Local Search and Combinatorial Optimization: From the Structural Analysis of a Problem to the Design of Efficient Algorithms

– – –

Recherche locale et optimisation combinatoire : de l'analyse structurelle d'un problème à la conception d'algorithmes efficaces

Marie-Éléonore Marmion

Advisors: Pr. Clarisse Dhaenens & Pr. Laetitia Jourdan

DOLPHIN Team

Friday, December 9th 2011

Université Lille 1 Sciences et Technologies

Laboratoire d'Informatique Fondamentale de Lille

cnrs

inria informatiques mathématiques

Background
Combinatorial Optimization Problem (COP)

Ω is a **discrete set of solutions**

$f : \Omega \longrightarrow \mathbb{R}$ is the **objective function**

- Minimization problem

Goal : find $s^* \in \Omega$ such as

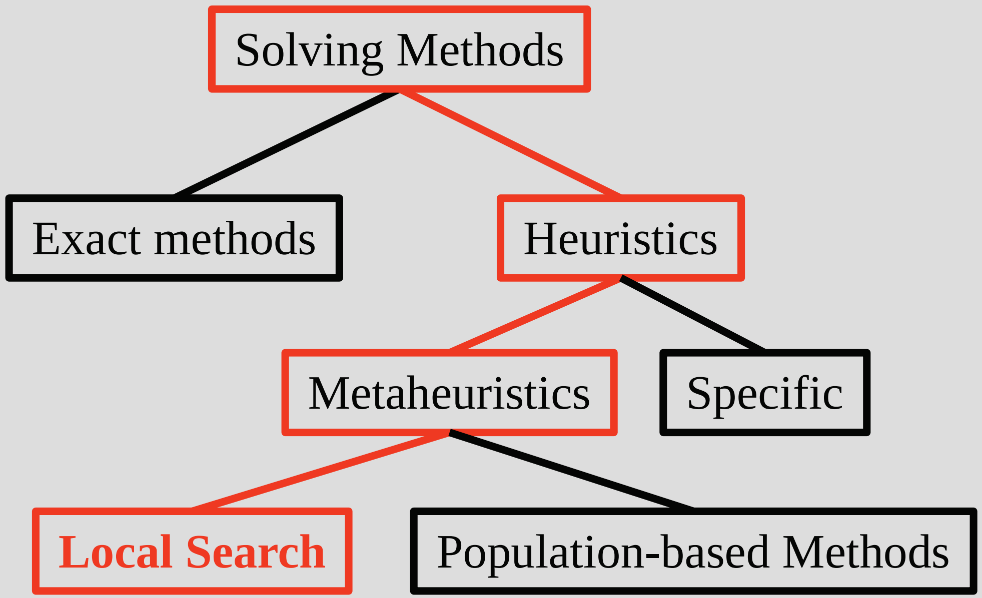
$$s^* = \texttt{argmin}_{s \in \Omega} \{f(s)\}$$

- $s^* \in \Omega$ is defined as **global optimum** iff

$$\forall s \in \Omega, f(s^*) \leq f(s)$$

Background
Solving Methods

- Exact methods
  - Optimal solution
  - Exponential complexity for NP-hard problems
- Heuristic methods
  - Good-quality solution
  - Reasonable time



Metaheuristics : **generic** methods

- **Local Search** (Hill Climbing, Tabu Search, Simulated Annealing...)
- Population-based methods (Genetic Algorithm, Ant Colony...)

Background

Designing Metaheuristics for NP-hard Optimization

Modelling the problem

- Solution representation
- Objective function(s)

Choosing the solving method

- Local search?
- Population-based method?

Setting the parameters

- Neighborhood relation
- Population size
- Tabu list size
- Mutation rate
- ...

Motivation

Designing metaheuristics for NP-hard optimization

3 main issues:

- How to **model** the problem?
- How to **choose** an efficient algorithm?
- How to **set** all parameters?

Needs:

Link between the **dynamics of metaheuristics**
and the underlying **structure of the problem**

→ Fitness Landscape Analysis

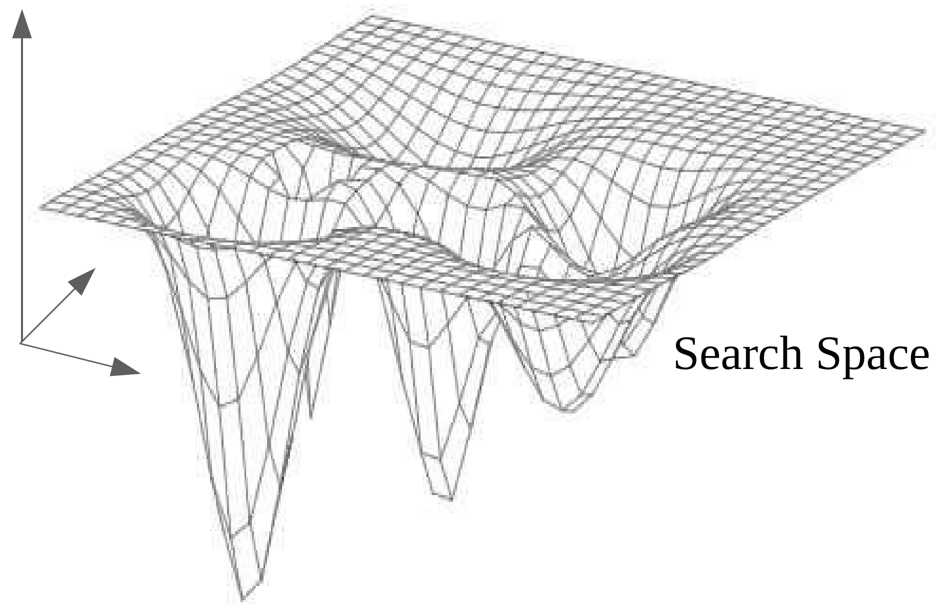
Background
Fitness Landscape

$(\Omega, \mathcal{N}, f)$ [Wright, 1932]

- Ω: **search space**
- $\mathcal{N}$: the **neighborhood function** connects solutions

  → 1 application of an operator

- f : the **objective function** gives the solution quality (fitness value)

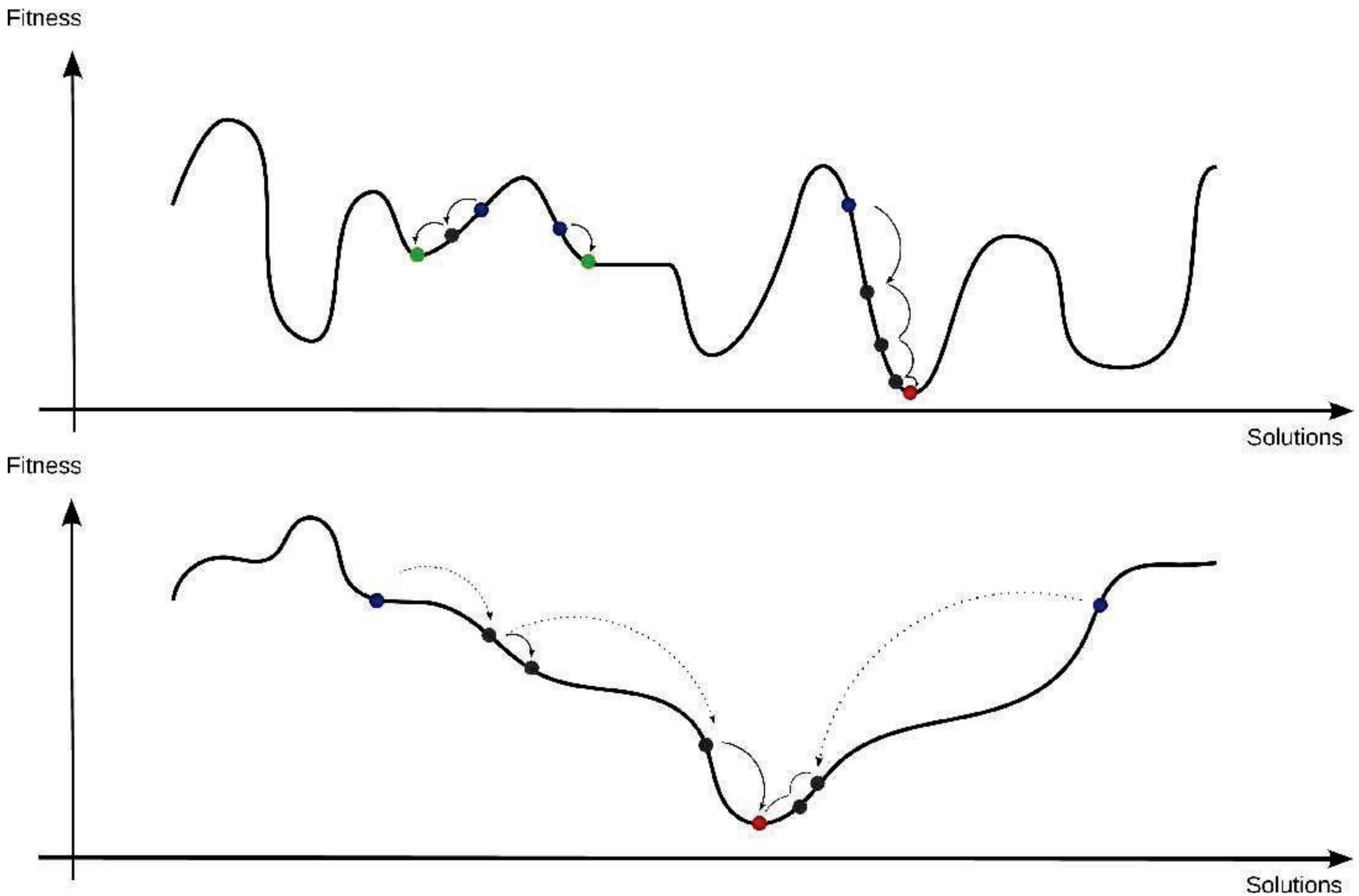


S. Wright, **The roles of mutation, inbreeding, crossbreeding and selection in evolution**, in *Proceedings of the Sixth International Congress on Genetics*, Vol. 1, 1932

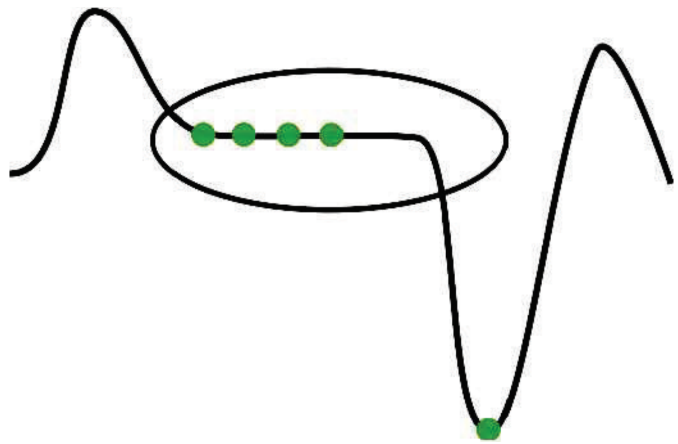
Background
Neighborhood Function

+ 1 Problem
  2 Neighborhood functions

} **2 Landscapes**



Observation & Questions

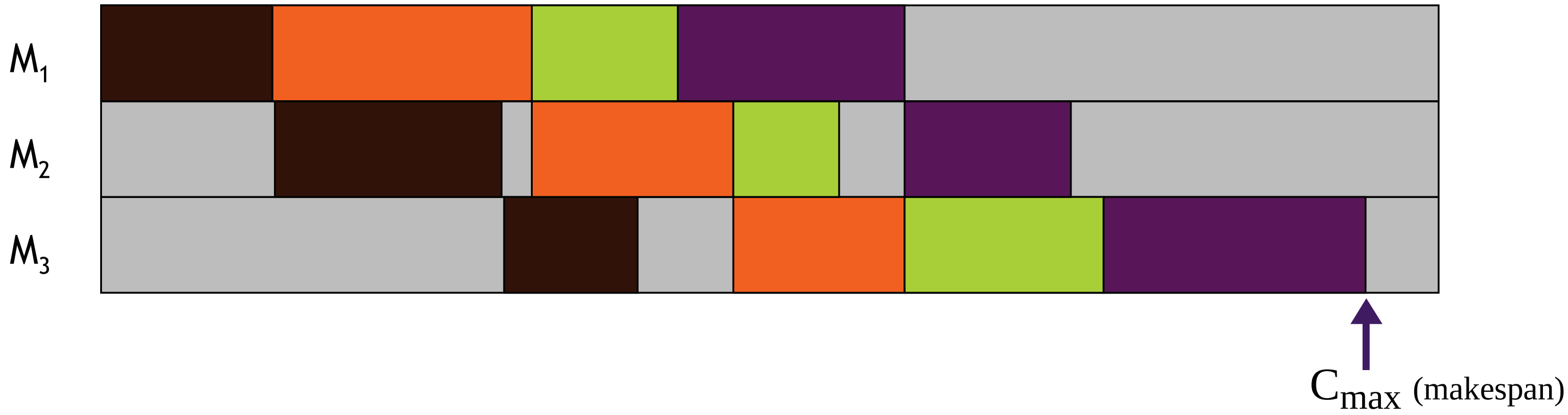


Many solutions have the **same fitness** values:

- Numerous?
- Neighbors?
- Local optima?

→ **Fitness landscape** analysis with **neutrality** consideration

Case study
Flowshop Scheduling Problem

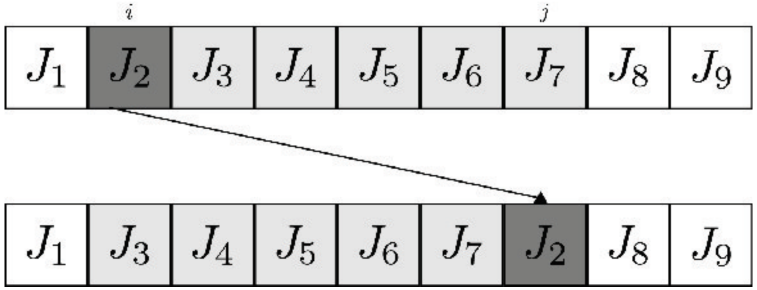


- N jobs - M machines
- Processing time of each job can be different on each machine
- Each job can be processed on at most one machine
- Each machine can process at most one job at a time
- Job order is the same on every machine: Representation = Permutation

→ Makespan minimization

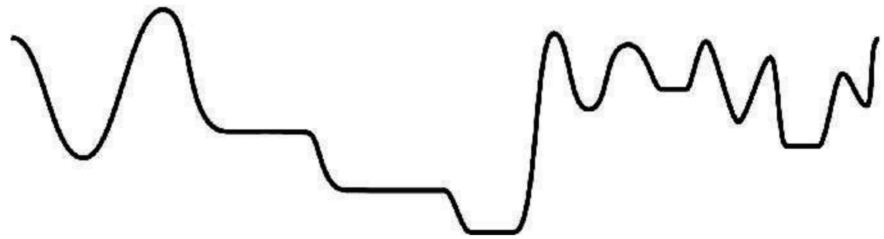
Flowshop Scheduling Problem
Neighborhood Operators

- **Insertion** (IN) operator

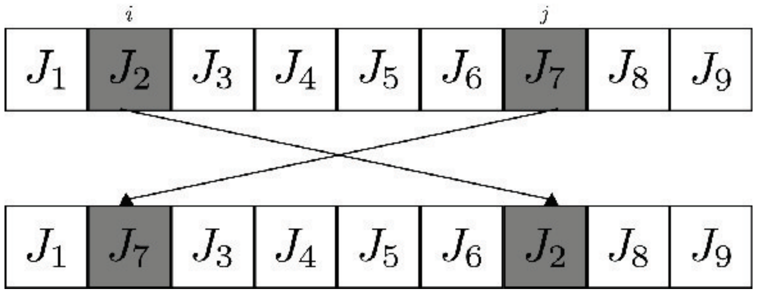


Neighborhood size: $(N-1)^2$

$(\Omega, \mathcal{N}_{\mathtt{IN}}, f)$: IN-FiL

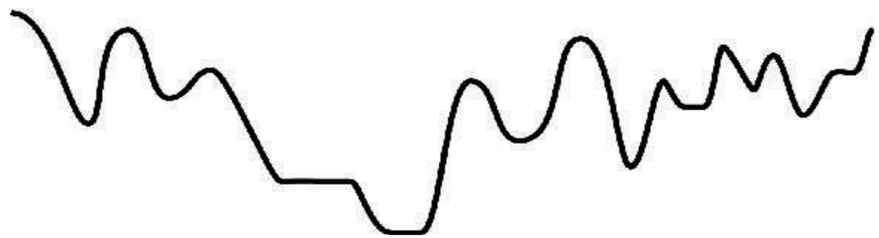


- **Exchange** (EX) operator



Neighborhood size: N (N-1) / 2

$(\Omega, \mathcal{N}_{\mathtt{EX}}, f)$: EX-FiL



Flowshop Scheduling Problem
Instances

Taillard Instances [Taillard, 1993]

- Jobs = 20, 50, 100, 200, 500
- Machines = 5, 10, 20
- Processing times are uniformly distributed in [1;99]

Structured Instances [Watson et al., 2002]

- Jobs = 20, 50, 100, 200
- Machines = 20
- Processing times are :
  - Job-correlated (jc)
  - Machine-correlated (mc)
  - Job/Machine-correlated (mxc)

How do the instance characteristics act on:

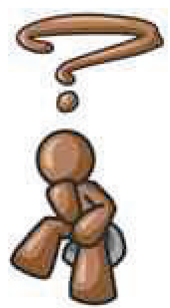
- the landscape **structure**?
- the metaheuristics **performance**?
- the **easiness** to find the global optimum?

→ Generic Approach

Flowshop Scheduling Problem
State of the Art

Highly studied in literature

- Exact methods: branch and bound [Ríos-Mercado et al., 1999]
- Building heuristics [Nawaz et al., 1983]
- **Local search** [Ruiz et al., 2007]
- **Population-based method** [Stützle, 1997 ; Ruiz et al., 2006]



Questions & Observations

- Many metaheuristics are used:
  - Why are they **efficient**? Is this efficiency **predictable**?
  - Can parameters be easily set?
- Many solutions with the **same fitness value**:
  - Is this property used to solve FSP?

Outline

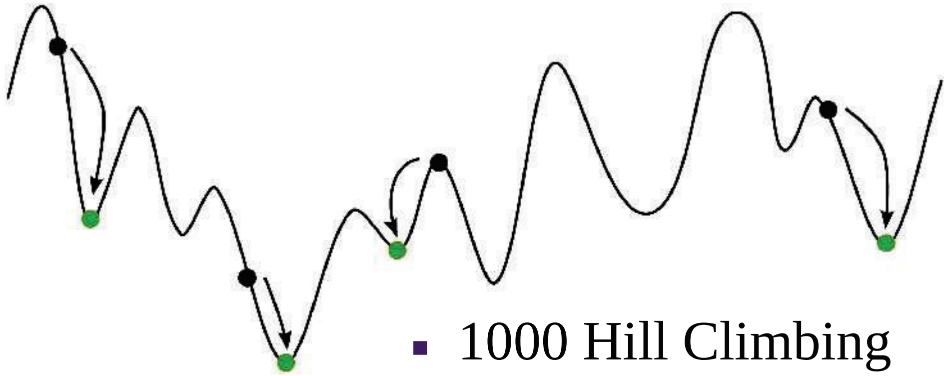
How can a landscape analysis help the design of efficient algorithms?

- Influence of the landscape on the performance of metaheuristics
- Characterization of neutral landscapes
- Design efficient local search methods with neutrality consideration

Experimental Design

2 Landscapes:

- IN-FiL (*Insertion* operator)
- EX-FiL (*Exchange* operator)



- 1000 Hill Climbing
- Taillard Instances

Classical indicators:

- Width and depth
- LO Quality
- Ruggedness

Metaheuristics:

[den Besten et al., 2001 ; Murata et al., 1996]

- Iterative Hill Climbing (ILS)
- Simulated Annealing
- Tabu Search
- Genetic Algorithm

Landscape and Performance Results

Indicators	IN-FiL	EX-FiL
Global Width (Average Distance between solutions)	–	+
Local Width (Average Distance between LO)	–	+
Depth (Average Step Length to find a LO)	+	–
Local Optima Quality (Average fitness values of LO)	+	–
Local Ruggedness (Average Autocorrelation Length)	–	+

Literature:

- $LS_{IN\text{-}FiL} \gg LS_{EX\text{-}FiL}$
- $GA_{IN\text{-}FiL} \gg GA_{EX\text{-}FiL}$
- $LS_{IN\text{-}FiL} \gg GA_{IN\text{-}FiL}$

INSERTION >> EXCHANGE

– Global Width
– Depth
– Local Optima Quality
– Local Ruggedness

IN-FiL favors more LS and GA, compared to EX-FiL

– Local Width
– Local Ruggedness

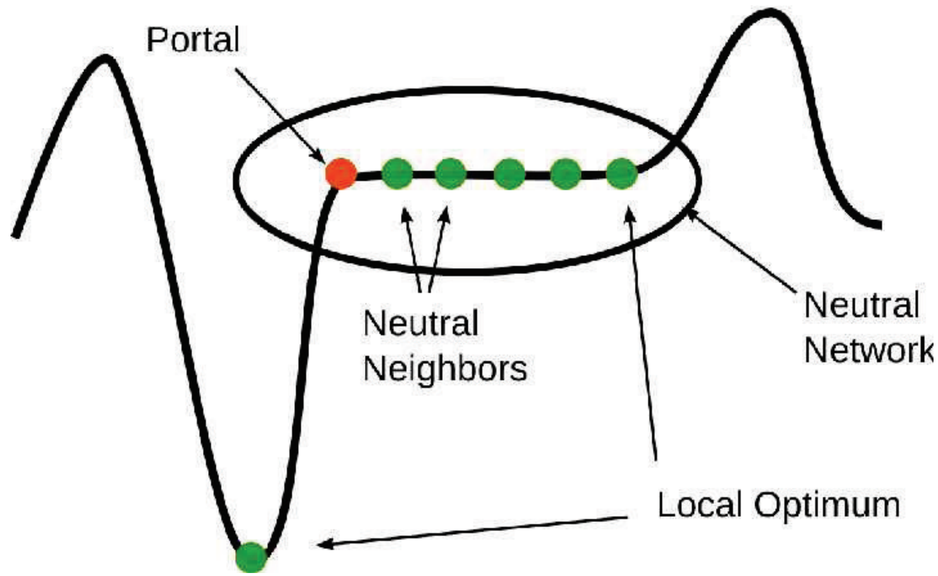
IN-FiL favors more LS than GA

Outline

How can landscape analysis help the design of efficient algorithms?

- Influence of the landscape on performance of metaheuristics
- Characterization of neutral landscapes
- Design efficient local search methods with neutrality consideration

Neutrality in Landscapes



- **Local optimum** (s*):
  iff no neighbor has a better fitness value

- **Neutral neighbor** of solution s:
  solution s' with the same fitness value

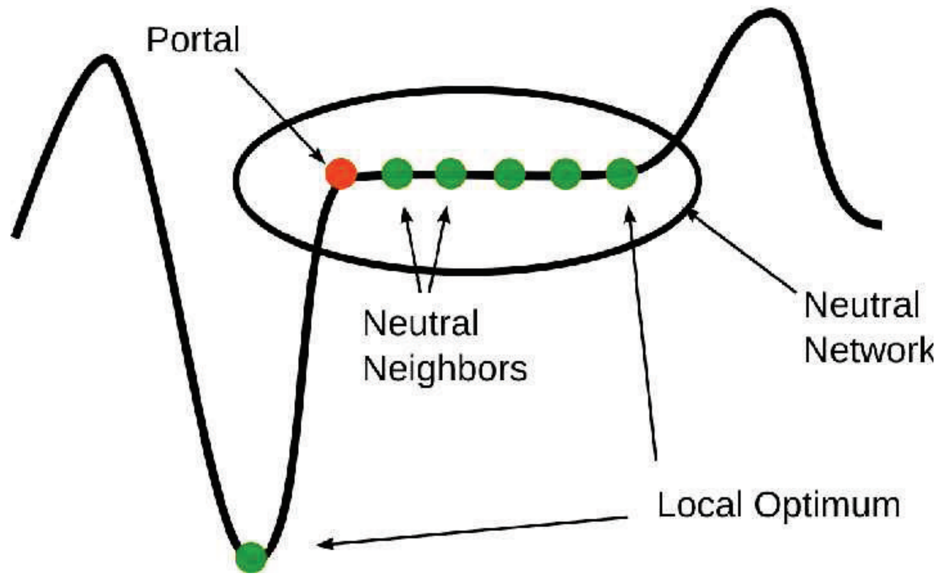
- **Neutral degree** of s: the number of its neutral neighbors

- **Neutral network (NN)**: connected sub-graph whose vertices are solutions with the same fitness value. Two vertices are connected if they are neutral neighbors.

- **Portal** in a NN: solution whose at least one neighbor has a better fitness value.

Neutrality in Landscapes
Questions

- Do solutions have a lot of **neutral neighbors**?
- Are the **neutral networks** (NN) large?
- Are there a lot of **portals** on a NN?
- Is it difficult to **reach a portal**?



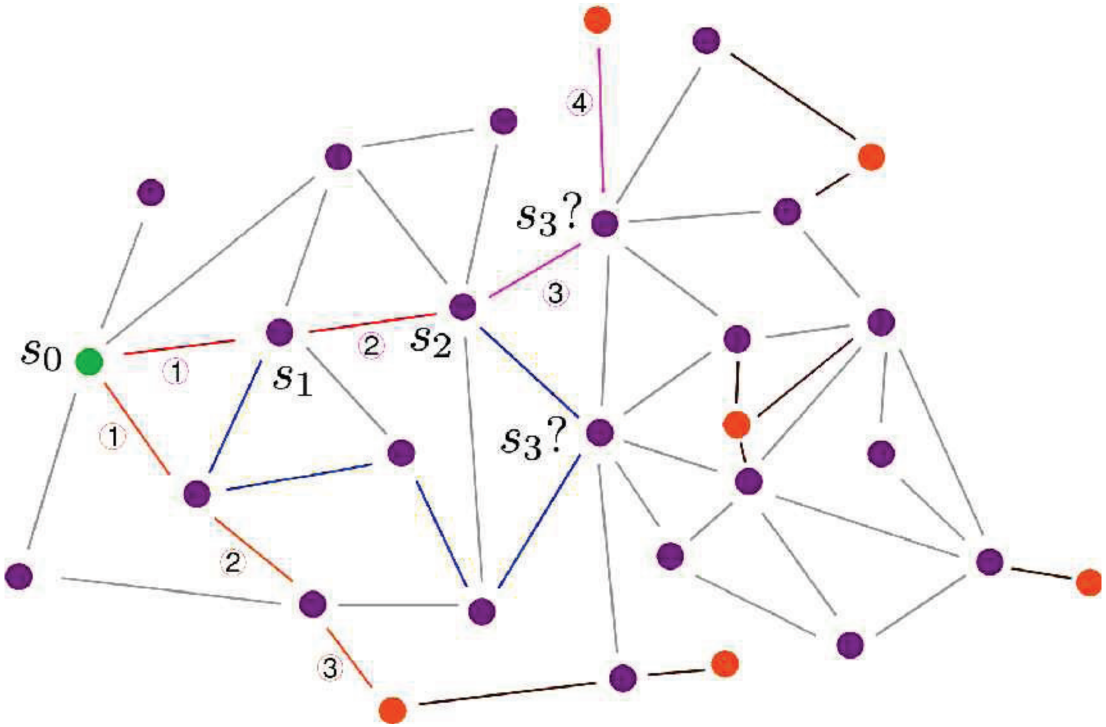
Neutrality in Landscapes
Neutral Networks Analysis

Neutral Walk

$$W_{neut} = (s_0, s_1, \ldots, s_m)$$

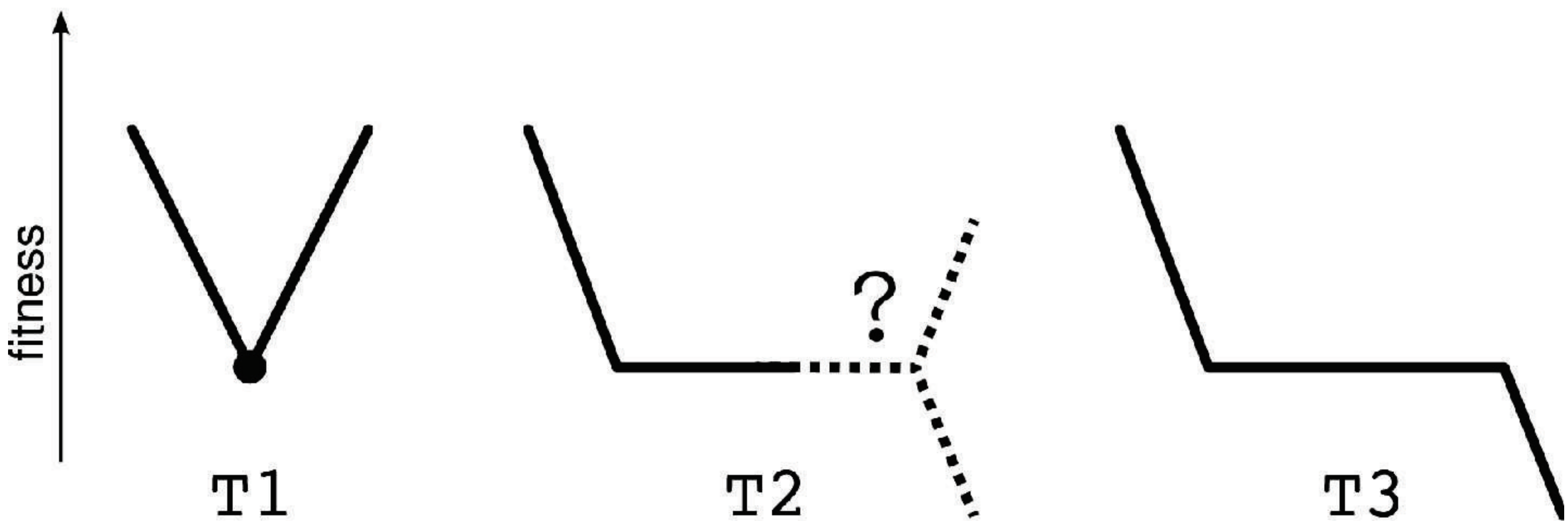
→ Sample NN

$$s_{i+1} \in \mathcal{N}(s_i)$$
$$f(s_{i+1}) = f(s_i)$$



- Neutral degree
- Ratio of the neutral degree
- Number of visited neutral solutions before finding a portal
- Portal presence on NN

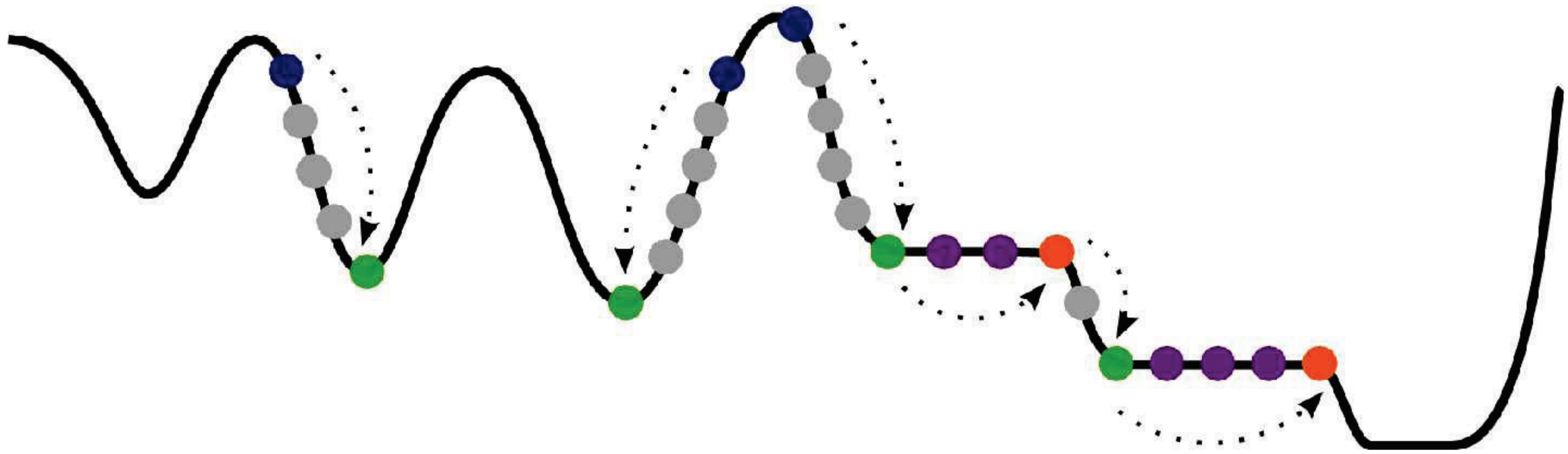
Neutrality in Landscapes
Neutral Networks Typology



M.-É. Marmion, L. Jourdan, C. Dhaenens, A. Liefooghe, S. Verel, On the Neutrality of Flowshop Scheduling Fitness Landscapes,
In LION 2011, LNCS, Springer, pages 238–252, 2011

Neutrality in Landscapes

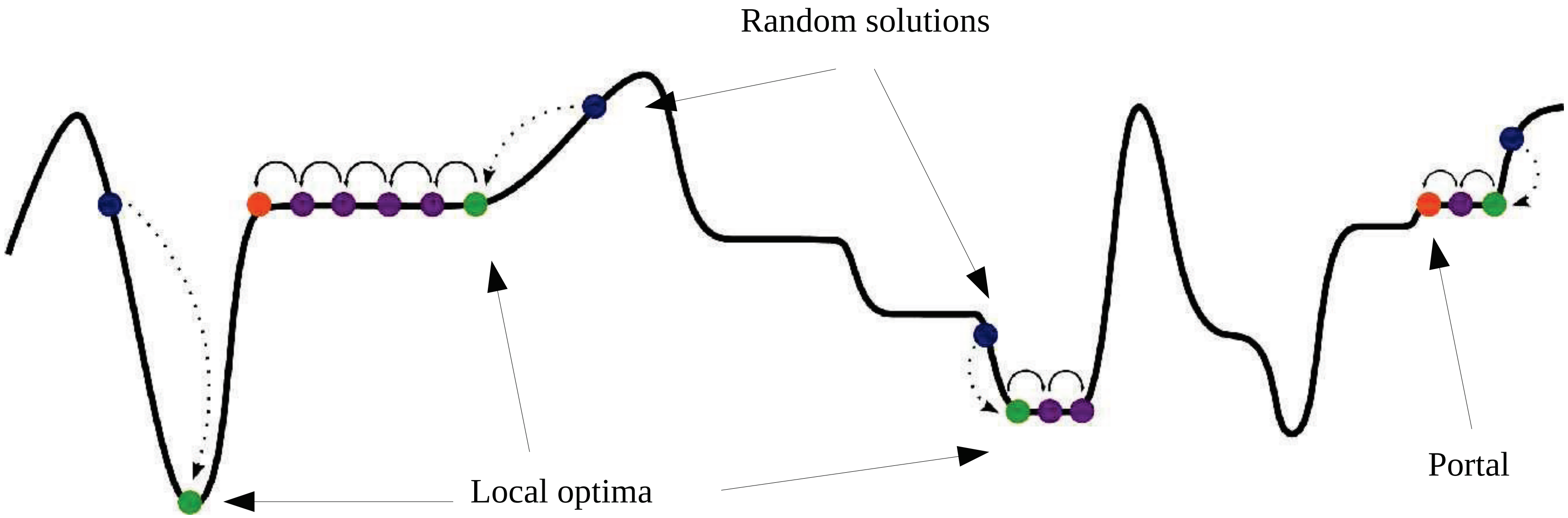
Neutral Networks exploration vs. Restart



- Is it **easy** and **fast** to find a portal?
- Do portals lead to solutions with **better quality**?

Neutrality in Landscapes
Experimental Design

Neutral Walks



Parameters:

- Neighborhood: Insertion operator
- Length according to the instance size

Instances:

- Taillard (11)
- Structured (12)

Run:

- 30 x #instances

Results

→ Taillard Instances

Neutral degree (ratio):

- Increases with # jobs
- Decreases with # machines
- Between 1 and 30%

Neutral networks:

- Large
- Almost no T1, T2
- T3: Portals are reached by random neutral walks

Observations & Conclusion:

- The lower the neutral degree, the faster the portals are found
- Faster to reach a portal with neutral walk than to reach a LO with Hill Climbing

→ Neutral solutions have to be considered

Neutrality in Landscapes
Results
→ Structured Instances

Neutral degree (ratio):

- Increases with correlation degree
- jc: medium neutrality
- mc, mxc: greater than 90%

Neutral networks:

- Large
- Global Optimum NN (mc, mxc)
- T3: Portals are reached by random neutral walks (jc)

Observations & Conclusion:

- Hill Climbing reaches Global Optimum NN easily
- Portals are reached with neutral walks quickly

→ Neutral solutions have to be considered

Outline

How can landscape analysis help the design of efficient algorithms?

- Influence of the landscape on metaheuristics performance
- Characterization of neutral landscapes
- Design efficient local search methods with neutrality consideration

Neutrality in Designing Local Search (1)

- Neutral degree of LO
- Number of NN of type T3
- Moving on NN vs Restart

Exploit LO Neutrality

+

Netcrawler [Barnett, 2001]: local search accepts the first neighbor with a **better or equal** fitness value

→ Local search that exploits neutrality from LO to continue the search

NILS
Algorithm

Neutrality-based Iterated Local Search

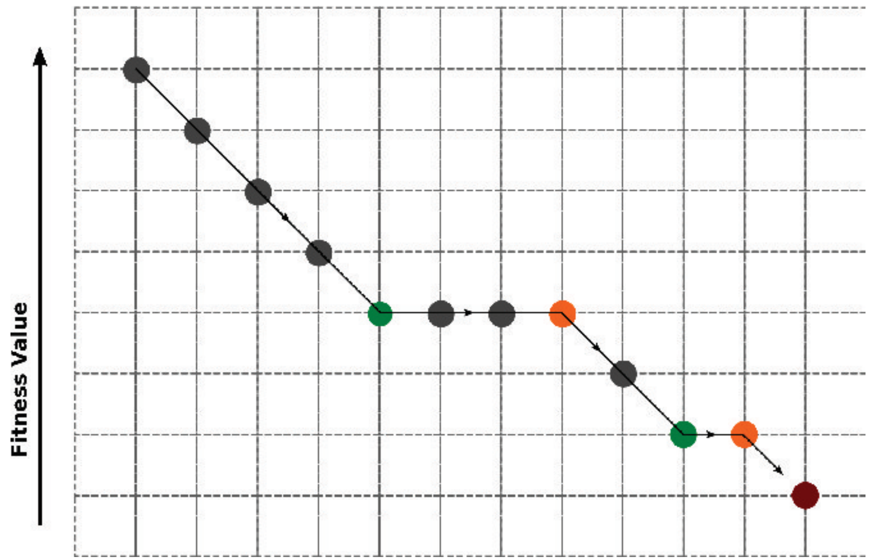
- Local search
  - First-Improvement Hill Climbing (FIHC)
  - **Stops on a LO**
- Perturbation
  - Escaping from local optimum
    - **Neutral moves** (exploitation of the neutral property)
    - Random moves (when portals are difficult to reach)

Trade-off exploitation / exploration

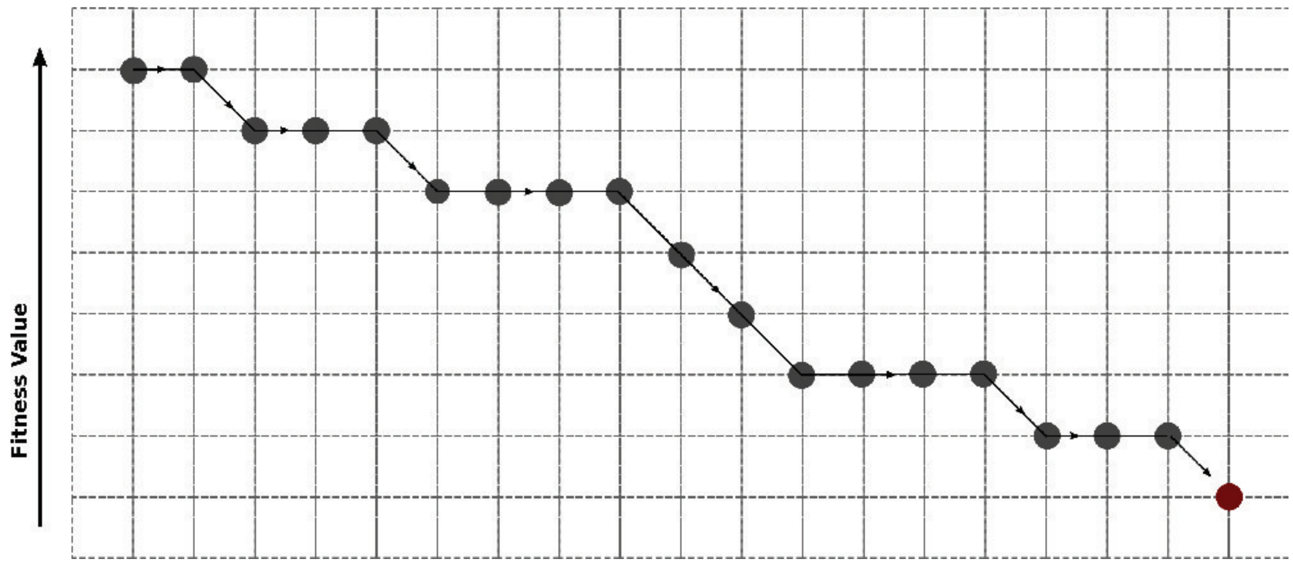
- **Parameter MNS** – Maximal Number of (neutral) Steps

NILS Dynamics

(NILS)



(NC)



NILS
Experimental Design

Stopping criterion:

- Number of evaluations (2.10^7)

Parameters:

- Neighborhood: Insertion operator
- Restart: 3 random Exchange-moves
- MNS values according to the instance size

Instances:

- Taillard (11)
- Structured (12)

Literature:

- Netcrawler (NC)
- Iterated Greedy (IG) [Ruiz and Stützle, 2007]

Run:

- 30 x #instances x #MNS

Validation with statistical tests:

- Student t-test
- Wilcoxon signed rank test

NILS
Results
→ Taillard Instances

NILS Performance:

- Better performance with **large MNS value** (not too high)
- **Best-known is found**: 6 instances
- Deviation to the best-known **under 1.22%**: other instances

Landscape Analysis:

- Many T3
- Portals on T3 are visited easily by random neutral walks

NILS vs. Literature:

- **Comparable performance** with NC and IG

NILS
Results
→ Structured Instances

NILS Performance:

- Better performance with **large MNS value** (not too high)
- **Best-known always found:**
  11 instances
- **New Best-known found**:
  1 instance

Landscape Analysis:

- Global Optimum NN is reached easily by HC
- Portals on T3 are visited easily by random neutral walks

NILS vs. ILS:

- **Higher performance**: best-known is always found

NILS
Discussion

NILS is:

- **Easy** to implement
- **Efficient** on neutral problems
- **Comparable performance** against literature algorithms

→ For **neutral problems**,
NILS seems to be **appropriate**

Question:

- Why only the last accepted solution of the NN is considered?

Neutrality in Designing Local Search (2)

- Neutral degree of LO
- Number of NN of type T3
- Moving on NN vs Restart

Exploit
Neutral Networks of LO

+

Guide the search to select the solution in the NN that leads faster to a portal

+

Evolvability [Altenberg, 1994] : the ability of random variations to sometimes produce improvement

→ Local search:

- Considering several solutions
- Guided over NN with respect to evolvability

Evolvability Measure

Average fitness value
in the neighborhood:

$$E(s) = \frac{1}{|\mathcal{N}(s)|} \sum_{s_i \in \mathcal{N}(s)} f(s_i)$$

Goal: find a portal by visiting few solutions

anti-correlation

Average fitness value of the neighborhood: high

Experimental Results:
(Taillard instances)

- Evolvability: **not random** between neighbors
- **Anti-correlation** between *# solutions to visit on NN* and *evolvability value*

Evolvability is better
when
Evolvability value is higher

Machine Learning
Multi-Armed Bandit

- Multi-Armed Bandit:
  Trade-off between **exploration** and **exploitation**

- Upper Confidence Bound (UCB)

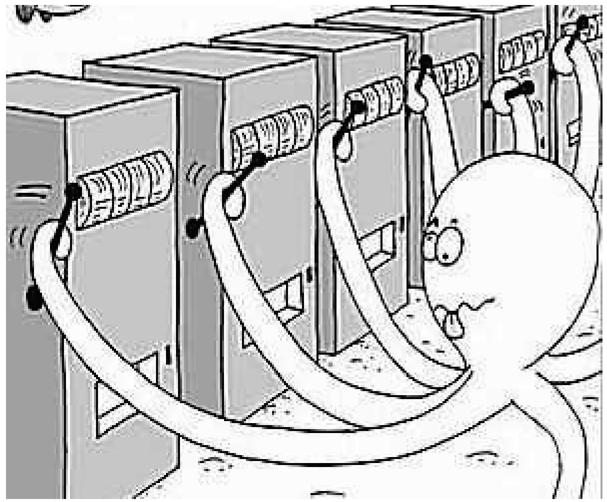
$$\arg\max_{i=1..A} \left(\hat{r}_{i,t} + C \sqrt{\frac{\log \sum_k n_{k,t}}{n_{i,t}}} \right)$$

A : Number of Arms

n_i : Number of Applications of the Arms i

$\hat{r}_i$: Credit Assignment

C : Trade-off parameter



Álvaro Fialho, Marc Schoenauer and Michele Sebag. ***Toward Comparison-based Adaptive Operator Selection****.*
GECCO'10, ACM Press : p. 767-774. July 2010

VEGAS Algorithm

Varying Evolvability-Guided Adaptive Search

$$\arg\max_{i=1..A}\left(\hat{r}_{i,t} + C\sqrt{\frac{\log \sum_k n_{k,t}}{n_{i,t}}}\right) \qquad \text{(UCB)}$$

A	Arms	⟷	Known solutions of the current NN NN too large: How to limit the number of arms? → size P
n_i	Applications of the arm i	⟷	1 neighbor evaluation
$\hat{r}_i$	Credit Assignment	⟷	Evolvability

VEGAS
Algorithm

Varying Evolvability-Guided Adaptive Search

- Local search
  - First-Improvement Hill Climbing (FIHC)
  - **Stops on a LO**
- Perturbation:
  - Escaping from LO : set $\mathcal{S}$ of the whole NN
    - Selecting s in $\mathcal{S}$ to evaluate a neighbor with **UCB**
    - Updating **Credit Assignment** (evolvability)
    - Updating $\mathcal{S}$

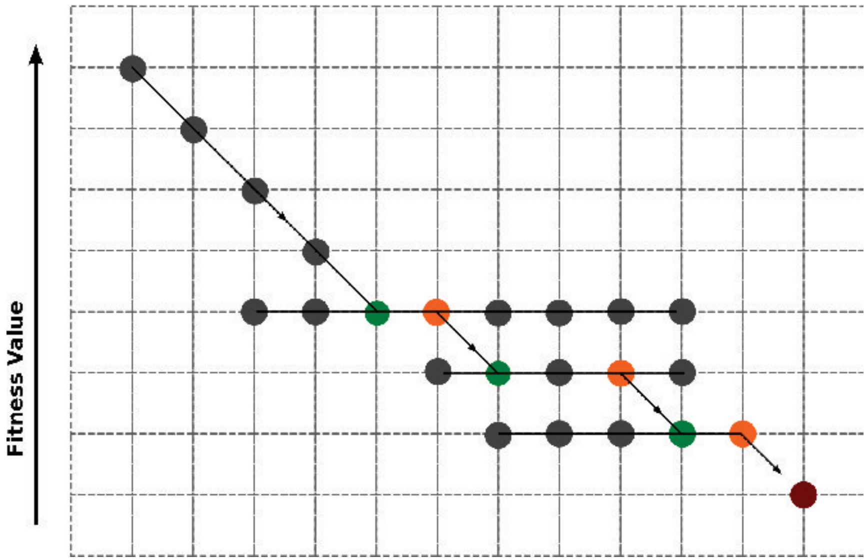
Trade-off exploitation / exploration

- **Parameters:** C, A, P

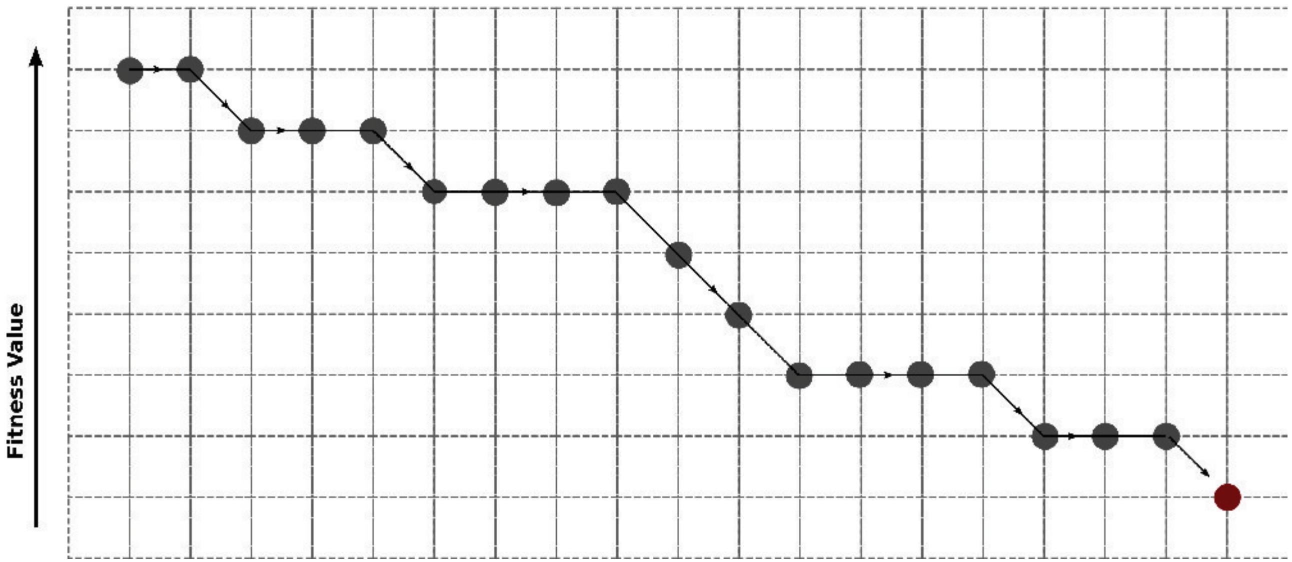
M.-É. Marmion, L. Jourdan, C. Dhaenens, A. Liefooghe, S. Verel, The Road to VEGAS: Guiding the Search over Neutral Networks, In GECCO 2011, ACM Press, pages 1979–1986, 2011

VEGAS Dynamics

(VEGAS)



(NC)



VEGAS
Experimental Design

Stopping criterion:

- Number of evaluations (2.10^7)

Parameters:

- Neighborhood: Insertion operator
- Restart: 3 random Exchange-moves
- C : trade-off values (3)
- P : MNS values (5)
- A : arbitrary values (3)

  & values from landscape analysis (2)

Instances:

- Taillard (11)

Literature:

- Netcrawler (NC)

Run:

- 30 x #instances x # C x # P x # A

Validation with statistical tests:

- Student t-test
- Wilcoxon signed rank test

VEGAS
Results
→ Taillard Instances

VEGAS Performance:

- Better performance with C that **encourages evolvability**, with **large** P value and with A **value from landscape analysis**
- **Best-known is found**: 6 instances
- Deviation to the best-known **under 1.56%**: other instances

VEGAS vs. Local Search Methods:

- **Random selection** gives very **bad performance** against VEGAS
- On average, VEGAS performance is lower than NC

VEGAS Discussion

VEGAS is:

- **Fair** on the NN
- **Quite efficient** on neutral problems
- More efficient than with a random selection

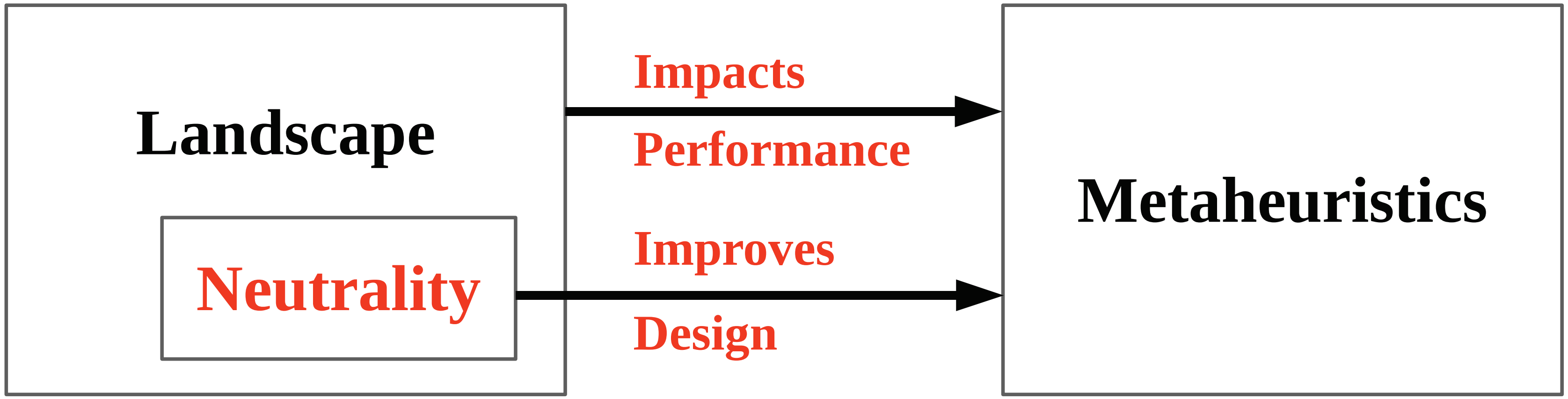
but...

- Defining evolvability measure is **difficult**
- UCB did not show its efficiency

→ For **neutral problems**, **VEGAS** seems to be an **appropriate** local search that has **to be improved**

Conclusion
General Contributions

How can landscape analysis help the design of efficient algorithms?



- **Fitness landscape analysis** impacts the **performance** of metaheuristics
- **Neutrality** has to be considered
- NILS: Neutrality can be **used easily** and **efficiently** in a local search
- VEGAS: the **whole Neutral Networks** can be considered in a local search with an advanced **guiding strategy**

Conclusion
Outlook

- What would be **performance of NILS or VEGAS** on other neutral problems?
- Can landscape analysis be done ***on-line***?
  - Choice of the neighborhood operator
  - Estimation of the parameters value
  - …
- How to **define evolvability** measure?
  - Landscape analysis?
- How to **guide** the search over neutral network?

Questions?