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

Arnaud Liefooghe
Univ. Lille – CRIStAL / Inria
Lille, France

arnaud.liefooghe@univ-lille1.fr

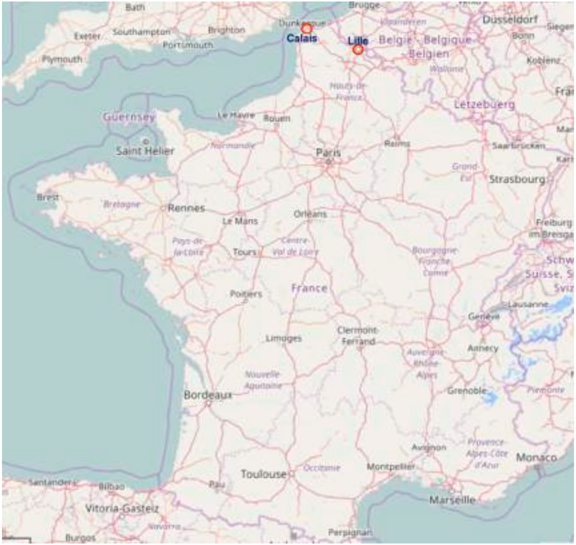
Sébastien Verel
Univ. Littoral Côte d'Opale – LISIC
Calais, France

verel@univ-littoral.fr

IEEE Congress on Evolutionary Computation
Donostia - San Sebastián, Spain

June 5, 2017

Where are we?



Bath
Exeter
Southampton
Brighton
Portsmouth
Plymouth
Brugge
Calais
Lille
Belgie - Belgique - Belgien
Düsseldorf
Bonn
Koblenz
Letzebuerg
Guernsey
Saint Helier
Le Havre
Rouen
Paris
Reims
Saarbrücken
Strasbourg
Brest
Rennes
Le Mans
Orléans
Tours
Poitiers
France
Limoges
Clermont-Ferrand
Genève
Lausanne
Annecy
Grenoble
Bordeaux
Toulouse
Montpellier
Avignon
Marseille
Monaco
Santander
Bilbao
Vitoria-Gasteiz
Burgos
Pau
Perpignan

Where are we?

LISIC
ulco
Calais
Lille
Université de Lille 1
CRIStAL
Inria
Université Lille Nord de France
Paris
Rennes
Brest
Bordeaux
Toulouse
Marseille
Strasbourg
Clermont-Ferrand
Limoges
Poitiers
Tours
Orléans
Le Mans
Grenoble
Annecy
Genève
Montpellier
Avignon
Monaco
France

Program for today

1. The Basics of Fitness Landscapes
   - Introductory example
   - Brief history and background
2. Geometries of Fitness Landscapes
   - Ruggedness and multimodality
   - Neutrality
3. Local Optima Network
   - Features from the network, algorithm design and performance
   - Performance prediction and algorithm portfolio
4. Multi-objective Fitness Landscapes
   - Brief overview of (evolutionary) multi-objective optimization
   - Features to characterize multi-objective fitness landscapes
   - Performance prediction and algorithm selection

Introductory example

Please visit the "game" at:

http://www-lisic.univ-littoral.fr/~verel/RESEARCH/fitness-landscape-game/index.html

or:

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

1. The Basics of Fitness Landscapes

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

Arnaud Liefooghe
Univ. Lille – CRIStAL / Inria
Lille, France
arnaud.liefooghe@univ-lille1.fr

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

IEEE Congress on Evolutionary Computation
Donostia - San Sebastián, Spain

June 5, 2017

Outline

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

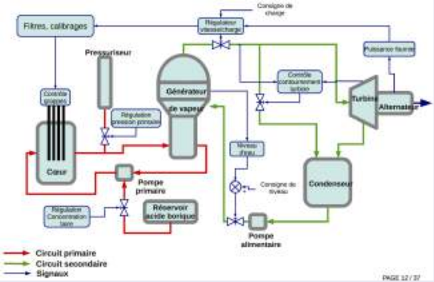
Real-world black-box optimization : an example

PhD of Mathieu Muniglia, Saclay Nuclear Research Centre (CEA), Paris

$x \longrightarrow$

$\longrightarrow f(x)$

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



Multi-physic simulator

$\longrightarrow \Delta_z P$

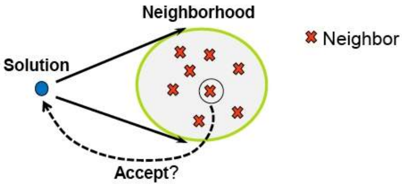
Search algorithms

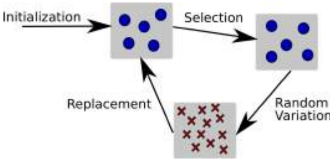
Principle

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

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

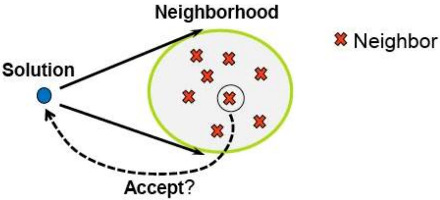
Local search

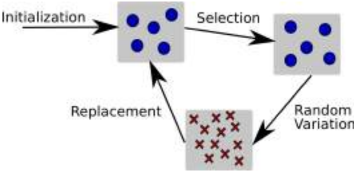




Metaheuristics

Local search methods using a neighborhood relation

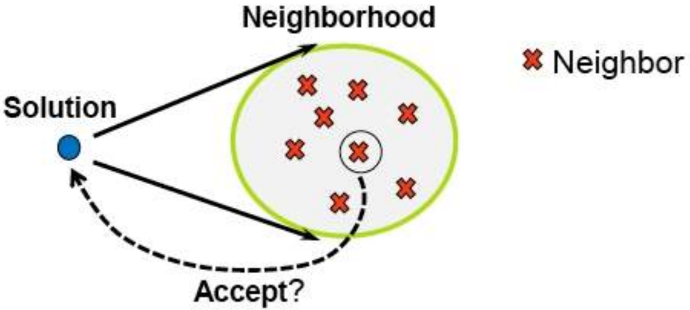




- **Single solution-based metaheuristics** : Hill climber, Simulated annealing, Tabu search, Iterated local search ...
- **Population-based metaheuristics** : Genetic algorithm, Genetic programming, Ant colony optimization ...

Stochastic algorithms with a single solution (Local Search)

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

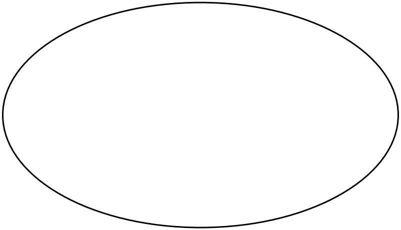


Main idea behind local search algorithms

Why using a local search strategy based on a neighborhood ?

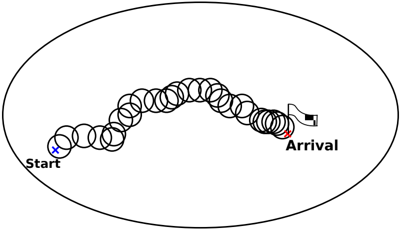
Main idea behind local search algorithms

Why using a local search strategy based on a neighborhood ?



Main idea behind local search algorithms

Why using a local search strategy based on a neighborhood ?

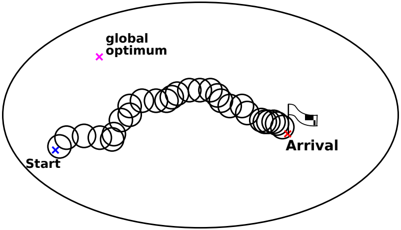


Split the global problem
into a sequence of (smaller/easier) local problems

- **Benefit** : reduce the complexity

Main idea behind local search algorithms

Why using a local search strategy based on a neighborhood ?



Split the global problem
into a sequence of (smaller/easier) local problems

- **Benefit** : reduce the complexity
- **Risk** : no guarantee to find the optimal solution

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

"the more we know of the **statistical properties** *of a class of fitness landscapes, the better equipped we will be for the* **design** *of effective search algorithms for such landscapes"*

L. Barnett, U. Sussex, PhD 2003.

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

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

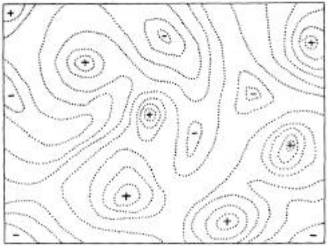


FIGURE 2.—Diagrammatic representation of the field of gene combinations in two dimensions instead of many thousands. Dotted lines represent contours with respect to adaptiveness.

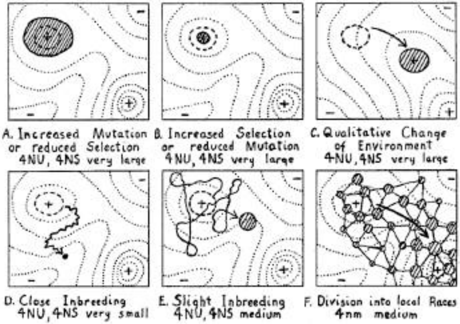
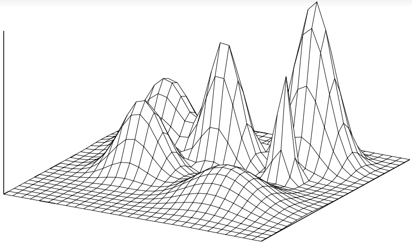


FIGURE 4.—Field of gene combinations occupied by a population within the general field of possible combinations. Type of history under specified conditions indicated by relation to initial field (heavy broken contour) and arrow.

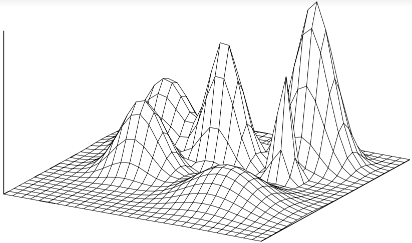
source : Encyclopaedia Britannica Online

Fitness landscapes in (evolutionary) biology



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

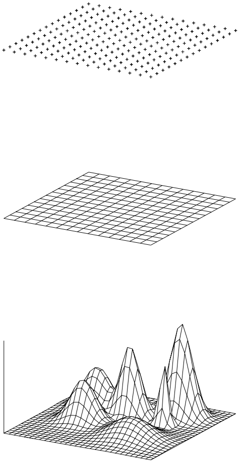
Fitness landscapes in (evolutionary) biology



- Metaphorical uphill struggle across a “fitness landscape”
  - mountain **peaks** represent high “fitness” (ability to survive)
  - **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

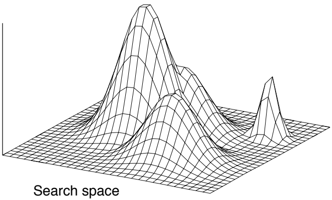
2 sides of Fitness Landscapes

- **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

Definition of fitness landscape for optimization [Sta02]



Definition

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

- **search space** :

$$X$$

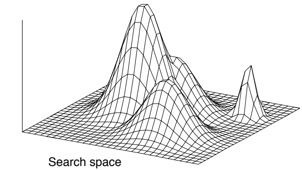
- **neighborhood relation** :

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

- **objective function** :

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

What is a neighborhood ?



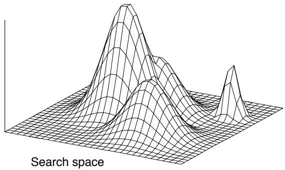
Neighborhood function :

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

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

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

What is a neighborhood ?



Neighborhood function :

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

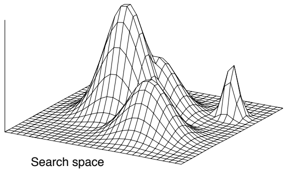
Set of "**neighbor**" solutions associated to each solution

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

or

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

What is a neighborhood ?



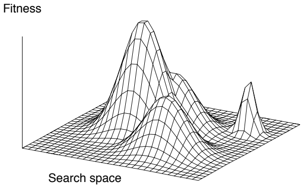
Neighborhood function :

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

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

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

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

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

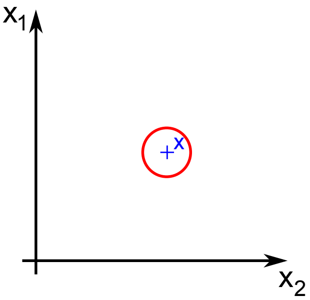
Traveling Salesman Problem :
find the shortest tour which cross one time every town

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

$$\mathcal{N}(x) = \{y \in X \mid \Pr(y = op_{2opt}(x)) > 0\}$$

Not so typical example : continuous optimization

Still an open question...



Search space : $X = [0,1]^d$

$$\mathcal{N}_\alpha(x) = \{y \in X \mid \|y - x\| \leqslant \alpha\} \text{ with } \alpha > 0$$

More than 1 operator. . . ?

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

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

More than 1 operator. . . ?

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

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

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

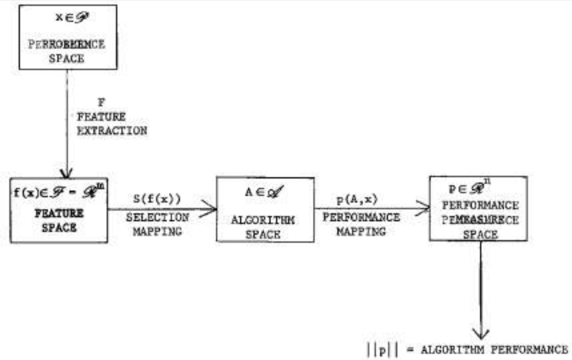
$$\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2 = \{y \in X \mid y = op_1(x) \text{ or } y = op_2(x)\}$$

- 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.

Positioning of fitness landscape analysis

Selection of local search algorithm

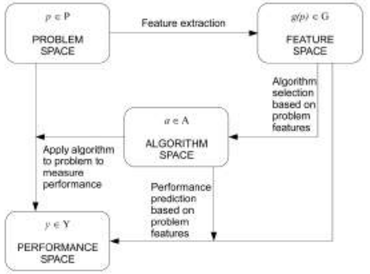


Figure 1.1: A framework for describing the general problems of algorithm selection and performance prediction based on problem features (based Rice's model [132]).

Malan, K. M., Engelbrecht, A. P. (2014). Fitness landscape analysis for metaheuristic performance prediction. In Recent advances in the theory and application of fitness landscapes (pp. 103-132). [ME14]

Positioning of fitness landscape analysis

Selection of local search algorithm

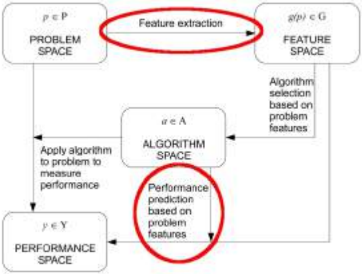


Figure 1.1: A framework for describing the general problems of algorithm selection and performance prediction based on problem features (based Rice's model [132]).

Malan, K. M., Engelbrecht, A. P. (2014). Fitness landscape analysis for metaheuristic performance prediction. In Recent advances in the theory and application of fitness landscapes (pp. 103-132). [ME14]

Fitness landscape analysis : features extraction *vs.* performance

Fitness landscape analysis

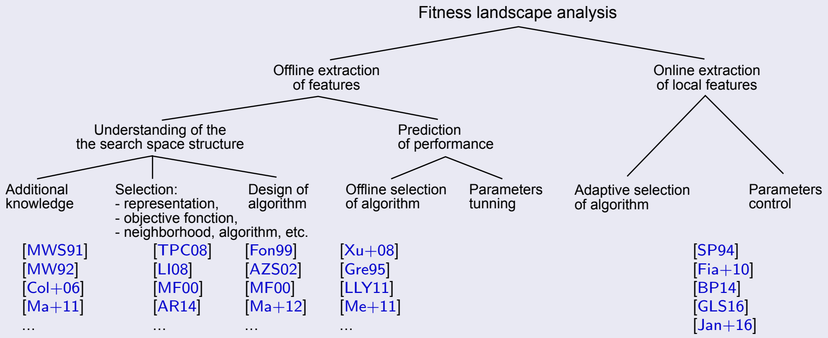
Algebraic approach, grey-box :

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

Statistical approach, black-box :

Problems ⇝ Features ⇝ Algorithm ⇝ Performances

Goals



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 ?

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 ?

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 ? . . .

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

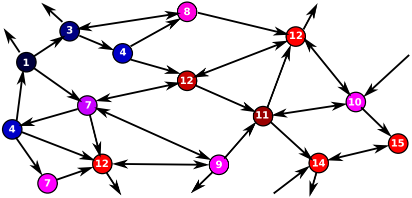
   Which algorithm shall I use ?
3. **Tuning the algorithm's parameters** :
   - *off-line* analysis of the fitness landscape structure

   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

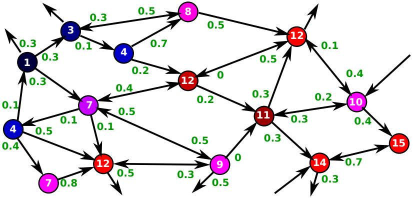


Remarks :

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

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 :

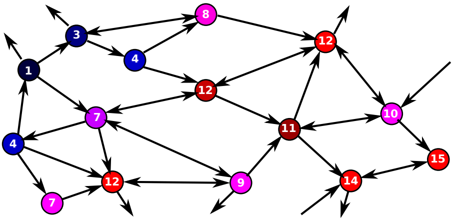
- Model of the search space
- Not specific to a particular local search
- A specific local search puts probability transitions on **edges**, according to f-values and the search *history*

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 . . .

⇒ 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

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

Next section : two main geometries

- multimodality and ruggedness
- neutrality

References I

KatherineM. Malan and AndriesP. Engelbrecht.
Fitness landscape analysis for metaheuristic performance prediction.
In Hendrik Richter and Andries Engelbrecht, editors, *Recent Advances in the Theory and Application of Fitness Landscapes*, volume 6 of *Emergence, Complexity and Computation*, pages 103–132. Springer Berlin Heidelberg, 2014.

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

References II

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.

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 and Algorithm Performance for Single- and Multi-objective Combinatorial Optimization

Arnaud Liefooghe
Univ. Lille – CRIStAL / Inria
Lille, France
arnaud.liefooghe@univ-lille1.fr

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

IEEE Congress on Evolutionary Computation
Donostia - San Sebastián, Spain

June 5, 2017

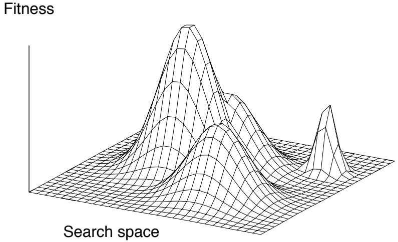
Outline

Multimodal fitness landscapes

Local optima $s^\star$

no neighboring solution with strictly better fitness value (maximization)

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



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

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

Sampling local optima by adaptive walks

Adaptive walk

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

Sampling local optima by adaptive walks

Adaptive walk

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

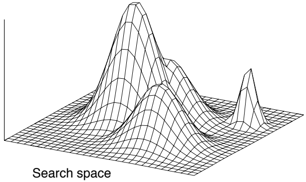
Hill-Climbing algorithm (first-improvement)

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 $x^\star$

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

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"

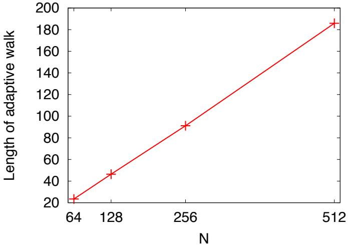
Optimization difficulty :
Number and size of the basins of attraction (Garnier *et al.* [GK02])

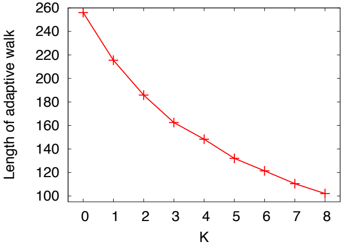
Feature to estimate the basins size :

- **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**

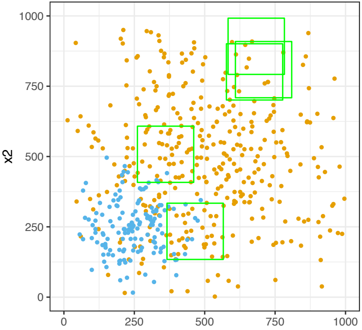
complexity : sample size $\times \ell \times |\mathcal{N}|$

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

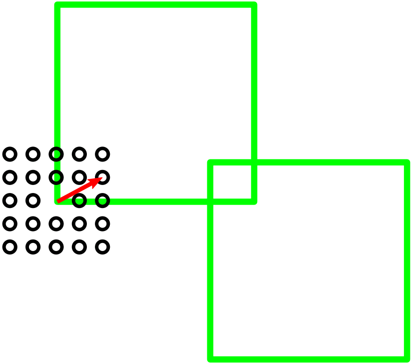
Different functions are already defined :

- `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

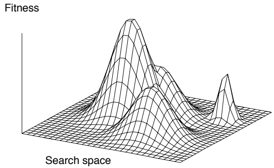
Pre-defined functions :

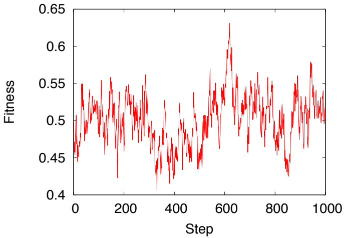
- `adaptive_length` :
  run the hill-climber and compute a data frame with the length of adaptive walks
- `main_adaptive_length_analysis` :
  Compute the adaptive length of two different SP problems

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 measure ruggedness





Random walk :

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

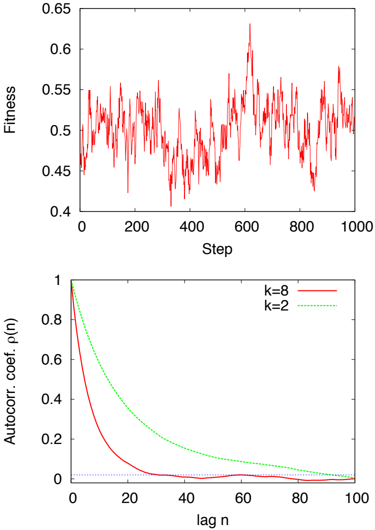
The idea :

- 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 $\approx 10^3$

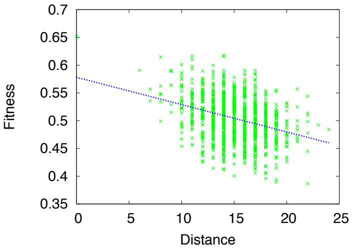
Results on rugged fitness landscapes (Stadler 96 [Sta96])

Ruggedness decreases with the size of those problems

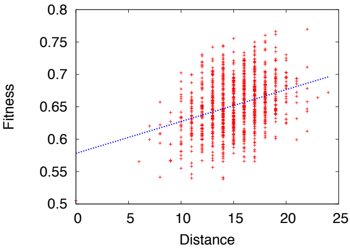
Problem	parameter	$\rho(1)$
symmetric TSP	n number of towns	$1 - \frac{4}{n}$
anti-symmetric TSP	n number of towns	$1 - \frac{4}{n-1}$
Graph Coloring Problem	n number of nodes α number of colors	$1 - \frac{2\alpha}{(\alpha-1)n}$
NK landscapes	N number of proteins K number of epistasis links	$1 - \frac{K+1}{N}$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

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

Correlation between fitness and distance to global optimum



"easy"



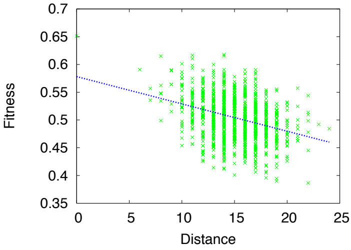
"hard"

Classification based on experimental studies

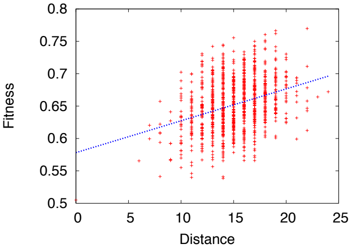
- $\rho < -0.15$: easy optimization
- $\rho > 0.15$: hard optimization
- $-0.15 < \rho < 0.15$: undecided zone

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

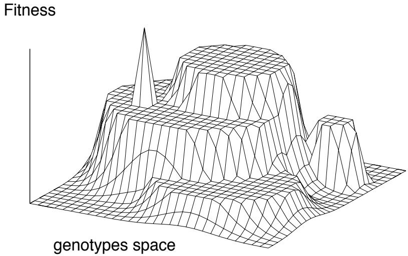
- Define the function `random_walk` to compute the fitness values during a random walk
- Execute line by line the `main` function to compute a sample of fitness value collected during a random walk
- Compare the first autocorrelation coefficient of the SP problems 1 and 2

Neutral fitness landscapes

Neutral theory (Kimura ≈ 1960 [Kim83])

Theory of mutation and random drift

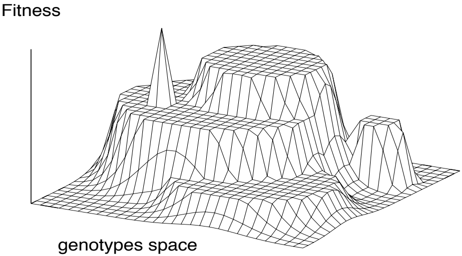
Many mutations have no effects on fitness-values



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

Neutral fitness landscapes

- Redundant problem (symmetries ...) [GS87]
- Problem "not well" defined, or dynamic environment [IT04]
- Unused variables, discrete values ...



Real-world problems :

- Robot controller
- Circuit design
- Genetic Programming
- Protein folding
- Learning problems
- Scheduling problems
- Graph problems...

Neutrality and difficulty

- In our knowledge, there is no definitive answer about the relation between neutrality and problem hardness
- Certainly, it is dependent on the "nature" of neutrality

Solving optimization problem and neutrality

3 ways to deals with neutrality :

- Decrease the neutrality : reduce the entropy barrier
- Increase the neutrality : reduce the fitness barrier
- Unchange the neutrality : use a specific algorithm

Sharp description of the geometry of neutral fitness landscapes is required

Neutrality and difficulty

We know for certain that :

- **No information** is better than **Bad information** :
  From a non-optimal solution, hard trap functions are more difficult than needle-in-a-haystack functions
- **Good information** is better than **No information** :
  Onemax problem is much easier than needle-in-a-haystack functions

Neutrality and difficulty

We know for certain that :

- **No information** is better than **Bad information** :
  From a non-optimal solution, hard trap functions are more difficult than needle-in-a-haystack functions
- **Good information** is better than **No information** :
  Onemax problem is much easier than needle-in-a-haystack functions

- When there is **No information** :
  you should have a good method to **create** it !

Objects of neutral fitness landscapes

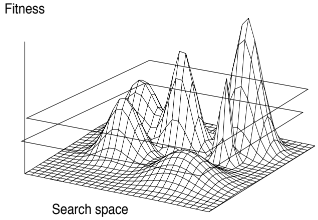
Description of multimodal fitness landscapes is based on :

- Local optima
- Basins of attraction

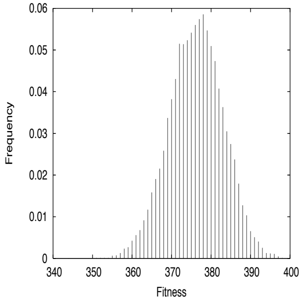
Description of neutral fitness landscapes is based on :

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

Neutral sets : density of states



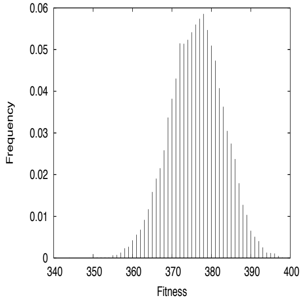
Set of solutions with same fitness



Density of states (D.O.S.)

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

Neutral sets : density of states



Informations given :

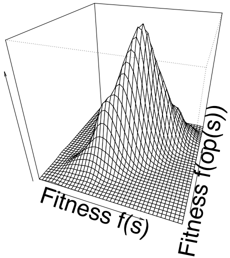
- Performance of random search
- Tail of the distribution is an indicator of difficulty :
  - the faster the decay, the harder the problem
- But do not care about the neighborhood relation

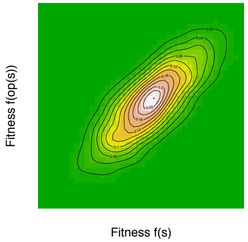
Features :

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

complexity : sample size

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





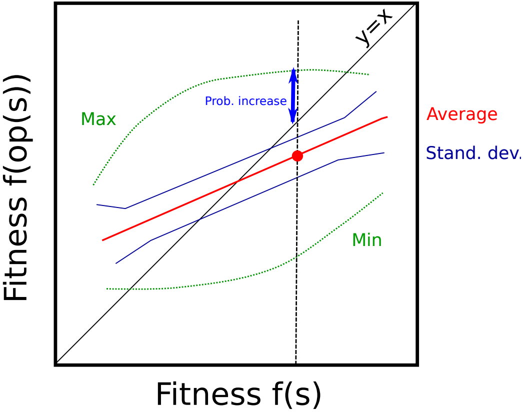
- $(X, \mathcal{F}, \text{Pr})$: probability space
- $op : X \rightarrow X$ stochastic operator of the local search

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

Fitness Cloud of *op*

Conditional probability density function of Y given X

Fitness cloud : a measure of evolvability



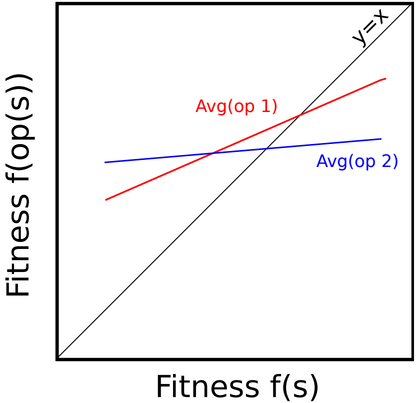
Evolvability

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

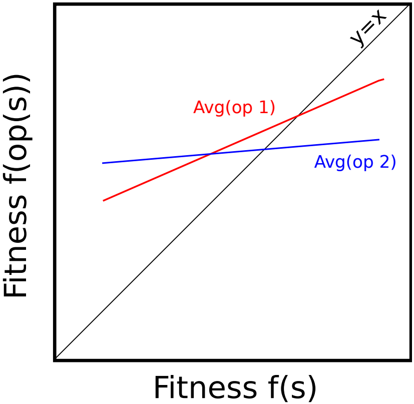
Average of evolvability



- Operator 1 ? ? Operator 2

Fitness cloud : comparing difficulty

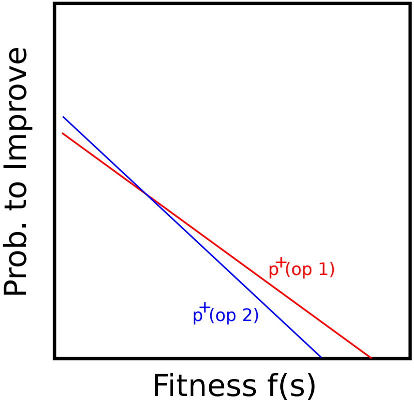
Average of evolvability



- 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

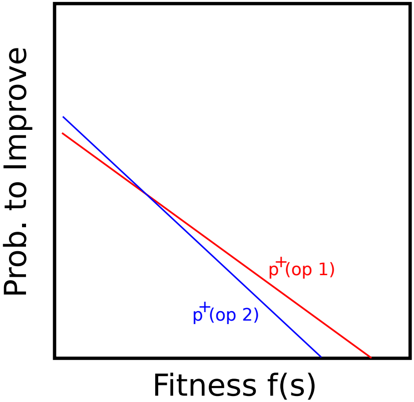
Probability to improve



- Operator 1 ? ? Operator 2

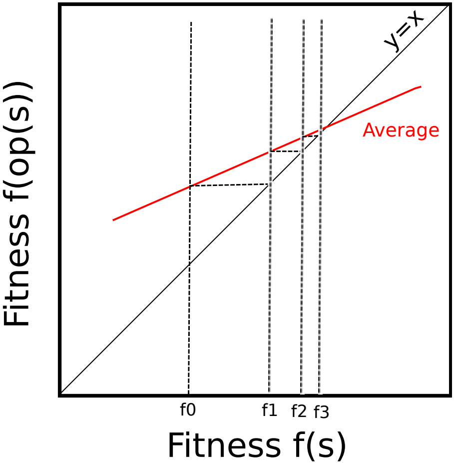
Fitness cloud : comparing difficulty

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

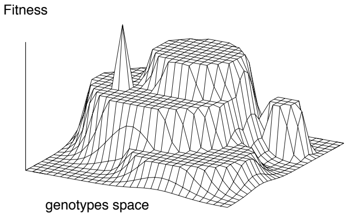


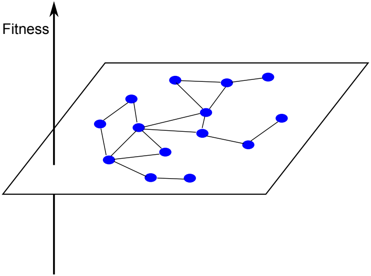
- Approximation (only approximation) of the fitness value after few steps of local operator
- Indication on the quality of the operator
- See fitness level technic

Where are we ?

- Neutral sets (done) :
  set of solutions with the same fitness-value
  $\Rightarrow$ No structure
- Fitness cloud (done) :
  Bivariate density $(f(s), f(op(s)))$
  $\Rightarrow$ Neighborhood relation **between** neutral sets
- Neutral networks (now) :
  $\Rightarrow$ neutral sets with neighborhood relation : graph

Neutral networks (Schuster 1994 [SFSH94])





Basic definition of Neutral Network

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

Definitions

Test of neutrality

$$isNeutral : S \times S \rightarrow \{true, false\}$$

For example, $isNeutral(x_1, x_2)$ is *true* if :

- $f(x_1) = f(x_2)$
- $|f(x_1) - f(x_2)| \leqslant 1/M$ with M is the search population size
- $|f(x_1) - f(x_2)|$ is under the evaluation error

Neutral neighborhood

of s is the set of neighbors which have the same fitness $f(s)$

$$\mathcal{N}_{neut}(s) = \{s' \in \mathcal{N}(s) \mid isNeutral(s, s')\}$$

Neutral degree of s

Number of neutral neighbors : $nDeg(s) = \sharp(\mathcal{N}_{neut}(s) - \{s\})$

Definitions

Neutral walk

$W_{neut} = (x_0, x_1, \ldots, x_m)$

- for all $i \in [\![0, m-1]\!]$, $x_{i+1} \in \mathcal{N}(x_i)$
- for all $(i, j) \in [\![0, m]\!]^2$, $isNeutral(x_i, x_j)$ is *true*

Neutral Network

graph $G = (N, E)$

- $N \subset X$: for all s and s' from N, there is a neutral walk belonging to N from s to s'
- $(x_1, x_2) \in E$ if they are neutral neighbors : $x_2 \in \mathcal{N}_{neut}(x_1)$

A fitness landscape is **neutral**
if there exist many solutions with a high neutral degree

Practice : computing the neutral rate

- The neutral rate is the proportion of neutral neighbors
- It can be estimated with a random walk :

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

Source code `exo03.R` :

- `main` :
  Code to compute the neutral rates

Questions

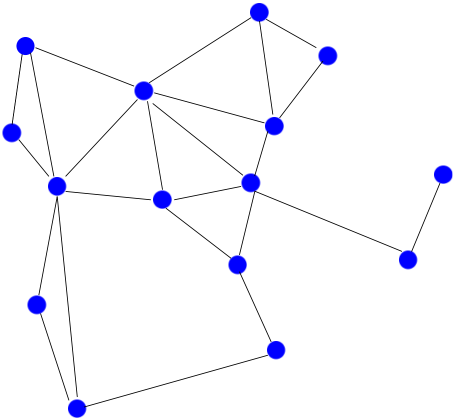
- Define the function `neutral_rate` to compute the neutral rate estimated with a random walk
- Execute the `main` function to compute the neutral rate
- Compare the neutrality of the SP problems 1 and 2

Features from the neutral network

Conventional graph metrics

1. **Size of NN** :
   - number of nodes of NN
2. **Neutral degree distribution** :
   - measure of the quantity of “neutrality”
3. **Autocorrelation of neutral degree** (Bastolla 03 [BPRV03]) during a neutral random walk :
   - comparaison with random graph
   - measure of the correlation structure of *NN*

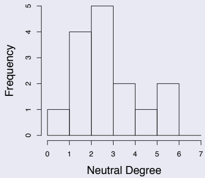
Features from the neutral network



1. **Size**

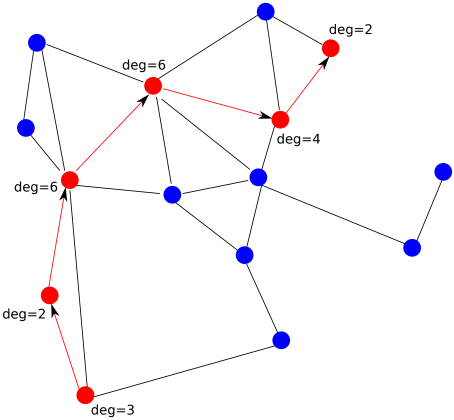
   avg, distribution ...

2. **Neutral degree distribution**



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

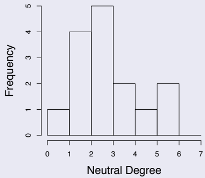
Features from the neutral network



1. **Size**

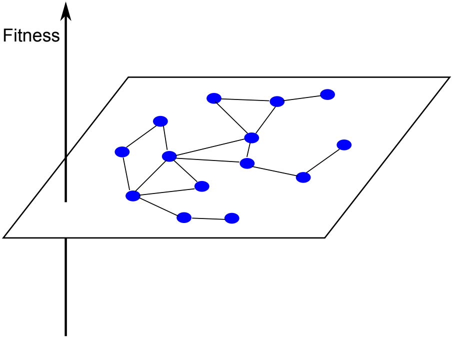
   avg, distribution . . .

2. **Neutral degree distribution**



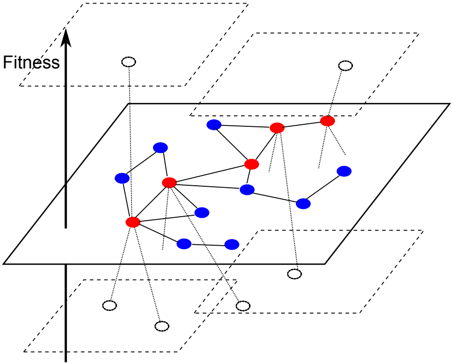
3. **Autocorrelation of the neutral degree**
   - random walk on NN
   - autocorr. of degrees

Features from the neutral network



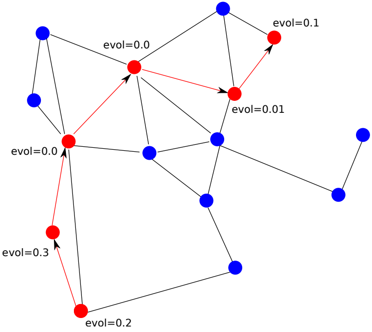
1. **Rate of innovation** (Huynen 96 [Huy96]) : the number of new accessible structures (fitness) per mutation
2. **Autocorrelation of evolvability** [VCC06] : autocorrelation of the sequence $(evol(x_0), evol(x_1), \ldots)$

Features from the neutral network



1. **Rate of innovation** (Huynen 96 [Huy96]) : the number of new accessible structures (fitness) per mutation
2. **Autocorrelation of evolvability** [VCC06] : autocorrelation of the sequence $(evol(x_0), evol(x_1), \ldots)$

Features between neutral networks



- **Autocorrelation of evolvability**
  - Autocorrelation of $(evol(x_0), evol(x_1), \ldots)$
  - Evolvability *evol* :
    - average fitness in the neighborhood
    - probability to improve
    - ...
- **What information ?**
  - if the correlation is high $\Rightarrow$ "easy" (you can use this information)
  - if the correlation is low $\Rightarrow$ "difficult"

Summary for neutral fitness landscape features

- Density of states
  Size of neutral sets
- Fitness cloud and related statistics
  Evolvability of solutions
- Neutral degrees distribution
  "How neutral is the fitness landscape ?"
- Autocorrelation of neutral degrees
  Network "structure"
- Autocorrelation of evolvability
  Evolution of evolvability on NN

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

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.

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.

References II

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.

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.

References III

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.

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.

References IV

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.

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.

References V

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

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 and Algorithm Performance for Single- and Multi-objective Combinatorial Optimization

Arnaud Liefooghe
Univ. Lille – CRIStAL / Inria
Lille, France
arnaud.liefooghe@univ-lille1.fr

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

IEEE Congress on Evolutionary Computation
Donostia - San Sebastián, Spain

June 5, 2017

Outline

Joint 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

$$\begin{array}{ccc} X & \xrightarrow{op} & X \\ p\downarrow & & \downarrow p \\ E & \xrightarrow{op'} & E \end{array}$$

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

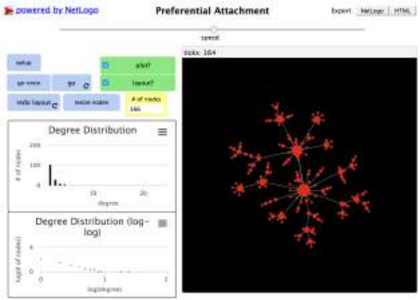
Methodology

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

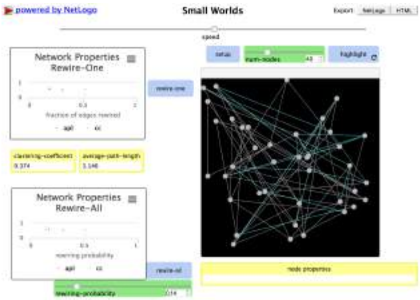
J. P. K. Doye, The network topology of a potential energy landscape : a static scale-free network., Phys. Rev. Lett., 88 :238701, 2002. [Doy02]

Complex networks

Scale free network
(Watts and Strogatz, 1998 [WS98])



Small world network
(Barabasi and Albert, 1999 [BA99])



Energy surface and inherent networks

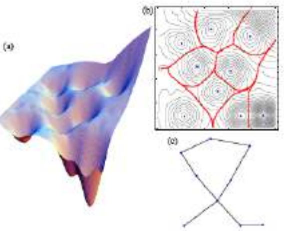
Inherent network

- **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

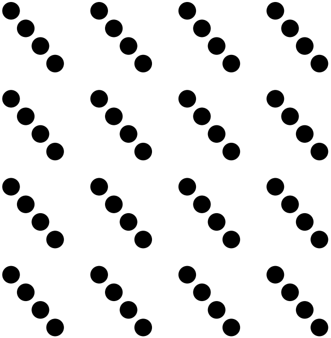
(c) Landscape as a network



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

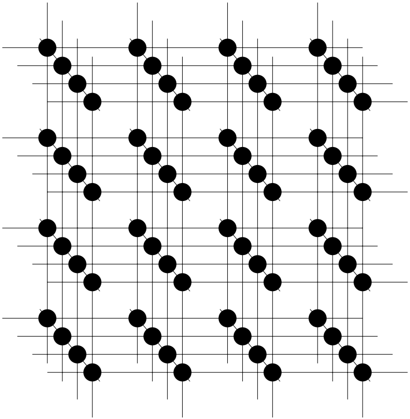
Example of a small NK landscape with $N = 6$ and $K = 2$



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

Basins of attraction in combinatorial optimization

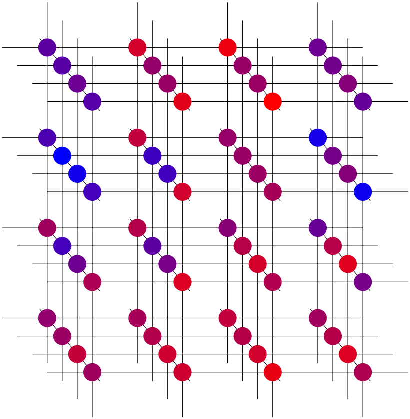
Example of a small NK landscape with $N = 6$ and $K = 2$



- 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 $K = 2$

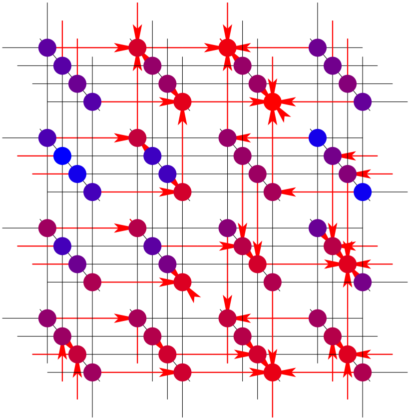


The color represents the fitness-values

- high fitness
- low fitness

Basins of attraction in combinatorial optimization

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



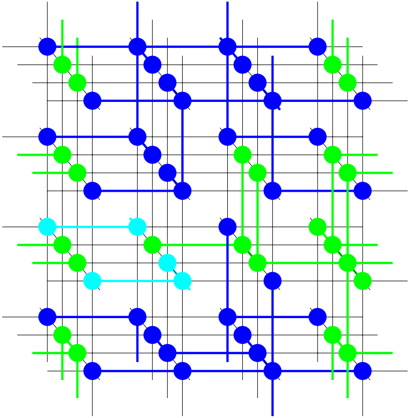
- Color represent fitness value
  - high fitness
  - low fitness
- → point towards the solution with highest fitness in the neighborhood

Exercise :

Why not making a Hill-Climbing walk on it ?

Basins of attraction in combinatorial optimization

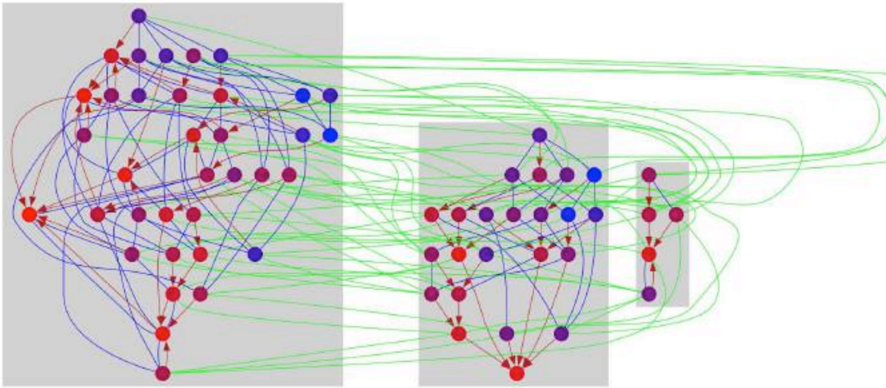
Example of small *NK* landscape with $N = 6$ and $K = 2$



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

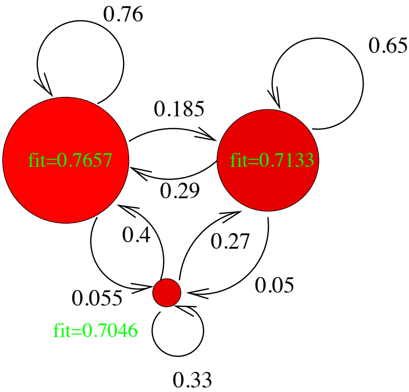
Basins of attraction in combinatorial optimization

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

Basin of attraction

Hill-Climbing algorithm (best-improvement)

Choose initial solution $x \in X$
repeat
  choose $x' \in \mathcal{N}(x)$ such that $f(x') = max_{y \in \mathcal{N}(x)} \; f(y)$
  if $f(x) < f(x')$ **then**
    $x \leftarrow x'$
  end if
until x is a Local optimum

Basin of attraction of $x^\star$:

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

Local optima network

Definition : Local Optima Network (LON)

Oriented weighted graph (V, E, w)

- Notes 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)

- Notes 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'

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

Prob. from solution s to basin b_j

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

Weights : Transition prob. from basin b_i to basin b_j

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

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'

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

For example, $X = \{0,1\}^N$ and bit-flip operator

if $x' \in \mathcal{N}(x)$, $p(x \to x') = \frac{1}{N}$, otherwise $p(x \to x') = 0$

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

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

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

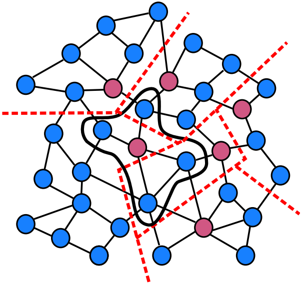
Escape edges

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

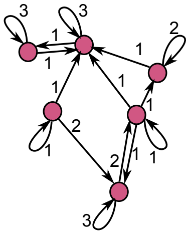
Escape edges

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



Basins of attraction features

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

NK-landscapes

[Kauffman 1993] [Kau93]

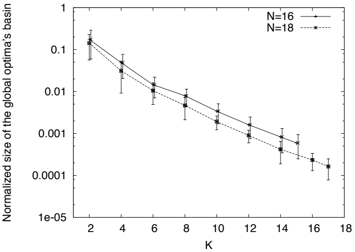
$$x \in \{0,1\}^n \qquad f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x_j, x_{i_1}, \ldots, x_{i_k})$$

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

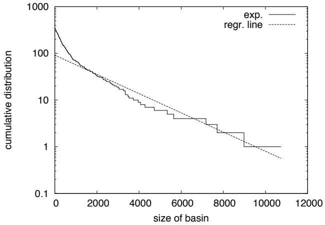
Global optimum basin size *vs.* non-linearity degree k



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

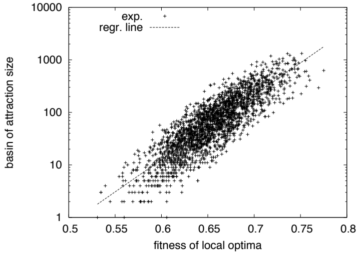
Distribution of basin sizes



Cumulative distribution of basins sizes for $n = 18$ and $k = 4$

- Log-normal cumulative distribution (**not uniform !**) :
  - large number of small basins
  - small number of large basins
- Effect of non-linearity :
  the distribution becomes more uniform with non-linearity degree k

Fitness of local optima *vs.* basin size

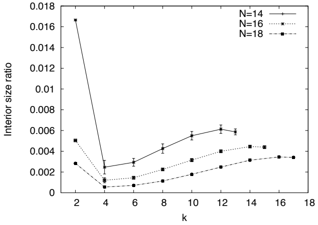


Correlation fitness of local optima *vs.* their corresponding basins sizes

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

Basin : Interior and border sizes



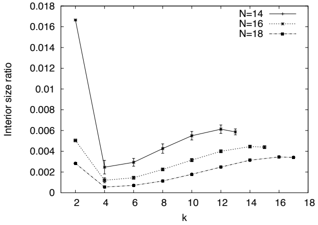
average of basins interior size ratio

Question :

Do basins look like a "mountain" with interior and border ?

solution $\in$ interior
if all neighbors are in the same basin

Basin : Interior and border sizes



average of basins interior size ratio

Question :

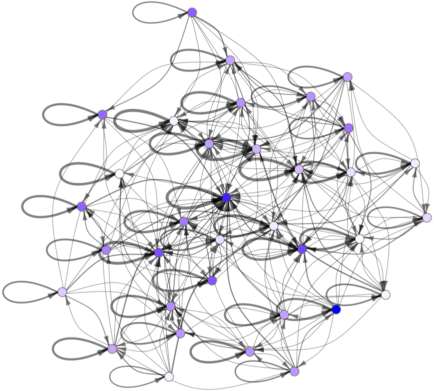
Do basins look like a "mountain" with interior and border ?

solution ∈ interior
if all neighbors are in the same basin

Answer

- Interior is very small
- Nearly all solutions ∈ border

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 $(x, y) \in E$

Some formal definitions

Weighted clustering coefficient

local density of the network

$$c^w(i) = \frac{1}{s_i(k_i - 1)} \sum_{j,h} \frac{w_{ij} + w_{ih}}{2} a_{ij} a_{jh} a_{hi}$$

where $s_i = \sum_{j \neq i} w_{ij}$, $a_{nm} = 1$ if $w_{nm} > 0$, $a_{nm} = 0$ if $w_{nm} = 0$ and $k_i = \sum_{j \neq i} a_{ij}$.

Disparity

dishomogeneity of nodes with a given degree

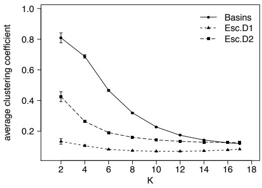
$$Y_2(i) = \sum_{j \neq i} \left(\frac{w_{ij}}{s_i} \right)^2$$

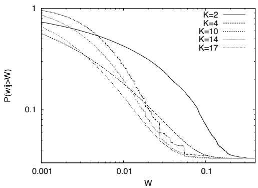
A LON-based fitness landscape analysis approach

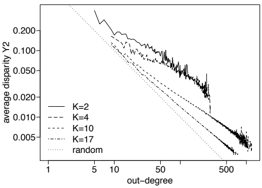
- Link between LON features and problem difficulty :
  small-size instances of NK-landscapes
- Analysis of the LON structure :
  small-size instances of NK-landscapes, QAP and Flowshop
- Design of one local search component :
  small-size instances of NK-landscapes and Flowshop
- Explaining performance with LON properties :
  simple correlation, small-size inst. of NK-landscapes, QAP
  multi-linear correlation, small-size instances of Flowshop
- Prediction performance with LON properties :
  large-size instances of NK-landscapes, QAP
- Algorithm portfolio :
  large-size instances of NK-landscapes, QAP

Structure of the local optima network

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

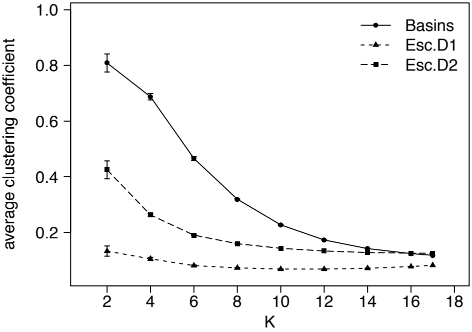






- LON is **not a random** network (NK, QAP, FSSP) :
  highly clustered network,
  distribution of weights and degrees have long tail . . .

Example : clustering coefficient for NK-landscapes



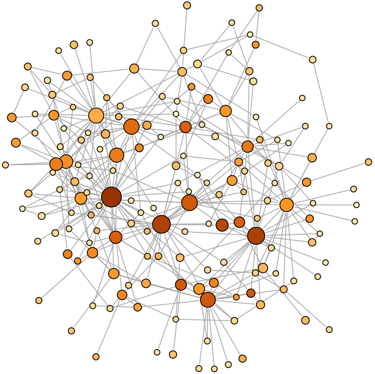
- Network highly clustered
- Clustering coefficient decreases with the degree of non-linearity k

LON to compare instance difficulty

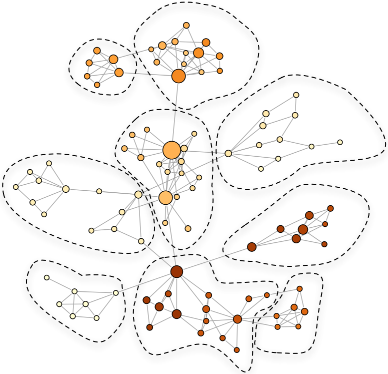
Local Optima Network for the Quadratic Assignment Problem (QAP) [DTVO11]

→ Community detection in LON :

Random instance



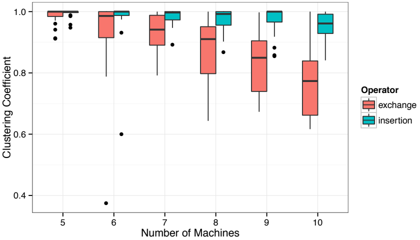
Real-like instance

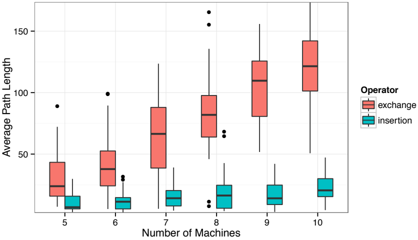


the structure of the LON is related to **problem difficulty**

LON to compare algorithm components (1)

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





LON to compare algorithm components (2)

comparaison of the hill-climbing's **pivot rule** for NK-landscapes

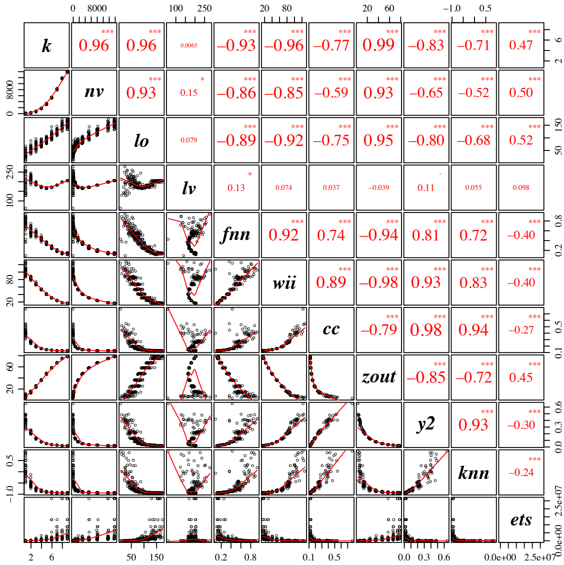
K	$\bar{n}_e/\bar{n}_v^2$		$\bar{Y}$		$\bar{d}$		$\bar{d}_{best}$	
	b-LON	f-LON	b-LON	f-LON	b-LON	f-LON	b-LON	f-LON
2	0.81	0.96	0.326	0.110	56	39	16	12
4	0.60	0.92	0.137	0.033	126	127	35	32
6	0.32	0.79	0.084	0.016	170	215	60	70
8	0.17	0.65	0.062	0.011	194	282	83	118
10	0.09	0.53	0.050	0.009	206	340	112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

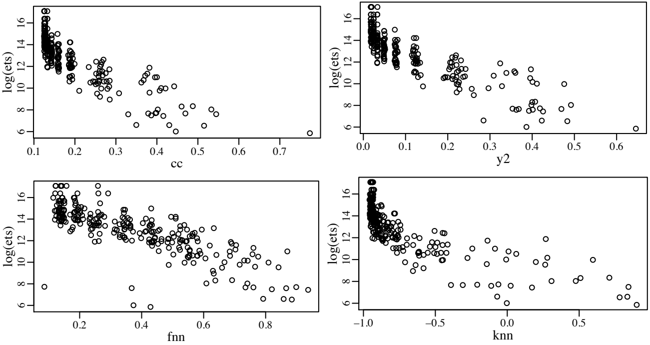


LON features *vs.* performance : simple correlation

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

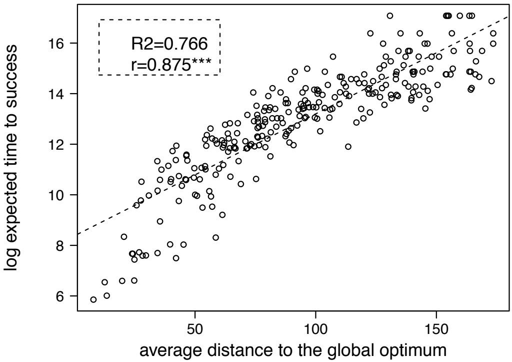
Performance : $ert = \mathbb{E}(T_s) + \left(\frac{1-p_s}{p_s}\right) T_{max}$

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]



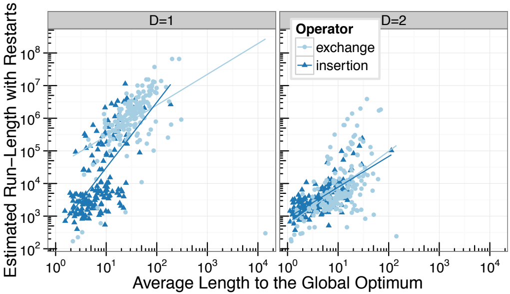
Expected running time

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

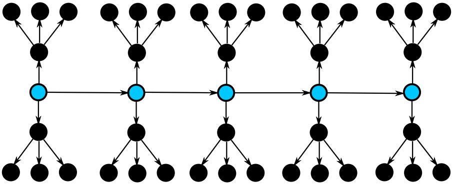
for the **Flowshop Scheduling Problem** using exhaustive selection

$\sharp P$	$\log(N_V)$	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

From the sampling of large-size complex network :

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



Procedure LONSampling(d, m, l)
$x_0 \leftarrow hc(x)$ with x random solution
for $t \leftarrow 0, \ldots l-1$ **do**
  Snowball(d, m, x_t)
  $x_{t+1} \leftarrow$ RandomWalkStep(x_t)
end for

Set of estimated LON features for large-size instances

	LON metrics
$\underline{\text{fit}}$	Average fitness of local optima in the network
$\underline{\text{wii}}$	Average weight of self-loops
$\underline{\text{zout}}$	Average outdegree
$\underline{y_2}$	Average disparity for outgoing edges
$\underline{\text{knn}}$	Weighted assortativity
$\underline{\text{wcc}}$	Weighted clustering coefficient
$\underline{\text{fnn}}$	Fitness-fitness correlation on the network
	Metrics from the sampling procedure
$\underline{\text{lhc}}$	Average length of hill-climbing to local optima
$\underline{\text{mlhc}}$	Maximum length of hill-climbing to local optima
$\underline{\text{nhc}}$	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* : 1^{st} 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)

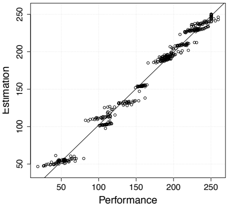
R^2 on cross-validation for NK-landscapes and QAP

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

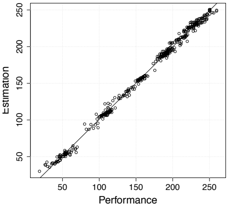
			NK					QAP				
Mod.	Feat.	Perf.	TS	SA	EA	ILS	avg	TS	SA	EA	ILS	avg
lm	basic	fit	0.8573	0.8739	0.8763	0.8874	0.8737	-38.42	-42.83	-41.63	-39.06	-40.48
lm	lon	fit	0.8996	0.9015	0.9061	0.8954	0.9007	**0.9995**	**1.0000**	**1.0000**	**0.9997**	0.9998
lm	all	fit	**0.9356**	**0.9455**	**0.9442**	**0.9501**	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

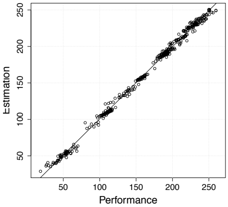
- On the 32 possibles cases (Mod. × Feat. × Algo.), the best set of features : *all* 27 times, *lon* 12 times, *basic* 6 times
- With linear model : *basic* set is never the one of the best set, *lon* features are more linearly correlated with performance
- Random forest model obtains higher regression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

A-L. Barabási and R. Albert.
Emergence of scaling in random networks.
Science, 286 :509–512, 1999.

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.

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.

References VI

D. J. Watts and S. H. Strogatz.
Collective dynamics of 'small-world' networks.
Nature, 393 :440–442, 1998.

4. Multi-objective Fitness Landscapes

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

Arnaud Liefooghe
Univ. Lille – CRIStAL / Inria
Lille, France
arnaud.liefooghe@univ-lille1.fr

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

IEEE Congress on Evolutionary Computation
Donostia - San Sebastián, Spain

June 5, 2017

Outline

1. ~~The Basics of Fitness Landscapes~~
2. ~~Geometries of Fitness Landscapes~~
3. ~~Local Optima Network~~
4. Multi-objective Fitness Landscapes
   - 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 typically hard
- understanding what makes a problem difficult, and how
- understanding what makes algorithms work well (or not)
- learning about the problem structure might lead to the design of better algorithms
- models to explain and predict the performance of algorithms based on (relevant) problem features
- models to understand the dynamics and the behavior of algorithms

Joint work

- Fabio Daolio, University of Stirling, UK
- Hernán Aguirre, Shinshu University, Japan
- Kiyoshi Tanaka, Shinshu University, Japan

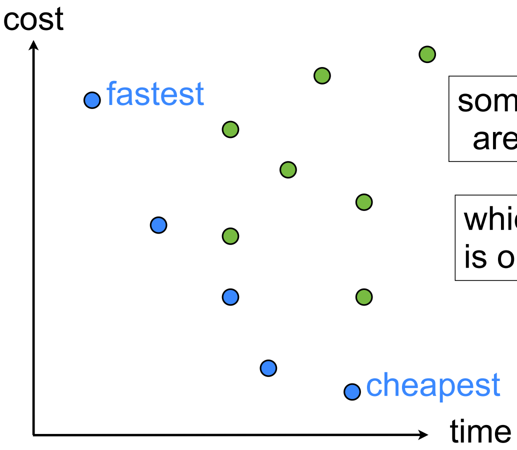
Many thanks! /// ありがとう ございます

I

(Evolutionary) multi-objective optimization

Shortest path

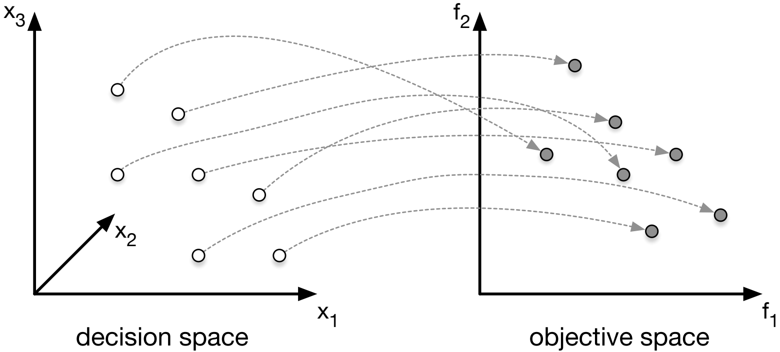
Shortest path



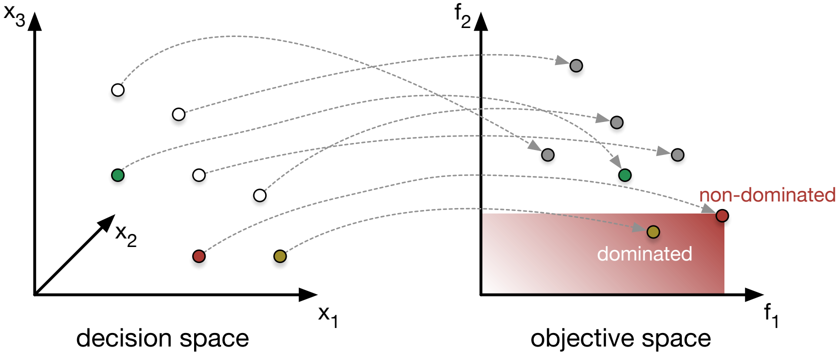
some paths are better

which path is optimal?

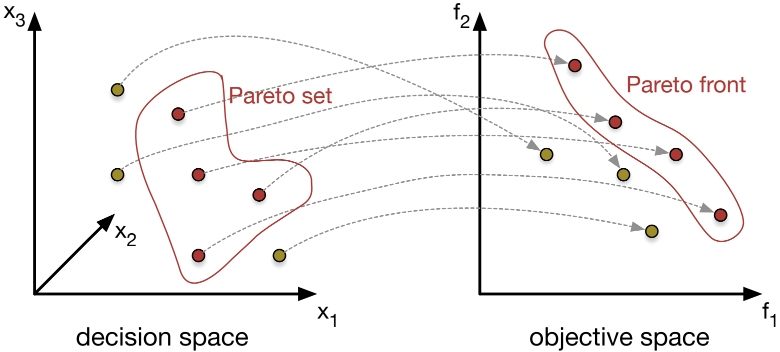
Definitions



Definitions



Definitions



Challenges

Why identifying the Pareto set is **difficult**? [Ehrgott 2005]

> intractability: the number of Pareto optimal solutions (non-dominated vectors) typically grows exponentially

> NP-completeness: deciding if a solution is Pareto optimal is difficult for many multi-objective optimization problems

What about a Pareto set **approximation**?

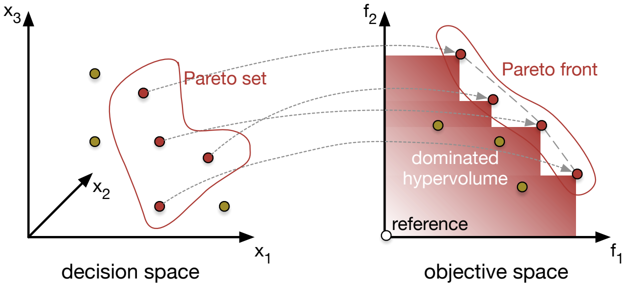
What is a good Pareto set approximation?

Rule of thumb

- closeness to the (exact) Pareto front
- well-distributed solutions in the objective space
- the more, the better?

Quality indicators

- scalar value that reflects an approximation set quality
- IGD, EPS, R-metrics, HV . . . (all have limitations and biases)

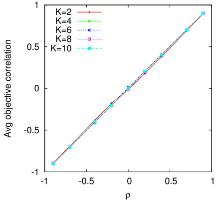


ρMNK-landscapes [Kauffman 1993; Aguirre & Tanaka 2007; Verel et al. 2010]

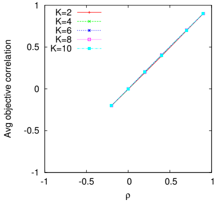
general-purpose family of multi-modal pseudo-boolean optimization functions
superposition of n Walsh functions of order k+1

$$\begin{aligned} \max \quad & f_i(x) = \frac{1}{n} \sum_{j=1}^{n} c_j^i(x_j, x_{j_1}, \ldots, x_{j_k}) \quad , \quad i \in \{1, \ldots, m\} \\ \text{s.t.} \quad & x_j \in \{0, 1\} \quad , \quad j \in \{1, \ldots, n\} \end{aligned}$$

$m = 2$



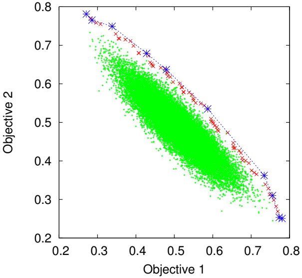
$m = 5$



Benchmark parameters:

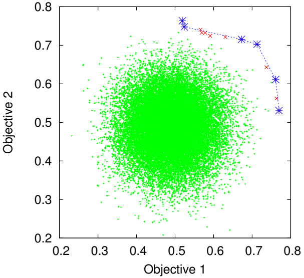
- problem size n (decision space dimension)
- problem non-linearity k < n (multi-modality, epistatic interactions)
- number of objective functions m (objective space dimension)
- objective correlation $\rho > -\frac{1}{m-1}$
  - multivariate uniform law
  - analytical and experimental proofs

Some intuitions on objective correlation ρ



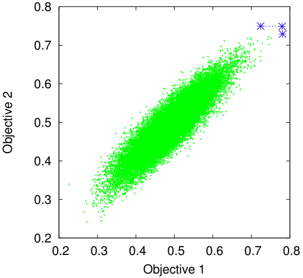
conflicting objectives

$\rho = -0.9$



independent objectives

$\rho = 0.0$



correlated objectives

$\rho = 0.9$

$m = 2 \quad n = 18 \quad k = 4$

EMO algorithm classes

Scalarizing approaches

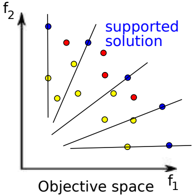
- multiple aggregations of the objectives (e.g. weighted-sum)
- beware of **unsupported** solutions
- MOSA, MOTS, TPLS, MOEA/D ...

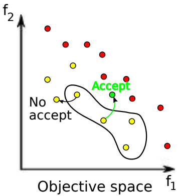
Dominance-based approaches

- search process guided by a dominance relation
- NSGA-II, SPEA2, PAES, PLS, SEMO, AεSεH ...

Indicator-based approaches

- search process guided by a quality indicator
- IBEA, IBMOLS, SMS-EMOA, HypE ...





Two prototypical dominance-based EMO algorithms

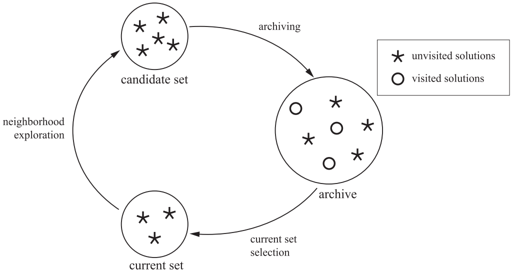
local search
multi-objective hill-climber
PLS
[Paquete et al. 2004]

$A \leftarrow \{x_0\}$
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

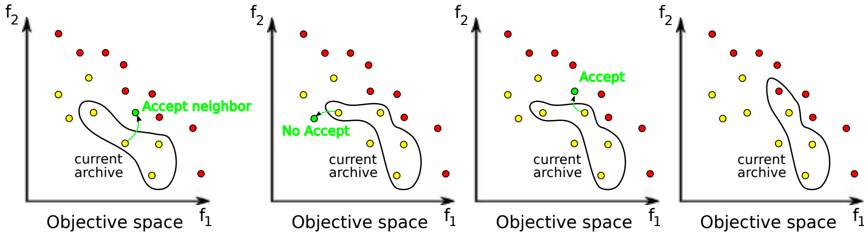
global search
multi-objective $(1+1)-$EA
G-SEMO
[Laumanns et al. 2004]

$A \leftarrow \{x_0\}$
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

Pareto local search (PLS) [Paquete et al. 2004]



- Archive solutions using the dominance relation
- Iteratively improve this archive by exploring its neighborhood



II

Problem features

Benchmark parameters

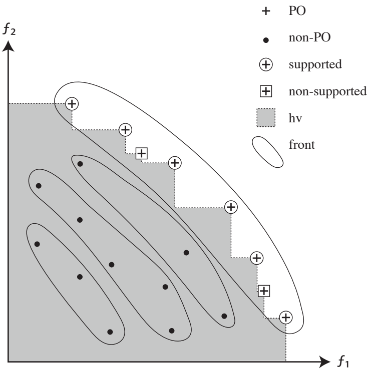
Parameters from ρmnk-landscapes

- n problem size
  (solution space dimension)
- k problem non-linearity
  (number of epistatic interactions)
- m number of objective functions
  (objective space dimension)
- ρ objective correlation
  (correlation between the objective function values)

$$\begin{array}{lll} \max & f_i(x) = \frac{1}{n}\sum_{j=1}^{n} c_j^i(x_j, x_{j_1}, \ldots, x_{j_k}) & , \quad i \in \{1, \ldots, m\} \\ \text{s.t.} & x_j \in \{0,1\} & , \quad j \in \{1, \ldots, n\} \end{array}$$

Global features from full enumeration (1)

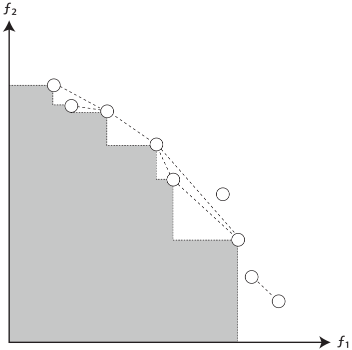
Features from the Pareto set/solution space



#po	Pareto optimal (PO) sol.
#supp	supported PO solutions
hv	PF's hypervolume
#fronts	non-dominated fronts
front_ent	entropy of front's size distribution

Global features from full enumeration (2)

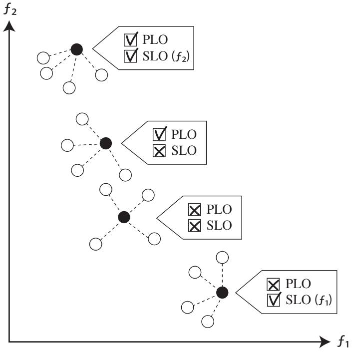
Features from the Pareto set/graph



podist_avg	avg Hamming distance
podist_max	max distance (diameter)
fdc	fitness-distance correlation
#cc	connected components
#sing	singletons
#lcc	largest connected comp.
lcc_dist	avg distance in LCC
lcc_hv	LCC's hypervolume

Global features from full enumeration (3)

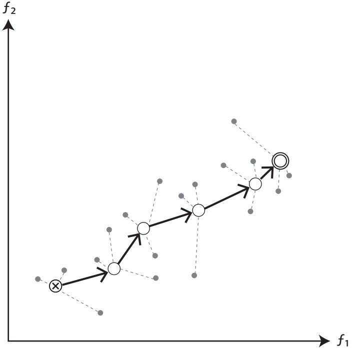
Local optimality



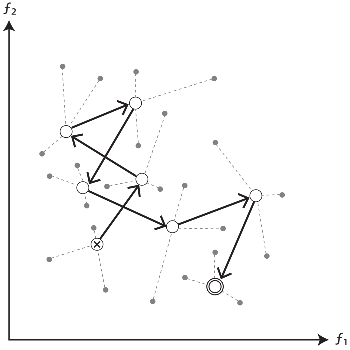
#plo	Pareto local optimal (PLO) solutions
#slo_avg	single-objective local optima (SLO) per objective (avg)

Local features from sampling (1)

Multi-objective adaptive/random walk



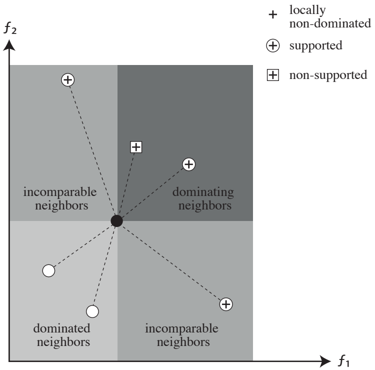
adaptive walk sampling (aws)



random walk sampling (rws)

Local features from sampling (2)

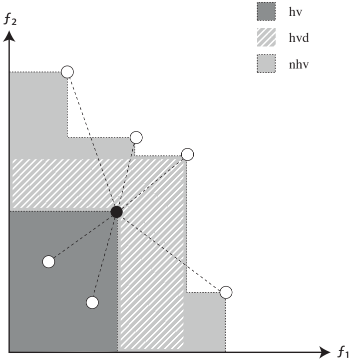
Dominance-based metrics



- locally non-dominated solutions in the neighborhood
- supported locally non-dominated solutions in the neighborhood
- neighbors dominated by the current solution
- neighbors dominating the current solution
- neighbors incomparable to the current solution
- average length of aws

Local features from sampling (3)

Hypervolume-based metrics



- (single) solution's hypervolume
- (single) solution's hypervolume difference
- neighborhood's hypervolume

Summary of problem features (1)

	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	*knowles2003*
#supp	proportion of supported solutions in the Pareto set	*knowles2003*
hv	hypervolume-value of the (exact) Pareto front	*aguirre2007*
#plo	proportion of Pareto local optimal (PLO) solutions	*paquete2007*
#slo_avg	average proportion of single-objective local optimal solutions per objective	
podist_avg	average Hamming distance between Pareto optimal solutions	*liefooghe2013*
podist_max	maximal Hamming distance between Pareto optimal solutions (diameter of the Pareto set)	*knowles2003*
po_ent	entropy of binary variables from Pareto optimal solutions	*knowles2003*
fdc	fitness-distance correlation in the Pareto set (Hamming dist. in solution space vs. Manhattan dist. in objective space)	*knowles2003*
#cc	proportion of connected components in the Pareto graph	*paquete2009*
#sing	proportion of isolated Pareto optimal solutions (singletons) in the Pareto graph	*paquete2009*
#lcc	proportional size of the largest connected component in the Pareto graph	*verel2011*
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	*aguirre2007*
front_ent	entropy of the non-dominated front's size distribution	

Summary of problem features (2)

	LOCAL FEATURES FROM RANDOM WALK **sampling** (rws) (17)	
hv_avg_rws	average (single) solution's hypervolume-value	
hv_r1_rws	first autocorrelation coefficient of (single) solution's hypervolume-values	*liefooghe2013*
hvd_avg_rws	average (single) solution's hypervolume difference-value	
hvd_r1_rws	first autocorrelation coefficient of (single) solution's hypervolume difference-values	*liefooghe2013*
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** (aws) (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	*verel2011*

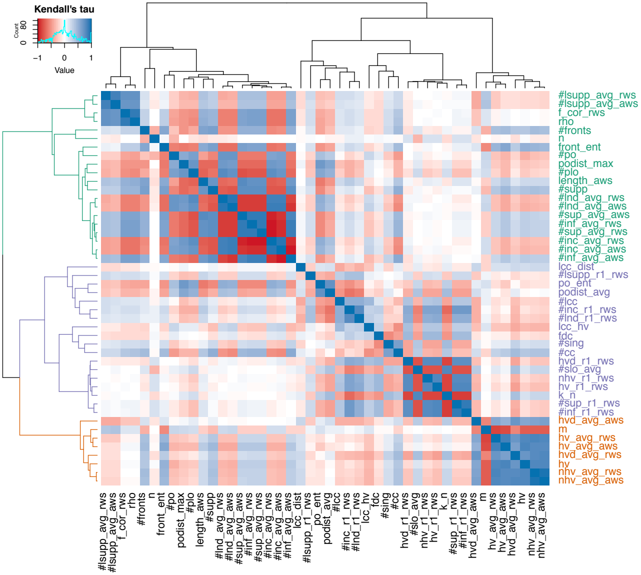
Experimental setup for enumerable instances

Small-size ρMNK-landscapes, factorial design, 30 instances

- problem size n $\in \{10, 11, 12, 13, 14, 15, 16\}$
- problem non-linearity k $\in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$
- number of objectives m $\in \{2, 3, 4, 5\}$
- objective correlation $\rho \in \{-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1\}$, $\rho > \frac{-1}{m-1}$

60 480 problem instances overall

Pairwise feature association (enumerable instances)



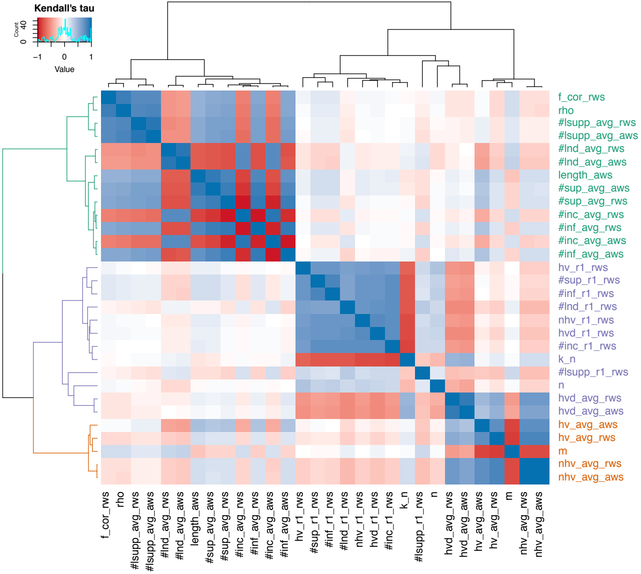
Experimental setup for large-size instances

Large-size ρMNK-landscapes, constrained random LHS DOE

- problem size $n \in [\![64, 256]\!]$
- problem non-linearity $k \in [\![0, 8]\!]$
- number of objectives $m \in [\![2, 5]\!]$
- objective correlation $\rho \in [-1, 1]$, $\rho > \frac{-1}{m-1}$

1 000 problem instances overall

Pairwise feature association (large-size instances)



III

Feature-based performance prediction

Experimental setup for large-size instances

Large-size ρMNK-landscapes, constrained random LHS DOE

- problem size $n \in [\![64, 256]\!]$
- problem non-linearity $k \in [\![0, 8]\!]$
- number of objectives $m \in [\![2, 5]\!]$
- objective correlation $\rho \in [-1, 1]$, $\rho > \frac{-1}{m-1}$

1 000 problem instances overall

GSEMO and IPLS algorithms

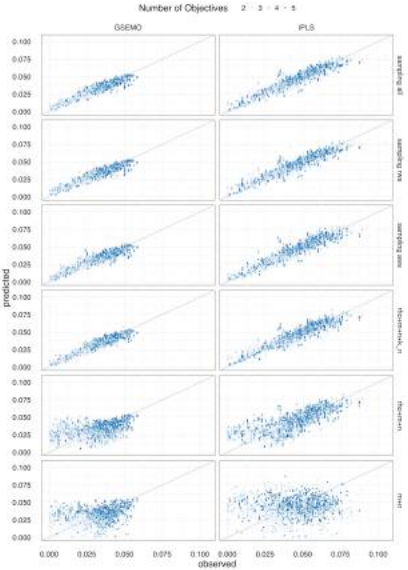
- 30 independent runs per instance
- Fixed budget of 100 000 evaluation calls
- epsilon approximation ratio to best-found non-dominated set

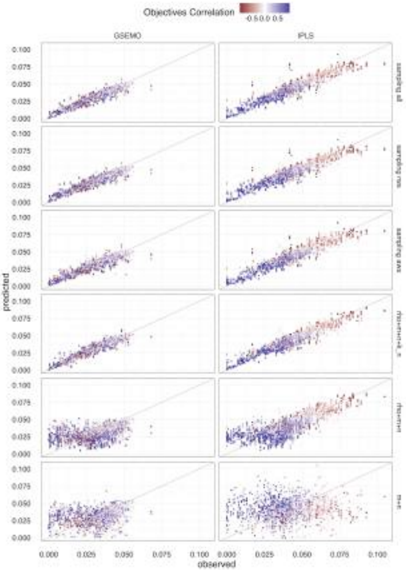
Prediction accuracy

cross validation with repeated subsampling, 50 iterations, 90/10 split

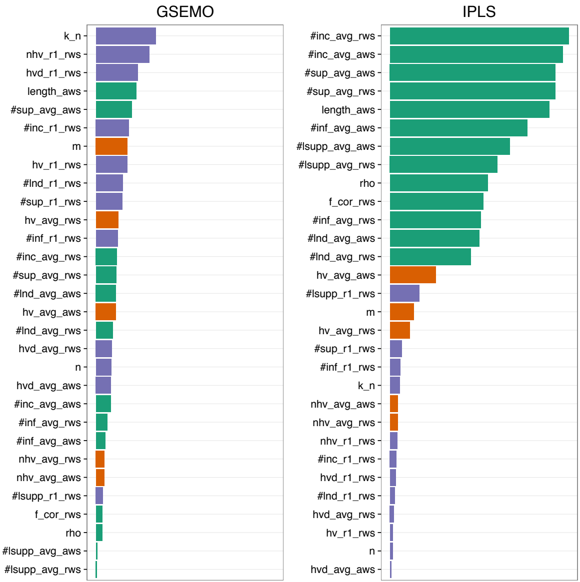
feature set	MAE	MSE	R^2	rank
		GSEMO		
all	**0.003049**	**0.000017**	**0.891227**	1
sampling all	0.003152	**0.000018**	**0.883909**	1.3
sampling rws	0.003220	0.000019	0.878212	2
sampling aws	0.003525	0.000023	0.854199	3
ρ+m+n+k/n	**0.003084**	**0.000017**	**0.892947**	1
ρ+m+n	0.009062	0.000148	0.065258	4
m+n	0.010813	0.000206	-0.303336	5
		IPLS		
all	**0.004290**	**0.000034**	**0.886568**	1
sampling all	**0.004359**	**0.000035**	**0.883323**	1
sampling rws	**0.004449**	0.000036	**0.879936**	1.3
sampling aws	0.004663	0.000039	0.871011	2
ρ+m+n+k/n	**0.004353**	**0.000033**	**0.889872**	1
ρ+m+n	0.008415	0.000119	0.600965	3
m+n	0.016959	0.000472	-0.568495	4

Predicted vs observed values (out-of-folds)





Features importance



Portfolio accuracy

cross validation with repeated subsampling, 50 iterations, 90/10 split

Portfolio: { GSEMO , IPLS }

feature set	error rate	rank
all	**0.0128**	1
sampling all	**0.0138**	1
sampling rws	**0.0150**	1
sampling aws	**0.0144**	1
ρ+m+n+k/n	**0.0134**	1
ρ+m+n	0.0824	2
m+n	0.1328	3
const=GSEMO	0.0880	
const=IPLS	0.7250	

IV

Complementary/concluding remarks

Set-based multi-objective fitness landscapes

A complementary view?

Key idea

- EMO algorithms are local search algorithms performing on sets [Zitzler et al 2010]

Set-based multi-objective fitness landscape $(\Sigma, \mathtt{N}, \mathtt{I})$ [Verel et al 2011]

- **Set-domain search space** (Σ)
  $\Sigma \subset 2^X$ is a set of feasible solution-sets
  (where X is the set of feasible solutions)
- **Set-domain neighborhood relation** $(\mathtt{N})$
  $\mathtt{N} : \Sigma \to 2^\Sigma$ is a neighborhood relation between solution-sets
- **Set-domain fitness function** $(\mathtt{I})$
  $\mathtt{I} : \Sigma \to \mathbb{R}$ is a unary quality indicator, e.g. hypervolume
  i.e. a fitness function measuring the quality of solution-sets

Conclusions and open issues

- Features to characterize multi-objective fitness landscapes
- Those features relate to problem difficulty
- Gaining knowledge about the multi-objective opt. problem
- Algorithm performance can be predicted using those features (for some problem/algorithm classes)

And now?

- Improving the design and configuration of EMO algorithms?
- Multi-objective algorithm portfolio?
- More theoretical works about EMO?
- Designing cheap features?
- ...

Concluding Remarks

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

Arnaud Liefooghe
Univ. Lille – CRIStAL / Inria
Lille, France
arnaud.liefooghe@univ-lille1.fr

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

IEEE Congress on Evolutionary Computation
Donostia - San Sebastián, Spain

June 5, 2017

Basic methodology of fitness landscapes analysis

- Density of states : pure random search, initialization ?
- Length of adaptive walks : multimodality ?
- Fitness autocorrelation : ruggedness ?
- Neutral degree distribution : neutrality ?
- Fitness cloud : Quality of the operator, evolvability ?
- Neutral walks and evolvability : neutral information ?
- Features from the local optima network : structure at LO level ?

Recent review : Katherine M. Malan, Information Sciences, (2013)

Basic methodology of fitness landscapes analysis

- Density of states : pure random search, initialization ?
- Length of adaptive walks : multimodality ?
- Fitness autocorrelation : ruggedness ?
- Neutral degree distribution : neutrality ?
- Fitness cloud : Quality of the operator, evolvability ?
- Neutral walks and evolvability : neutral information ?
- Features from the local optima network : structure at LO level ?
- ... be creative from your algorithm and problem point of view
- ... be careful on the computed measures : one measure is not enough, and must be very well understood

Recent review : Katherine M. Malan, Information Sciences, (2013)

Sofware to perform fitness landscape analysis

Framework **ParadisEO**

http://paradiseo.gforge.inria.fr

C++ software framework for the the reusable of metaheuristics
(local search, EA, continuous, discrete, parallel, fitness landscape. . .)

```
moAutocorrelationSampling<Neighbor> sampling(randomInitialization,
                                             neighborhood,
                                             evalFunction,
                                             neighborEvaluation,
                                             nbStep);

sampling();

sampling.fileExport(str_out);
```

Summary on fitness landscapes

Fitness landscape is a representation of

- search space
- notion of neighborhood
- fitness of solutions

Summary on fitness landscapes

Fitness landscape is a representation of

- search space
- notion of neighborhood
- fitness of solutions

Goal :

- **local description** : fitness between neighboring solutions
  Ruggedness, local optima, fitness cloud, neutral networks, local optima networks. . .
- . . . to deduce **global features** :
  - Difficulty !
  - Decide (and control) a good choice for the representation, variation operator and fitness function

Open issues

- How to control the **parameters** of the algorithm with the local description of fitness landscape ?
- Links between **neutrality** and time complexity (difficulty) ?
- Can fitness landscape describe the dynamics of a **population** of solutions ?
- Fitness landscape for **parallel** algorithm (island model) ?
- What about **crossover** ?
- **Multi-objective**, **continuous** optimization problems. . .
- Links with theoretical approaches
- ...

Open issues

- Which "**aggregation of variables**" shows relevant properties of the optimization problem according to the local search heuristic ?

$$\begin{array}{ccc} X & \xrightarrow{op} & X \\ p\downarrow & & \downarrow p \\ Z & \xrightarrow{op_Z} & Z \end{array}$$

Open issues

- Which "**aggregation of variables**" shows relevant properties of the optimization problem according to the local search heuristic ?

$$\begin{array}{ccc} X & \xrightarrow{op} & X \\ p\downarrow & & \downarrow p \\ Z & \xrightarrow{op_Z} & Z \end{array}$$

Thanks for your attention !