space
  - Potentially loose some information
  - Relevant information remains when

$$p(op(x)) \approx op'(p(x))$$

Key idea : complex system tools

Principle of variable aggregation

A model for dynamical systems with two scales (time/space)

- Split the state space according to the different scales
- Study the system at the large scale

$$\begin{tikzcd} X \arrow[r, "op"] \arrow[d, "p"'] & X \arrow[d, "p"] \\ E \arrow[r, "op'"] & E \end{tikzcd}$$

Variable aggregation for fitness landscape

- At solutions level (small scale) :
  - Stochastic local search operator
  - Exponential number of solutions
  - Exponential size of the stochastic matrix of the process (Markov chain)
- Projection on a relevant space :
  - Reduce the size of state space
  - Potentially loose some information
  - Relevant information remains when

$$p(op(x)) \approx op'(p(x))$$

Key idea : complex system tools

Complex network

Bring the tools from complex networks analysis to study the structure of combinatorial fitness landscapes

Methodology

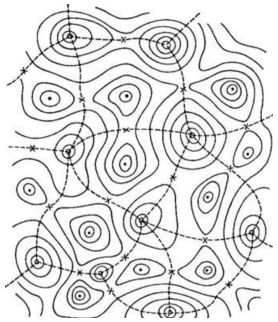
- **Design a network** that represents the landscape
  - Nodes : local optima
  - Edges : a notion of adjacency between local optima
- **Extract features** :
  - "complex" network analysis
- **Use the network features** :
  - search algorithm design, difficulty ...

J. P. K. Doye, The network topology of a potential energy landscape : a static scale-free network., Phys. Rev. Lett., 88 :238701, 2002. [Doy02]

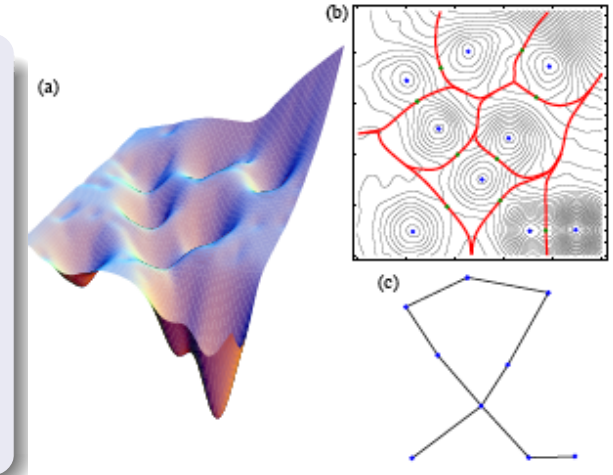
Energy surface and inherent networks

Inherent network

- **Nodes** : energy minima
- **Edges** : two nodes are connected if the energy barrier separating them is sufficiently low (transition state)



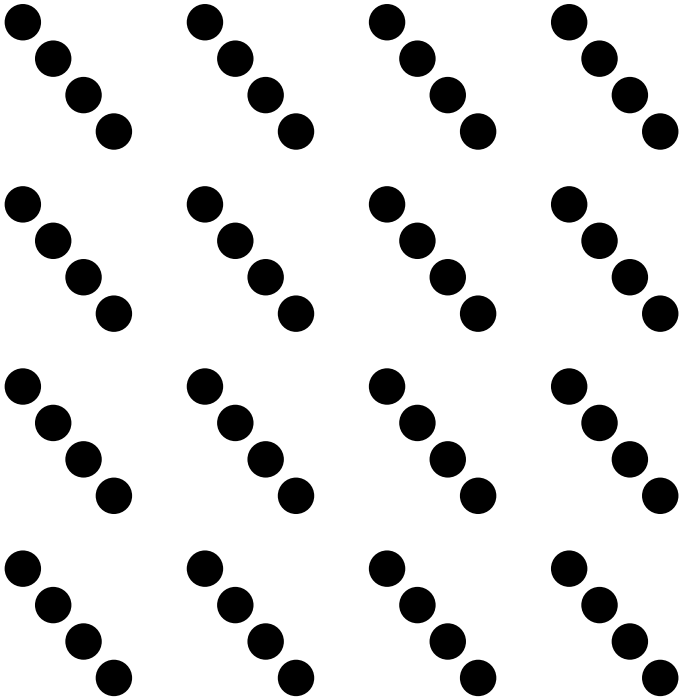
(a) Energy surface
(b) Contours plot : partition of states space into basins of attraction
(c) Landscape as a network



F. H Stillinger, T. A Weber. Packing structures and transitions in liquids and solids. Science, 225.4666 , p. 983-9, 1984. [SW84]
J. P. K. Doye, The network topology of a potential energy landscape : a static scale-free network. Phys. Rev. Lett., 88 :238701, 2002. [Doy02]

Basins of attraction in combinatorial optimization

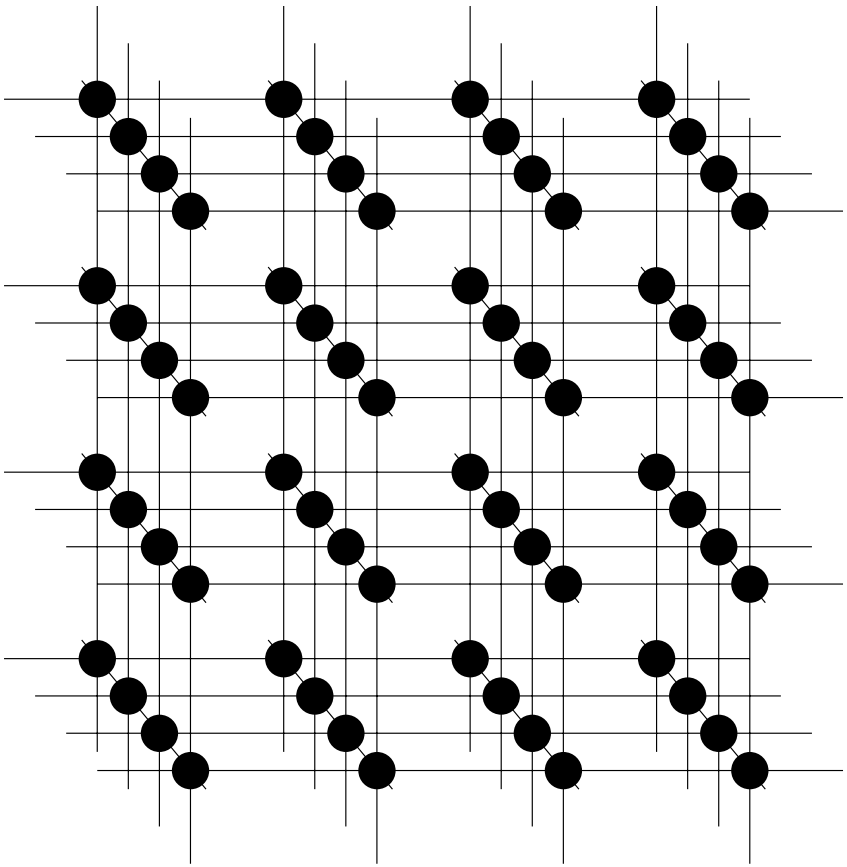
Example of a small NK landscape with $N = 6$ and $K = 2$



- Bit strings of length $N = 6$
- $2^6 = 64$ solutions
- one point = one solution

Basins of attraction in combinatorial optimization

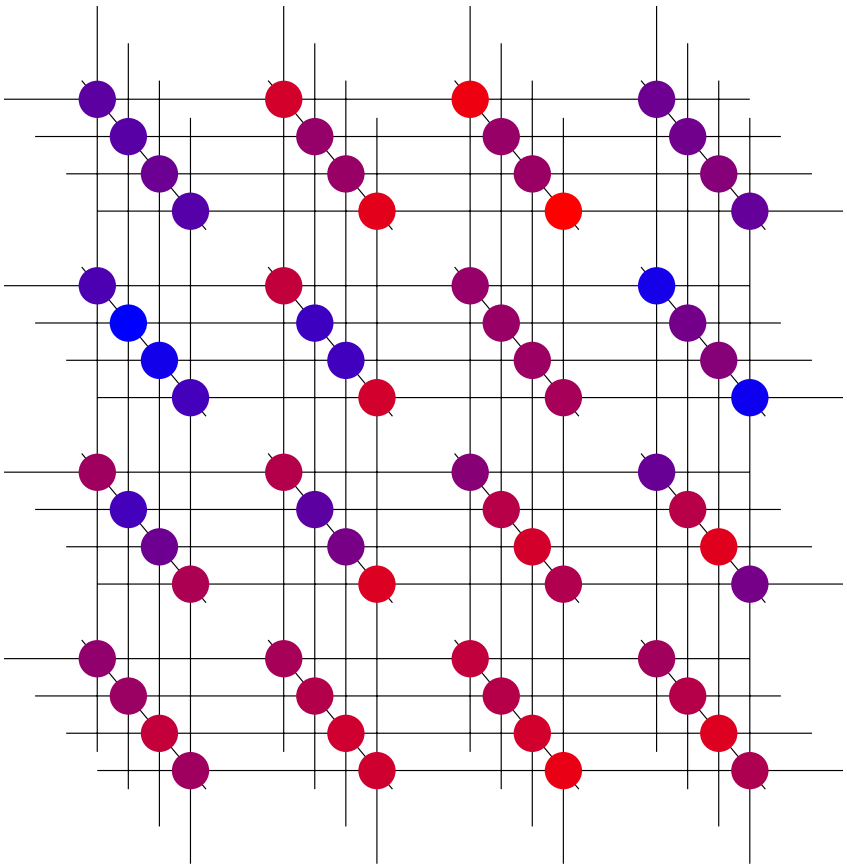
Example of a small NK landscape with $N = 6$ and $K = 2$



- Bit strings of length $N = 6$
- Neighborhood size $= 6$
- Line between points = solutions are neighbors
- Hamming distances between solutions are preserved (except for at the border of the cube)

Basins of attraction in combinatorial optimization

Example of small NK landscape with $N = 6$ and $K = 2$

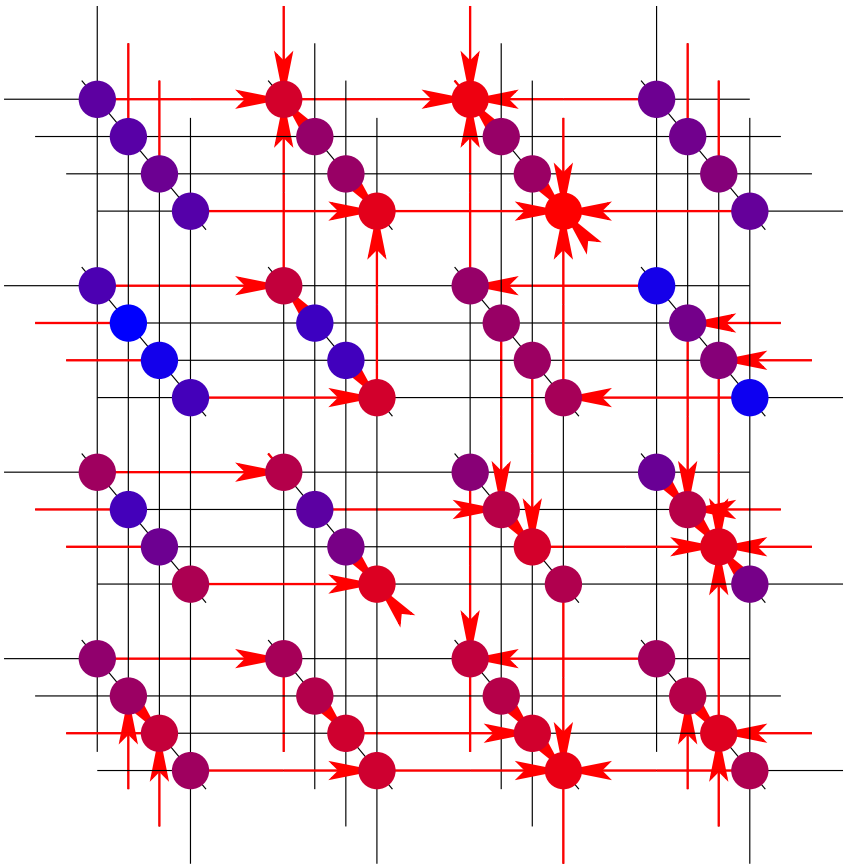


The color represents the fitness-values

- high fitness
- low fitness

Basins of attraction in combinatorial optimization

Example of small NK landscape with $N = 6$ and $K = 2$



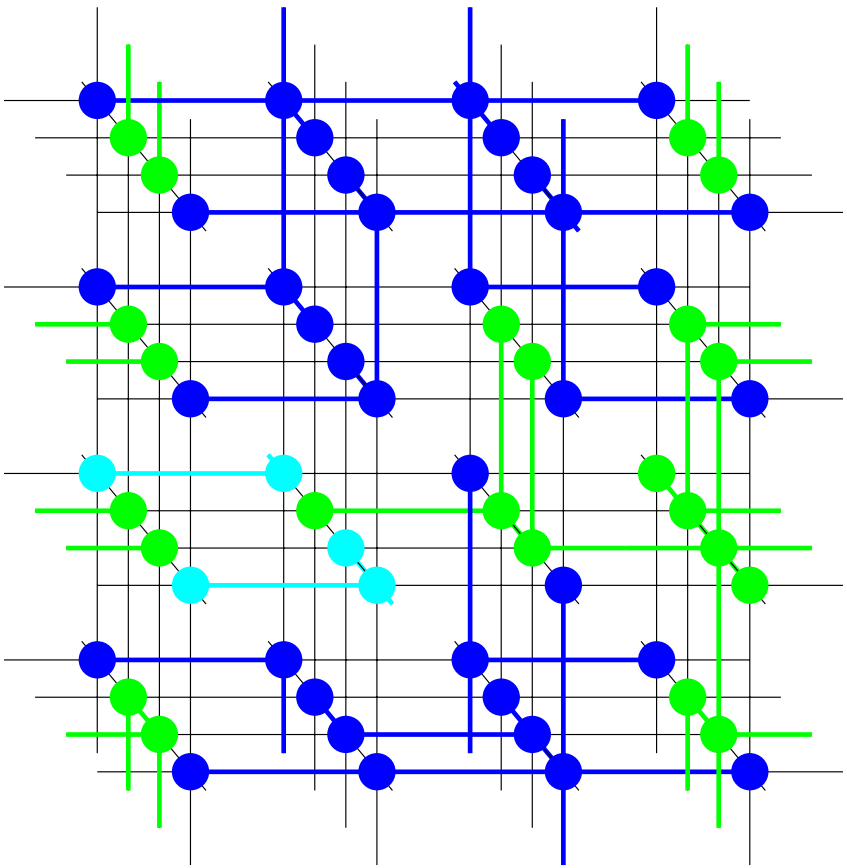
- Color represent fitness value
  - high fitness
  - low fitness
- $\longrightarrow$ point towards the solution with highest fitness in the neighborhood

Exercise :

Why not making a Hill-Climbing walk on it ?

Basins of attraction in combinatorial optimization

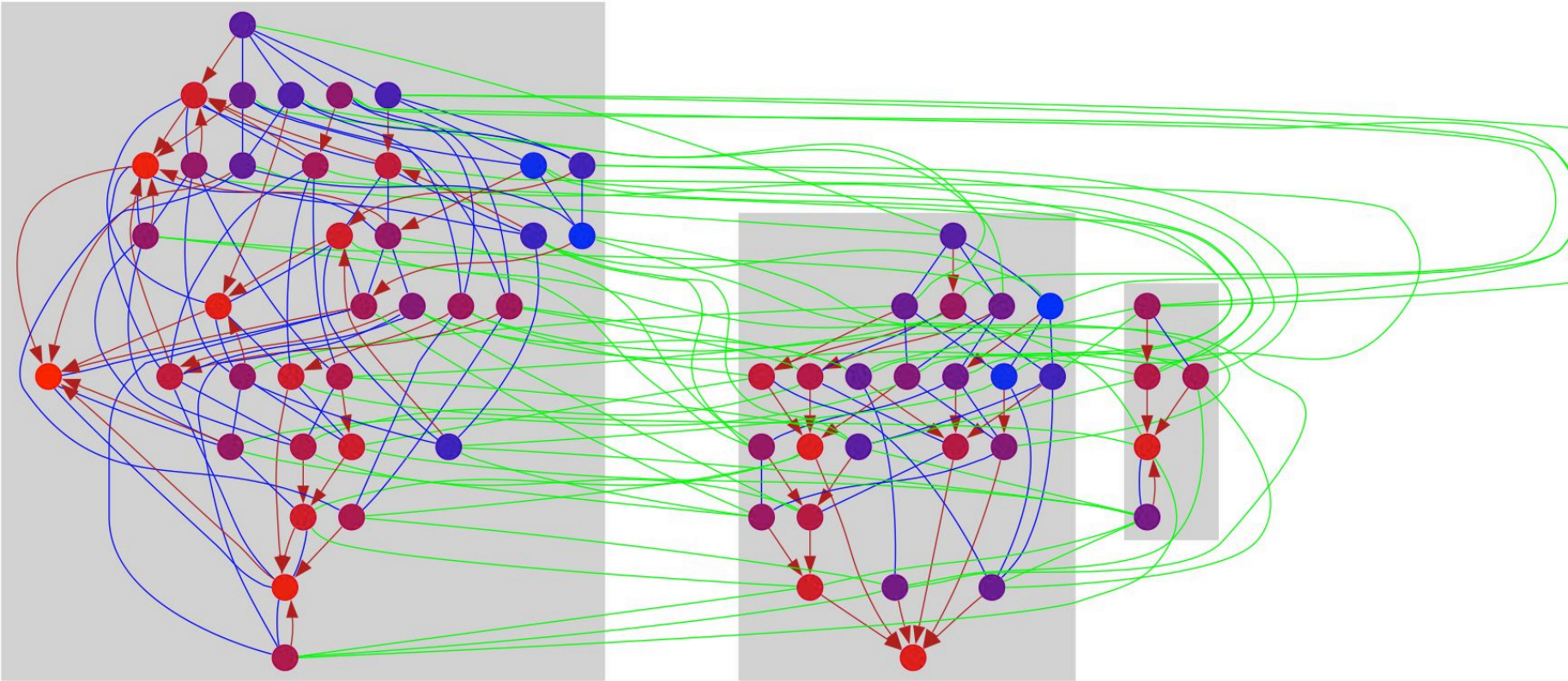
Example of small NK landscape with $N = 6$ and $K = 2$



- Each color corresponds to one basin of attraction
- Basins of attraction are interlinked and overlapped
- Basins have no "interior"

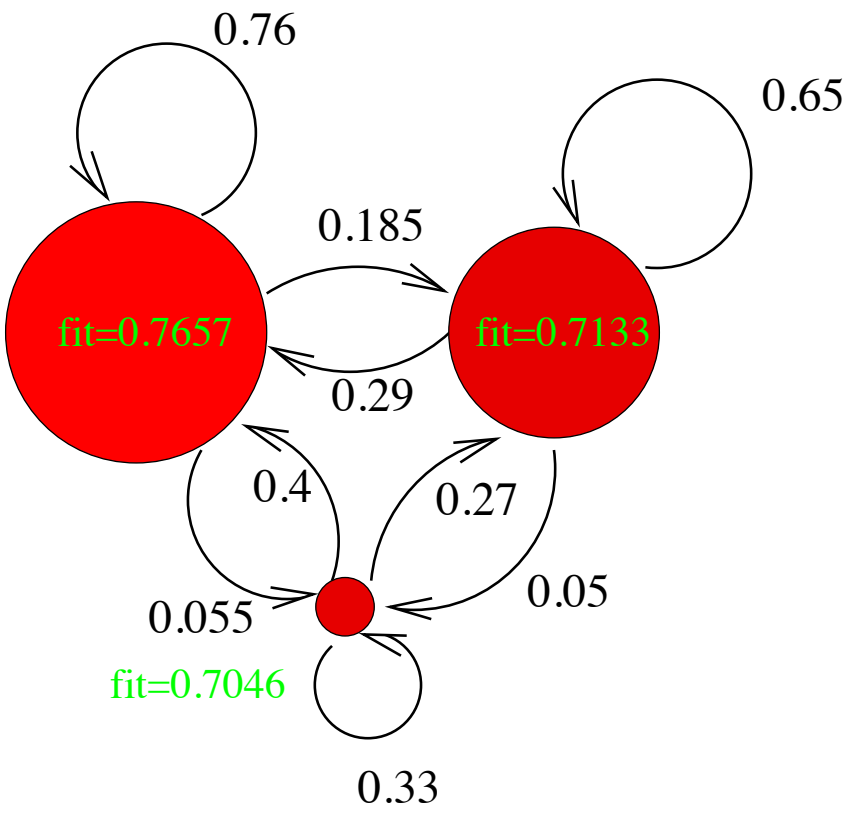
Basins of attraction in combinatorial optimization

Example of small NK landscape with $N = 6$ and $K = 2$



- Basins of attraction are interlinked and overlapped !
- Most neighbors of a given solution are outside its basin

Local optima network



- Nodes :
  local optima
- Edges :
  transition probabilities

Local optima network

Definition : Local Optima Network (LON)

Oriented weighted graph (V, E, w)

- Nodes V : set of local optima $\{LO_1, \ldots, LO_n\}$
- Edges E : notion of connectivity between local optima

Local optima network

Definition : Local Optima Network (LON)

Oriented weighted graph (V, E, w)

- Nodes V : set of local optima $\{LO_1, \ldots, LO_n\}$
- Edges E : notion of connectivity between local optima

2 possible definitions for edges

- **Basin-transition edges** :
  transition between random solutions from basin b_i to basin b_j
  ([OTVD08], [VOT08], [TVO08], [VOT10])
- **Escape edges** :
  transition from Local Optimum i to basin b_j
  (EA 2011, GECCO 2012, PPSN 2012, EA 2013 [DVOT13])

Basin-transition edges : random transition between basins

Edges

e_{ij} between LO_i and LO_j if $\exists\ x_i \in b_i$ and $x_j \in b_j$: $x_j \in \mathcal{N}(x_i)$

Prob. from solution x to solution x'

$$p(x \to x') = \Pr(x' = op(x))$$

Prob. from solution s to basin b_j

$$p(x \to b_j) = \sum_{x' \in b_j} p(x \to x')$$

Weights : Transition prob. from basin b_i to basin b_j

$$w_{ij} = p(b_i \to b_j) = \frac{1}{\sharp b_i} \sum_{x \in b_i} p(s \to b_j)$$

LON with escape edges

Definition : Local Optima Network (LON)

Orienter weighted graph (V, E, w)

- Notes V : set of local optima $\{LO_1, \ldots, LO_n\}$
- Edges E : notion of connectivity between local optima

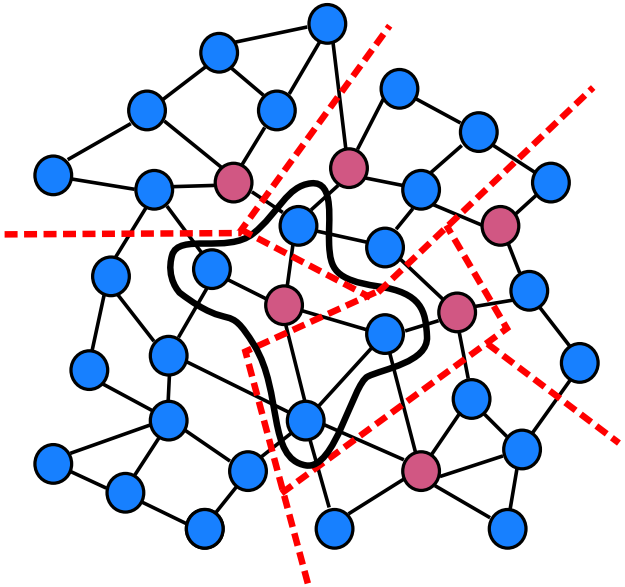
Escape edges

Edge e_{ij} between LO_i and LO_j
if $\exists x : distance(LO_i, x) \leqslant D$ and $x \in b_j$

Weights

$$w_{ij} = \sharp\{x \in X \mid d(LO_i, x) \leqslant D,\ x \in b_j\}$$

can be normalized by the number of solutions at distance D



LON with escape edges

Definition : Local Optima Network (LON)

Orienter weighted graph (V, E, w)

- Notes V : set of local optima $\{LO_1, \ldots, LO_n\}$
- Edges E : notion of connectivity between local optima

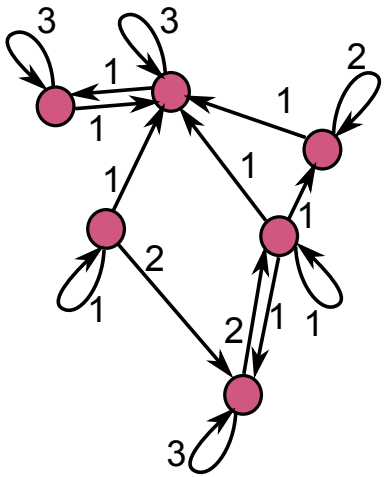
Escape edges

Edge e_{ij} between LO_i and LO_j
if $\exists x : distance(LO_i, x) \leqslant D$ and $x \in b_j$

Weights

$$w_{ij} = \sharp\{x \in X \mid d(LO_i, x) \leqslant D,\ x \in b_j\}$$

can be normalized by the number of solutions at distance D



Methodology

- Design, and understand LON metrics
  on tunable enumerable problem instances
  nk-landscapes, qap, ubqp, flow-shop
- Understand, and predict algorithm performances
  on enumerable instances
- Define sampling techniques for large size instance
- Understand, and predict algorithm performances
  on large instances

NK-landscapes

[Kauffman 1993] [Kau93]

$$x \in \{0,1\}^n \qquad f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x_j, x_{i_1}, \ldots, x_{i_k})$$

Two parameters

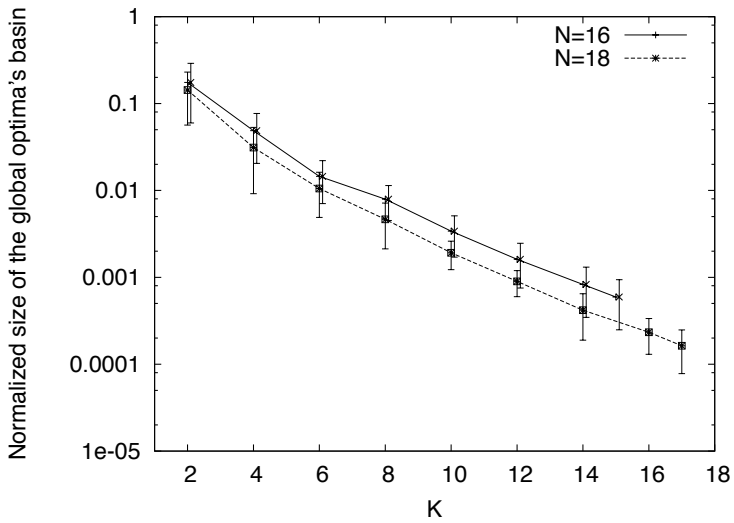
- Problem size n
- Non-linearity k $<$ n
  (multi-modality, epistatic interactions)
  - $k = 0$: linear problem, one single maxima
  - $k = n - 1$: random problem, number of local optima $\frac{2^N}{N+1}$

note : similar results for QAP and flowshop

Basins of attraction features

- **Basin of attraction** :
  - Size :
    average, distribution ...
  - Fitness of local optima :
    average, distribution, correlation ...

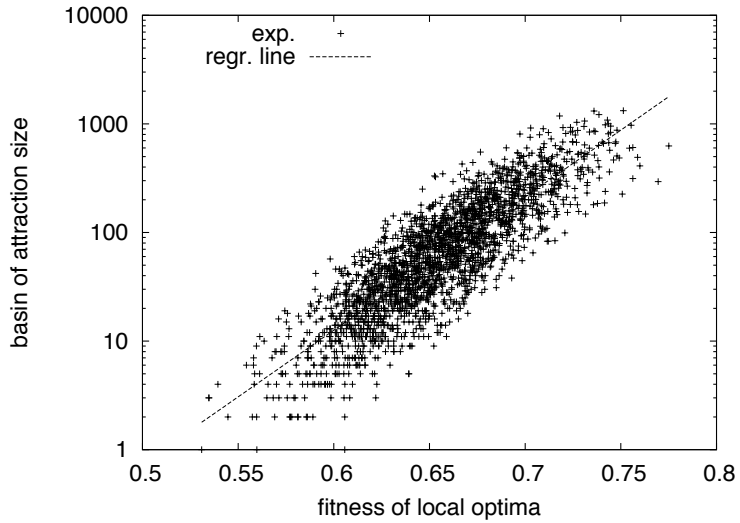
Global optimum basin size *vs.* non-linearity degree k



Size of the global maximum basin as a function of non-linearity degree k

- Basin size of maximum decreases exponentially with non-linearity degree
- $\Rightarrow$ Difficulty of (best-improvement) hill-climber from a random solution

Fitness of local optima *vs.* basin size

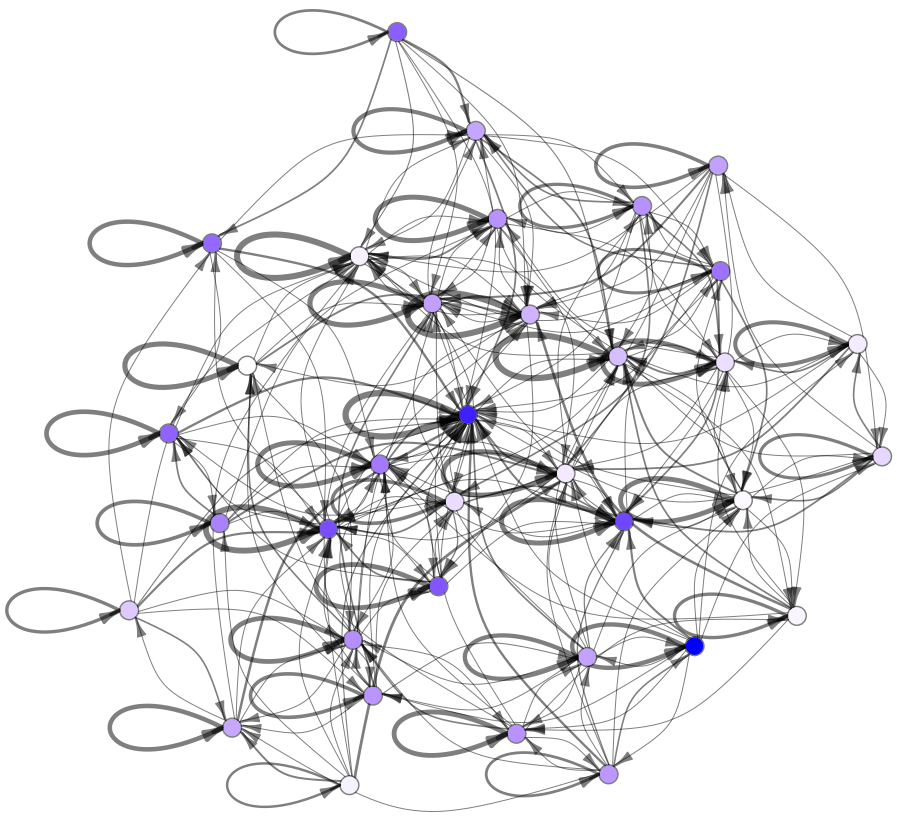


Correlation fitness of local optima *vs.* their corresponding basins sizes

The highest, the largest !

- On average, the global optimum is easier to find than one given other local optimum
- . . . but more difficult to find, as the number of local optima increases exponentially with k

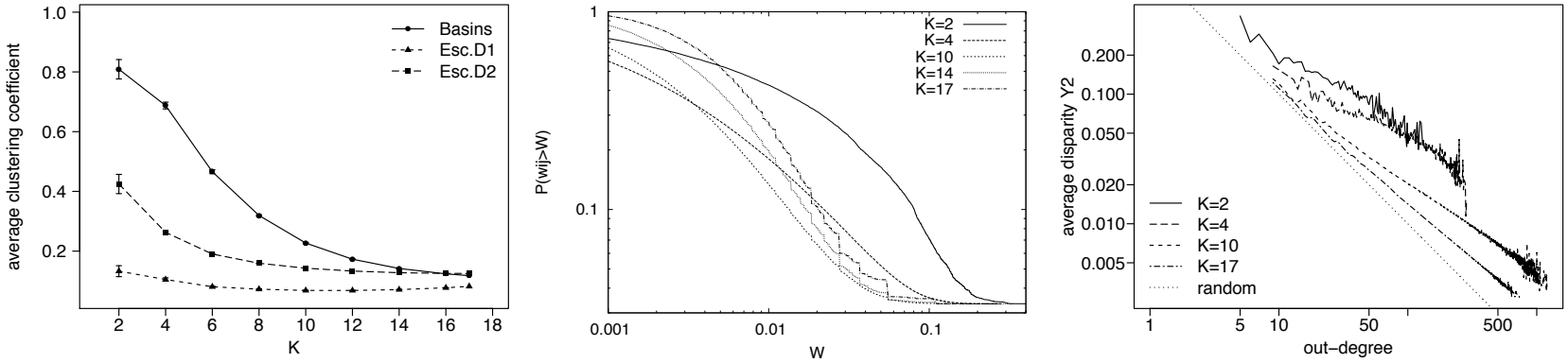
Features form the local optima network



- nv : ♯vertices
- lv : avg path length
  $d_{ij} = 1/w_{ij}$
- lo : path length to best
- fnn : fitness corr.
  $(f(x), f(y))$ with $(x, y) \in E$
- wii : self loops
- wcc : weighted clust. coef.
- $zout$: out degree
- $y2$: disparity
- knn : degree corr.
  $(deg(x), deg(y))$ with $(x, y) \in E$

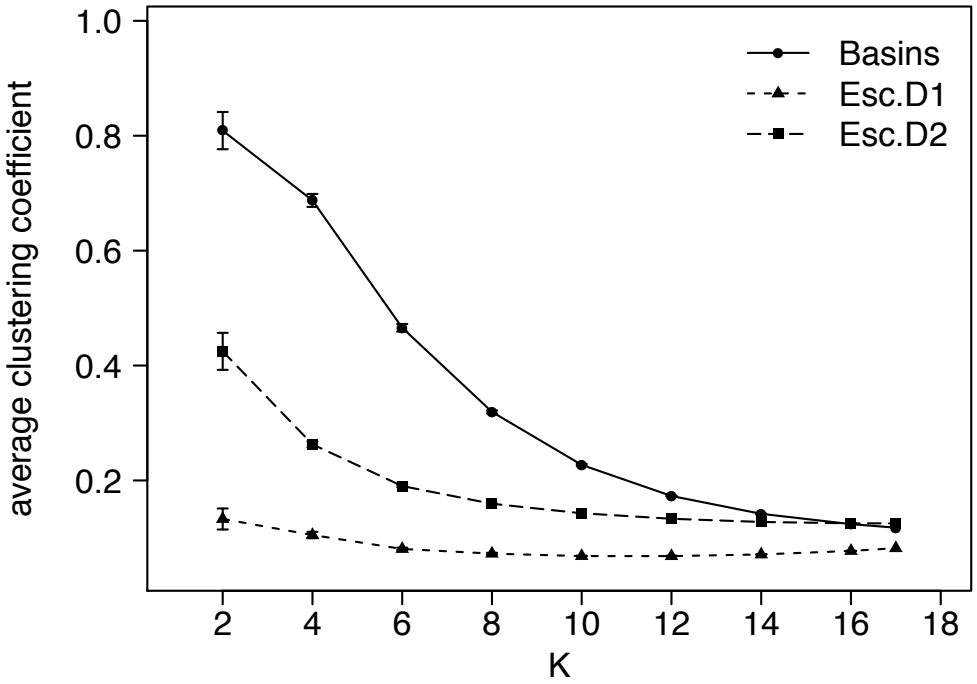
Structure of the local optima network

- NK-landscapes (small instances) :
  most of features are correlated with k
  relevance of the LON definition



- LON is **not a random** network (NK, QAP, FSSP) :
  highly clustered network,
  distribution of weights and degrees have long tail . . .

Example : clustering coefficient for NK-landscapes



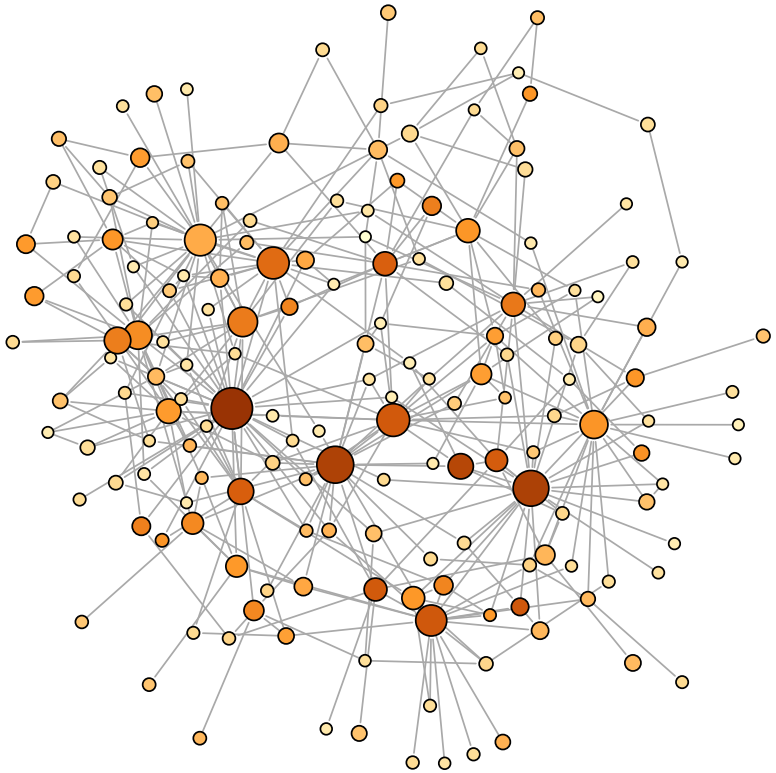
- Network highly clustered
- Clustering coefficient decreases with the degree of non-linearity k

LON to compare instance difficulty

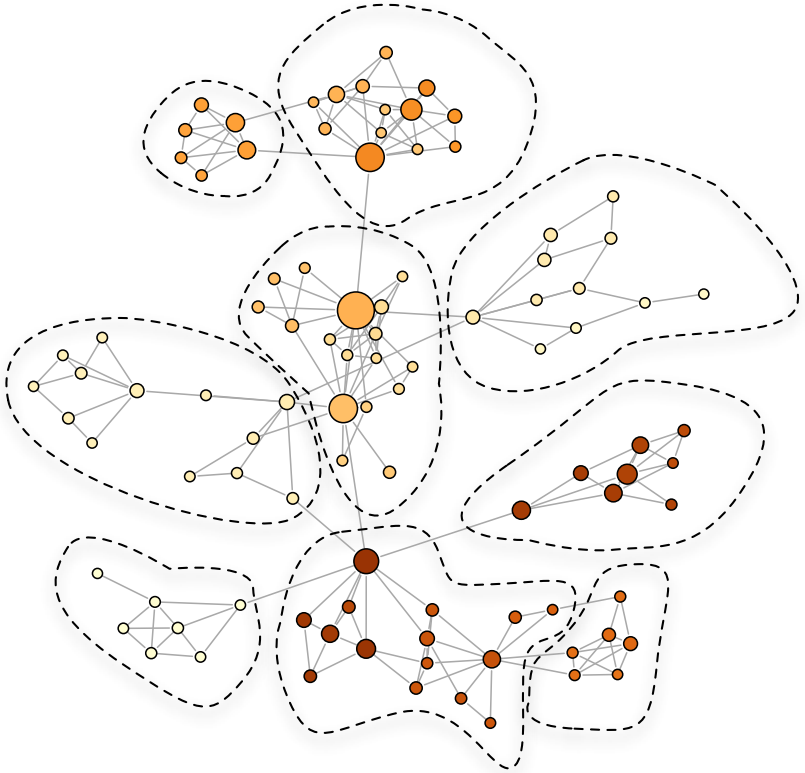
Local Optima Network for the Quadratic Assignment Problem (QAP) [DTVO11]

→ Community detection, Funnel, Fractal dimension

Random instance



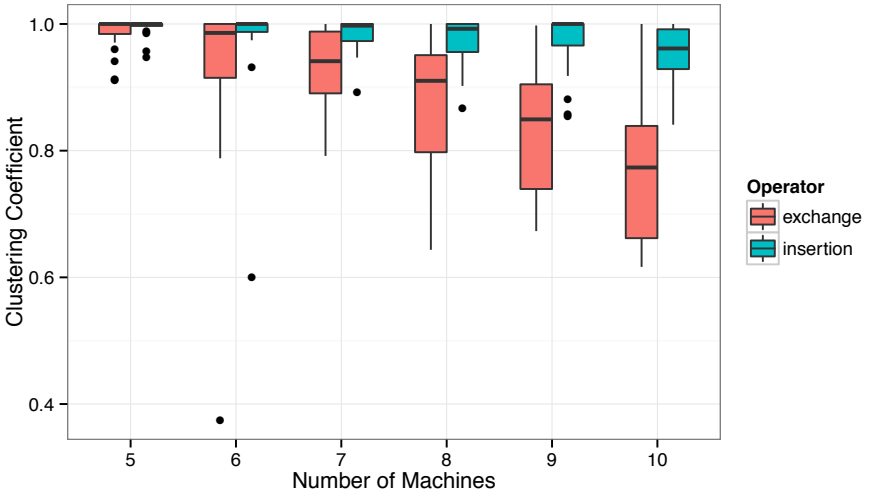
Real-like instance

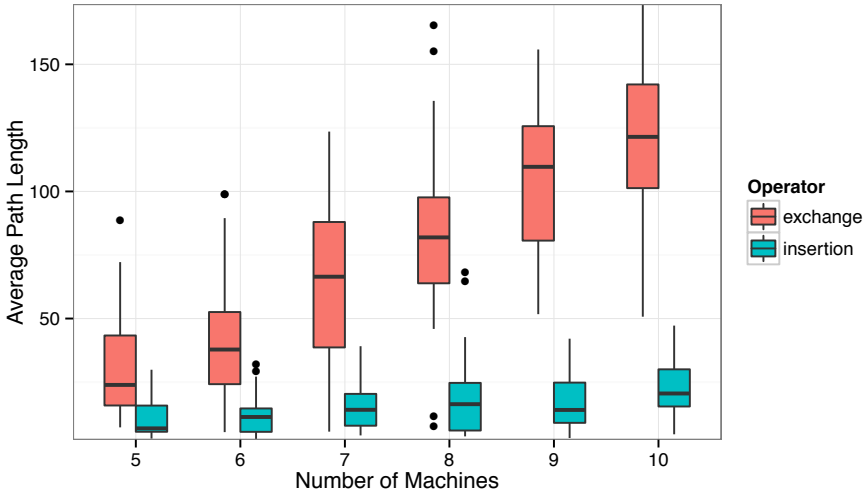


the structure of the LON is related to **problem difficulty**

Configuration : LON to compare algorithm components (1)

comparaison of **operators** for the Flowshop Scheduling Problem





Configuration : LON to compare algorithm components (2)

comparaison of the hill-climbing's **pivot rule** for NK-landscapes : First *vs.* Best improvement HC

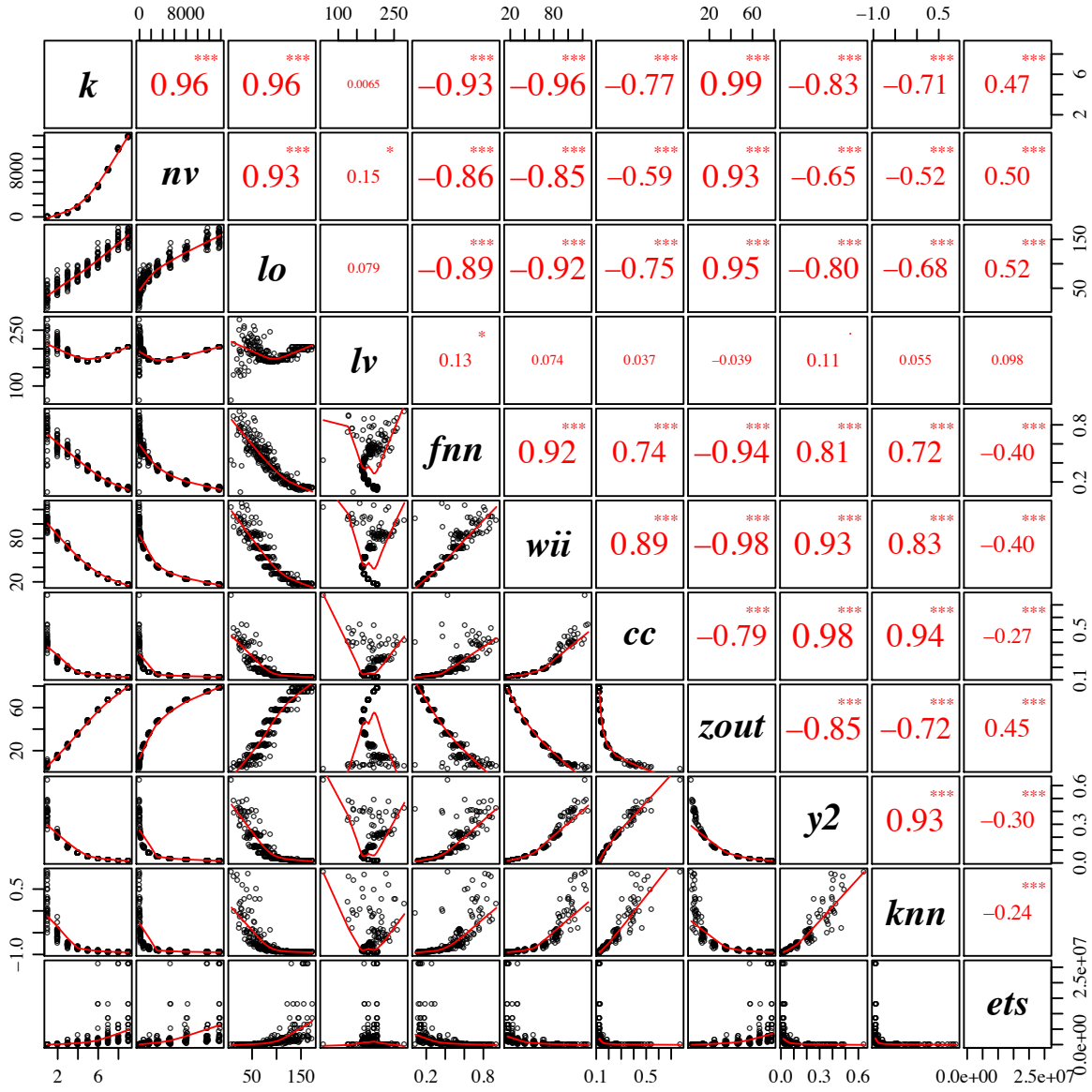
K	$\bar{n}_e / \bar{n}_v^2$		$\bar{Y}$		$\bar{d}$		$\bar{d}_{best}$	
	b-LON	f-LON	b-LON	f-LON	b-LON	f-LON	b-LON	f-LON
2	0.81	0.96	0.326	0.110	56	39	16	12
4	0.60	0.92	0.137	0.033	126	127	35	32
6	0.32	0.79	0.084	0.016	170	215	60	70
8	0.17	0.65	0.062	0.011	194	282	83	118
10	0.09	0.53	0.050	0.009	206	340	112	183
12	0.05	0.44	0.043	0.008	207	380	143	271

Information given by the local optima network

Advanced questions

- Can we explain the performance from LON features ?
- Can we predict the performance from LON features ?
- Can we select the relevant algorithm from LON features ?

Correlation matrix (small size problem instances)

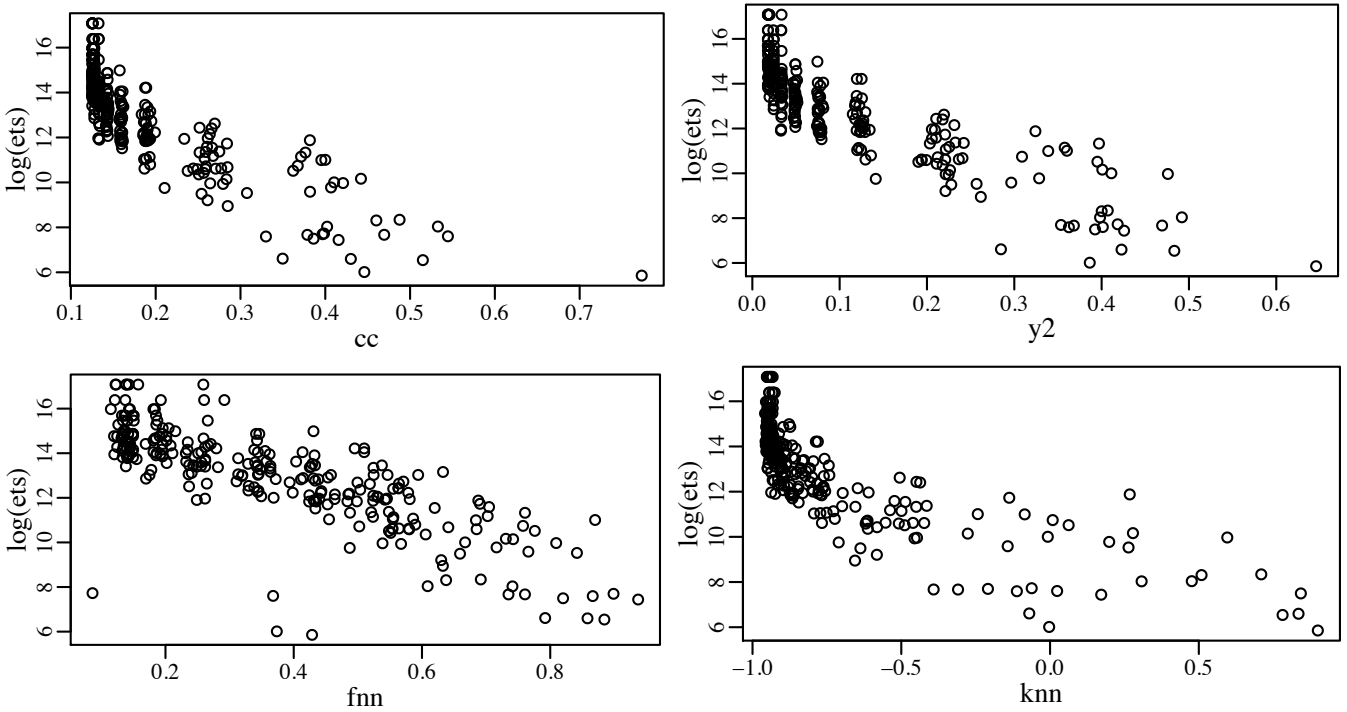


LON features *vs.* performance : simple correlation

Algorithm : Iterated Local Search on NK-landscapes with $N = 18$

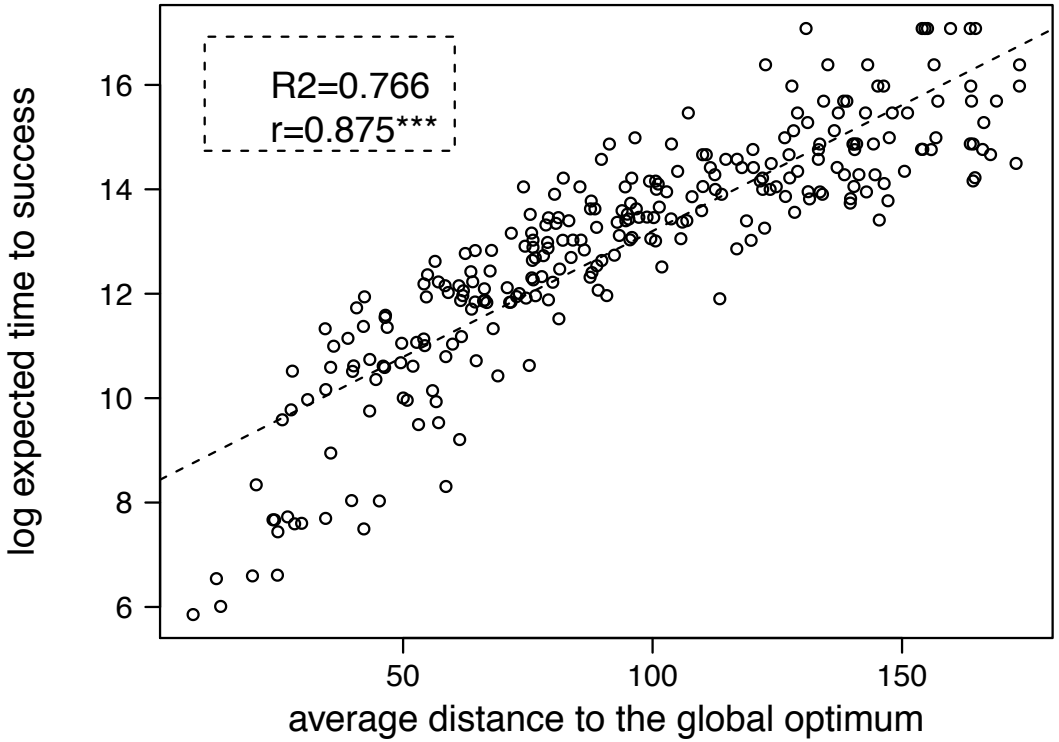
Performance : $ert = \mathbb{E}(T_s) + \left(\frac{1-p_s}{p_s}\right) T_{max}$

n_v	$\bar{d}_{best}$	$\bar{d}$	fnn	w_{ii}	$\bar{C}^w$	zout	$\bar{Y}$	knn
0.885	0.915	0.006	−0.830	−0.883	−0.875	0.885	−0.883	−0.850



ILS performance vs LON metrics

NK-landscapes [DVOT12]



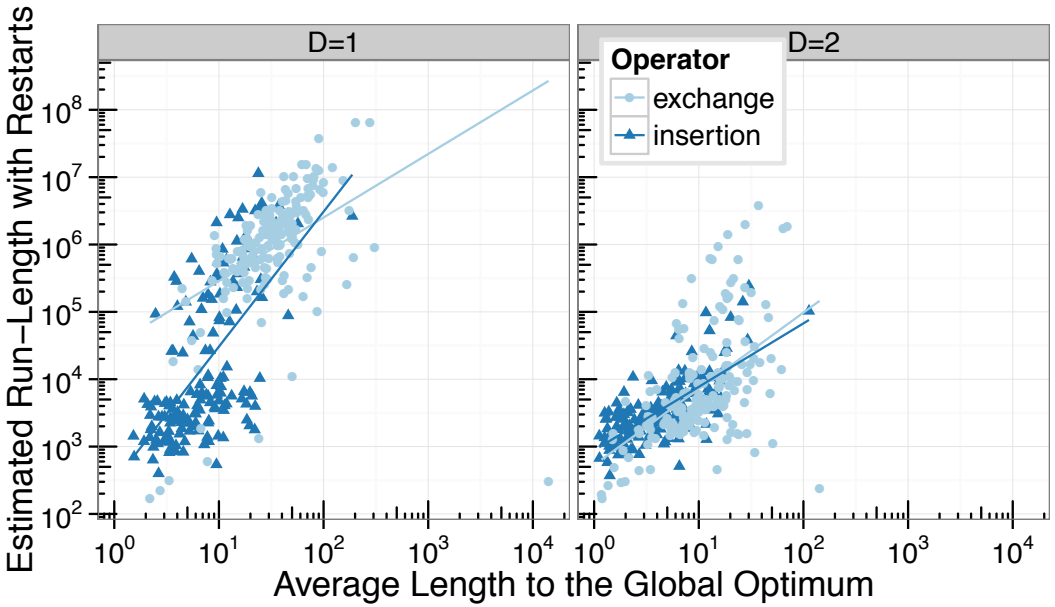
Expected running time

vs.

Average shortest path to the global optimum

ILS performance vs LON metrics

Flow-Shop Scheduling Problem [EA'13]



Expected running time

vs.

Average shortest path to the global optimum

LON features *vs.* performance : multi-linear regression

1. Multiple **linear** regression on all possible predictors :

$$\log(ert) = \beta_0 + \beta_1 k + \beta_2 \log(nv) + \beta_2 lo + \cdots + \beta_{10} knn + \varepsilon$$

2. Step-wise **backward elimination** of each predictor in turn

Predictor	$\widehat{\beta}_i$	Std. Error	p-value
(Intercept)	10.3838	0.58512	$9.24 \cdot 10^{-47}$
lo	0.0439	0.00434	$1.67 \cdot 10^{-20}$
zout	−0.0306	0.00831	$2.81 \cdot 10^{-04}$
y2	−7.2831	1.63038	$1.18 \cdot 10^{-05}$
knn	−0.7457	0.40501	$6.67 \cdot 10^{-02}$

Multiple R^2 : 0.8494, Adjusted R^2 : 0.8471

LON features *vs.* performance : multi-linear regression

for the **Flowshop Scheduling Problem** using exhaustive selection

$\sharp P$	$\log(N_V)$	CC^w	F_{nn}	k_{nn}	r	$\log(L_{opt})$	$\log(L_V)$	w_{ii}	Y_2	k_{out}	C_p	$adjR^2$
1						2.13					265.54	0.574
2		−5.18				1.43					64.06	0.675
3						1.481	0.895			−0.042	16.48	0.700
4		−2.079				1.473	0.540			−0.032	8.75	0.704
5		−2.388			−1.633	1.470	0.528			−0.030	5.97	0.706

Sampling methodology for large-size instances

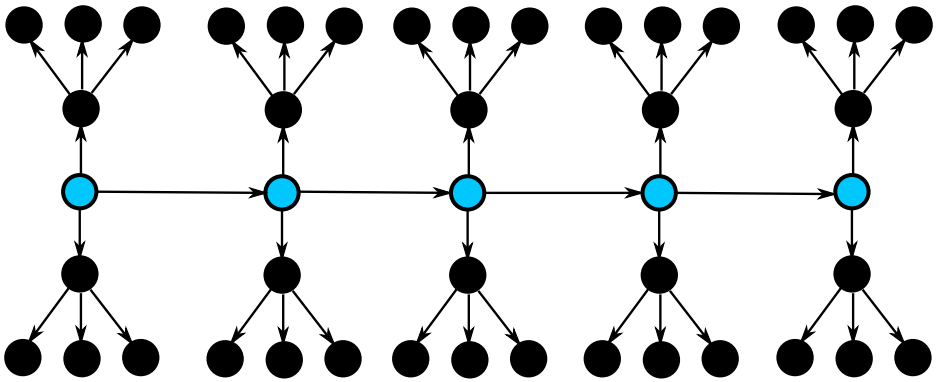
Two mains techniques (Thomson *et al.* [TOVV20]) :

- Random walk on local optima network
- Adaptive walk lon local optima network

Sampling methodology for large-size instances

From the sampling of large-size complex network :

- Random walk on the network
- Breadth-First-Search



Set of estimated LON features for large-size instances

	LON metrics
fit	Average fitness of local optima in the network
wii	Average weight of self-loops
zout	Average outdegree
y_2	Average disparity for outgoing edges
knn	Weighted assortativity
wcc	Weighted clustering coefficient
fnn	Fitness-fitness correlation on the network
	Metrics from the sampling procedure
lhc	Average length of hill-climbing to local optima
mlhc	Maximum length of hill-climbing to local optima
nhc	Number of hill-climbing paths to local optima

Performance prediction based on estimated features

- Optimization scenario using off-the-shelf metaheuristics :
  TS, SA, EA, ILS on 450 instances for NK and QAP
- Performance measures :
  average fitness / average rank
- Regression model :
  multi-linear model / random forest
- Set of features :
  - *basic* : 1st autocorr. coeff. of fitness (rw of length 10^3)
    Avg. fitness of local optima (10^3 hc)
    Avg. length to reach local optima (10^3 hc)
  - *lon* : see previous
  - *all* : *basic* and *lon* features
- Quality measure of regression :
  R^2 on cross-validation (repeated random sub-sampling)

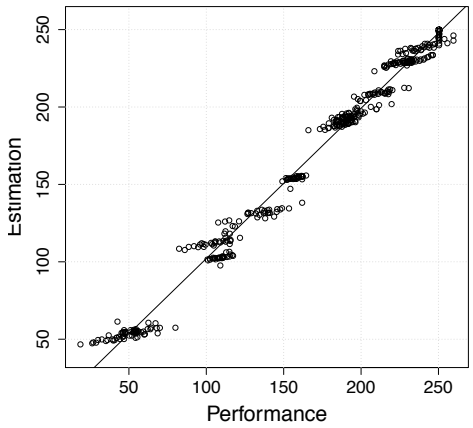
R^2 on cross-validation for NK-landscapes and QAP

Sampling parameters : length $\ell = 100$, sampled edge $m = 30$, deep $d = 2$

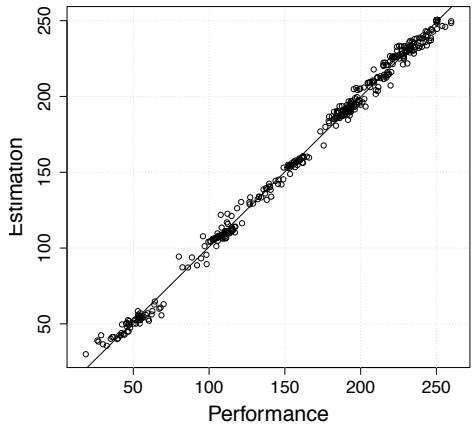
			NK					QAP				
Mod.	Feat.	Perf.	TS	SA	EA	ILS	avg	TS	SA	EA	ILS	avg
lm	basic	fit	0.8573	0.8739	0.8763	0.8874	0.8737	-38.42	-42.83	-41.63	-39.06	-40.48
lm	lon	fit	0.8996	0.9015	0.9061	0.8954	0.9007	**0.9995**	**1.0000**	**1.0000**	**0.9997**	0.9998
lm	all	fit	**0.9356**	**0.9455**	**0.9442**	**0.9501**	0.9439	**0.9996**	**0.9997**	**0.9999**	**0.9997**	0.9997
lm	basic	rank	0.8591	0.9147	0.6571	0.6401	0.7678	0.2123	0.8324	-0.0123	0.4517	0.3710
lm	lon	rank	0.9517	**0.9332**	**0.7783**	**0.7166**	0.8449	**0.7893**	**0.9673**	**0.8794**	**0.9015**	0.8844
lm	all	rank	**0.9534**	**0.9355**	**0.7809**	**0.7177**	0.8469	0.6199	0.9340	0.8577	**0.9029**	0.8286
rf	basic	fit	**0.9043**	0.9104	0.9074	**0.8871**	0.9023	0.8811	0.8820	0.8806	0.8801	0.8809
rf	lon	fit	0.8323	0.8767	0.8567	0.8116	0.8443	0.9009	0.9025	0.9027	0.9019	0.9020
rf	all	fit	0.8886	**0.9334**	**0.9196**	**0.8778**	0.9048	**0.9431**	**0.9445**	**0.9437**	**0.9429**	0.9436
rf	basic	rank	**0.9513**	0.9433	0.7729	**0.8075**	0.8687	**0.9375**	**0.9653**	0.8710	0.9569	0.9327
rf	lon	rank	0.9198	0.9291	0.7979	0.7798	0.8566	0.9308	0.9630	**0.8820**	0.9601	0.9340
rf	all	rank	**0.9554**	**0.9465**	**0.8153**	0.8151	0.8831	**0.9381**	**0.9668**	**0.8779**	**0.9643**	0.9368

Observed vs. estimated performance

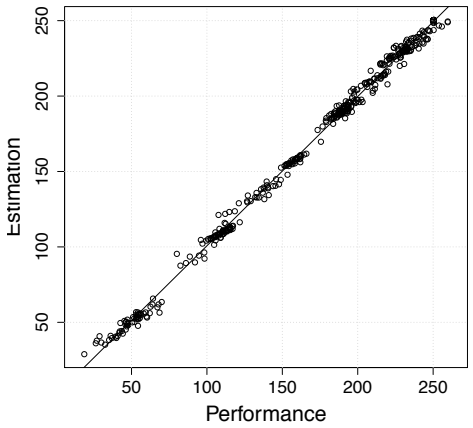
- On the 32 possibles cases (Mod. × Feat. × Algo.),
  the best set of features : *all* 27 times, *lon* 12 times, *basic* 6 times
- With linear model : *basic* set is never the one of the best set,
  lon features are more linearly correlated with performance
- Random forest model obtains higher regression quality :
  basic can be one of the best set (2 times)
  Nevertheless, 7/8 cases, *all* features are the best one



basic, $R^2 = 0.9327$



lon, $R^2 = 0.9601$



all, $R^2 = 0.9643$

Portfolio scenario

- Portfolio of 4 metaheuristics : TS, SA, EA, ILS
- Classification task : selection of one of the best metaheuristic
- Models : logit, random forest, svm
- Quality of classification :
  error rate (algo. is not one of the best) on cross-validation

		Avg. error rate				
Probl.	Feat.	logit	rf	svm	cst	rnd
NK	basic	0.0379	0.0278	**0.0158**	0.4711	0.6749
	lon	**0.0203**	**0.0249**	**0.0168**		
	all	0.0244	**0.0269**	**0.0165**		
QAP	basic	**0.0142**	0.0107	0.0771	0.4222	0.6706
	lon	**0.0156**	**0.0086**	**0.0456**		
	all	**0.0161**	0.0106	**0.0431**		

Conclusions and perspectives

- The structure of the local optima network ...
  ...can explain problem difficulty
- LON-features can be used for performance prediction
- The sampling methodology gives relevant estimation of LON features for performance prediction and algorithm portfolio

Perspectives

- Reducing the cost and improving the efficiency of the sampling
- Other (real-world, black-box) problems and algorithms
- Understanding the link between the problem definition and the LON structure
- Studying the LON as a fitness landscape at a large scale

References I

J. P. K. Doye.
The network topology of a potential energy landscape : a static scale-free network.
Phys. Rev. Lett., 88 :238701, 2002.

Fabio Daolio, Marco Tomassini, Sébastien Verel, and Gabriela Ochoa.
Communities of Minima in Local Optima Networks of Combinatorial Spaces.
Physica A : Statistical Mechanics and its Applications, 390(9) :1684 – 1694, July 2011.

References II

Fabio Daolio, Sébastien Verel, Gabriela Ochoa, and Marco Tomassini.
Local optima networks and the performance of iterated local search.
In
Proceedings of the fourteenth international conference on Genetic an
pages 369–376, Philadelphia, United States, July 2012. ACM.

Fabio Daolio, Sébastien Verel, Gabriela Ochoa, and Marco Tomassini.
Local Optima Networks of the Permutation Flow-Shop Problem.
In Springer, editor,
International Conference on Artificial Evolution (EA 2013),

References III

Lecture Notes in Computer Science, pages –, Bordeaux, France, October 2013.

S. A. Kauffman.
The Origins of Order.
Oxford University Press, New York, 1993.

Gabriela Ochoa, Marco Tomassini, Sébastien Verel, and Christian Darabos.
A Study of NK Landscapes' Basins and Local Optima Networks.
In Proceedings of the 10th annual conference on Genetic and evolutionary computation Genetic And Evolutionary Computation Conference, pages 555–562, Atlanta États-Unis d'Amérique, 07 2008. ACM New York, NY, USA.
best paper nomination.

References IV

Frank H Stillinger and Thomas A Weber.
Packing structures and transitions in liquids and solids.
Science(Washington, DC), 225(4666) :983–9, 1984.

SL Thomson, G Ochoa, S Verel, and N Veerapen.
Inferring future landscapes : Sampling the local optima level.
Evolutionary computation, page 1, 2020.

Marco Tomassini, Sébastien Verel, and Gabriela Ochoa.
Complex-network analysis of combinatorial spaces : The NK landscape case.
Physical Review E : Statistical, Nonlinear, and Soft Matter Physics, 78(6) :066114, 12 2008.
89.75.Hc ; 89.75.Fb ; 75.10.Nr.

References V

Sébastien Verel, Gabriela Ochoa, and Marco Tomassini.
The Connectivity of NK Landscapes' Basins : A Network Analysis.
In Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems Artificial Life XI, pages 648–655, Winchester France, 08 2008. MIT Press, Cambridge, MA.
tea team.

Sébastien Verel, Gabriela Ochoa, and Marco Tomassini.
Local Optima Networks of NK Landscapes with Neutrality.
IEEE Transactions on Evolutionary Computation, volume 14(6) :783 – 797, November 2010.

4. Multi-objective Fitness Landscapes

Bilel Derbel	Arnaud Liefooghe	Sébastien Verel
bilel.derbel@univ-lille.fr	arnaud.liefooghe@univ-lille.fr	verel@univ-littoral.fr
Univ. Lille – CRIStAL / Inria Lille		Univ. Littoral Côte d'Opale – LISIC
Université de Lille	MODÔ INTERNATIONAL ASSOCIATED LABORATORY SHINSHU univ + univ LILLE	ulco UNIVERSITÉ DU LITTORAL CÔTE D'OPALE

Outline

1. ~~The Basics of Fitness Landscapes~~
2. ~~Geometries of Fitness Landscapes~~
3. ~~Local Optima Network~~
4. Multi-objective Fitness Landscapes

F Daolio H Aguirre K Tanaka

信州大学 SHINSHU UNIVERSITY

- Brief overview of (evolutionary) multi-objective optimization
- Features to characterize multi-objective fitness landscapes
- Performance prediction and algorithm selection

Motivations

- Multi-objective optimization problems are hard
- Understanding what makes a problem difficult, and how
- Understanding what makes algorithms work well (or not)
- Learning about the problem structure to design better algorithms
- Models to explain and predict the performance of algorithms based on (relevant) problem features
- Models to understand the dynamics and behavior of algorithms

Global Picture

Features

- Expensive features to understand difficulty (problems vs. algorithms)
- Low-cost features for prediction (performance, selection, configuration)

Problems

- Multi-objective NK landscapes (pmnk-landscapes)
- Multi-objective quadratic assignment problem (mQAP)
- Multi-objective scheduling, quadratic programming…

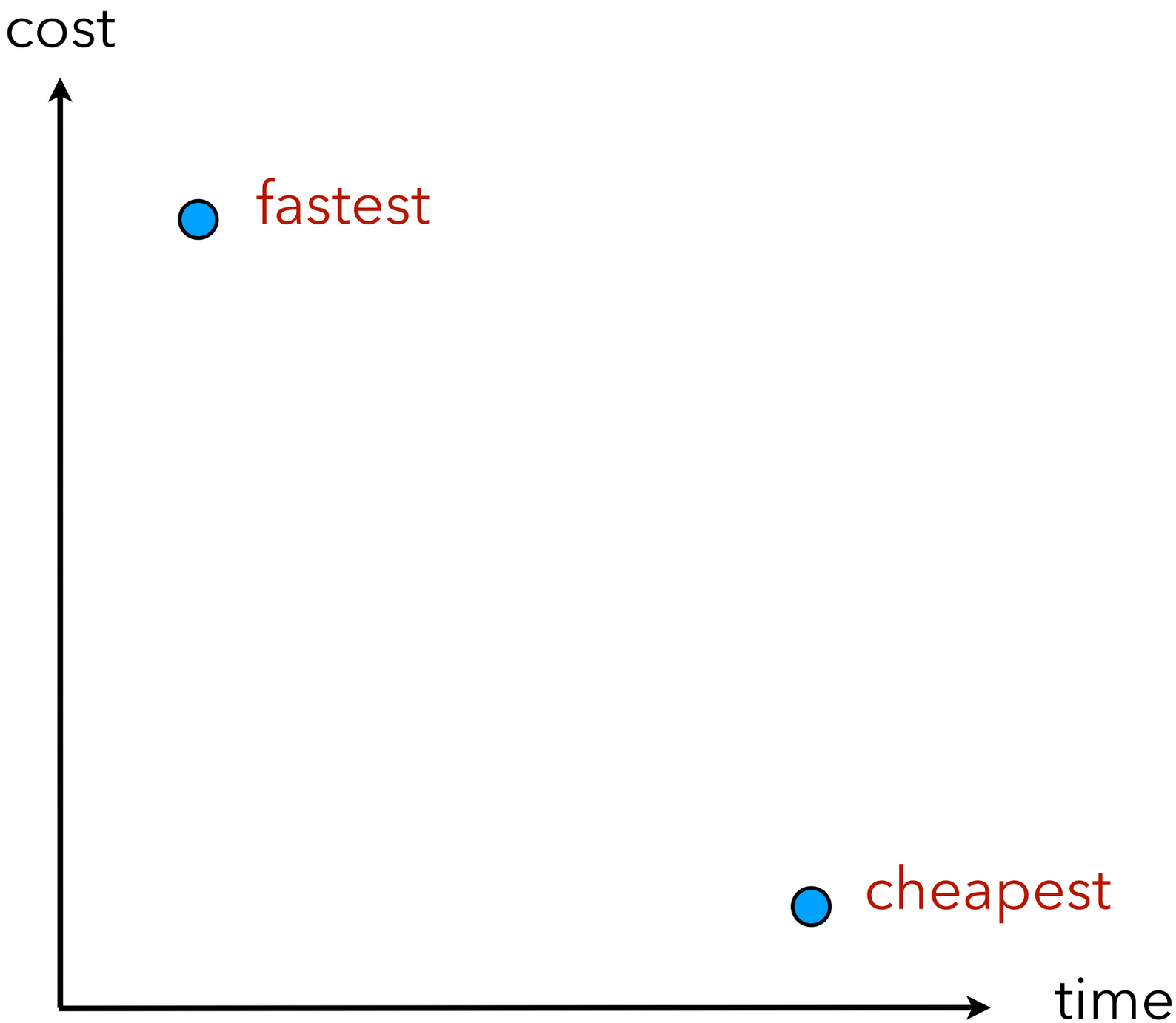
Algorithms

- Global vs. local dominance-based search algorithms
- State-of-the-art EMO algorithms (decomposition, dominance, indicator)

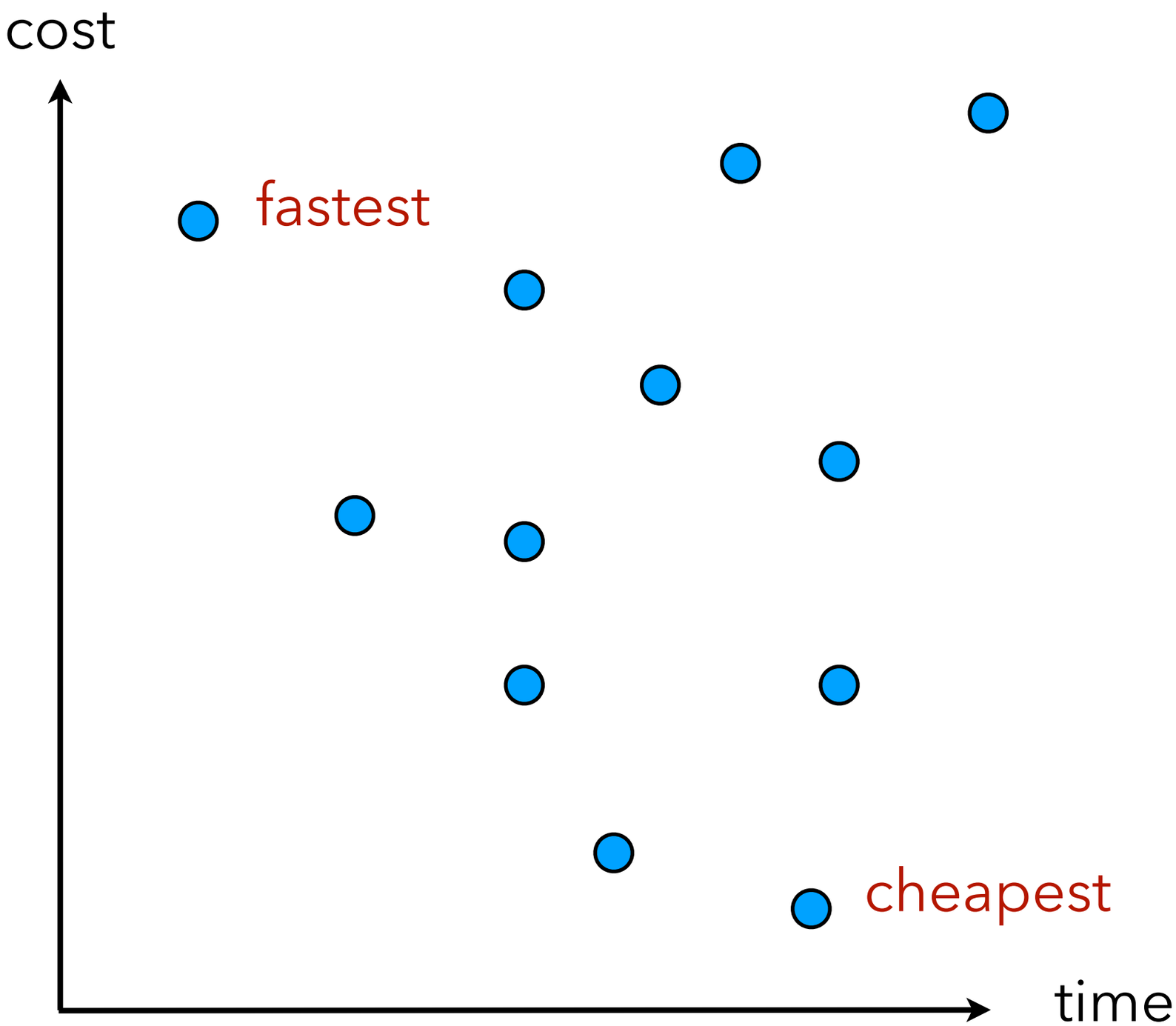
Multi-objective Optimization

Multi-objective Shortest Path

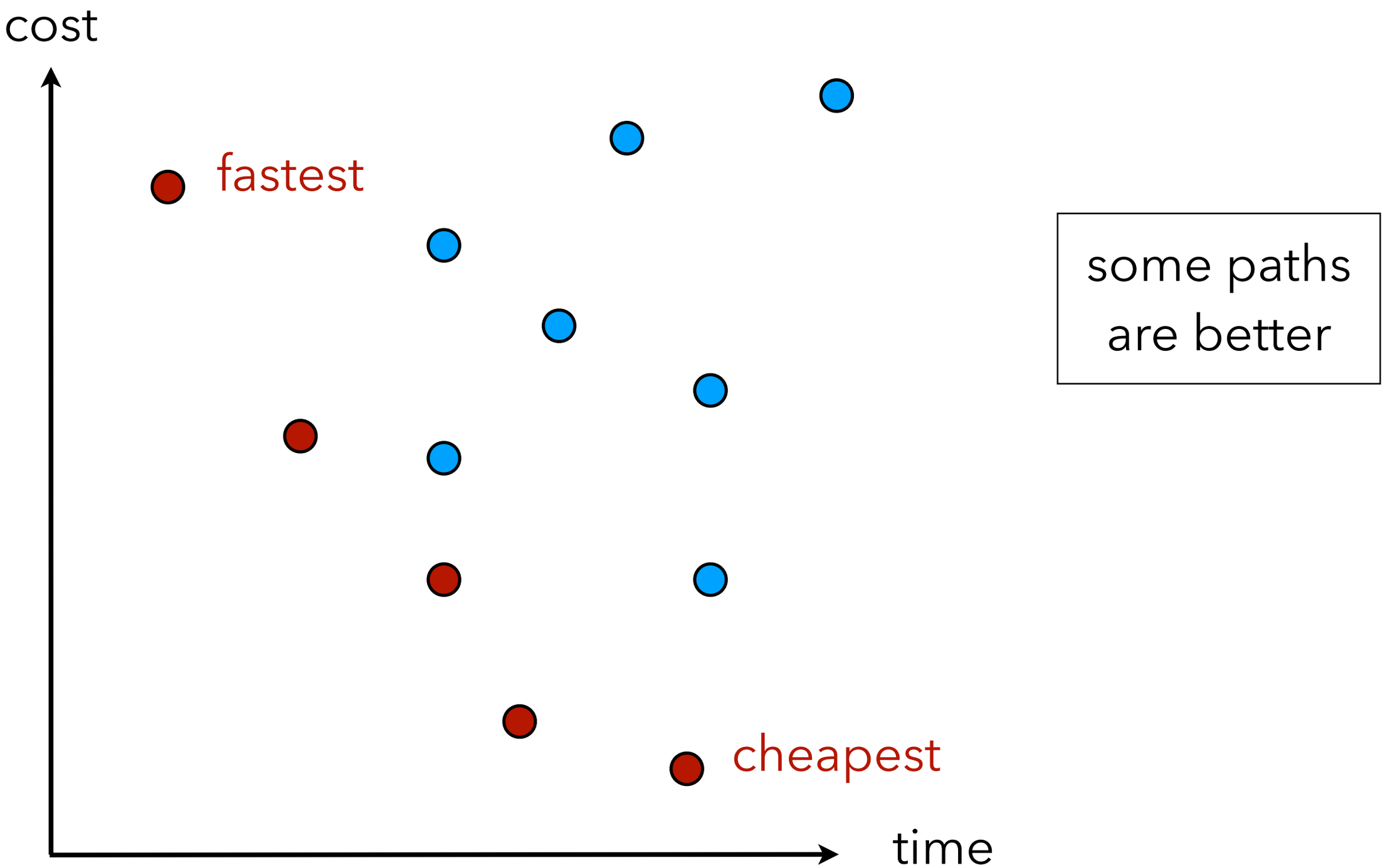
Multi-objective Shortest Path



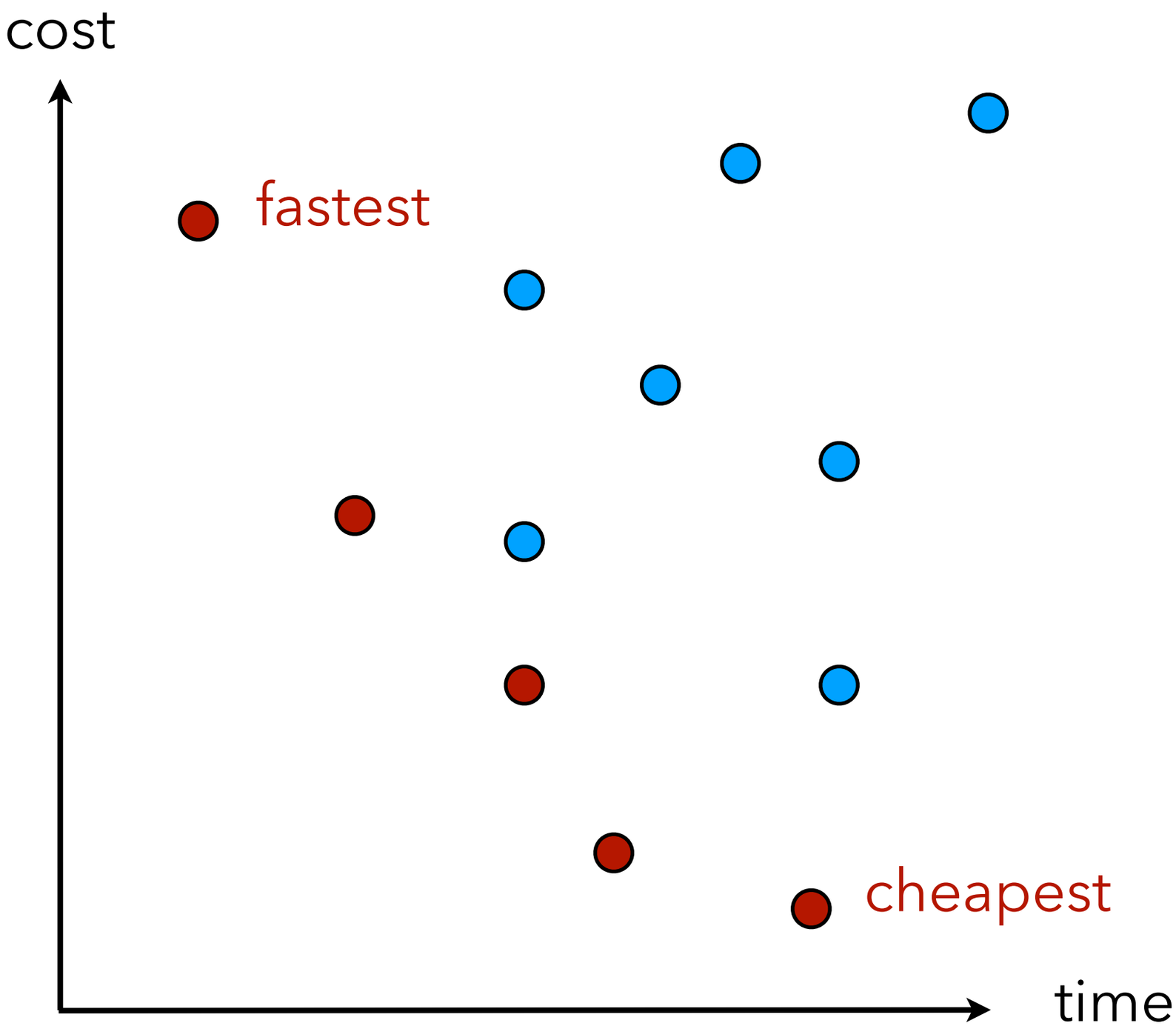
Multi-objective Shortest Path



Multi-objective Shortest Path



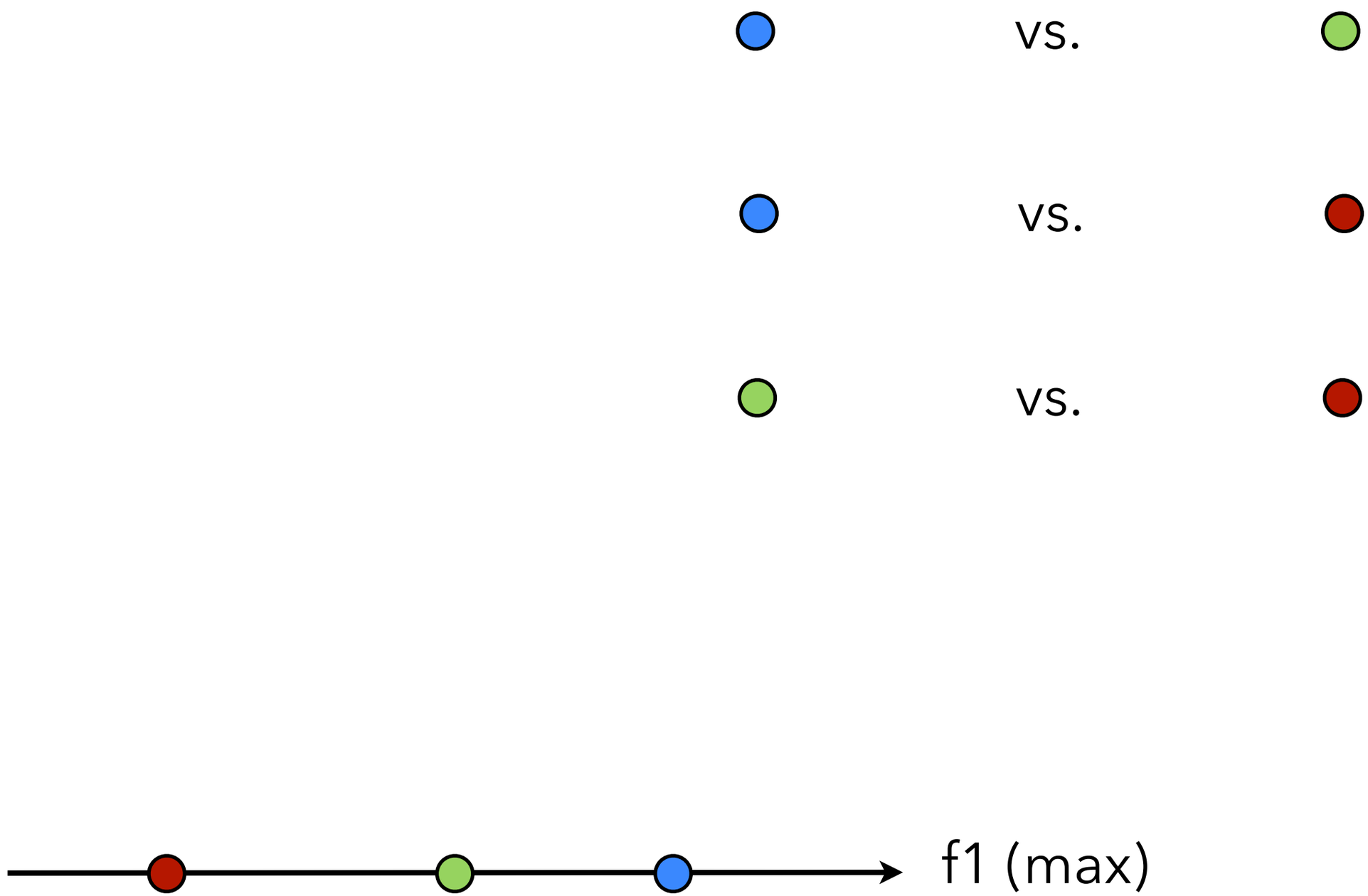
Multi-objective Shortest Path



some paths are better

which path is optimal?

Single vs. Multi-objective



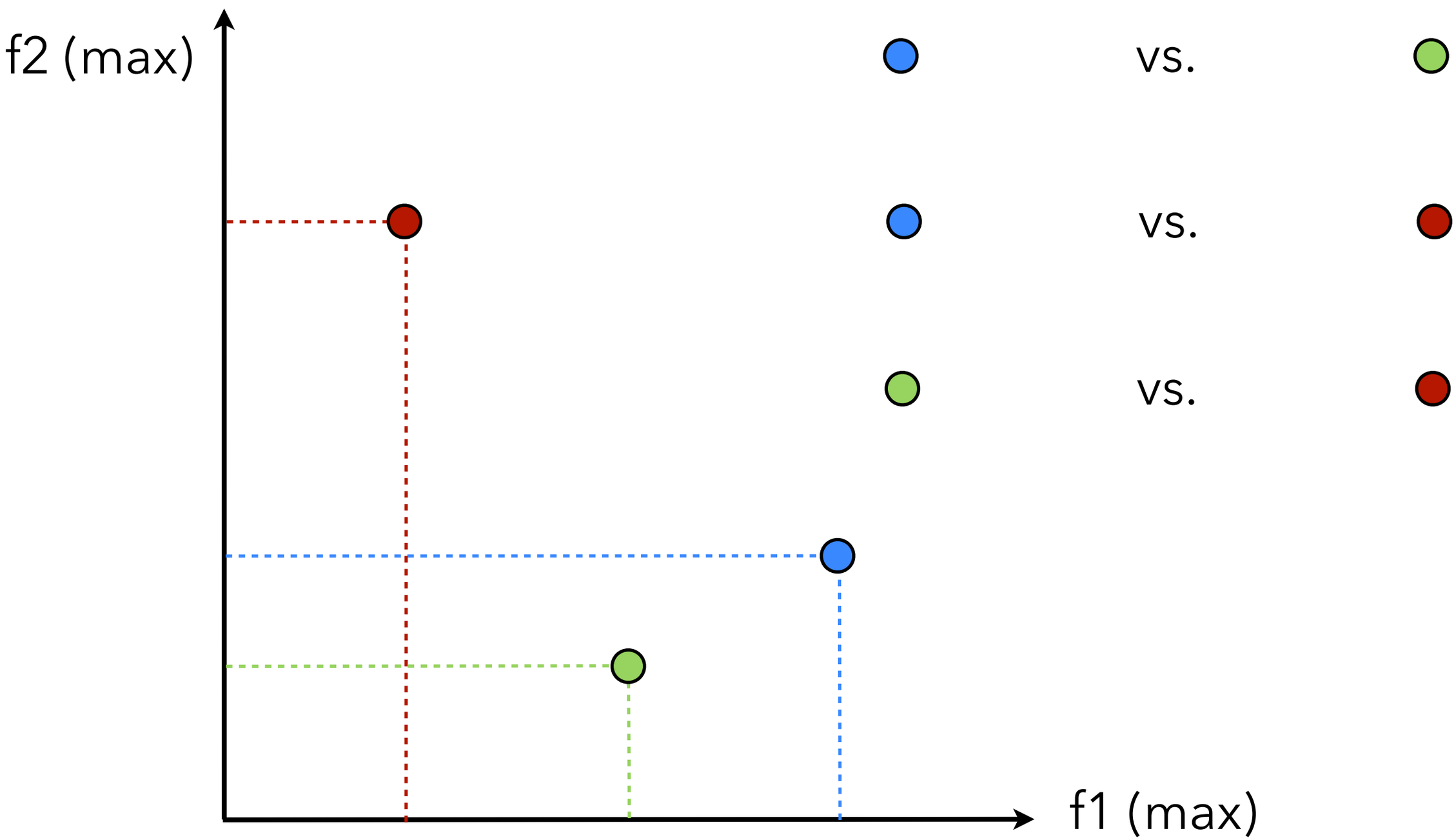
Single vs. Multi-objective

better than

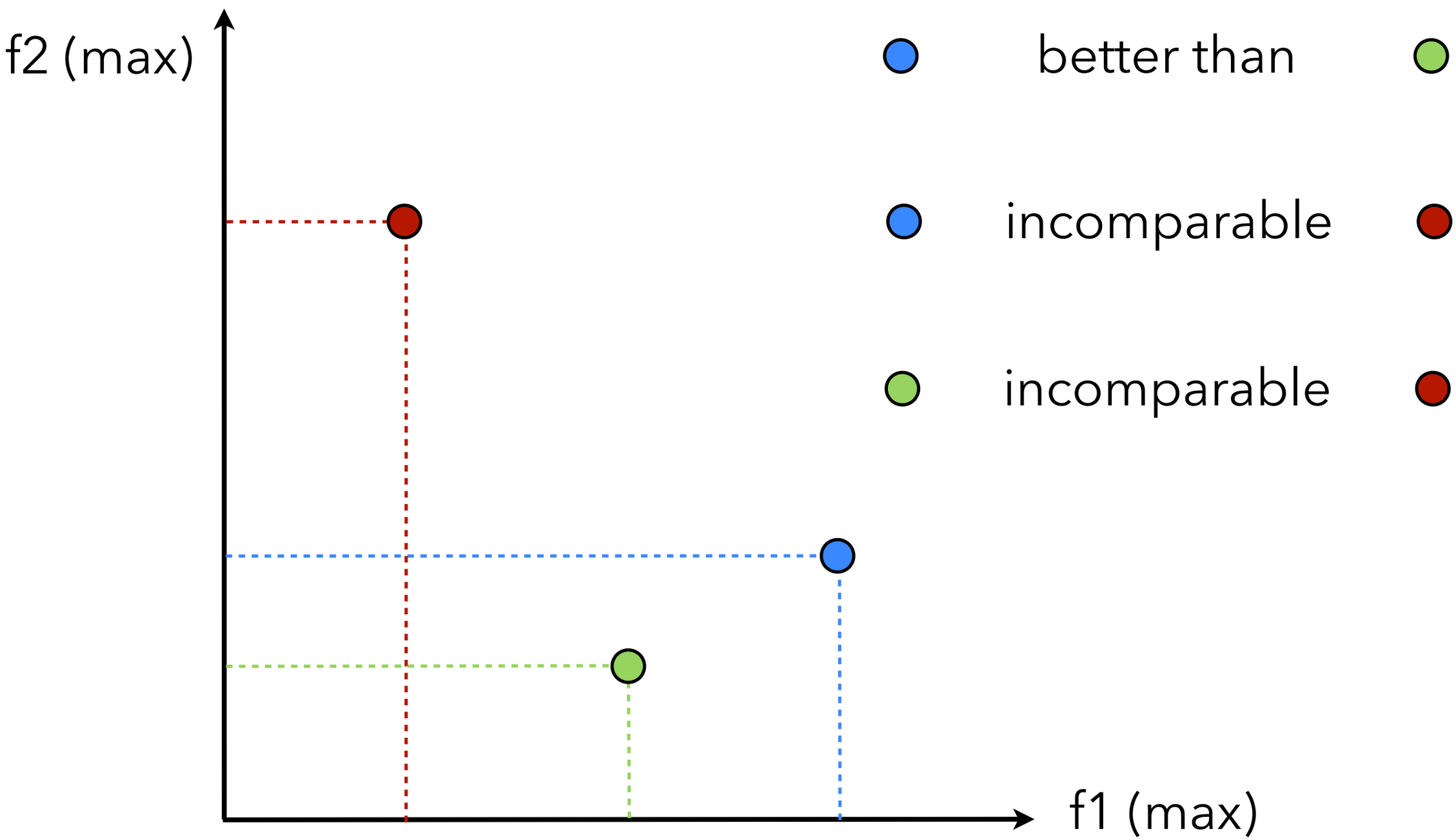
better than

better than

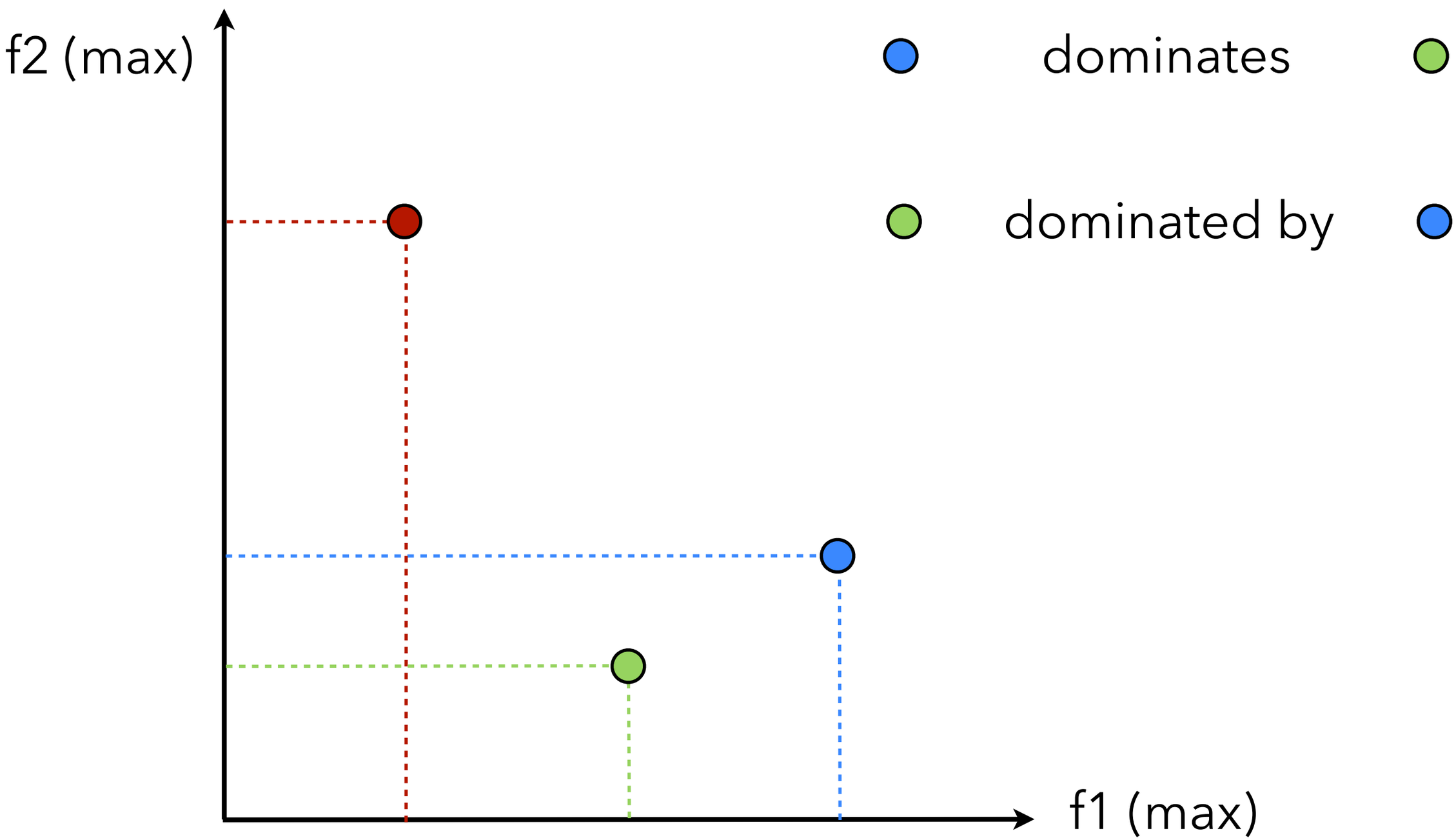
Single vs. Multi-objective



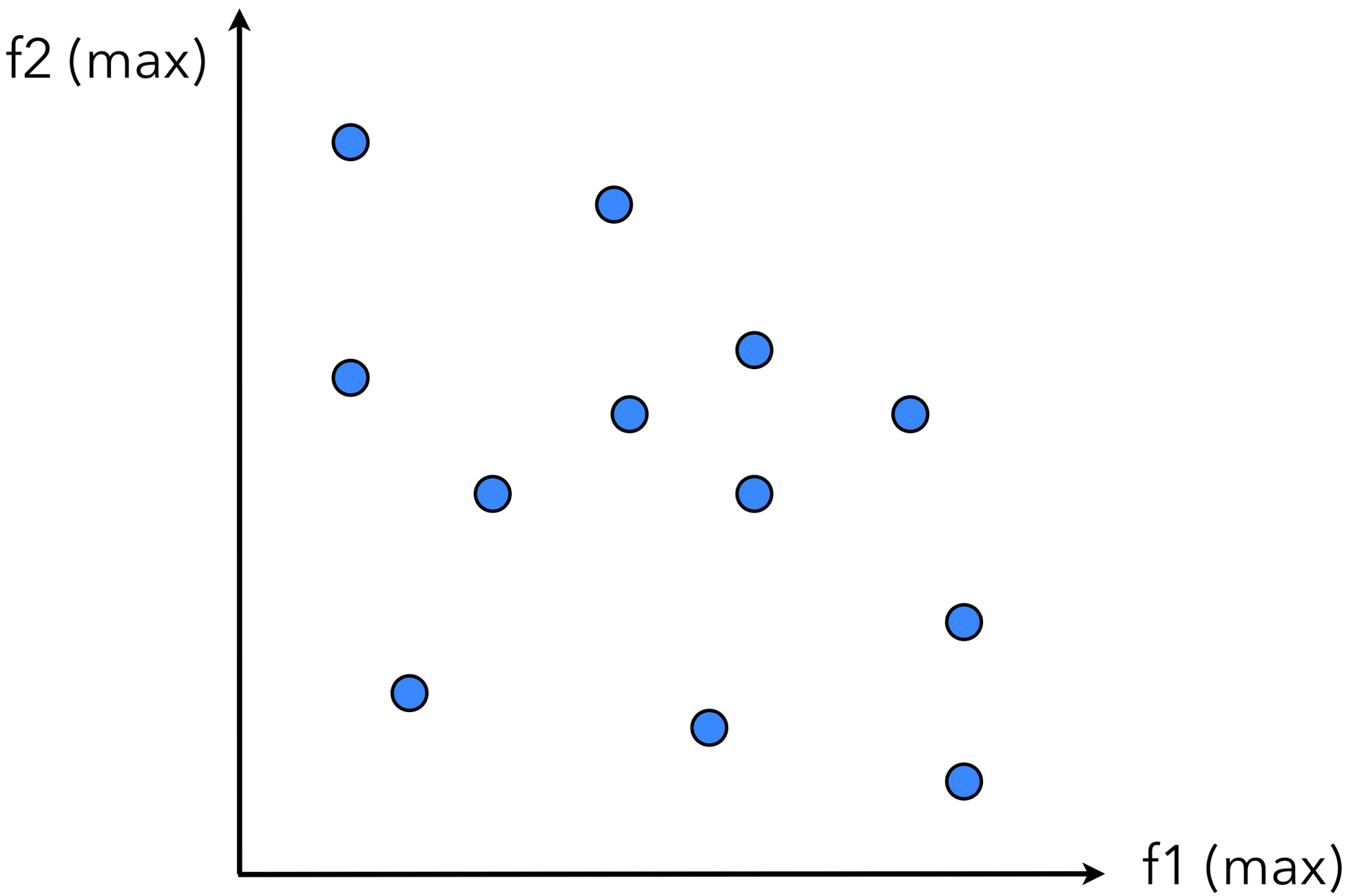
Single vs. Multi-objective



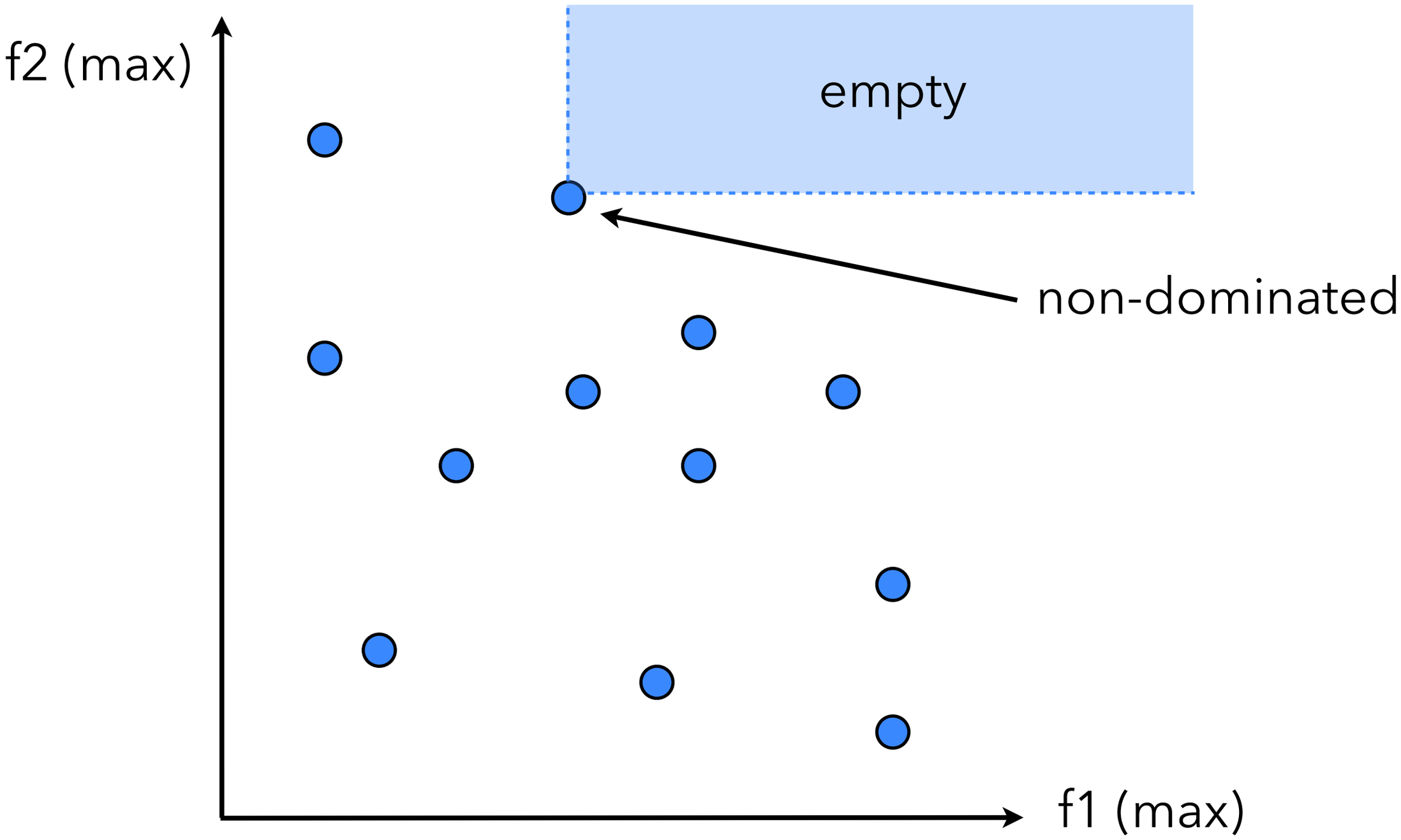
Pareto Dominance



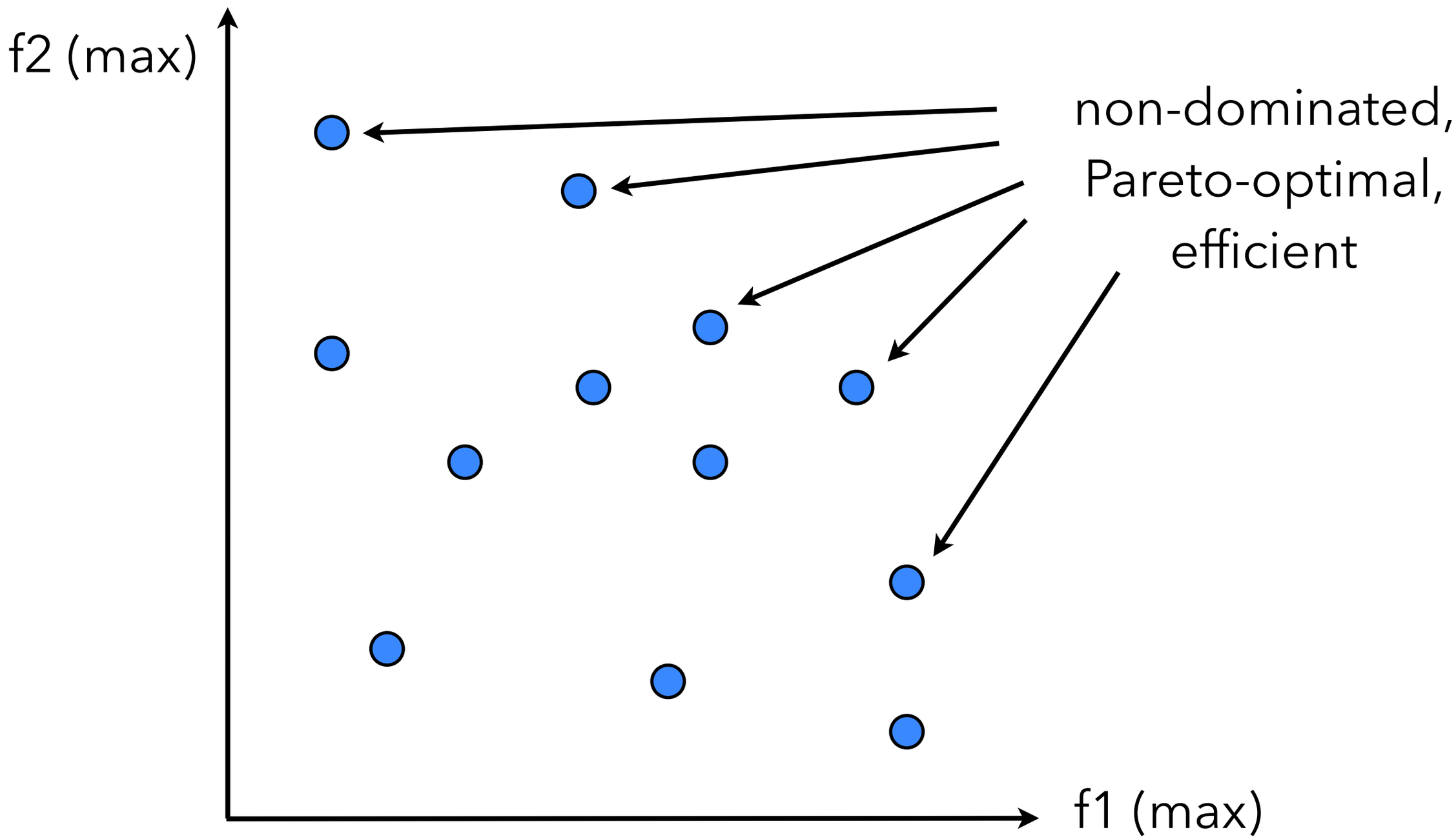
Pareto Dominance



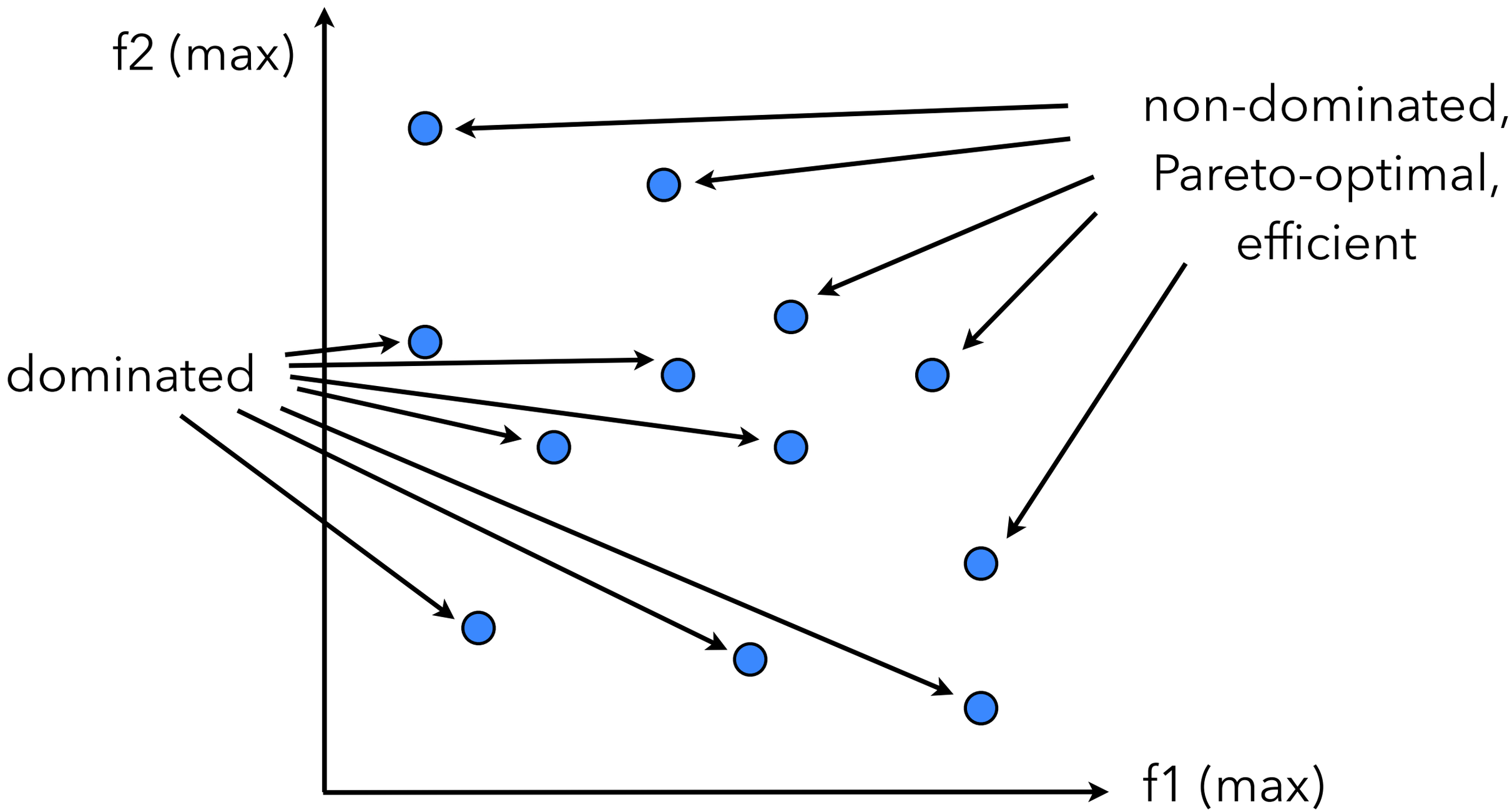
Pareto Dominance



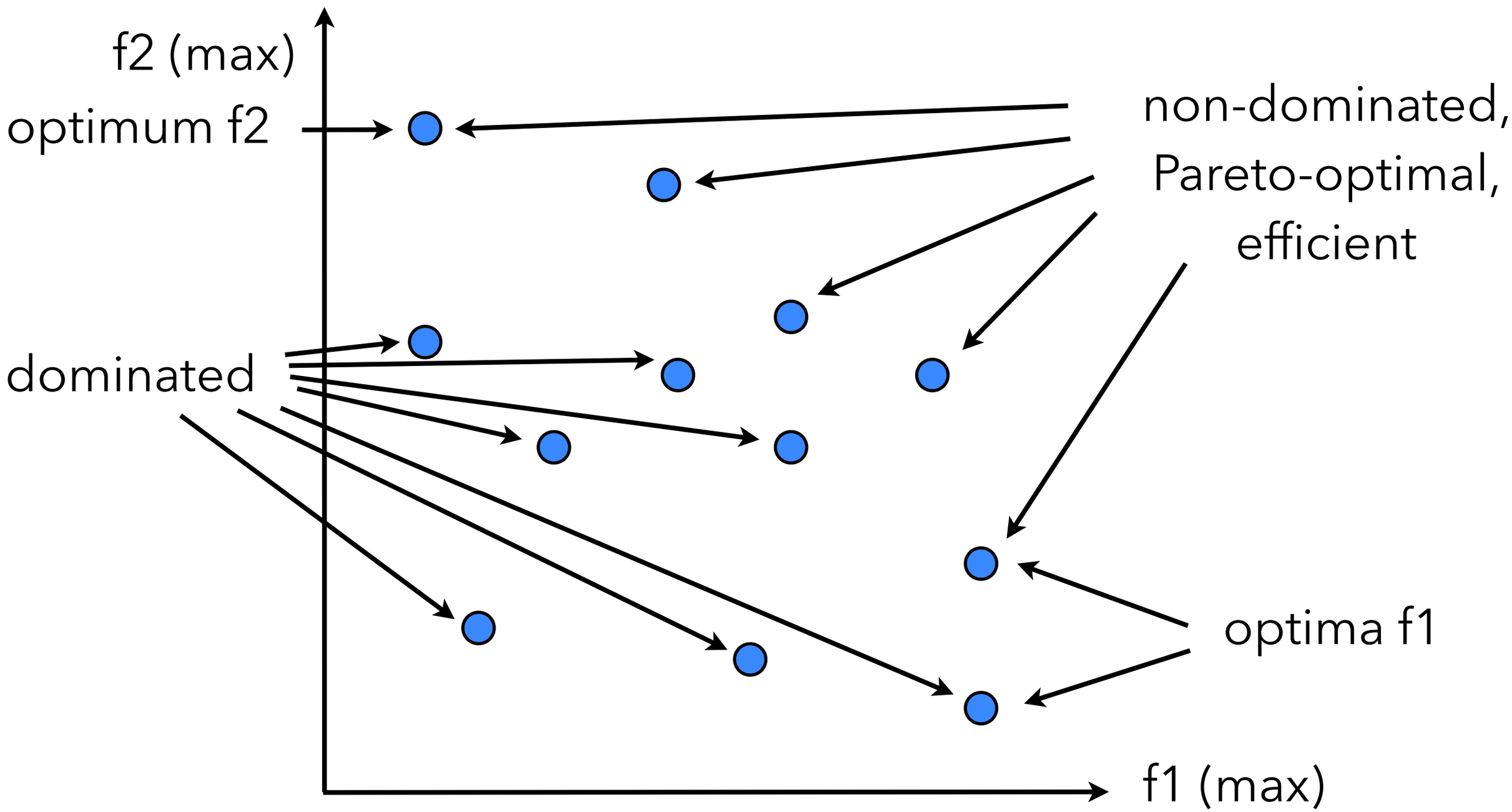
Pareto Dominance



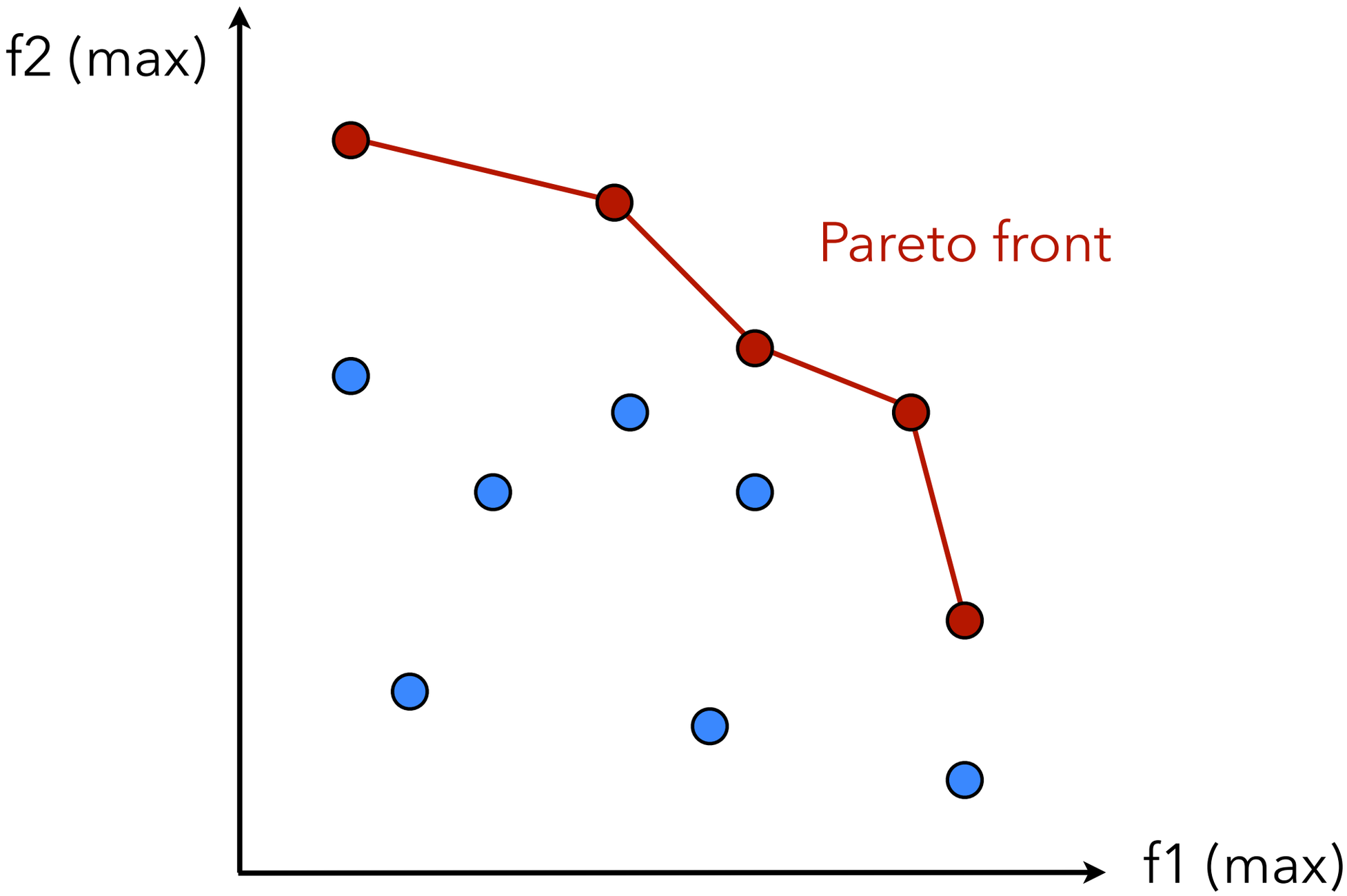
Pareto Dominance



Pareto Dominance

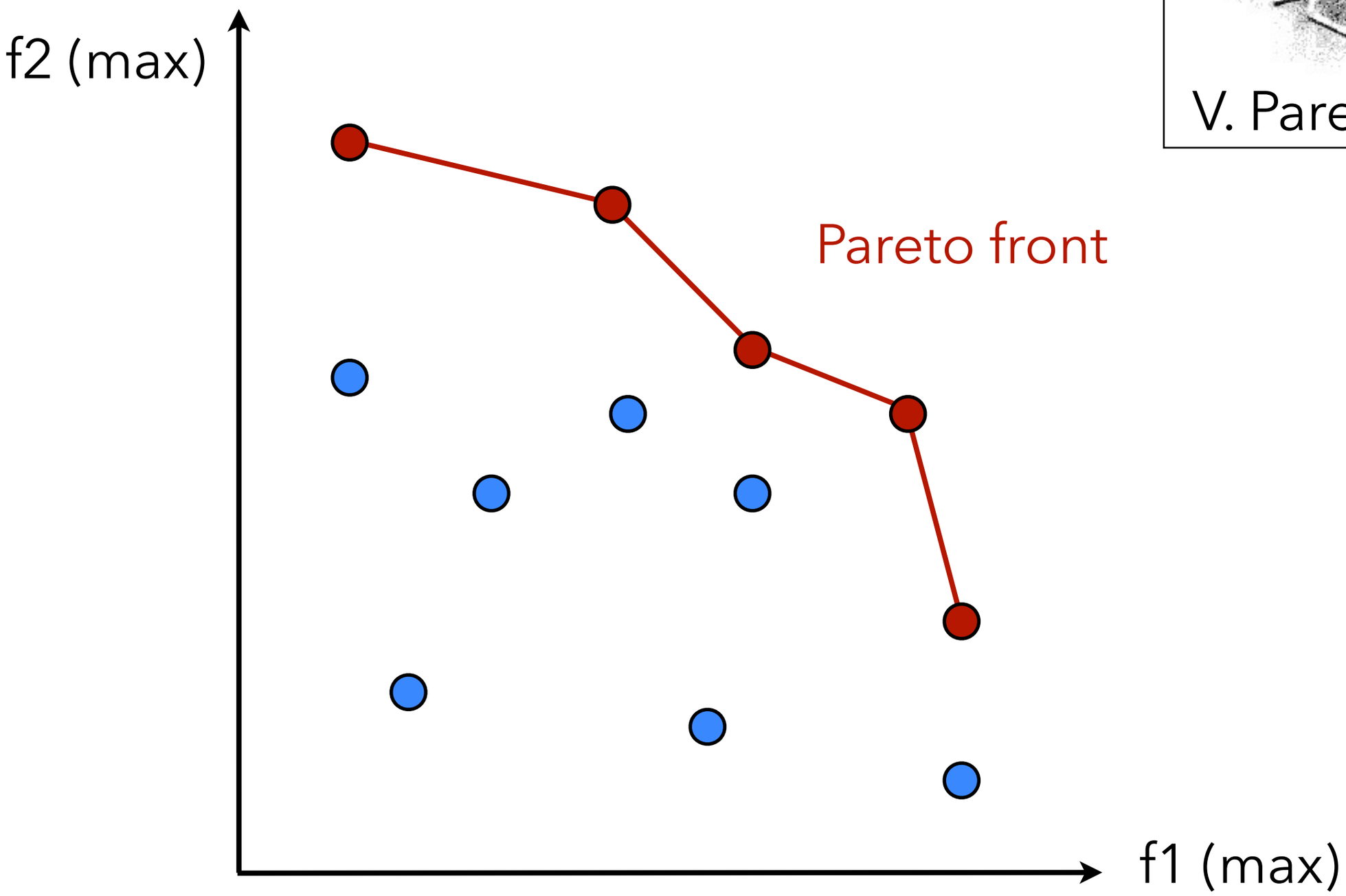


Pareto Front



Pareto Front

V. Pareto, 1886



Challenges

- Variables: many, heterogeneous, intricate structure
- Objectives: many, heterogeneous, black-box (expensive)
- NP-completeness: deciding if a solution is Pareto optimal is difficult for many multi-objective optimization problems
- Intractability: number of Pareto optimal solutions (non-dominated vectors) typically grows exponentially

What about a Pareto set approximation?

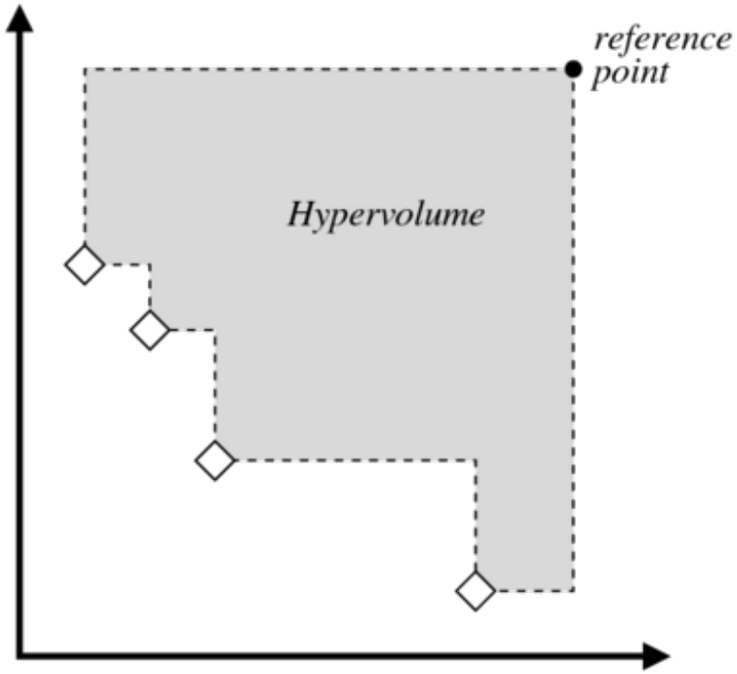
Pareto Set Approximation

Rule of thumb

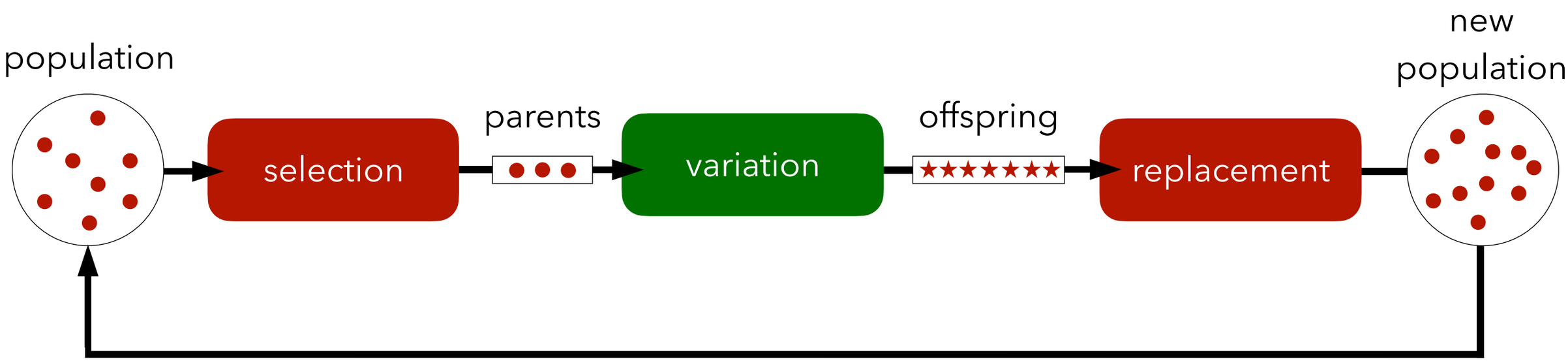
- closeness to the (exact) Pareto front
- well-distributed solutions in the objective space

Quality indicators

- scalar value that reflects approximation quality
  e.g. HV, EPS, IGD, R-metrics



EMO Algorithms



EMO Algorithms

(1) Decomposition approaches

- multiple aggregations of the objectives (e.g. weighted-sum)

  e.g. MOSA, MOTS, TPLS, MOEA/D

(2) Dominance-based approaches

- search process guided by a dominance relation

  e.g. NSGA-II, SPEA2, PAES, PLS, SEMO, AεSεH

(3) Indicator-based approaches

- search process guided by a quality indicator

  e.g. IBEA, IBMOLS, SMS-EMOA, HypE

Global vs. Local EMO Search

local search

multi-objective hill-climber

PLS

[Paquete et al. 2004]

global search

multi-objective $(1+1)-$EA

G-SEMO

[Laumanns et al. 2004]

$A \leftarrow \{x_0\}$
repeat
  select $x \in A$ at random
  for all x' s.t. $||x - x'||_1 = 1$ **do**
    $A \leftarrow$ non-dominated
      solutions from $A \cup \{x'\}$
  end for
until stop

$A \leftarrow \{x_0\}$
repeat
  select $x \in A$ at random
  $x' \leftarrow x$
  flip each bit x'_i with a rate $\frac{1}{n}$
  $A \leftarrow$ non-dominated
    solutions from $A \cup \{x'\}$
until stop

What Makes a Multi-objective Optimization Problem Difficult?

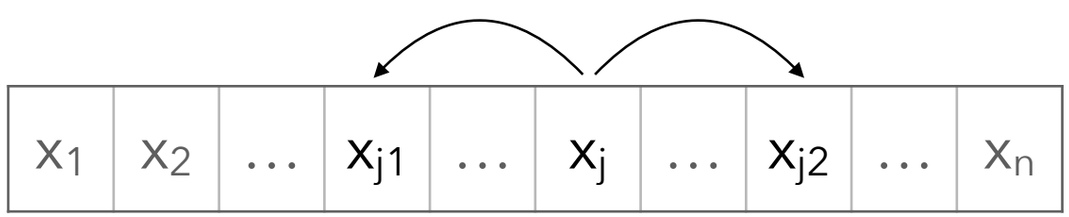
(Single-objective) nk Landscapes

[Kauffman 1993]

$$\max f(x) = \frac{1}{n}\sum_{j=1}^{n} c_j\,(x_j, x_{j_1}, \ldots, x_{j_k})$$

$$\textbf{s.t. } x_j \in \{0,1\} \qquad j \in \{1,\ldots,n\}$$

- number of variables n
- non-linearity k < n (variable interactions, epistasis)
- contributions values (c_j) ~ [0,1]



x_j x_{j1} x_{j2}	c_j
000	0.9
001	0.1
…	…
111	0.6

pmnk Landscapes

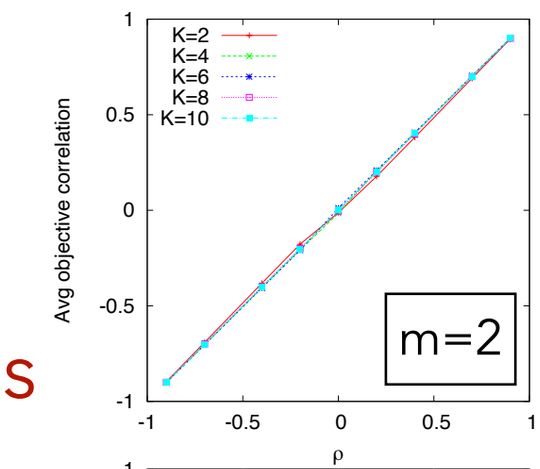
[Verel et al. 2013]

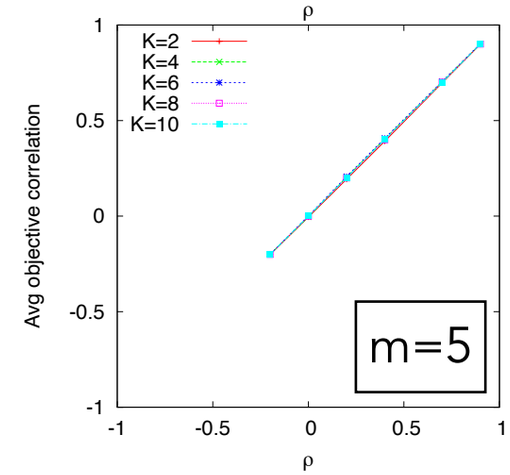
$$\max f_i(x) = \frac{1}{n}\sum_{j=1}^{n} c_j^i\,(x_j, x_{j_1}, \ldots, x_{j_k}) \qquad i \in \{1,\ldots,m\}$$

$$\textbf{s.t. } x_j \in \{0,1\} \qquad j \in \{1,\ldots,n\}$$

- number of variables n
- non-linearity k < n
- number of objectives m
- objective correlation ρ > −1 / (m−1)

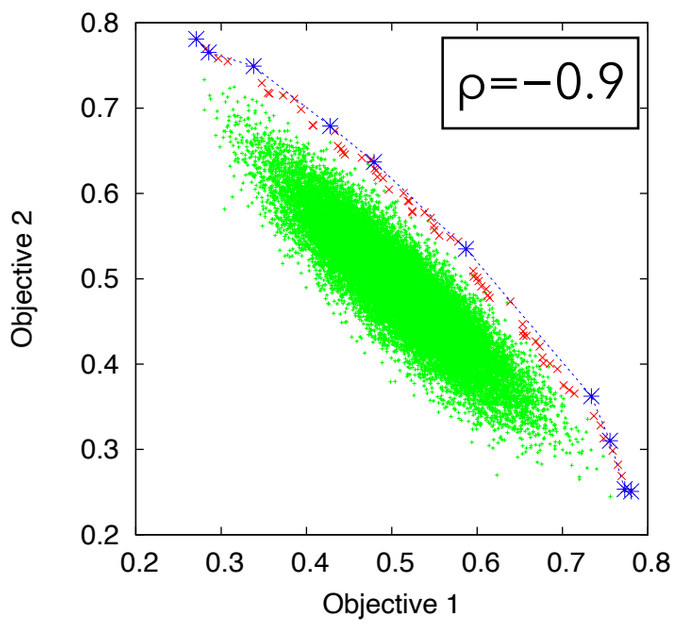
<u>unknown</u> for black-box problems



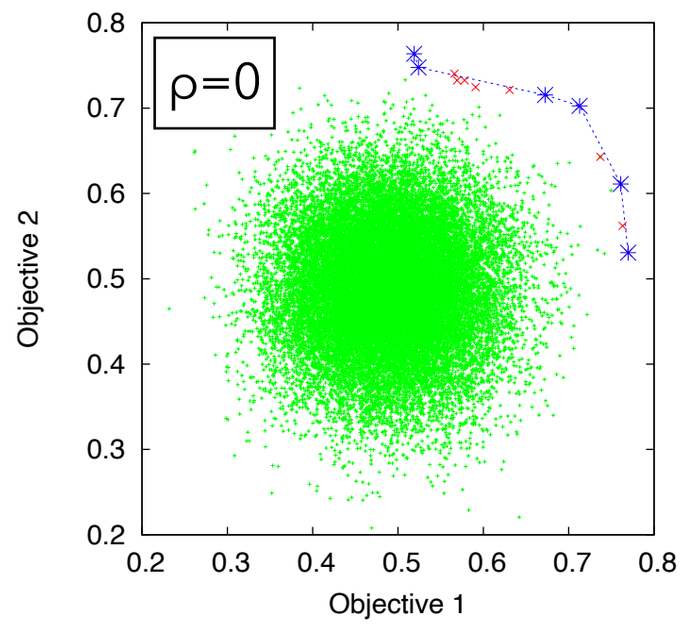


http://mocobench.sf.net

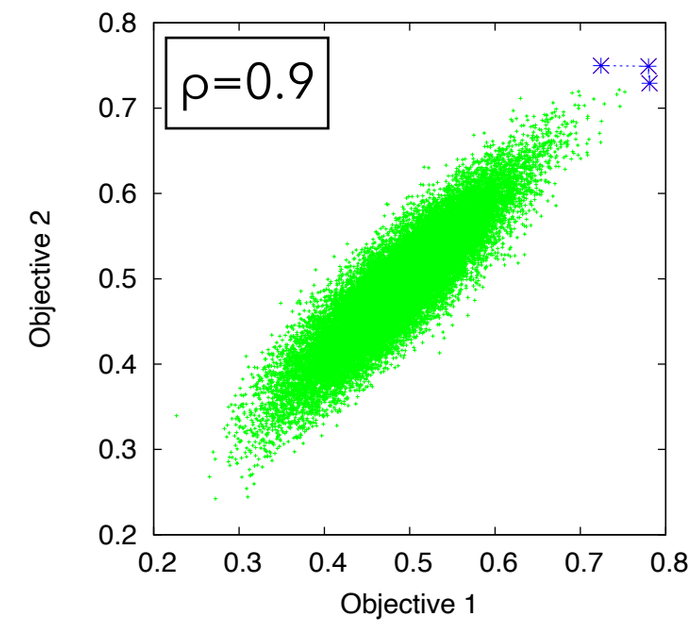
Objective Correlation (ρ)



conflicting objectives



independent objectives



correlated objectives

Experimental Setup

1000 landscapes generated by a design of experiments

- number of variables $n \in [\![64, 256]\!]$
- non-linearity $k \in [\![0, 8]\!]$
- number of objectives $m \in [\![2, 5]\!]$
- objective correlation $\rho \in [-1/(m-1), 1]$

large landscapes

Experimental Setup

Algorithms

- **NSGA-II** vs. **IBEA** vs. **MOEA/D** (default setting, population size = 100)

Performance

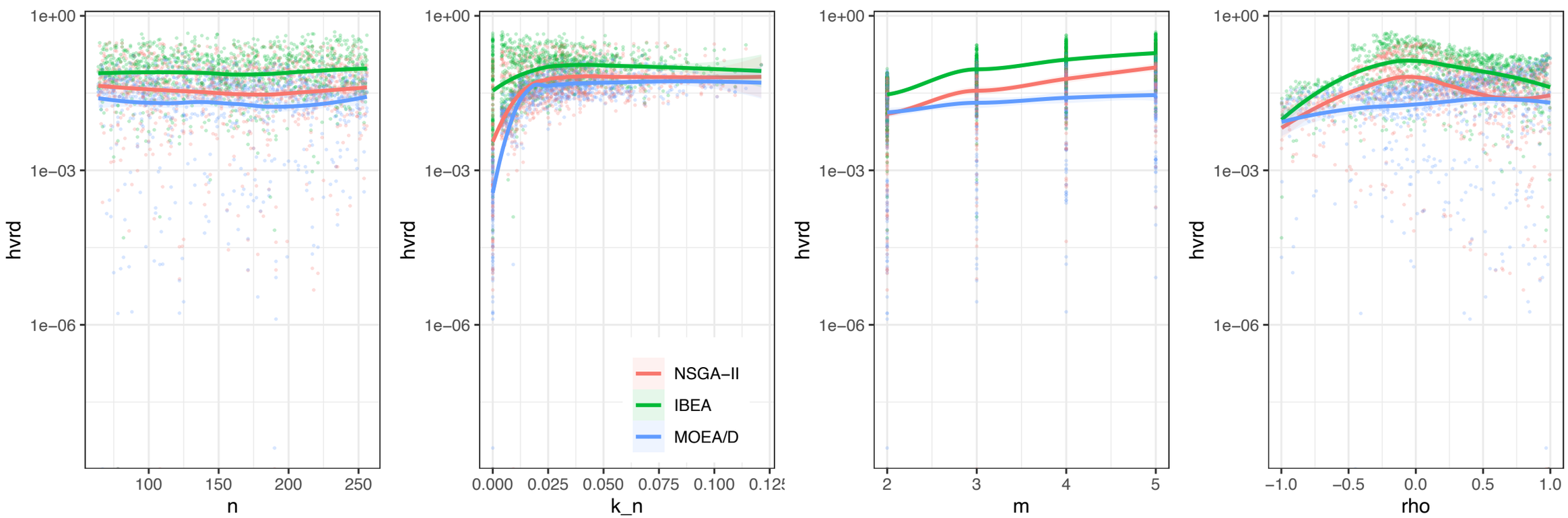
- 20 independent runs per instance, fixed budget of 1 000 000 evaluations
- (Expected) hypervolume relative deviation (hvrd)

Statistics

- Correlation = Spearman (non-parametric)
- Regression = random forest (default setting)

large landscapes

n, k, m, ρ vs. Performance

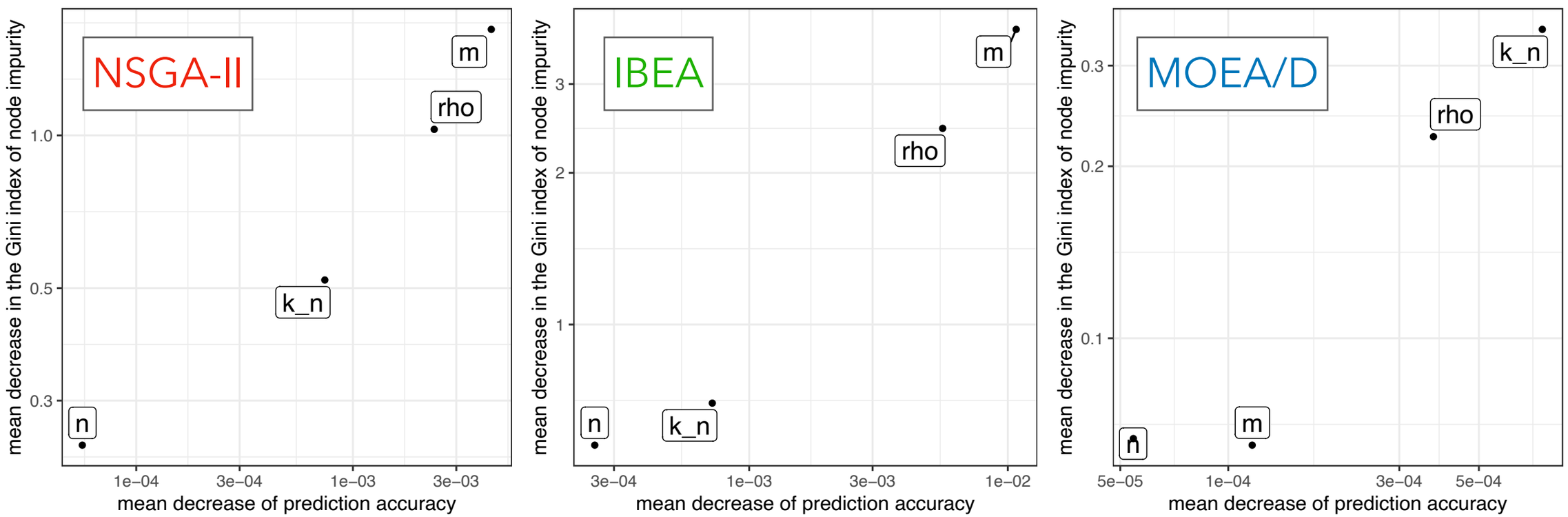


	cor (n, hvrd)	cor (k_n, hvrd)	cor (m, hvrd)	cor (ρ, hvrd)
NSGA-II	−0.01	0.40	0.66	−0.03
IBEA	0.03	0.18	0.78	−0.08
MOEA/D	−0.07	0.70	0.23	0.38

large landscapes

Variable Importance

Model (regression, RF) = hvrd ~ (n, k_n, m, ρ)



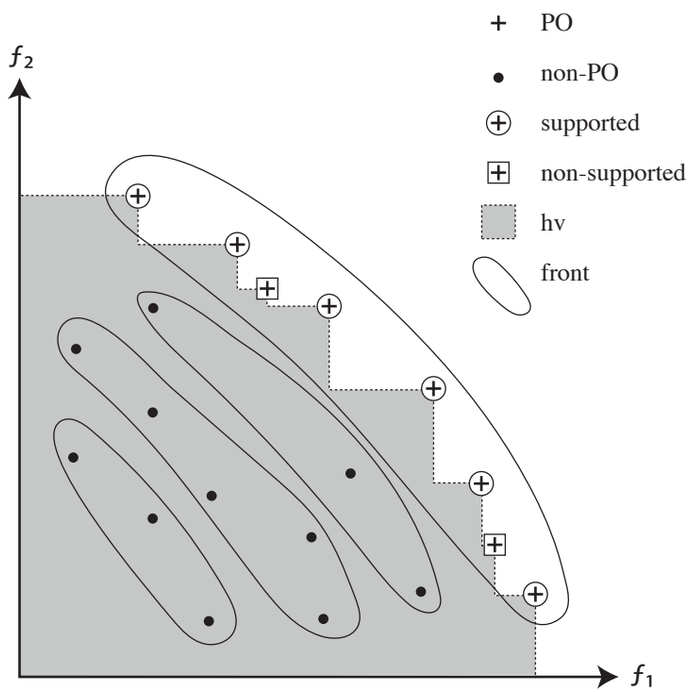
% var explained: 87.63

% var explained: 87.77

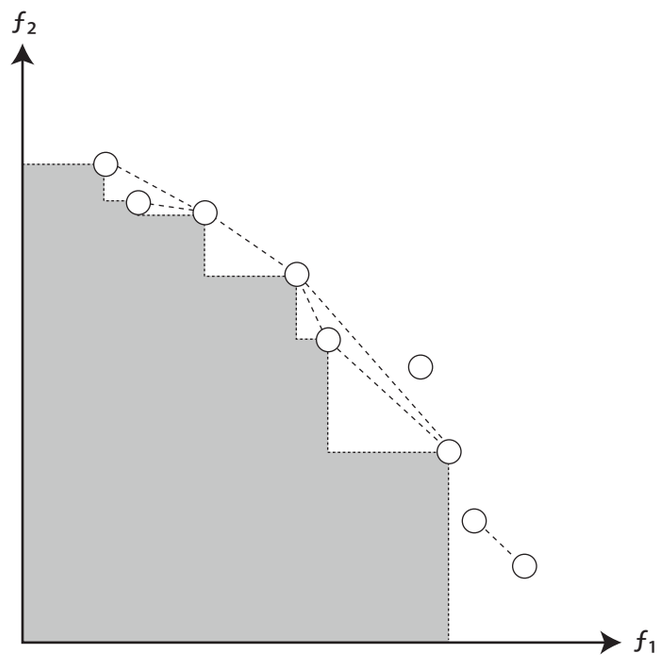
% var explained: 77.81

Multi-objective Landscape Features

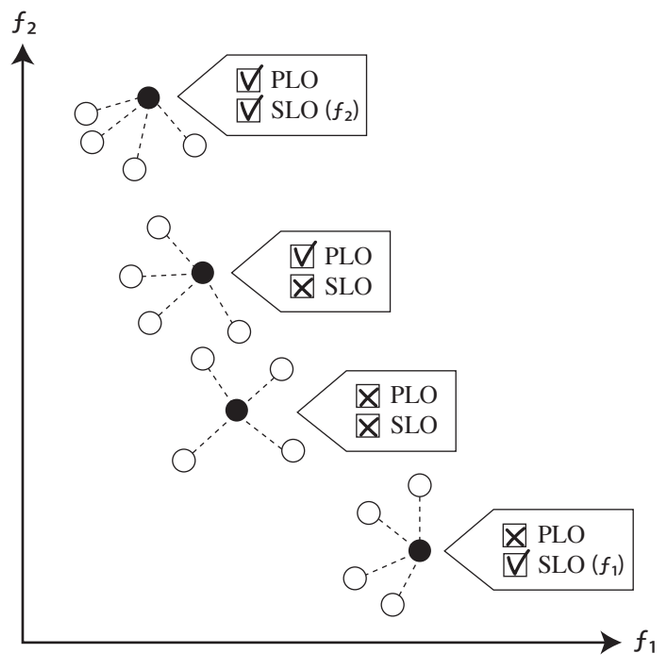
Global Features



Features from the solution space and Pareto set



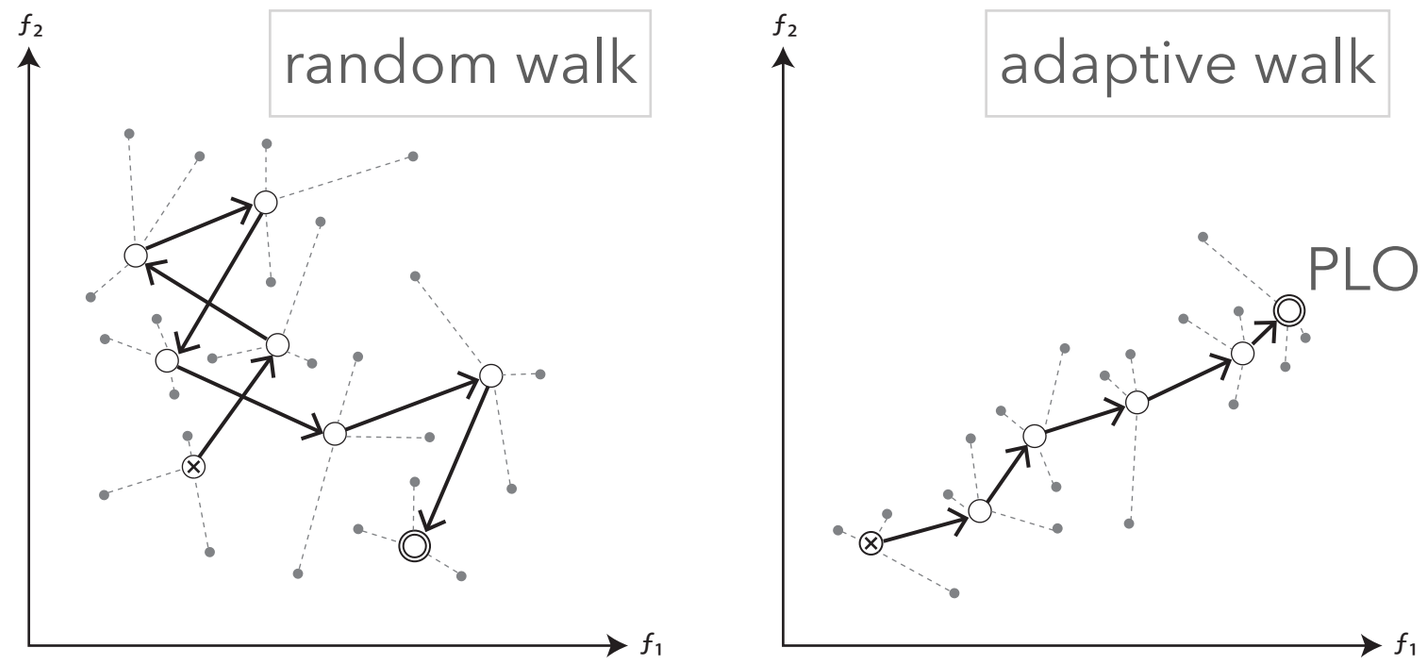
Features from the Pareto graph



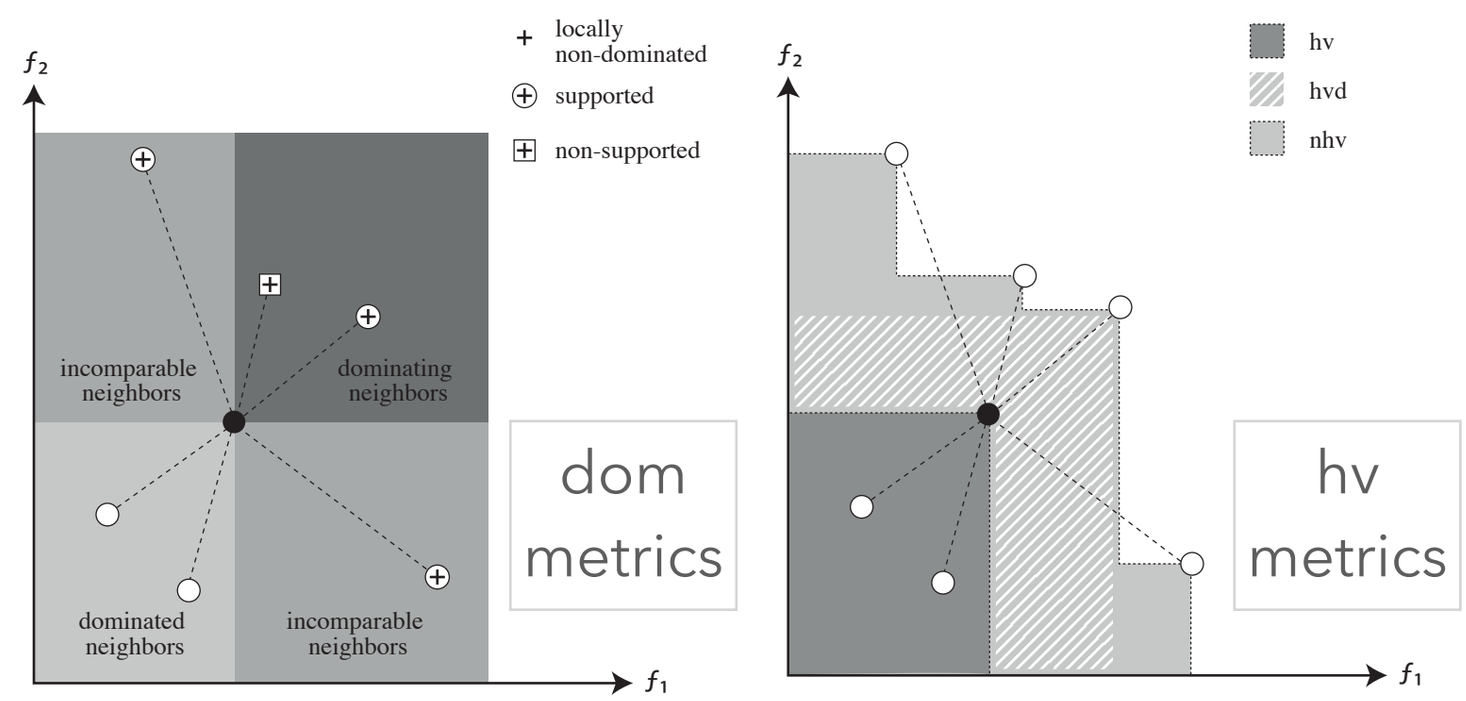
Multimodality / Local optimality

Local Features

1. sampling
   - walk $(x_0, x_1, \ldots, x_l)$ s.t. $x_t \in N(x_{t-1})$



2. measures
   - avg
   - r1 (first autocorrelation)
   - length l



	BENCHMARK **parameters** (4)
n	number of (binary) variables
k_n	proportional number of variable interactions (epistatic links) : k/n
m	number of objectives
ρ	correlation between the objective values
	GLOBAL FEATURES FROM **full enumeration** (16)
#po	proportion of Pareto optimal (PO) solutions
#supp	proportion of supported solutions in the Pareto set
hv	hypervolume-value of the (exact) Pareto front
#plo	proportion of Pareto local optimal (PLO) solutions
#slo_avg	average proportion of single-objective local optimal solutions per objective
podist_avg	average Hamming distance between Pareto optimal solutions
podist_max	maximal Hamming distance between Pareto optimal solutions (diameter of the Pareto set)
po_ent	entropy of binary variables from Pareto optimal solutions
fdc	fitness-distance correlation in the Pareto set (Hamming dist. in solution space vs. Manhattan dist. in objective space)
#cc	proportion of connected components in the Pareto graph
#sing	proportion of isolated Pareto optimal solutions (singletons) in the Pareto graph
#lcc	proportional size of the largest connected component in the Pareto graph
lcc_dist	average Hamming distance between solutions from the largest connected component
lcc_hv	proportion of hypervolume covered by the largest connected component
#fronts	proportion of non-dominated fronts
front_ent	entropy of the non-dominated front's size distribution
	LOCAL FEATURES FROM RANDOM WALK **sampling** (17)
hv_avg_rws	average (single) solution's hypervolume-value
hv_r1_rws	first autocorrelation coefficient of (single) solution's hypervolume-values
hvd_avg_rws	average (single) solution's hypervolume difference-value
hvd_r1_rws	first autocorrelation coefficient of (single) solution's hypervolume difference-values
nhv_avg_rws	average neighborhood's hypervolume-value
nhv_r1_rws	first autocorrelation coefficient of neighborhood's hypervolume-value
#lnd_avg_rws	average proportion of locally non-dominated solutions in the neighborhood
#lnd_r1_rws	first autocorrelation coefficient of the proportion of locally non-dominated solutions in the neighborhood
#lsupp_avg_rws	average proportion of supported locally non-dominated solutions in the neighborhood
#lsupp_r1_rws	first autocorrelation coefficient of the proportion of supported locally non-dominated solutions in the neighborhood
#inf_avg_rws	average proportion of neighbors dominated by the current solution
#inf_r1_rws	first autocorrelation coefficient of the proportion of neighbors dominated by the current solution
#sup_avg_rws	average proportion of neighbors dominating the current solution
#sup_r1_rws	first autocorrelation coefficient of the proportion of neighbors dominating the current solution
#inc_avg_rws	average proportion of neighbors incomparable to the current solution
#inc_r1_rws	first autocorrelation coefficient of the proportion of neighbors incomparable to the current solution
f_cor_rws	estimated correlation between the objective values
	LOCAL FEATURES FROM ADAPTIVE WALK **sampling** (9)
hv_avg_aws	average (single) solution's hypervolume-value
hvd_avg_aws	average (single) solution's hypervolume difference-value
nhv_avg_aws	average neighborhood's hypervolume-value
#lnd_avg_aws	average proportion of locally non-dominated solutions in the neighborhood
#lsupp_avg_aws	average proportion of supported locally non-dominated solutions in the neighborhood
#inf_avg_aws	average proportion of neighbors dominated by the current solution
#sup_avg_aws	average proportion of neighbors dominating the current solution
#inc_avg_aws	average proportion of neighbors incomparable to the current solution
length_aws	average length of Pareto-based adaptive walks

Experimental Setup

60480 instances generated by factorial design

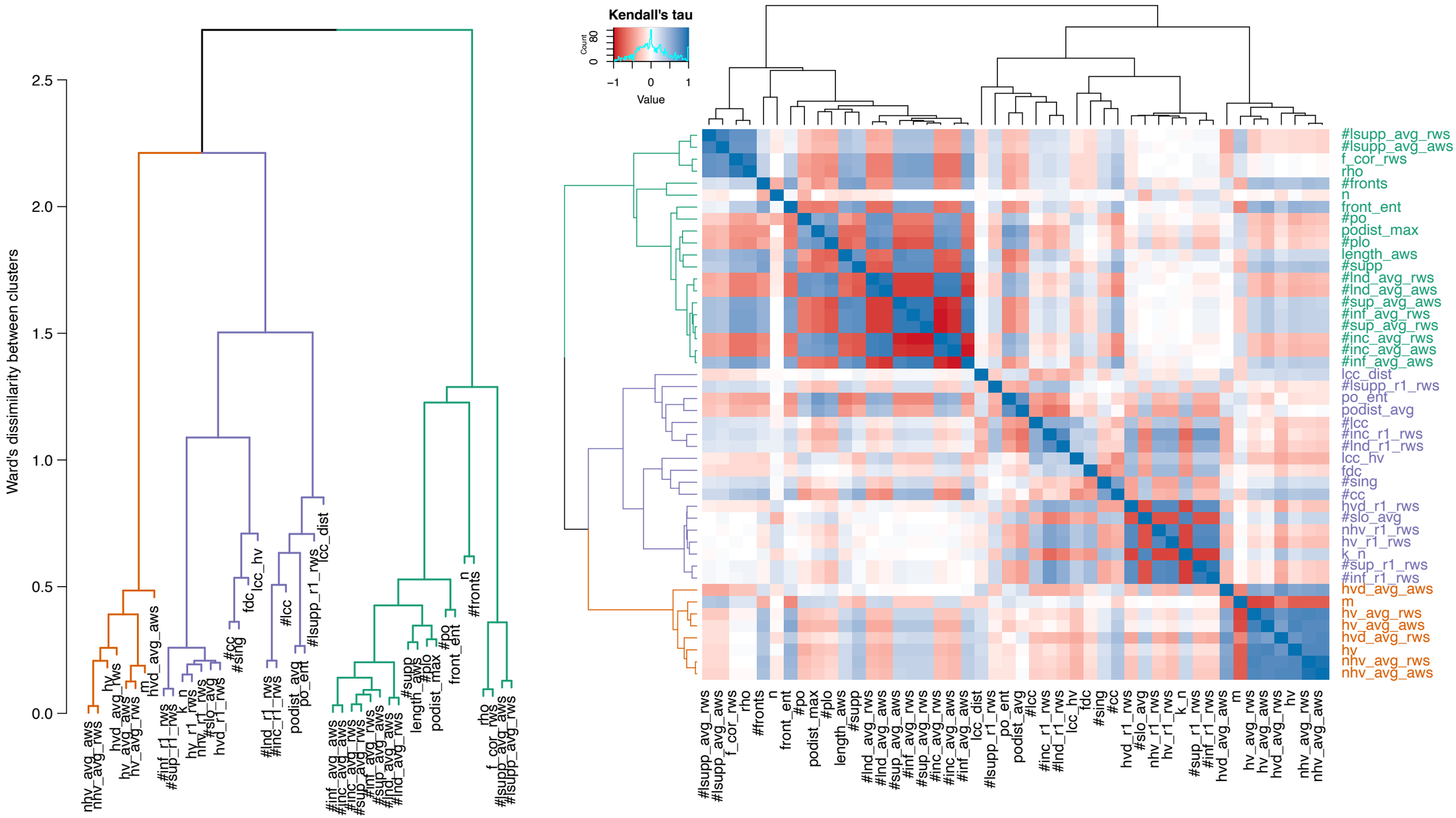
(30 per setting)

- number of variables $n \in \{10, 11, 12, 13, 14, 15, 16\}$
- non-linearity $k \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$
- number of objectives $m \in \{2, 3, 4, 5\}$
- objective correlation $\rho > -1/(m-1)$

  $\rho \in \{-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1\}$

small landscapes

Pairwise Feature Correlation



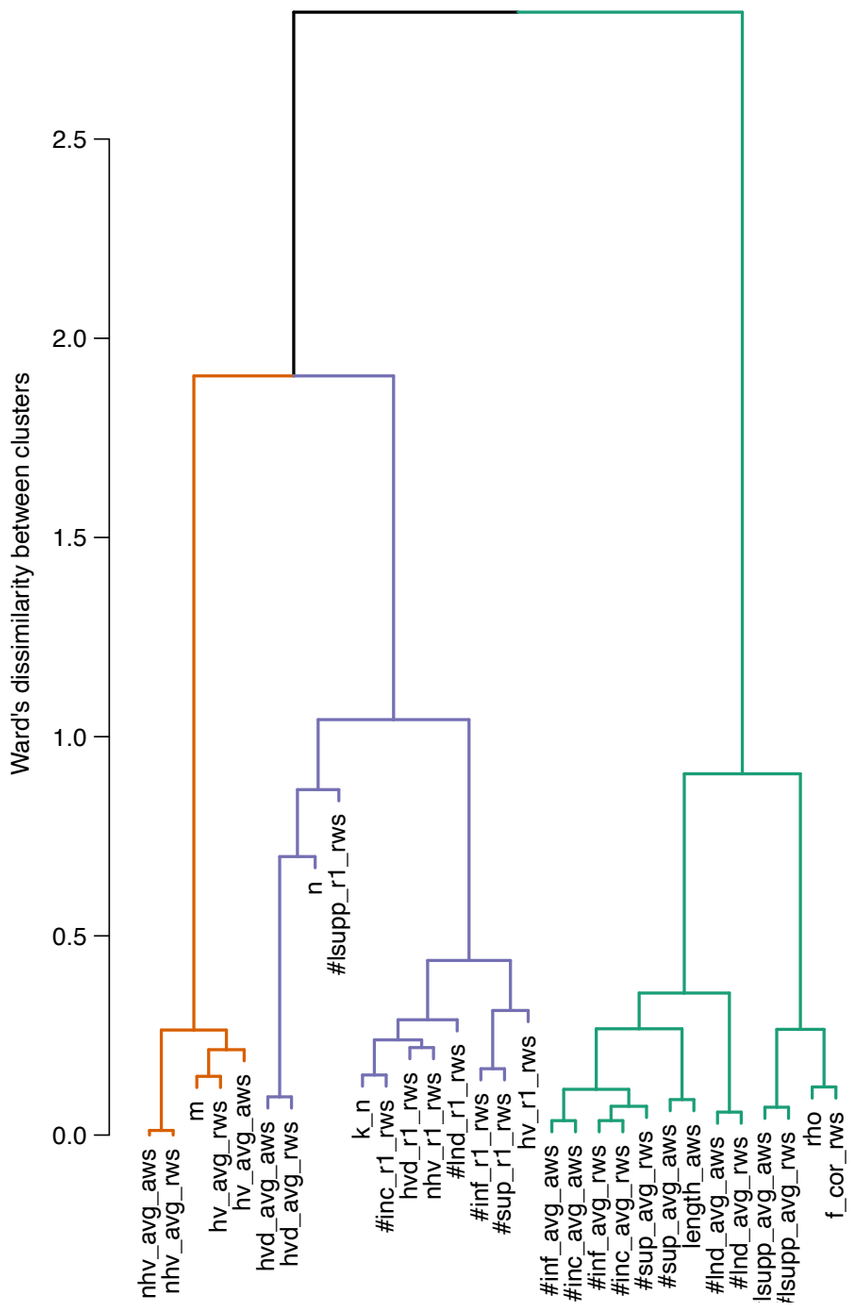
Experimental Setup

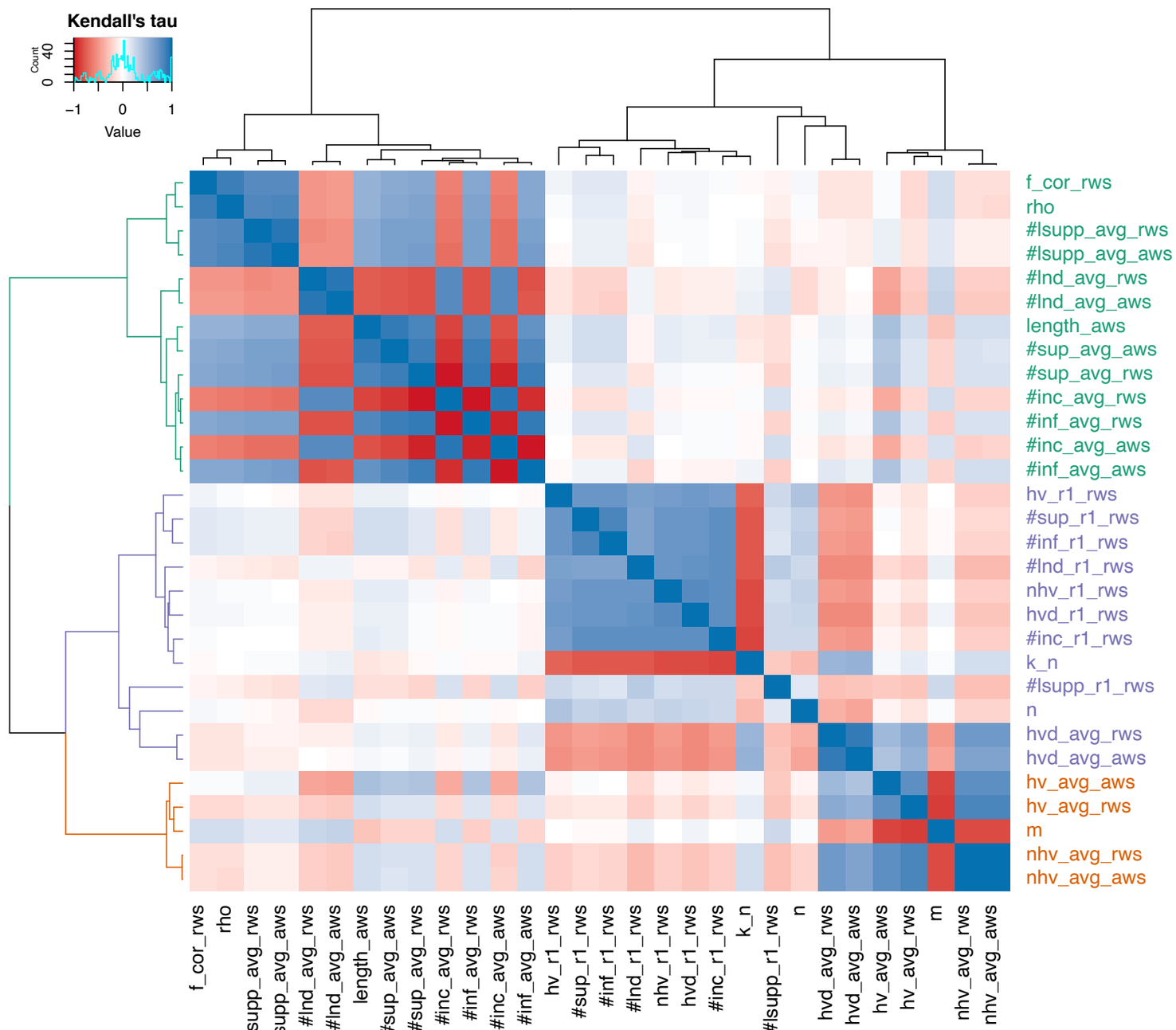
1000 landscapes generated by a design of experiments

- number of variables $n \in [\![64, 256]\!]$
- non-linearity $k \in [\![0, 8]\!]$
- number of objectives $m \in [\![2, 5]\!]$
- objective correlation $\rho \in [-1/(m-1), 1]$

large landscapes

Pairwise Feature Correlation





Experimental Setup

Algorithms

- G-SEMO + I-PLS

Performance

- 30 independent runs per instance, fixed budget of 100 000 evaluations
- (Expected) epsilon approximation ratio to best non-dominated set

Statistics

- Regression = extremely randomized trees

large landscapes

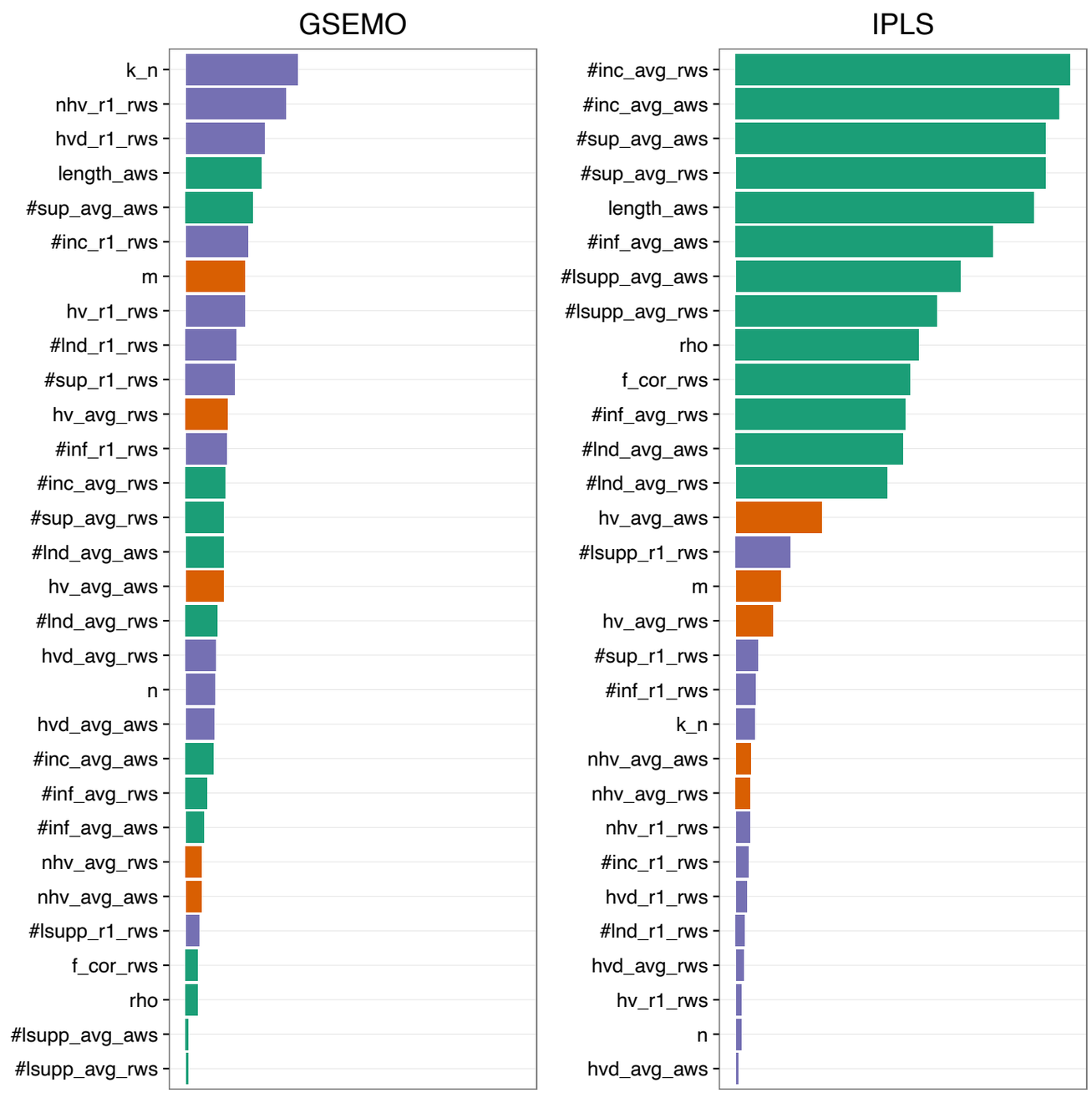
Prediction Accuracy

algo.	set of features	MAE avg	MAE std	MSE avg	MSE std	R^2 avg	R^2 std	adjusted R^2 avg	adjusted R^2 std	rank
G-SEMO	all features	**0.003049**	0.000285	**0.000017**	0.000004	0.891227	0.024584	0.843934	0.035273	1
	local features	**0.003152**	0.000295	**0.000018**	0.000004	0.883909	0.026863	0.838126	0.037457	1
	local features (random walk)	**0.003220**	0.000314	0.000019	0.000004	0.878212	0.028956	0.849287	0.035833	1.5
	local features (adaptive walk)	0.003525	0.000329	0.000023	0.000006	0.854199	0.032339	0.834089	0.036799	5
	$\{\rho, \mathtt{m}, \mathtt{n}, \mathtt{k_n}\}$	**0.003084**	0.000270	**0.000017**	0.000003	0.892947	0.020658	0.888440	0.021528	1
	$\{\mathtt{m}, \mathtt{n}\}$	0.010813	0.000830	0.000206	0.000030	-0.303336	0.188046	-0.330209	0.191923	6
I-PLS	all features	**0.004290**	0.000430	**0.000034**	0.000008	0.886568	0.026980	0.837249	0.038710	1
	local features	**0.004359**	0.000423	**0.000035**	0.000008	0.883323	0.027274	0.837309	0.038030	1
	local features (random walk)	**0.004449**	0.000394	**0.000036**	0.000008	0.879936	0.026335	0.851421	0.032589	1
	local features (adaptive walk)	0.004663	0.000403	0.000039	0.000008	0.871011	0.025903	0.853219	0.029476	3.5
	$\{\rho, \mathtt{m}, \mathtt{n}, \mathtt{k_n}\}$	**0.004353**	0.000320	**0.000033**	0.000006	0.889872	0.024505	0.885235	0.025537	1
	$\{\mathtt{m}, \mathtt{n}\}$	0.016959	0.001473	0.000472	0.000077	-0.568495	0.228629	-0.600836	0.233343	6

random subsampling cross-validation
(50 iter., 90/10 split)

large landscapes

Feature Importance



mean decrease in node impurity

large landscapes

Experimental Setup

Algorithms

- **NSGA-II** vs. **IBEA** vs. **MOEA/D** (default setting, population size = 100)

Performance

- 20 independent runs per instance, fixed budget of 1 000 000 evaluations
- (Expected) hypervolume relative deviation (hvrd)

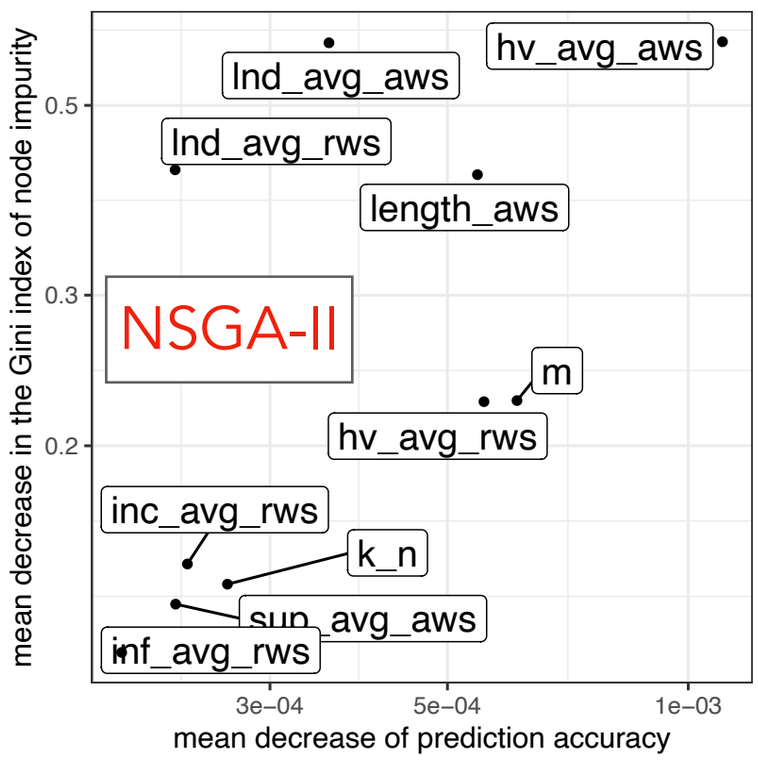
Statistics

- Classification = extremely randomized trees, decision tree (default)

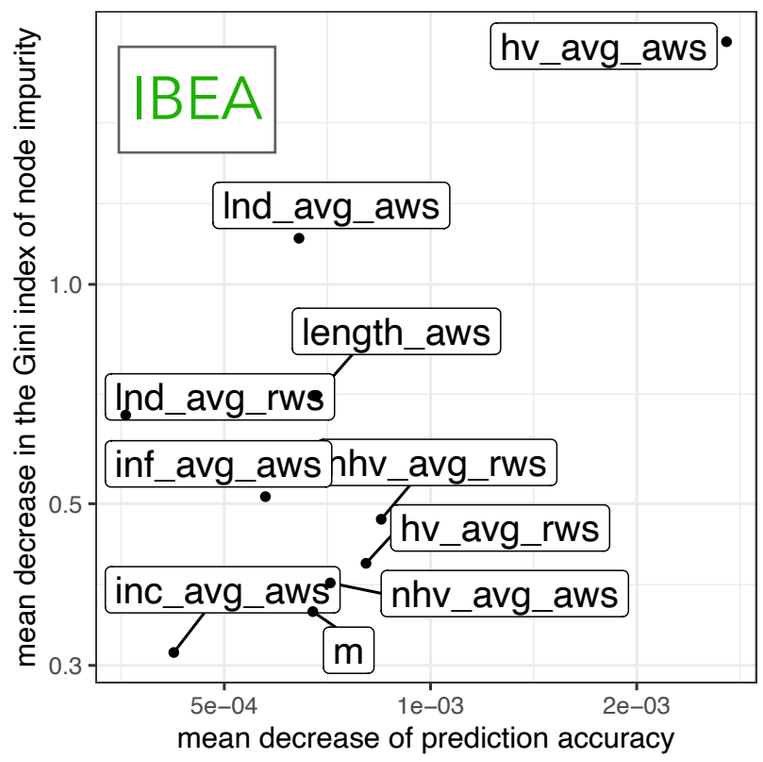
large landscapes

Feature Importance (top 10)

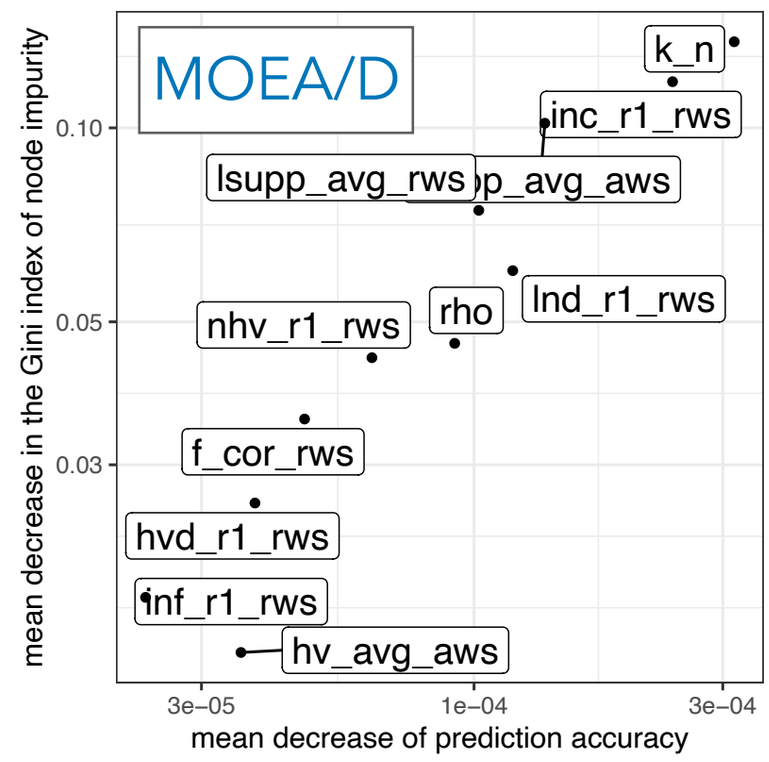
Model (regression, RF) = hvrd ~ (n, k_n, m, ρ, {features})



% var explained: 94.43



% var explained: 96.77



% var explained: 82.86

large landscapes

Algorithm Selection

Algorithm portfolio = {NSGA-II, IBEA, MOEA/D}

Model (classif, RF) = {algo} ~ (n, k_n, m, ρ, {features})

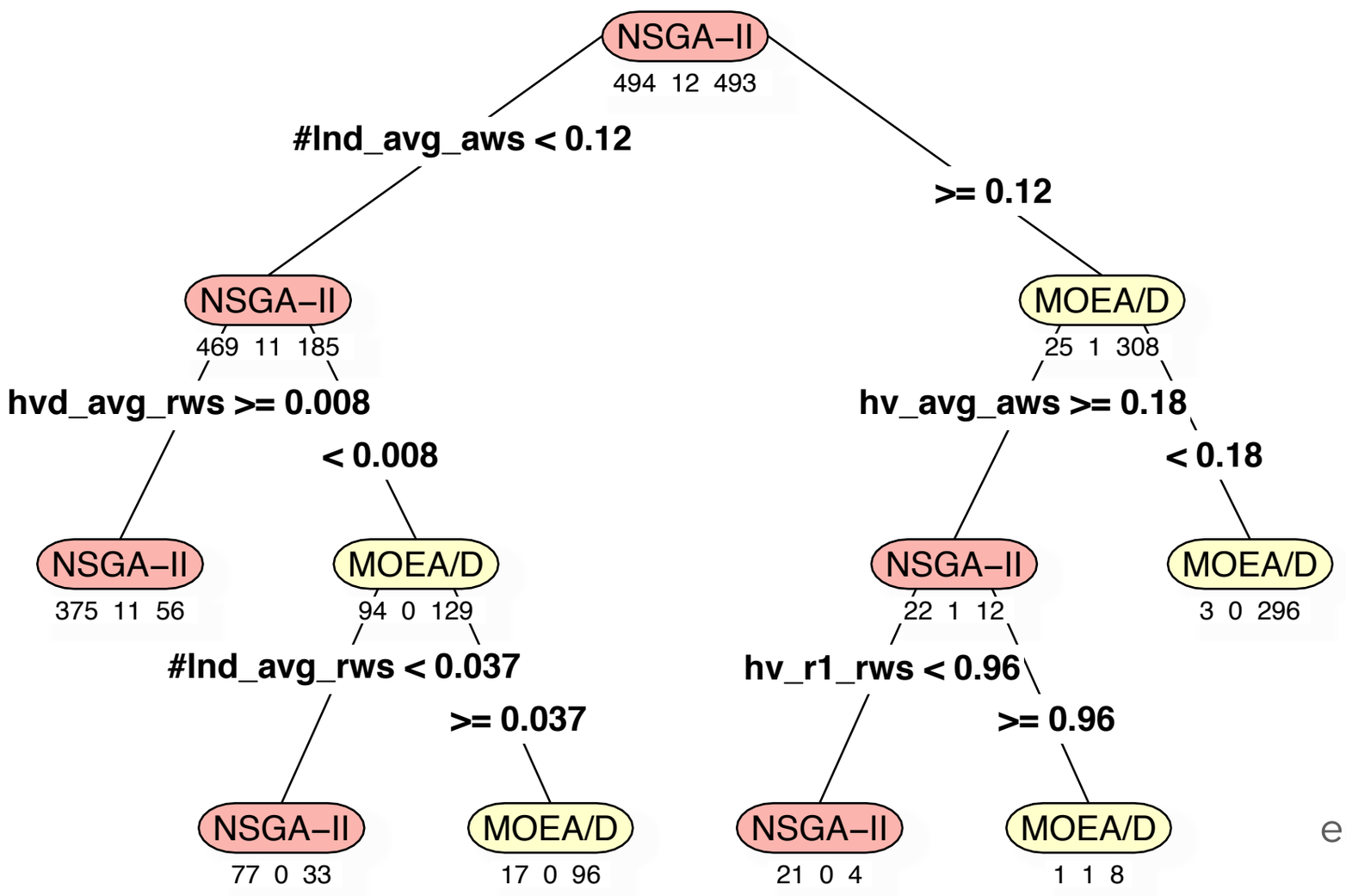
set of features	error rate of best average performance mean	std	rank	error rate of best statistical rank mean	std	rank
all features	**0.122222**	0.031033	1	**0.012727**	0.014110	1
local features	**0.123030**	0.030521	1	**0.013737**	0.014103	1
local features (random walk)	**0.118788**	0.029187	1	**0.013333**	0.012149	1
local features (adaptive walk)	**0.130303**	0.029308	1	**0.015354**	0.014026	1
$\{\rho, \mathtt{m}, \mathtt{n}, \mathtt{k_n}\}$	**0.125859**	0.028875	1	**0.014141**	0.013382	1
$\{\mathtt{m}, \mathtt{n}\}$	0.413333	0.045533	6	0.197374	0.043778	6

random subsampling cross-validation
(50 iter., 90/10 split)

large landscapes

Algorithm Selection

Model (classif, decision tree) = {algo} ~ (n, k_n, m, ρ, {features})



error = 12.61%
(best in avg)

Does it Generalize to Other Problems?

Multi-objective QAP

[Knowles et al. 2002]

$$\min_{x \in X} f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{x_i x_j} e_{ij}^{k} \qquad k \in \{1, \ldots, m\}$$

- number of variables n
- number of objectives m
- objective correlation ρ
- instance type type

unknown

Experimental Setup

1000 instances generated by a design of experiments

- number of variables $n \in [\![30 , 100]\!]$
- number of objectives $m \in [\![2 , 5]\!]$
- objective correlation $\rho \in [-1 , 1]$
- instance type $type \in \{ \text{uniform} , \text{real-like} \}$

Experimental Setup

Algorithms

- **NSGA-II** vs. **IBEA** vs. **MOEA/D** (default setting, population size = 100)

Performance

- 20 independent runs per instance, fixed budget of 1 000 000 evaluations
- (Expected) hypervolume relative deviation (hvrd)

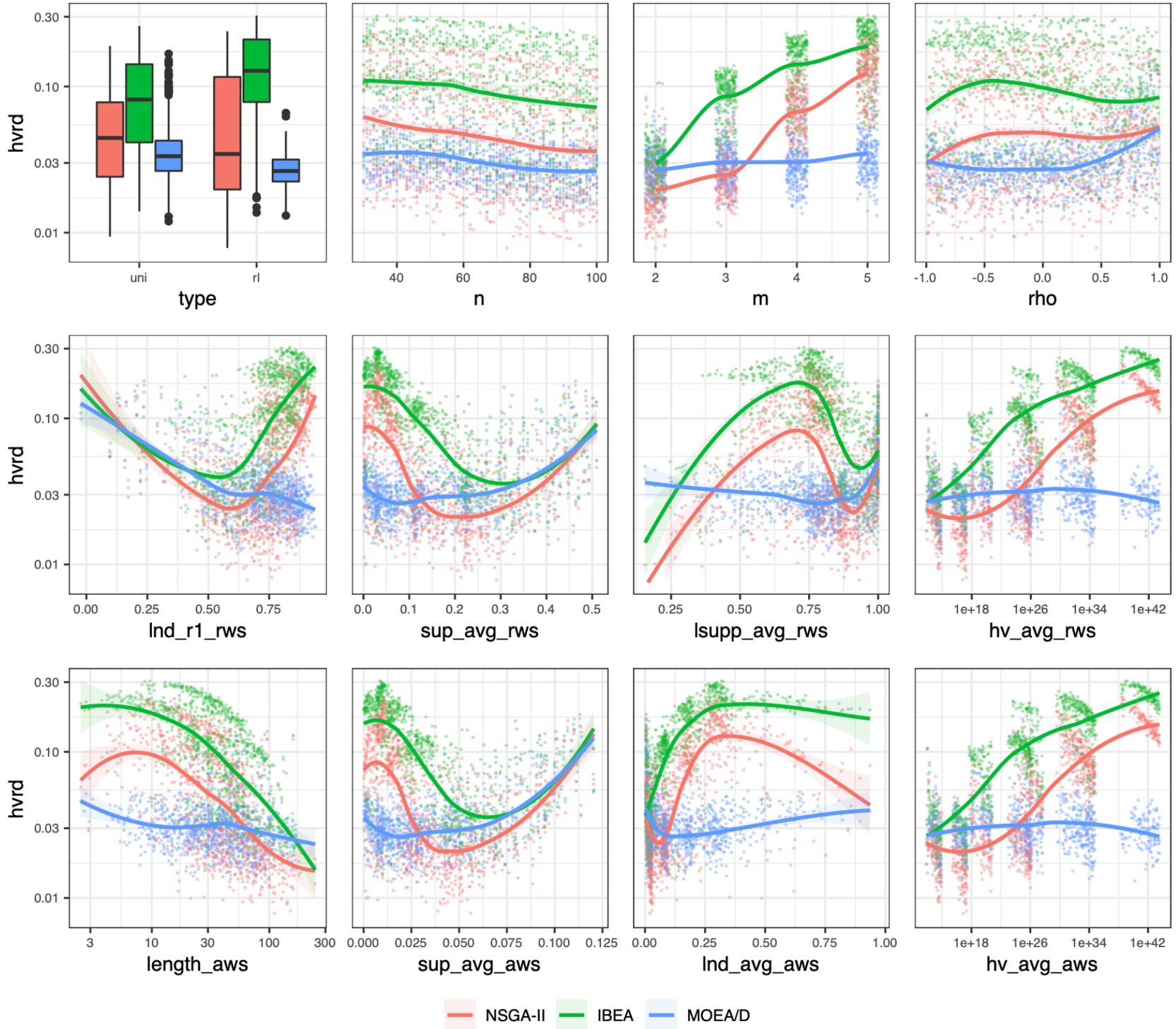
Statistics

- Correlation = Spearman (non-parametric)
- Prediction = random forest, decision tree (default setting)

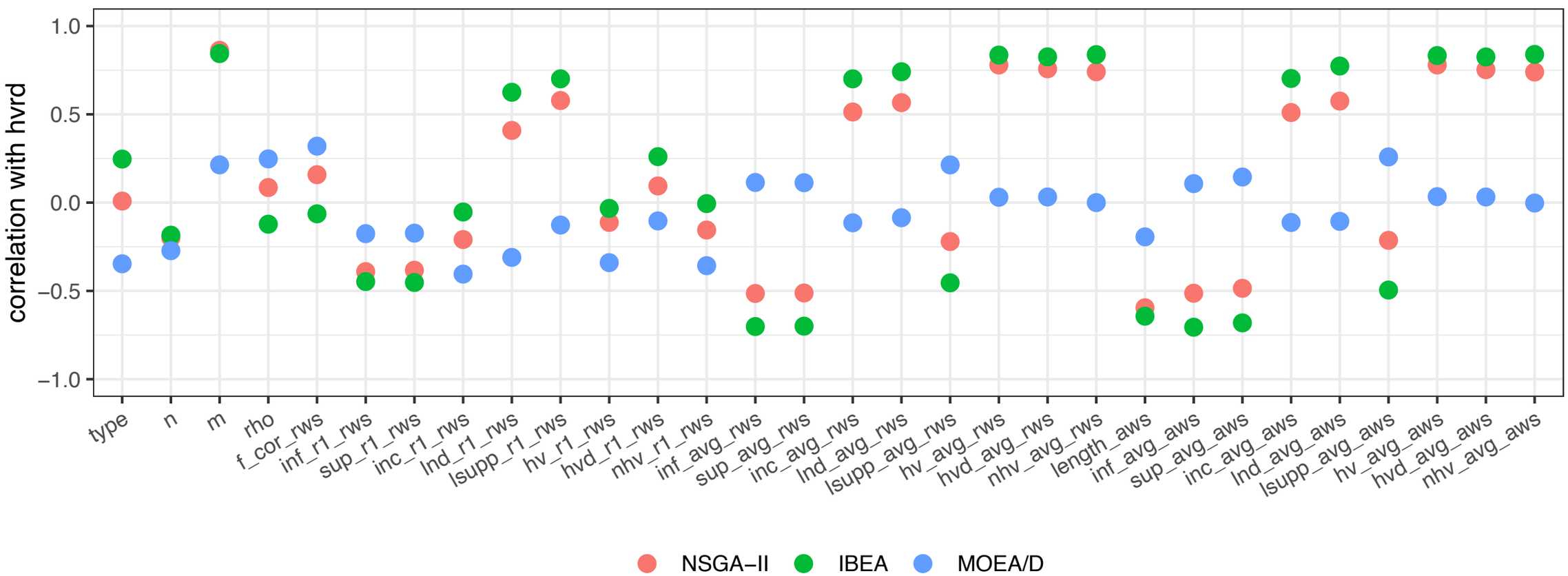
Correlation among Features

	length_aws	nhv_avg_aws	hvd_avg_aws	hv_avg_aws	lsupp_avg_aws	lnd_avg_aws	inc_avg_aws	sup_avg_aws	inf_avg_aws	nhv_avg_rws	hvd_avg_rws	hv_avg_rws	lsupp_avg_rws	lnd_avg_rws	inc_avg_rws	sup_avg_rws	inf_avg_rws	nhv_r1_rws	hvd_r1_rws	hv_r1_rws	lsupp_r1_rws	lnd_r1_rws	inc_r1_rws	sup_r1_rws	inf_r1_rws	f_cor_rws	rho	m	n	type
nhv_avg_aws		–0.3																												
hvd_avg_aws			1	–0.3																										
hv_avg_aws				1	1	–0.3																								
lsupp_avg_aws					–0.3	–0.3	–0.3	0.5																						
lnd_avg_aws						–0.8	0.6	0.7	0.6	–0.7																				
inc_avg_aws							1	–0.9	0.5	0.5	0.5	–0.8																		
sup_avg_aws								–1	–1	0.9	–0.6	–0.6	–0.5	0.7																
inf_avg_aws									1	–1	–1	0.9	–0.5	–0.5	–0.5	0.8														
nhv_avg_rws										–0.5	–0.5	0.5	0.6	–0.3	1	1	1	–0.3												
hvd_avg_rws											1	–0.5	–0.6	0.5	0.6	–0.3	1	1	1	–0.3										
hv_avg_rws												1	1	–0.5	–0.6	0.5	0.7	–0.3	1	1	1	–0.3								
lsupp_avg_rws													–0.2	–0.2	–0.2	0.9	0.9	–0.9	–0.8	0.9	–0.2	–0.2	–0.2	0.6						
lnd_avg_rws														–0.9	0.6	0.6	0.6	–1	–1	1	1	–0.8	0.6	0.6	0.6	–0.8				
inc_avg_rws															1	–0.9	0.6	0.5	0.5	–1	–1	1	1	–0.9	0.5	0.6	0.5	–0.7		
sup_avg_rws																–1	–1	0.9	–0.6	–0.5	–0.5	1	1	–1	–1	0.9	–0.5	–0.5	–0.5	0.7
inf_avg_rws																	1	–1	–1	0.9	–0.6	–0.5	–0.5	1	1	–1	–1	0.9	–0.5	–0.6	–0.5	0.7
nhv_r1_rws																		0.1	0.1	–0.1	–0.1	0.1	0.3	0.3	0.4	0.1	0	–0.1	0	–0.1	0.3	0.3	0.4	0.6
hvd_r1_rws																			0.6	–0.2	–0.2	0.2	0.2	–0.1	0.5	0.5	0.5	–0.2	–0.3	0.2	0.3	–0.2	0.5	0.5	0.5	0.2
hv_r1_rws																			0.6	0.8	0.1	0.1	–0.1	0	0.1	0.3	0.3	0.3	0.1	0	–0.1	0	0	0.3	0.3	0.3	0.6
lsupp_r1_rws																			0.3	0.4	0.3	–0.4	–0.4	0.4	0.4	–0.2	0.7	0.7	0.8	–0.4	–0.4	0.4	0.5	–0.3	0.7	0.7	0.8	–0.2
lnd_r1_rws																			0.7	0.5	0.6	0.6	–0.5	–0.5	0.5	0.6	–0.3	0.8	0.8	0.8	–0.5	–0.6	0.5	0.6	–0.4	0.8	0.8	0.8	–0.1
inc_r1_rws																			0.6	0.3	0.6	0.4	0.7	0.2	0.2	–0.2	–0.2	0.3	0.2	0.2	0.3	0.2	0.2	–0.2	–0.1	0.1	0.2	0.2	0.3	0.6
sup_r1_rws																			0.7	0.1	0	0.6	0.3	0.6	0.7	0.7	–0.7	–0.7	0.7	–0.1	–0.1	–0.1	0.7	0.6	–0.7	–0.6	0.6	–0.1	–0.1	–0.1	0.9
inf_r1_rws																			0.9	0.7	0.1	0	0.6	0.3	0.5	0.7	0.7	–0.7	–0.7	0.7	–0.1	–0.1	–0.1	0.7	0.6	–0.7	–0.6	0.6	–0.1	–0.1	–0.1	0.9
f_cor_rws																			0.4	0.4	0	–0.2	0	0.2	0.1	0	0.5	0.5	–0.5	–0.5	0.7	0.1	0.1	0.1	0.5	0.5	–0.5	–0.4	0.7	0.1	0.1	0.1	0.3
rho																			0.7	0.5	0.5	0.1	–0.1	0	0.1	0	0.1	0.6	0.6	–0.6	–0.5	0.8	0.1	0.1	0.1	0.6	0.6	–0.6	–0.5	0.8	0.1	0.1	0.1	0.4
m																			0.1	0.2	–0.3	–0.3	–0.1	0.5	0.6	0.1	0.3	0	–0.6	–0.6	0.6	0.7	–0.3	0.9	0.9	0.9	–0.6	–0.6	0.6	0.7	–0.3	0.9	0.9	0.9	–0.6
n																			0	0	0	0.5	0.5	0.5	0.4	0.1	0.8	0.5	0.7	0	0	0	0	0	0.2	0.2	0.2	0.1	–0.1	–0.1	0	0	0.2	0.2	0.2	0.6
type																			0	0	0	–0.1	0.3	0.3	0.7	0.6	0.5	0.4	0.4	0.6	0	0	0	0	0.1	0.3	0.3	0.4	0	0	0	0.1	–0.1	0.3	0.3	0.4	0.3

Features vs. Performance

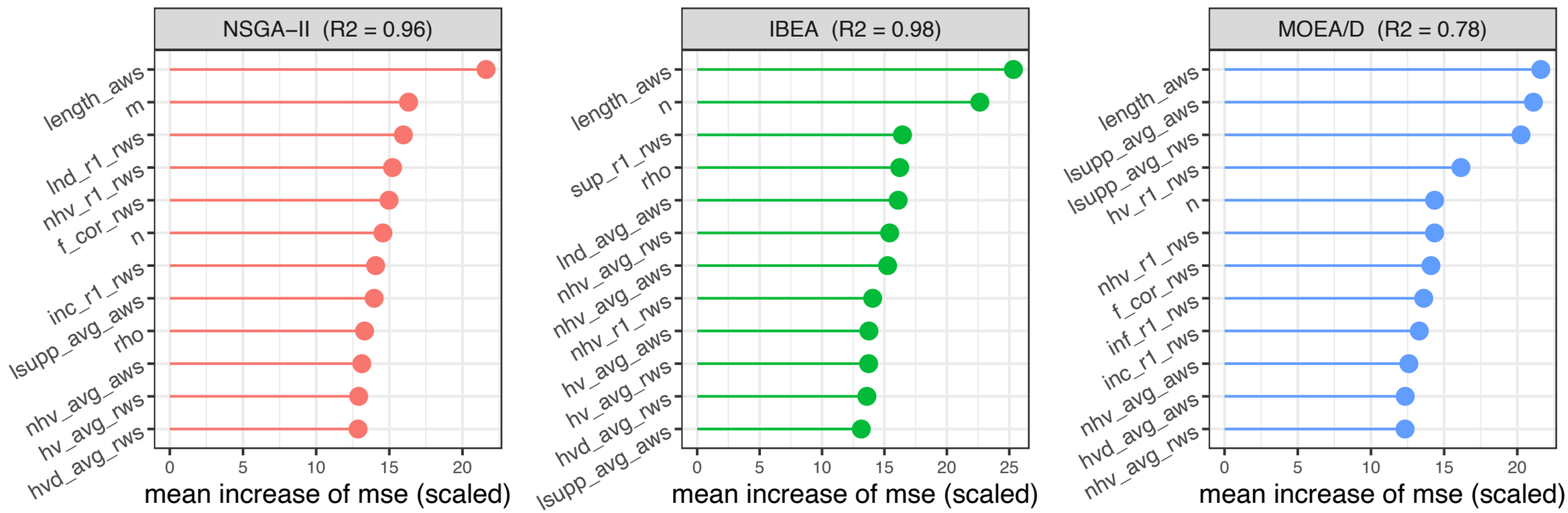


Features vs. Performance



Importance of Features

Model (regression, RF) = hvrd ~ (n, m, ρ, type, {features})



Algorithm Selection

Algorithm portfolio = {**NSGA-II**, **IBEA**, **MOEA/D**}

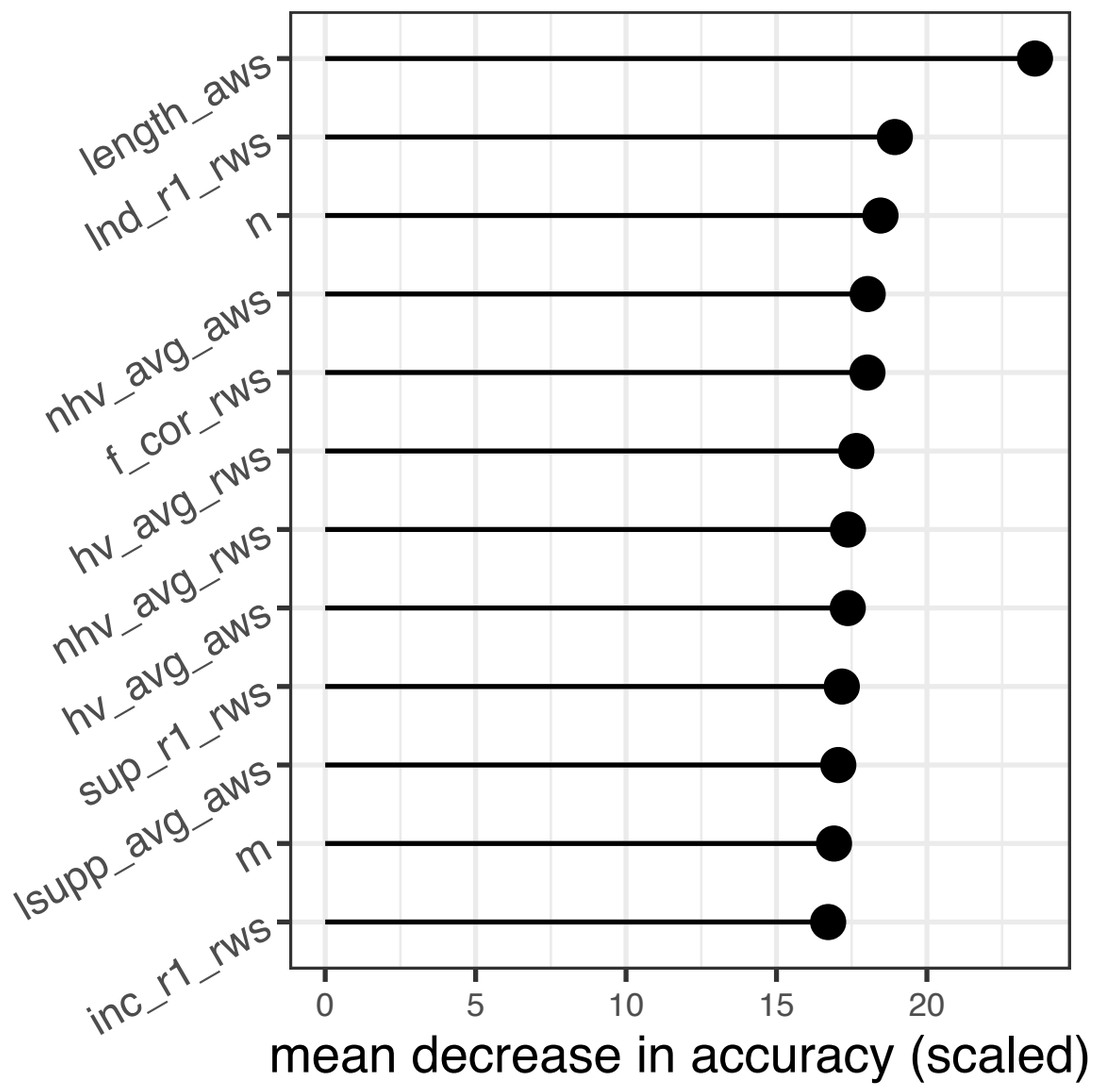
Model (classif, RF) = {algo} ~ (n, m, ρ, type, {features})

subset of features	classification error	error predicting statistical best
{n, m}	.1962	.0332
{type, n, m, ρ}	.1197	.0072
{⋆_rws, n, m}	.1114	**.0062**
{⋆_aws, n, m}	.1125	.0065
{⋆_rws, length_aws, n, m}	**.1089**	**.0056**
{⋆_rws, ⋆_aws, n, m}	**.1077**	**.0063**
{⋆_rws, ⋆_aws, type, n, m, ρ}	**.1078**	**.0063**
random classifier	.6667	.3810
dummy classifier (MOEA/D)	.4200	.1040

random subsampling cross-validation
(100 repetitions, 80/20% split)

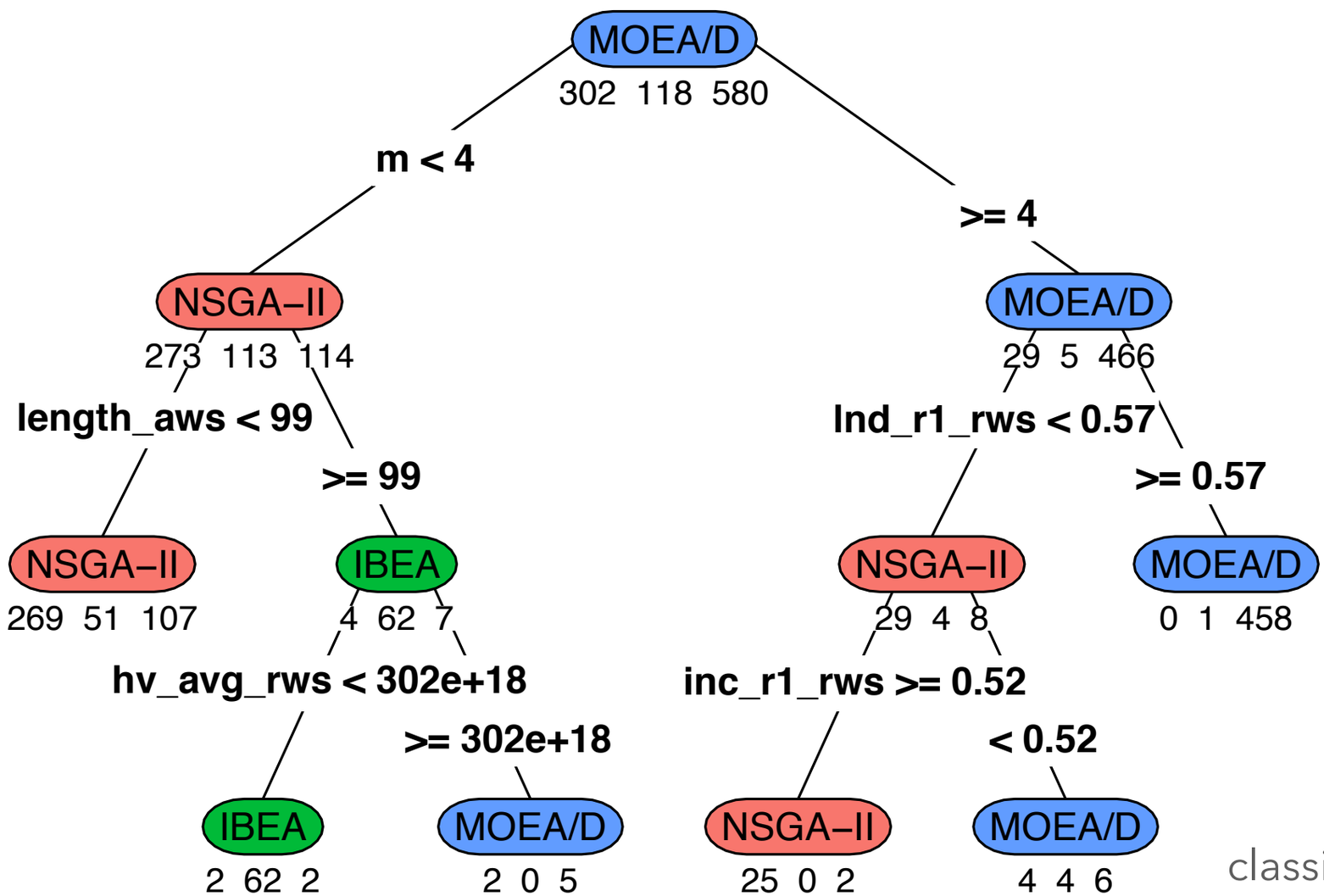
Importance of Features

Model (classif, RF) = {algo} ~ (n, m, ρ, type, {features})



Importance of Features

Model (classif, decision tree) = {algo} ~ (n, m, ρ, type, {features})



classif error = 17.5%

Algorithm Selection

Algorithm portfolio = {**NSGA-II**, **IBEA**, **MOEA/D**}

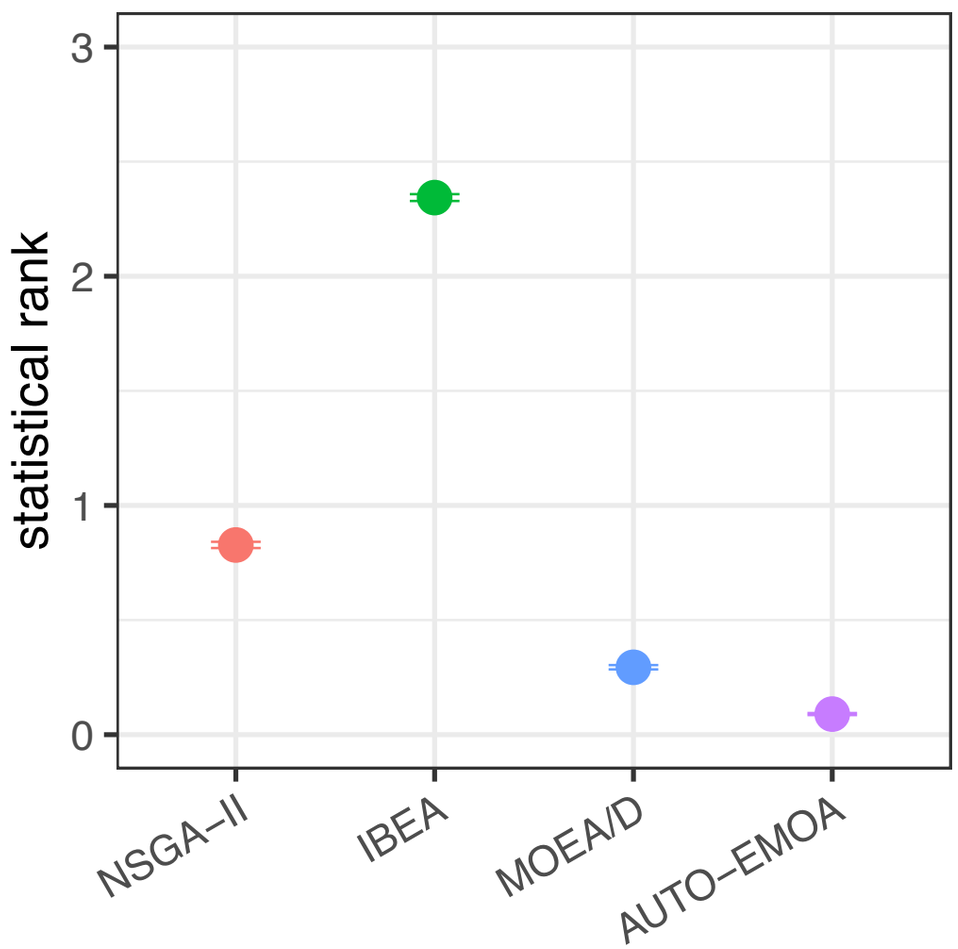
Model (classif, RF) = {algo} ~ (n, m, ρ, type, {features})

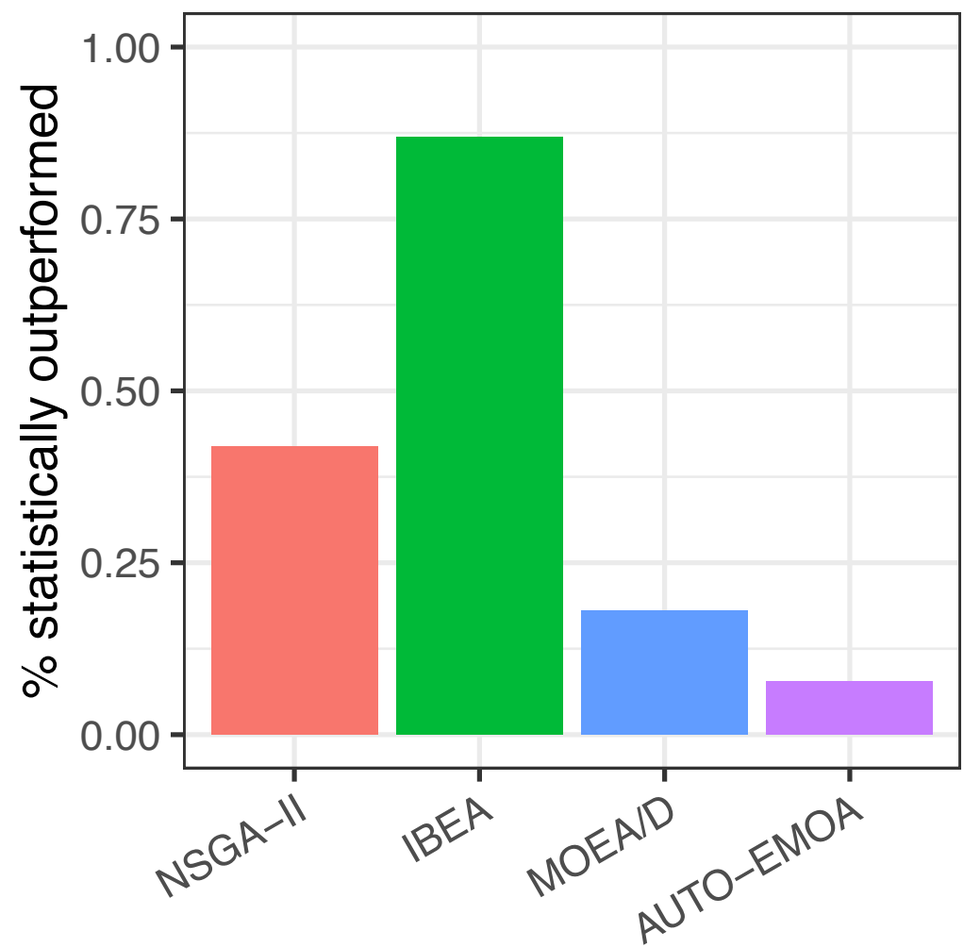
subset of features	classification error	error predicting statistical best
{n, m}	.1962	.0332
{type, n, m, ρ}	.1197	.0072
{⋆_rws, n, m}	.1114	**.0062**
{⋆_aws, n, m}	.1125	.0065
{⋆_rws, length_aws, n, m}	**.1089**	**.0056**
{⋆_rws, ⋆_aws, n, m}	**.1077**	**.0063**
{⋆_rws, ⋆_aws, type, n, m, ρ}	**.1078**	**.0063**
random classifier	.6667	.3810
dummy classifier (MOEA/D)	.4200	.1040

random subsampling cross-validation
(100 repetitions, 80/20% split)

Automated Selection

- Low-cost features extracted from search budget (~ 5%)





Multi-objective Landscapes

- Design and quantify landscape features that capture problem difficulty and search performance

- Understand why a problem is hard, why an algorithm is good

- Scalability (#var, #obj) partly explain complexity, importance of ruggedness and multimodality

References

- H. Aguirre, K. Tanaka, "**Working principles, behavior, and performance of MOEAs on MNK-landscapes**," European Journal of Operational Research, vol. 181, no. 3, pp. 1670–1690, 2007.
- F. Daolio, A. Liefooghe, S. Verel, H. Aguirre, K. Tanaka, "**Problem features versus algorithm performance on rugged multiobjective combinatorial fitness landscapes**," Evolutionary Computation, vol. 25, no. 4, pp. 555–585, 2017.
- D. Garrett, D. Dasgupta, "**Multiobjective landscape analysis and the generalized assignment problem**," in LION 2, LNCS vol. 5313, pp. 110–124, Trento, Italy, 2007.
- D. Garrett and D. Dasgupta, "**Plateau connection structure and multiobjective metaheuristic performance**," in CEC 2009, pp. 1281–1288, , 2009.
- S. Kauffman, **The Origins of Order**. Oxford University Press, 1993.
- J. Knowles and D. Corne, "**Instance generators and test suites for the multiobjective quadratic assignment problem**," in EMO 2003, LNCS vol. 2632, pp. 295–310, Faro, Portugal, 2003.
- A. Liefooghe, S. Verel, B. Derbel, H. Aguirre, and K. Tanaka, "**Dominance, indicator and decomposition based search for multi-objective QAP: landscape analysis and automated algorithm selection**," International Conference on Parallel Problem Solving from Nature (PPSN 2020), Lecture Notes in Computer Science (LNCS), Leiden, The Netherlands, 2020.
- A. Liefooghe, F. Daolio, S. Verel, B. Derbel, H. Aguirre, and K. Tanaka, "**Landscape-aware performance prediction for evolutionary multi-objective optimization**," IEEE Transactions on Evolutionary Computation, early access.
- L. Paquete, T. Schiavinotto, and T. Stützle, "**On local optima in multiobjective combinatorial optimization problems**," Annals of Operations Research, vol. 156, no. 1, pp. 83–97, 2007.
- L. Paquete and T. Stützle, "**Clusters of non-dominated solutions in multiobjective combinatorial optimization: An experimental analysis**," in Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications, LNEMS vol. 618, pp. 69–77, 2009.
- S. Verel, A. Liefooghe, L. Jourdan, and C. Dhaenens, "**On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives**," European Journal of Operational Research, vol. 227, no. 2, pp. 331–342, 2013.

(almost) Final Slide: Open Issues

- Landscape-aware automated algorithm **selection** and **configuration**
- Landscape-aware **parameter control**
- Fitness landscapes for **stochastic operators**
- **Population** and **set-based** fitness landscapes, **parallel**, **crossover**…
- Multi-objective landscapes **visualization**
- **Continuous** fitness landscapes
- Links with other **theoretical** approaches

PPSN 2020
Session 4 (Thu)