Fitness Landscapes and Graphs: Multimodularity, Ruggedness and Neutrality

Sébastien Verel

DOLPHIN team - INRIA Lille-Nord Europe
I3S laboratory - University of Nice-Sophia Antipolis / CNRS, France
http://www.i3s.unice.fr/~verel

July, 8 2012
1 Introduction:
   context, goals, related theoretical works, definition
2 Fitness landscapes of multimodal, and neutral problems, a case study
3 Local optima network
4 Fitness landscapes for continuous and multobjective problems
Optimization framework

**Inputs**

- **Search space**: Set of all feasible solutions, \( S \)
- **Objective function**: Quality criterium
  \[ f : S \rightarrow \mathbb{R} \]

**Goal**

Find the best solution according to the criterium

\[ s^* = \text{argmax} \ f \]
Optimization framework

**Inputs**

- **Search space**: Set of all feasible solutions, $S$
- **Objective function**: Quality criterion $f : S \rightarrow \mathbb{R}$

**Goal**

Find the best solution according to the criterion

$$s^* = \operatorname{argmax} f$$

**But, sometime, the set of all best solutions, good approximation of the best solution, good 'robust' solution...**
Black box Scenario

We have only \{ (s_0, f(s_0)), (s_1, f(s_1)), ... \} given by an "oracle". No information is either not available or needed on the definition of objective function.

- Objective function given by a computation, or a simulation.
- Objective function can be irregular, non differentiable, non continuous, etc.
Black box Scenario

We have only \( \{(s_0, f(s_0)), (s_1, f(s_1)), \ldots\} \) given by an "oracle"
No information is either not available or needed on the definition of objective function

- Objective function given by a computation, or a simulation
- Objective function can be irregular, non differentiable, non continuous, etc.
- (Very) large search space for discrete case (combinatorial optimization), i.e. NP-complete problems
Search algorithms

**Principle**

**Enumeration of the search space**

- A lot of ways to enumerate the search space
- Using random sampling: Monte Carlo techniques
- Local search techniques:

```
Solution
```

```
Initialization → Selection → Replacement → Random Variation
```

```
Accept?
```

```
neighborhood
```

```
neighbor
```
Search algorithms

Principle

Enumeration of the search space

- A lot of ways to enumerate the search space
- Using random sampling: Monte Carlo techniques
- Local search techniques:

If objective function $f$ has no properties: random search
If not...
Fitness landscapes : Motivations

Why using fitness landscapes?

- To analyse the structure of the search space
- To study problem (search) difficulty in combinatorial optimisation:
  information on runtime for a given problem and a class of LS
- To design effective search algorithms

L. Barnett, U. Sussex, DPhil Diss. 2003
"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"
Fitness landscapes in biology

Biological evolution:
- a metaphorical uphill struggle across a "fitness landscape"
- mountain peaks represent high "fitness", or ability to survive,
- valleys represent low fitness.
- evolution proceeds:
  population of organisms performs an "adaptive walk"
Fitness landscapes in biology

In biology:
- Modelisation of species evolution

Used to model dynamical systems:
- statistical physic,
- molecular evolution,
- ecology, etc
Fitness landscapes in biology

2 sides for Fitness Landscapes:

- Powerful **metaphor**: most profound concept in evolutionary dynamics
  - give pictures of evolutionary process
  - be careful of misleading pictures: "smooth landscape without noise"

- **Quantitative** concept: predict the evolutionary paths
  - Quasispecies equation: mean field analysis with differential equations
  - Stochastic process: markov chain
  - Network analysis
In combinatorial optimization

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

- $S$: set of admissible solutions,
- $\mathcal{N}: S \rightarrow 2^S$: neighborhood function,
- $f: S \rightarrow \mathbb{R}$: fitness function.

$2^S$ is the set of sets: $\mathcal{N}(s) \subset S$
Fitness landscapes in Evolutionary Computation

2 sides for Fitness Landscapes:

- **Powerful metaphor**: most profound concept
  - give pictures of the search dynamic:
    "if the fitness landscapes have big valleys, I can use this algorithm"
  - be careful of misleading pictures: set of smooth mountains

- **Quantitative** concept: predict the evolutionary dynamic
  - Quasispecies equation: mean field analysis with differential equations
  - Stochastic process: markov chain
  - Network analysis
What is a neighborhood?

**Neighborhood function:**

\[ \mathcal{N} : S \rightarrow 2^S \]

Set of "neighbor" solutions associated to each solution

\[ \mathcal{N}(x) = \{ y \in S \mid \mathbb{P}(y = \text{op}(x)) > 0 \} \]
What is a neighborhood?

**Neighborhood function:**

\[ \mathcal{N} : S \rightarrow 2^S \]

Set of "neighbor" solutions associated to each solution

\[ \mathcal{N}(x) = \{ y \in S \mid \mathbb{P}(y = op(x)) > 0 \} \]

or

\[ \mathcal{N}(x) = \{ y \in S \mid \mathbb{P}(y = op(x)) > \epsilon \} \]
What is a neighborhood?

**Neighborhood function:**

\[ \mathcal{N} : S \rightarrow 2^S \]

Set of "neighbor" solutions associated to each solution

\[ \mathcal{N}(x) = \{ y \in S \mid \mathbb{P}(y = \text{op}(x)) > 0 \} \]

or

\[ \mathcal{N}(x) = \{ y \in S \mid \mathbb{P}(y = \text{op}(x)) > \epsilon \} \]

or

\[ \mathcal{N}(x) = \{ y \in S \mid d(y, x) \leq 1 \} \]
Example of neighborhood: bit strings

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

*Algorithm:* simple GA, hill-climbing, or simulated annealing, etc.

$\mathcal{N}(01101) = \{$

01100,  
01111,  
01001,  
00101,  
11101, 

$\}$$

Important!

Definition of neighborhood must be based on the local search operator used in the algorithm

Neighborhood $\Leftrightarrow$ Operator

$\mathcal{N}(x) = \{y \in S \mid d_{Hamming}(y, x) = 1\}$
Example of neighborhood: permutations

- **Search space**: 
  \[ S = \{ \sigma \mid \sigma \text{ permutations} \} \]

- **Algorithm**: simple EA
  operator: 2-opt

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

\[ \mathcal{N}(x) = \{ y \in S \mid \text{IP}(y = op_{2opt}(x)) > 0 \} \]
Example of neighborhood

- **Search space**: 
  \[ S = \{ \sigma | \sigma \text{ permutations } \} \]

- **Algorithm**: simple EA
  - operators: 2-opt and 3-opt

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

\[ \mathcal{N}(x) = \{ y \in S | \Pr(y = op_{2\text{opt}}(x)) > 0 \text{ or } \Pr(y = op_{3\text{opt}}(x)) > 0 \} \]
Example of neighborhood: memetic algorithms

- **Algorithm**: memetic algorithm, EA + operator hill-climbing

\[ \mathcal{N}(x) = \{ y \in S \mid y = op_{HC}(x) \} \]
Example of neighborhood: memetic algorithms

- **Algorithm**: memetic algorithm, EA + operator hill-climbing
  \[ \mathcal{N}(x) = \{ y \in S \mid y = op_{HC}(x) \} \]

- **Algorithm**: memetic algorithm, EA + operator hill-climbing and bit-flip mutation

2 possibilities:

- Study 2 landscapes:
  - one for \( HC \) operator, one for bit-flip mutation

- Study 1 landscape:
  \[ \mathcal{N}(x) = \{ y \in S \mid y = op_{HC}(x) \text{ or } \mathbb{P}(y = op_{bit-flip}(x)) > \epsilon \} \]
Example of neighborhood: memetic algorithms

- **Algorithm**: memetic algorithm, EA + operator hill-climbing

\[ \mathcal{N}(x) = \{y \in S \mid y = \text{op}_{HC}(x)\} \]

- **Algorithm**: memetic algorithm, EA + operator hill-climbing and bit-flip mutation

2 possibilities:

- Study 2 landscapes:
  - one for \( HC \) operator, one for bit-flip mutation

- Study 1 landscape:

\[ \mathcal{N}(x) = \{y \in S \mid y = \text{op}_{HC}(x) \text{ or } \mathbb{P}(y = \text{op}_{bit\text{-flip}}(x)) > \epsilon\} \]

It depends on what you want to know
Goal of the fitness landscapes study

- "Geometry" (features) of fitness landscape ⇒ dynamics of a local search algorithm
- Geometry is linked to the problem difficulty:
  - If there are a lot of local optima, the probability to find the global optimum is lower.
  - If the fitness landscape is flat, discovering better solutions is rare.
  - What is the best search direction in the landscape?

Study of the fitness landscape features allows to study the performance of search algorithms
Goal of the fitness landscapes study

1. To compare the difficulty of two search spaces:
   - One problem with 2 (or more) possible codings: \((S_1, N_1, f_1)\)
     and \((S_2, N_2, f_2)\)
     different coding, mutation operator, fitness function, etc.
     Which one is easier to solve?
Goal of the fitness landscapes study

1. To compare the difficulty of two search spaces:
   - One problem with 2 (or more) possible codings: \((S_1, N_1, f_1)\)
     and \((S_2, N_2, f_2)\)
   - different coding, mutation operator, fitness function, etc.
   - Which one is easier to solve?

2. To select the algorithm:
   - analysis of features of the landscape
   - Which algorithm can I use or design?
Goal of the fitness landscapes study

1. To compare the difficulty of two search spaces:
   - One problem with 2 (or more) possible codings: \((S_1, N_1, f_1)\) and \((S_2, N_2, f_2)\)
   - different coding, mutation operator, fitness function, etc.
   - Which one is easier to solve?

2. To select the algorithm:
   - analysis of features of the landscape
   - Which algorithm can I use or design?

3. To tune the parameters:
   - *off-line* analysis of structure of fitness landscape
   - Which is the best mutation operator? the size of the population? etc.
Goal of the fitness landscapes study

1. To compare the difficulty of two search spaces:
   - One problem with 2 (or more) possible codings: \((S_1, \mathcal{N}_1, f_1)\) and \((S_2, \mathcal{N}_2, f_2)\)
   - different coding, mutation operator, fitness function, etc.
   - Which one is easier to solve?

2. To select the algorithm:
   - analysis of features of the landscape
   - Which algorithm can I use or design?

3. To tune the parameters:
   - \textit{off-line} analysis of structure of fitness landscape
   - Which is the best mutation operator? the size of the population? etc.

4. To control the parameters during the run:
   - \textit{on-line} analysis of structure of fitness landscape
   - Which is the optimal mutation rate according to the estimation of structure?
Fitness landscapes point of view

$$FL = (\text{Sol.}, \text{Neighbors}, \text{Fitness})$$
Fitness landscapes point of view

\[ FL = (\text{Sol.}, \text{Neighbors}, \text{Fitness}) \]

No particular heuristic, heuristics based on the same neighborhood relation

- Sample the neighborhood to have information on **local features** of the search space
- From local information: deduce some **global features** like general shape of search space, "difficulty", etc.
Goal of the fitness landscapes study

Study of the **geometry** of the landscape allows to study the **difficulty**, and design a good optimisation algorithm.

Fitness landscape is a graph \((S, \mathcal{N}, f)\) where the nodes have a value (fitness) : can be "pictured" as a "real" landscape.

Two main geometries have been studied :
- multimodal and ruggedness
- neutral
Multimodal Fitness landscapes

Local optima $s^*$ (maximization):

no neighbor solution with higher fitness value

$$\forall s \in \mathcal{N}(s^*), f(s) < f(s^*)$$
Multimodal Fitness landscapes

Local optima \( s^* \) (maximization):

no neighbor solution with higher fitness value

\[
\forall s \in \mathcal{N}(s^*), f(s) \leq f(s^*)
\]
**Multimodal Fitness landscapes**

Adaptive walk: \((s_0, s_1, \ldots)\) where \(s_{i+1} \in \mathcal{N}(s_i)\) and \(f(s_i) < f(s_{i+1})\)

**Hill-Climbing (HC) algorithm**

Choose initial solution \(s \in S\)

repeat

choose \(s' \in \mathcal{N}(s)\) such that \(f(s') = \max_{x \in \mathcal{N}(s)} f(x)\)

if \(f(s) < f(s')\) then

\(s \leftarrow s'\)

end if

until \(s\) is a Local optimum

Basin of attraction of \(s^*\):

\[ \{ s \in S \mid \text{HillClimbing}(s) = s^* \} . \]
Multimodal Fitness landscapes

Optimisation difficulty: number and size of attractive basins (Garnier et al [10])

The idea:
- if the size of attractive basin of global optima is relatively "small"
- the problem is difficult to optimize

The measure:
- Length of adaptive walks (distribution, avg, etc.)
Walking on fitness landscapes

Random walk: \((s_1, s_2, \ldots)\) such that \(s_{i+1} \in \mathcal{N}(s_i)\) and equiprobability on \(\mathcal{N}(s_i)\)

- Fitness seems to be very "chaotic"
- Analysis the fitness during the random walk as a signal

fitness vs. step of a random walk (example of max-SAT problem)
Rugged/smooth fitness landscapes

Autocorrelation of time series of fitnesses \( f(s_1), f(s_2), \ldots \) along a random walk \( (s_1, s_2, \ldots) \) \cite{35}:

\[
\rho(n) = \frac{E[(f(s_i) - \bar{f})(f(s_{i+n}) - \bar{f})]}{\text{var}(f(s_i))}
\]

autocorrelation length \( \tau = \frac{1}{\rho(1)} \)

- small \( \tau \): rugged landscape
- long \( \tau \): smooth landscape
Results on rugged fitness landscapes (Stadler 96 [26])

<table>
<thead>
<tr>
<th>Problem</th>
<th>parameter</th>
<th>$\rho(1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>symmetric TSP</td>
<td>$n$ number of towns</td>
<td>$1 - \frac{4}{n}$</td>
</tr>
<tr>
<td>anti-symmetric TSP</td>
<td>$n$ number of towns</td>
<td>$1 - \frac{4}{n-1}$</td>
</tr>
<tr>
<td>Graph Coloring Problem</td>
<td>$n$ number of nodes</td>
<td>$1 - \frac{2\alpha}{(\alpha-1)n}$</td>
</tr>
<tr>
<td></td>
<td>$\alpha$ number of colors</td>
<td></td>
</tr>
<tr>
<td>NK landscapes</td>
<td>$N$ number of proteins</td>
<td>$1 - \frac{K+1}{N}$</td>
</tr>
<tr>
<td></td>
<td>$K$ number of epistasis links</td>
<td></td>
</tr>
</tbody>
</table>

Ruggedness decreases with the size of those problems: small variation has less effect on the fitness values.

Weinberger, Stadler, Whitley, Sutton: elementary landscapes
Fitness Distance Correlation (FDC) (Jones 95 [15])

Correlation between distance to global optimum and fitness

Classification based on experimental studies:
- $\rho < -0.15$, easy optimization
- $\rho > 0.15$, hard optimization
- $-0.15 < \rho < 0.15$, undecided zone
Neutral Fitness Landscapes

Neutral theory (Kimura ≈ 1960 [17])

Theory of mutation and random drift

A considerable number of mutations have no effects on fitness values

- plateaus
- neutral degree
- neutral networks
  [Schuster 1994 [25], RNA folding]
Neutral Fitness Landscapes
Combinatorial optimization

- Redundant problem (symmetries, ...) (Goldberg 87 [12])
- Problem “not well” defined or dynamic environment (Torres 04 [14])

Fitness

Applicative problems:
- Robot controller
- Circuit design
- Genetic programming
- Protein Folding
- Learning problems
Neutrality and difficulty

- In our knowledge, there is no definitive answer about neutrality / problem hardness
- Certainly, it is dependent on the nature of neutrality of the fitness landscape

⇒ Sharp description of the geometry of neutral fitness landscapes is needed
Neutrality and difficulty

We know for certain that:

- **No information** is better than **Bad information**: Hard trap functions are more difficult than needle-in-a-haystack functions
- **Good information** is better than **No information**
Neutrality and difficulty

We know for certain that:

- **No information** is better than **Bad information**:
  Hard trap functions are more difficult than needle-in-a-haystack functions
- **Good information** is better than **No information**

- When there is **No information**:
  you should have a good method to find it!
In the following

Description of neutral fitness landscapes:

- Neutral sets:
  set of solutions with the same fitness

- Neutral networks:
  add neighborhood information
Neutral sets: Density Of States

Set of solutions with fitness value

Density of states (D.O.S.)

- Introduce in physics (Rosé 1996 [24])
- Optimization (Belaidouni, Hao 00 [4])
Neutral sets: Density Of States

Density of states (D.O.S.)

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
Neutral sets: Fitness Cloud

- \((S, \mathcal{F}, \mathbb{P})\): probability space
- \(op: S \rightarrow S\) 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: Measure of evolvability

Evolvability

Ability to evolve: fitness in the neighborhood compared to the fitness of the solution

- Probability of finding better solutions
- Average fitness of better neighbor solutions
- Average and standard deviation of fitnesses
Fitness cloud: Comparison of difficulty
Fitness cloud: Comparison of difficulty

- Operator 1 > Operator 2
- Because Average 1 more correlated to fitness
- Linked to autocorrelation
- Average is often a line:
  - See works on Elementary Landscapes (D. Wihtley and others)
  - See Negative Slope Coefficient (NSC)
Fitness cloud
Prediction of fitness (CEC 2003)

- Approximation (only approximation) of the fitness value after few steps of local operator
- Indication on the quality of the operator
Neutral fitness landscapes

- Neutral sets (done):
  set of solutions with the same fitness
  ⇒ No structure

- Fitness cloud (done):
  Bivariate density \((f(s), f(op(s)))\)
  ⇒ Neighborhood relation between neutral sets

- Neutral networks (to be done):
  ⇒ Neighborhood structure into the neutral sets: Graph
Neutral networks (Schuster 1994 [25])

Fitness

genotypes space

Neutral Network

Doors
Definitions (simple version)

**Neutral neighborhood**

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

$$\mathcal{N}_{\text{neut}}(s) = \{ s' \in \mathcal{N}(s) \mid f(s) = f(s') \}$$

**Neutral degree of $s$**

Number of neutral neighbors: $nDeg(s) = \#(\mathcal{N}_{\text{neut}}(s) - \{s\})$. 
Definitions

**Test of neutrality**

\[
\text{isNeutral} : S \times S \to \{\text{true}, \text{false}\}
\]

For example, \(\text{isNeutral}(s_1, s_2)\) is true if:
- \(f(s_1) = f(s_2)\).
- \(|f(s_1) - f(s_2)| \leq 1/M\) with \(M\) is the search population size.
- \(|f(s_1) - f(s_2)|\) is under the evaluation error.

**Neutral neighborhood**

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

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

**Neutral degree of \(s\)**

Number of neutral neighbors: \(n\text{Deg}(s) = \#(\mathcal{N}_{neut}(s) - \{s\})\).
Definitions

**Neutral walk**

\[ \mathcal{W}_{neut} = (s_0, s_1, \ldots, s_m) \]

- for all \( i \in [0, m - 1] \), \( s_{i+1} \in \mathcal{N}(s_i) \)
- for all \((i, j) \in [0, m]^2\), \( isNeutral(s_i, s_j) \) is true.

**Neutral Network**

\[ G = (N, E) \]

- \( N \subset S \): for all \( s \) and \( s' \) from \( V \), there is a neutral walk belonging to \( V \) from \( s \) to \( s' \),
- \( (s_1, s_2) \in E \) if they are neutral neighbors: \( s_2 \in \mathcal{N}_{neut}(s_1) \)

A fitness landscape is neutral if there are many solutions with high neutral degree.
Neutral Networks (NN) : Inside Metrics

Classical graph metrics :

- **Size of NN** :
  number of nodes of NN,

- **Neutral degree distribution** :
  - measure of the quantity of "neutrality"

- **Autocorrelation of neutral degree** (Bastolla 03 [3]) :
  during neutral random walk
  - comparison with random graph,
  - measure of the correlation structure of $NN$
Neutral Networks: Inside Metrics

- **Size**: 15 solutions
- **Distribution of size overall landscapes**
- **Neutral degree distribution**

---

Neutral Degree | Frequency
---|---
0 | 1
1 | 2
2 | 3
3 | 4
4 | 4
5 | 3
6 | 1
7 | 0

---

S. Verel
Fitness landscapes and graphs
Neutral Networks: Inside Metrics

- **Size**: 15 solutions
  Distribution of size overall landscapes
- **Neutral degree distribution**
- **Autocorrelation of neutral degree**:
  - random walk on NN
  - autocorrelation of degrees
Neutral Networks: Outside Metrics

- Number of portals (exits toward better solutions), spreading of portals
- Autocorrelation of evolvability [33]: autocorrelation of the sequence \( \text{evol}(s_0), \text{evol}(s_1), \ldots \).
Neutral Networks: Outside Metrics

- Autocorrelation of evolvability:
  - Evolvability: \( evol = \text{avg fitness in the neighborhood} \)
  - Autocorrelation of \( (evol(s_0), evol(s_1), \ldots) \).

- Informations:
  - if high correlation \( \Rightarrow \) "easy"
    (you can use this information)
  - if low correlation \( \Rightarrow \) "difficult"
From analysis to design: a case study

PhD work of Marie-Eléonore Marmion (December 2011)

An example:
From fitness landscapes analysis to design of efficient local search
A case study: The Permutation Flowshop Scheduling Problem (FSP)

<table>
<thead>
<tr>
<th></th>
<th>$J_1$</th>
<th>$J_2$</th>
<th>$J_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_3$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_4$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $N$ jobs, $M$ machines
- Processing time of each job can be different on each machine
- Each job can be processed on at most one machine
- Each machine can process at most one job at a time
- Job order is the same on every machine:
  Representation $= $ Permutation
- $\Rightarrow$ Makespan minimization
Neighborhood relation

Insertion operator

Neighborhood size: \((N - 1)^2\)

Something strange in the state-of-art local search (Iterated Greedy)
So a fitness analysis...
# Neutrality

## Question 1

Is there some neutrality in the problem?

## Neutral degree (ratio) of local optima

<table>
<thead>
<tr>
<th>Sample: set of local optima</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>20</td>
</tr>
</tbody>
</table>
Question 1

Is there some neutrality in the problem?

Neutral degree (ratio) of local optima

<table>
<thead>
<tr>
<th>N</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>65 (18%)</td>
<td>792 (33%)</td>
<td>3920 (40%)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7 (2%)</td>
<td>96 (4%)</td>
<td>784 (8%)</td>
<td>6732 (17%)</td>
</tr>
<tr>
<td>20</td>
<td>4 (1%)</td>
<td>24 (1%)</td>
<td>98 (1%)</td>
<td>396 (1%)</td>
</tr>
</tbody>
</table>

Yes! There is some neutrality.

Continue the analysis to find information...
Neutral Networks

Question 2
The neutral networks have some structure?

Autocorrelation of neutral degrees

![Graph showing autocorrelation of neutral degrees](image-url)
Neutral Networks

Question 2

The neutral networks have some structure?

Autocorrelation of neutral degrees

Yes! Neutral Networks are not random graphs.

Continue the analysis to find information...
Portals, exits

Question 3
Can we reach a portal?

Typology of neutral networks

- No T1 for $N = 50, 100, 200$
- $< 20\%$ T1 and T2 for $N = 20$, $< 3\%$ T2 for $N = 50, 100, 200$
- $> 97\%$ T3
Portals, exits

Question 3
Can we reach a portal?

Typology of neutral networks

- No T1 for \( N = 50, 100, 200 \)
- \(< 20\% \) T1 and T2 for \( N = 20 \), \(< 3\% \) T2 for \( N = 50, 100, 200 \)
- \( > 97\% \) T3

Yes! There is some portals.
## Neutrality

### Question 4
Is it easy to find a portal?

### Average number of steps before a portal
- **During random neutral walks**

<table>
<thead>
<tr>
<th>N</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>M = 5</td>
<td>17</td>
<td>33</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>14</td>
<td>17</td>
<td>30</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>
Neutrality

Question 4

Is it easy to find a portal?

Average number of steps before a portal

- During random neutral walks

<table>
<thead>
<tr>
<th>N</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>M = 5</td>
<td>17</td>
<td>33</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>14</td>
<td>17</td>
<td>30</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

Yes! The portals are closed to local optima.
Design efficient Local search

NILS : Neutral Iterated Local Search

- Local Search : First-improvement Hill-Climbing (until local optimum)
- Perturbation : to escape from plateau
  - Neutral moves (until maximum number of steps)
  - Kick move (from the IG local search)

⇒ Very competitive vs. the state-of-art IG, and we understand why.
Summary of metrics

- Neutral degrees distribution:
  "How neutral is the fitness landscape?"

- Autocorrelation of neutral degrees: network “structure”

- Portals, exits:
  Escaping of plateau to find better solutions

- Autocorrelation of evolvability:
  Information on the links between NN
Basic Methodology of fitness landscapes analysis

- Density of States: pure random search, initialization?
- Length of adaptive walks: multimodality?
- Autocorrelation of fitness: ruggedness?
- Neutral Degree Distribution: neutrality?
- Fitness Cloud: Quality of the operator, evolvability?
- Fitness Distance Correlation from best known
- Neutral walks and evolvability: neutral information?
Basic Methodology of fitness landscapes analysis

- Density of States: pure random search, initialization?
- Length of adaptive walks: multimodality?
- Autocorrelation of fitness: ruggedness?
- Neutral Degree Distribution: neutrality?
- Fitness Cloud: Quality of the operator, evolvability?
- Fitness Distance Correlation from best known
- Neutral walks and evolvability: neutral information?
- ... 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 understand
Software to perform fitness landscape analysis

Framework ParadisEO 1.3


and tutorials:

```java
moAutocorrelationSampling<Neighbor> sampling(randomInit,
    neighborhood,
    fullEval,
    incrementalEval,
    nbStep);

sampling();

sampling.fileExport(str_out);
```
Motivation and general idea: Levels of description

- **Fitness landscapes**: based on an huge number of solutions
- **One metric**: based on one real number, or curve to catch all the complexity
- **Local optima Network**: based on local optima

High level | Medium level | Low level
-----------|--------------|-------------
One metric | Local Optima Network | Fitness landscape
FDC, autocorrelation, etc. | local optima, basins of attraction | Solutions
Overview and Motivation

- Bring the tools of \textit{complex networks} analysis to the study the structure of combinatorial fitness landscapes
- \textbf{Goals}: Understand problem difficulty, design effective heuristic search algorithms
- \textbf{Methodology}: Extract a network that represents the landscape (Inspiration from energy landscapes (Doye, 2002)\textsuperscript{1})
  - \textit{Vertices}: local optima
  - \textit{Edges}: a notion of adjacency between basins
- Conduct a network analysis
- Relate (exploit?) network features to search algorithm design

**Small – world networks** (Watts and Strogatz, 1998)

- Neither ordered nor completely random
- Nodes highly clustered yet path length is small
- Network topological measures:
  - $C$: clustering coefficient, measure of local density
  - $l$: shortest path length, global measure of separation

**Scale – free networks** (Barabasi and Albert, 1999)

- The distribution of the number of neighbours (the degree distribution) is right – skewed with a heavy tail
- Most of the nodes have less-than-average degree, whilst a small fraction of hubs have a large number of connections
- Described mathematically by a power-law
Energy surface and inherent networks (Doye, 2002)

Inherent network:

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

- **Model of 2D energy surface**
- **Contour plot, partition of the configuration space into basins of attraction surrounding minima**
- **landscape as a network**
Basins of attraction in combinatorial optimisation
Example of 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 optimisation
Example of 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 optimisation
Example of small NK landscape with $N = 6$ and $K = 2$

Color represent fitness value
- high fitness
- low fitness
Basins of attraction in combinatorial optimisation
Example of small \( NK \) landscape with \( N = 6 \) and \( K = 2 \)

- Color represent fitness value
  - \( \bigcirc \) high fitness
  - \( \bullet \) low fitness
- \( \rightarrow \) point towards the solution with highest fitness in the neighborhood

Exercise:
Why not make a Hill-Climbing walk on it?
Basins of attraction in combinatorial optimisation

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 optimisation

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

- Basins of attraction are interlinked and overlapped!
- Most neighbours of a given solution are outside its basin
Local optima network

- **Nodes**: local optima
- **Edges**: transition probabilities
Basin of attraction

Hill-Climbing (HC) algorithm

Choose initial solution \( s \in S \)

repeat

choose \( s' \in \mathcal{N}(s) \) such that \( f(s') = \max_{x \in \mathcal{N}(s)} f(x) \)

if \( f(s) < f(s') \) then

\( s \leftarrow s' \)

end if

until \( s \) is a Local optimum

Basin of attraction of \( s^* \):

\[ \{s \in S \mid \text{HillClimbing}(s) = s^* \}. \]
local optima network

<table>
<thead>
<tr>
<th>Local optima network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes: set of local optima $S^*$</td>
</tr>
<tr>
<td>Edges: notion of connectivity between basins of attraction</td>
</tr>
<tr>
<td>$e_{ij}$ between $i$ and $j$ if there is at least a pair of neighbours $s_i$ and $s_j \in \mathcal{N}(s_i)$ such that $s_i \in b_i$ and $s_j \in b_j$ (GECCO 2008 [21])</td>
</tr>
<tr>
<td>weights $w_{ij}$ is attached to the edges, account for transition probabilities between basins (ALIFE 2008 [34], Phys. Rev. E 2008 [30], CEC 2010)</td>
</tr>
<tr>
<td>Escape edges: $e_{ij}$ between $i$ and $j$ if Basins $j$ which can be reached from the local optima $i$ (Artificial Evolution 2011 [31])</td>
</tr>
</tbody>
</table>
Basin edges: Weights of edges

- From each $s$ and $s'$, $p(s \rightarrow s') = \mathbb{P}(s' = \text{op}(s))$
  
  For example, $S = \{0, 1\}^N$ and bit-flip operator
  
  - if $s' \in \mathcal{N}(s)$, $p(s \rightarrow s') = \frac{1}{N}$
  - if $s' \notin \mathcal{N}(s)$, $p(s \rightarrow s') = 0$
From each $s$ and $s'$, $p(s \rightarrow s') = \mathbb{P}(s' = \text{op}(s))$

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

- if $s' \in \mathcal{N}(s)$, $p(s \rightarrow s') = \frac{1}{N}$
- if $s' \notin \mathcal{N}(s)$, $p(s \rightarrow s') = 0$

Probability that a configuration $s \in S$ has a neighbor in a basin $b_j$

$$p(s \rightarrow b_j) = \sum_{s' \in b_j} p(s \rightarrow s')$$
Basin edges: Weights of edges

- From each $s$ and $s'$, $p(s \rightarrow s') = IP(s' = op(s))$
  
  For example, $S = \{0, 1\}^N$ and bit-flip operator
  
  - if $s' \in \mathcal{N}(s)$, $p(s \rightarrow s') = \frac{1}{N}$
  - if $s' \not\in \mathcal{N}(s)$, $p(s \rightarrow s') = 0$

- Probability that a configuration $s \in S$ has a neighbor in a basin $b_j$

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

- $w_{ij}$: Total probability of going from basin $b_i$ to basin $b_j$ is the average over all $s \in b_i$ of the transition prob. to $s' \in b_j$:

  $p(b_i \rightarrow b_j) = \frac{1}{\#b_i} \sum_{s \in b_i} p(s \rightarrow b_j)$
Escape edges

Given distance function \( d \), and positive integer \( D > 0 \).

- \( e_{ij} \) between \( LO_i \) and \( LO_j \):
  - if it exists a solution \( s \) such that: \( d(s, LO_i) \leq D \) and \( h(s) = LO_j \).

- Weight \( w_{ij} \) of \( e_{ij} \):
  - \( \# \{ s \in S \mid d(s, LO_i) \leq D \text{ and } h(s) = LO_j \} \)
  - normalized by \( \# \{ s \in S \mid d(s, LO_i) \leq D \} \).

\[ \Rightarrow \text{local optima network: weighted oriented graph} \]
Basin edges vs. Escape edges

Basin edges

Escape edges
$NK$ fitness landscapes: ruggedness and epistasis

$NK$-landscapes: Model of problems

- $N$ size of the bit-strings
- $K$ from 0 to $N-1$, $NK$ landscapes can be tuned from smooth to rugged (easy to difficult respectively):
  - $K = 0$ no correlations, $f$ is an additive function, and there is a single maximum
  - $K = N-1$ landscape completely random, the expected number of local optima is $\frac{2^N}{N+1}$
  - Intermediate values of $K$ interpolate between these two extreme cases and have a variable degree of epistasis (i.e. gene interaction)
Methods

- Extracted and analysed networks
  - $N \in \{14, 16, 18\}$,
  - $K \in \{2, 4, \ldots, N - 2, N - 1\}$
  - 30 random instances for each case

- Measures:
  - Statistics on basins sizes and fitness of optima
  - Network features: clustering coefficient, shortest path to the global optimum, weight distribution, disparity, boundary of basins
Remember: Multimodal Fitness landscapes

Optimisation difficulty:
number and size of attractive basins (Garnier et al [10])

The idea:
- if the size of attractive basin of global optima is relatively "small"
- the problem is difficult to optimize
Global optimum basin size versus K

- Trend: the basin shrinks very quickly with increasing K.
- for higher K, more difficult for a search algorithm to locate the basin of attraction of the global optimum.

Size of the basin corresponding to the global maximum for each K
Analysis of basins: basin size

- Trend: small number of large basin, large number of small basin
- Log-normal cumulative distribution: not uniform!
- Slope of correlation increases with $K$
- When $K$ large: basin sizes are nearly equals the distribution becomes more uniform

Cumulative distribution of basins sizes for $N = 18$ and $K = 4$
Analysis of basins: basin size

- Trend: small number of large basin, large number of small basin
- log-normal cumulative distribution
- slope of correlation increases with K
- when K large: basin sizes are nearly equals
Analysis of basins: fitness vs. basin size

- Trend: clear positive correlation between the fitness values of maxima and their basins’ sizes.

- The highest, the largest

- On average, the global optimum easier to find than one other local optimum
- But more difficult to find, as the number of local optima increases exponentially with increasing K.
General network statistics

**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**

Dischomogeneity of nodes with a given degree

\[
Y_2(i) = \sum_{j \neq i} \left( \frac{w_{ij}}{s_i} \right)^2
\]
### General network statistics $N = 16$

<table>
<thead>
<tr>
<th>$K$</th>
<th># nodes</th>
<th># edges</th>
<th>$C^w$</th>
<th>$Y$</th>
<th>$d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>33$_{15}$</td>
<td>516$_{358}$</td>
<td>0.96$_{0.0245}$</td>
<td>0.326$_{0.0579}$</td>
<td>56$_{14}$</td>
</tr>
<tr>
<td>4</td>
<td>178$_{33}$</td>
<td>9129$_{2930}$</td>
<td>0.92$_{0.0171}$</td>
<td>0.137$_{0.0111}$</td>
<td>126$_8$</td>
</tr>
<tr>
<td>6</td>
<td>460$_{29}$</td>
<td>41791$_{4690}$</td>
<td>0.79$_{0.0154}$</td>
<td>0.084$_{0.0028}$</td>
<td>170$_3$</td>
</tr>
<tr>
<td>8</td>
<td>890$_{33}$</td>
<td>93384$_{4394}$</td>
<td>0.65$_{0.0102}$</td>
<td>0.062$_{0.0011}$</td>
<td>194$_2$</td>
</tr>
<tr>
<td>10</td>
<td>1,470$_{34}$</td>
<td>162139$_{4592}$</td>
<td>0.53$_{0.0070}$</td>
<td>0.050$_{0.0006}$</td>
<td>206$_1$</td>
</tr>
<tr>
<td>12</td>
<td>2,254$_{32}$</td>
<td>227912$_{2670}$</td>
<td>0.44$_{0.0031}$</td>
<td>0.043$_{0.0003}$</td>
<td>207$_1$</td>
</tr>
<tr>
<td>14</td>
<td>3,264$_{29}$</td>
<td>290732$_{2056}$</td>
<td>0.38$_{0.0022}$</td>
<td>0.040$_{0.0003}$</td>
<td>203$_1$</td>
</tr>
<tr>
<td>15</td>
<td>3,868$_{33}$</td>
<td>321203$_{2061}$</td>
<td>0.35$_{0.0022}$</td>
<td>0.039$_{0.0004}$</td>
<td>200$_1$</td>
</tr>
</tbody>
</table>

- **Clustering Coefficient**: For high $K$, transition between a given pair of neighboring basins is less likely to occur.
- **Disparity**: For high $K$ the transitions to other basins tend to become equally likely, an indication of the randomness of the landscape.
Weights distribution: transition probability between basins

- Weights are small
- For high $K$ the decay is faster
- Low $K$ has longer tails
- On average, the transition probabilities are higher for low $K$ (less local optima)

Distribution of the network weights $w_{ij}$ for outgoing edges with $j \neq i$ in log-x scale, $N = 18$
Weight distribution remain in the same basin

Question:
Is it easy to escape a basin?

- Weights to remains in the same are large compare to \( w_{ij} \) with \( i \neq j \)
- \( w_{jj} \) are higher for low \( K \)
- Easier to leave the basin for high \( K \) : high "natural" exploration
- But : number of local optima increases fast with \( K \)
Interior and border size

**Question:**
Do basins look like a "montain" with interior and border?

Solution is in the interior if all neighbors are in the same basin.

Average of the mean size of basins interiors
Interior and border size

Average of the mean size of basins interiors

Question:

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

solution is in the interior if all neighbors are in the same basin

Answer:

- Interior is very small
- Nearly all solution are in the border
Shortest path length between local optima

![Graph showing the average path length between nodes for different values of N and K.](image)

Question:
Are the basins "far" from each other?

- Increase with N (number of nodes increases exponentially)
- For a given N, increase with K up to $K = 10$, then stagnates
Shortest path length to global optima

Average path length to the global optimum from all the other basins

Question:
Is the global optimum basin is far?

- More relevant for optimisation
- Increase steadily with increasing $K$
Summary on local optima network

- Medium level of description: proposed characterization of combinatorial landscapes as networks
- A new model for landscape analysis
- New findings about basin’s structure: sizes, fitness vs. size, etc.
- Related some network features to search difficulty
Summary on local optima network

- Related some *network features* to *search difficulty*

- Weights edges of NK-landscapes LON
- Effective frequencies of TS and ILS from one basin to another

Wednesday, July 11th, 14:32 "Local Optima Networks and the Performances of ILS", F. Daolio *et al.*
Continuous problems

Fitness landscape – FiL \((X, N, f)\)

- \(X = \mathbb{R}^n\)
- \(N\) defined by gaussian mutation
- \(f : \mathbb{R}^n \rightarrow \mathbb{R}\)

Continuous fitness landscapes

Main difficulty

- Which step size to define the neighborhood? $10^{-3}$?
  - Different step size, different landscape?
  - Use the standard definition of LO?
    $\exists \epsilon > 0, \forall x, |x - x^*| \leq \epsilon \Rightarrow f(x) \leq f(x^*)$
- Toward measure like $\|f(x) - f(y)\| / \|x - y\|$?
- But is it better than derivate function?
Continuous fitness landscapes

Main difficulty

- Which step size to define the neighborhood? $10^{-3}$?
- Different step size, different landscape?
- Use the standard definition of LO?
  $\exists \epsilon > 0, \forall x, |x - x^*| \leq \epsilon \Rightarrow f(x) \leq f(x^*)$
- Toward measure like $\|f(x) - f(y)\|/\|x - y\|$?
- But is it better than derivate function?

Personal feeling

Main different between combinatorial and continuous FiL

- No search "direction" in combinatorial problem
- Probability to improve the current solution
- Shape of the attraction basins
Multiobjective optimization

What is the structure of a multiobjective search space?

Have you seen the multiobjective tutorial by Dimo Brockhoff?
Multiobjective optimization

Multiobjective optimization problem

- $\mathcal{X}$: set of feasible solutions in the decision space
- $M \geq 2$ objective functions $f = (f_1, f_2, \ldots, f_M)$ (to maximize)
- $\mathcal{Z} = f(\mathcal{X}) \subseteq \mathbb{R}^M$: set of feasible outcome vectors in the objective space
Definition

Pareto dominance relation

A solution \( x \in \mathcal{X} \) dominates a solution \( x' \in \mathcal{X} \) \( (x' \prec x) \) iff

1. \( \forall i \in \{1, 2, \ldots, M\}, f_i(x) \geq f_i(x') \)
2. \( \exists j \in \{1, 2, \ldots, M\} \) such that \( f_j(x) > f_j(x') \)
Pareto set, Pareto front

- **Decision space**
  - $x_1$
  - $x_2$

- **Objective space**
  - $f_1$
  - $f_2$

- **Pareto optimal solution**
- **Pareto optimal set**
- **Pareto front**
Set-based multiobjective fitness landscape

Definition

A set-based multiobjective fitness landscape is defined as a triplet \((\Sigma, N, I)\) such that:

- \(\Sigma \subset 2^X\) is a set of feasible solution-sets (where \(X\) is the set of feasible solutions)
- \(N : \Sigma \rightarrow 2^\Sigma\) is a neighborhood relation between solution-sets
- \(I : \Sigma \rightarrow \mathbb{R}\) is a unary quality indicator, i.e. a fitness function measuring the quality of solution-sets

State-of-the-art tools from single-objective FiL can be reused
Set-based search space

### Illustrative examples

- **Population-based approaches**
  \[ \Sigma = \{ \sigma \in 2^X : |\sigma| = \mu \}, \text{ where } \mu \text{ is the population size} \]

- **Bounded archive-based approaches**
  \[ \Sigma = \{ \sigma \in 2^X : |\sigma| \leq \mu \}, \text{ where } \mu \text{ is the max archive size} \]

- **Dominance-based approaches** (with mutually *n-d* solutions)
  \[ \Sigma = \{ \sigma \in 2^X : \forall s, s' \in \sigma, s \not\prec s' \} \]

- **Bounded archive- + Dominance-based approaches**
  \[ \Sigma = \{ \sigma \in 2^X : |\sigma| \leq \mu \text{ and } \forall s, s' \in \sigma, s \not\prec s' \} \]

- **No restriction**
  \[ \Sigma = 2^X \]

...
Set-based neighborhood relation

Illustrative examples

- **Set-level replacement neighborhood**
  \[ \sigma' = \sigma \cup \{s'\} \setminus \{s\} \text{ such that } s \in \sigma, \ s' \in \mathcal{N}(s) \]
  \[ \rightarrow |\mathcal{N}(\sigma)| \leq |\sigma| \cdot \sum_{s \in \sigma} |\mathcal{N}(s)| \]

- **Set-level insertion neighborhood**
  \[ \sigma' = \sigma \cup \{s'\} \text{ such that } s \in \sigma, \ s' \in \mathcal{N}(s) \]
  \[ \rightarrow |\mathcal{N}(\sigma)| \leq \sum_{s \in \sigma} |\mathcal{N}(s)| \]

- **Set-level deletion neighborhood**
  \[ \sigma' = \sigma \setminus \{s\} \text{ such that } s \in \sigma \]
  \[ \rightarrow |\mathcal{N}(\sigma)| = |\sigma| \]

- \ldots (recombination)
Set-based fitness function

Illustrative example: **hypervolume** ($I_H$)

- Compliant with the (weak) Pareto dominance relation
  \[ A \prec B \Rightarrow I_H(A) \leq I_H(B) \]
- A single parameter: the reference point
- Minimal solution-set maximizing $I_H$
  \[ \text{arg max}_{\sigma \in \Sigma} I_H(\sigma) \]

\[ \Rightarrow \text{subset of the Pareto optimal set} \]
Main Fitness Landscape measures can be used

**Ruggedness, autocorrelation**

→ Based on *random walk* sampling

- Is the fitness of neighboring solutions random?

Autocorrelation length $\tau = \frac{1}{\rho}$, with $\rho$ correlation $(f(s), f(s'))$, $s' \in \mathcal{N}(s)$

small $\tau$: *rugged* landscape  |  large $\tau$: *smooth* landscape

**Local optima**

→ Based on *adaptive walk* sampling

- What is the number of local optima, and the size of basins?

$(s_0, s_1, \ldots, s_L)$ where $s_{i+1} \in \mathcal{N}(s_i)$, $f(s_i) < f(s_{i+1})$

small $L$: high multi-modality  |  large $L$: low multi-modality

(many local optima, small basin size)  |  (few local optima, large basin size)
What is the structure of the search space?

Facts:
- Very large search space
- Very large neighborhood
- High computational cost to compute the fitness
What is the structure of the search space?

**Facts**
- Very large search space
- Very large neighborhood
- High computational cost to compute the fitness

**Open questions**
(combinatorial) multiobjective problems:
- Escaping from local optima?
- Exploring the neighborhood is an optimization problem?
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 neighbor solutions
  Ruggedness, local optima, fitness cloud, neutral networks, local optima networks...
- and to deduce **global features**:
  - Difficulty!
  - To decide (design, tune or control) a good choice of the representation, operator and fitness function
Open questions

- How to control the parameters and/or operators of the algorithm with the local description of fitness landscape?
- Can fitness landscape describe the dynamics of a population of solutions?
- Links between neutrality and fitness difficulty?
- Which intermediate description shows relevant properties of the optimization problem according to the local search heuristic?

Integration of the FL tools into the open framework *paradisEO*

http://paradiseo.gforge.inria.fr
L. Barnett.
Ruggedness and neutrality - the NKp family of fitness landscapes.
In C. Adami, R. K. Belew, H. Kitano, and C. Taylor, editors, 

Lionel Barnett.
Netcrawling - optimal evolutionary search with neutral networks.

Statistical properties of neutral evolution.
*Journal Molecular Evolution*, 57(S):103–119, August 2003.
Meriem Belaidouni and Jin-Kao Hao.
An analysis of the configuration space of the maximal constraint satisfaction problem.

P. Collard, M. Clergue, and M. Defoin Platel.
Synthetic neutrality for artificial evolution.

J. C. Culberson.
Mutation-crossover isomorphisms and the construction of discrimination function.
J. P. K. Doye.
The network topology of a potential energy landscape: a static scale-free network.

Characterizing the network topology of the energy landscapes of atomic clusters.

Ricardo Garcia-Pelayo and Peter F. Stadler.
Correlation length, isotropy, and meta-stable states.
Santa Fe Institute Preprint 96-05-034.

Josselin Garnier and Leila Kallel.
Efficiency of local search with multiple local optima.
P. Gitchoff and G. Wagner.

David E. Goldberg and Philip Segrest.
Finite markov chain analysis of genetic algorithms.

M. Huynen.
Exploring phenotype space through neutral evolution.

E. Izquierdo-Torres.
The role of nearly neutral mutations in the evolution of dynamical neural networks.
T. Jones.  
*Evolutionary Algorithms, Fitness Landscapes and Search.*  

S. A. Kauffman.  
*The Origins of Order.*  

M. Kimura.  
*The Neutral Theory of Molecular Evolution.*  

J. Lobo, J. H. Miller, and W. Fontana.  

M. Newman and R. Engelhardt.  
Effect of neutral selection on the evolution of molecular species.
Erik Van Nimwegen, James P. Crutchfield, and Martijn Huynen.
Metastable evolutionary dynamics: Crossing fitness barriers or escaping via neutral paths?

Gabriela Ochoa, Marco Tomassini, Sébastien Verel, and Christian Darabos.
A Study of NK Landscapes’ Basins and Local Optima Networks.
best paper nomination.
M. Defoin Platel.
*Homologie en Programmation Génétique - Application à la résolution d’un problème inverse.*

Eduardo Rodriguez-Tello, Jin-Kao Hao, and Jose Torres-Jimenez.
A new evaluation function for the minla problem.

Helge Rosé, Werner Ebeling, and Torsten Asselmeyer.
The density of states - a measure of the difficulty of optimisation problems.

From sequences to shapes and back: a case study in RNA secondary structures.

**Peter F. Stadler.**
Landscapes and their correlation functions.

**Peter F. Stadler and W. Schnabl.**
The landscape of the traveling salesmen problem.

**Peter F. Stadler and Gunter P. Wagner.**
Algebraic theory of recombination spaces.

**Terry Stewart.**
Extrema selection: Accelerated evolution on neutral networks.


tea team.

**E. D. Weinberger.**
Correlated and uncorrelated fitness landscapes and how to tell the difference.

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