Examples of fitness landscape analysis

Fitness landscape analysis for understanding and designing local search heuristics

Sébastien Verel

LISIC - Université du Littoral Côte d'Opale, Calais, France
http://www-lisic.univ-littoral.fr/~verel/

The 51st CREST Open Workshop
Tutorial on Landscape Analysis
University College London

27th, February, 2017

Real-world problems and fitness landscape analysis

Two combinatorial black-box problems :

- Engineering design problem :
  Design of the control program of rods
  Toward landscape aware parameter settings
- Cellular automata problem :
  Design of a complex system program
  One more step to understand why it is possible

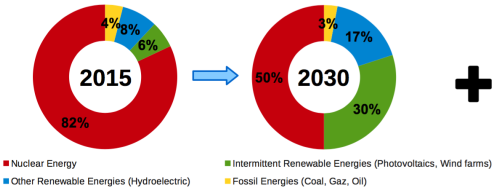
Real-world problems and fitness landscape analysis

Two combinatorial black-box problems :

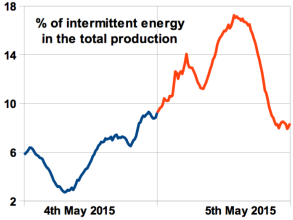
- Engineering design problem :
  joined work M. Muniglia, J.-C. Le Pallec, J.-M. Do
  Design of the control program of rods
  Toward landscape aware parameter settings
- Cellular automata problem :
  joined work M. Clergue, E. Formenti
  Design of a complex system program
  One more step to understand why it is possible

Energy production problem

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



(Source : ADEME)



(Source : RTE eco2mix)

Large scale deployment of **intermittent renewable energies** in France

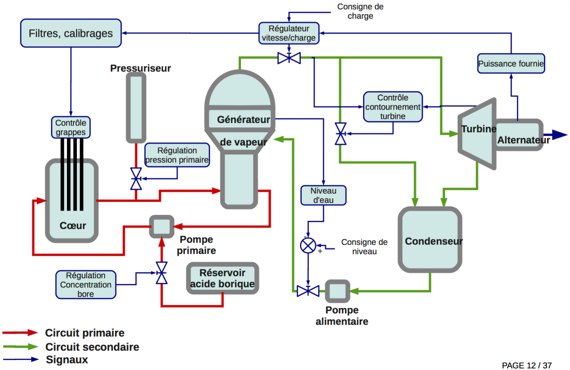
Highly fluctuating production rate (up to 3 times the average)

Challenge of the NPP control

Optimize the nuclear power plant (NPP)
toward better manageability,
so they can cope with huge power variations

Real-world black-box combinatorial optimization problem

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

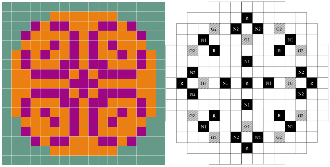


Multi-physic simulator

- core : neutronics, thermalhydraulics, fuel,
- boron management
- steam generator model

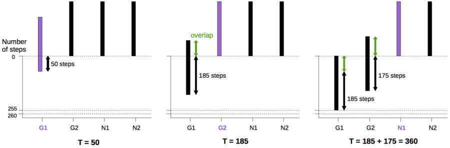
Expensive computation : 40 minutes of simulation

Program of the control rods

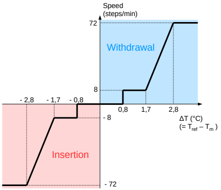


- Power Shimming rods :
  4 groups : G1, G2, N1, N2
- Temperature Regulation rods :
  1 group : R

Insertion sequence of the PS rods :



Speed program of the TR rods :

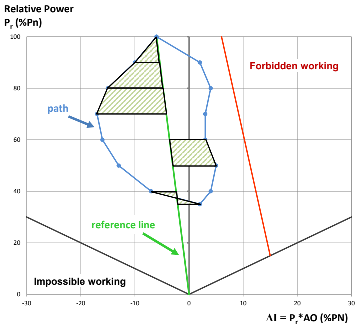


Parametric program defined by $n = 11$ integers :

	PSR Overlaps			PSR Velocities				TRR V.			
	o_1	o_2	o_3	v_1	v_2	v_3	v_4	V	v	mb	db
lower b.	0	0	0	10	10	10	10	3	3	7	8
upper b.	255	255	255	110	110	110	110	13	13	117	16
ref. val.	185	175	160	60	60	60	60	72	8	27	8

Nuclear Reactor Operation Optimization problem [MDG+16]

Control diagram



$(P_{r,i}, \Delta I_i)$: state of the core

Objective function

$x \in X$, set of integers vectors,

$$f(x) =$$

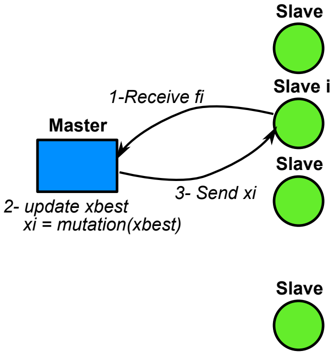
$$\frac{1}{4} \sum_i |P^2_{r,i+1} - P^2_{r,i}| \cdot \left(D(\Delta I_{i+1}) + D(\Delta I_i) \right)$$

where :

- $D(\Delta I_i) = |\Delta I_i - \Delta I_i^{ref}|$
- ΔI_i^{ref} : the power axial imbalance

Asynchronous distributed $(1+\lambda)$-Evolution Strategy

Master-slaves architecture



Algorithm on Master

$\{x_1, \ldots, x_\lambda\} \leftarrow$ **Initialization**()
for $i = 1..\lambda$ **do**
  Send (Non-blocking) x_i to slave S_i
end for
$x_{best} \leftarrow \emptyset$, and $f_{best} \leftarrow \infty$
repeat
  if there is a pending mess. from S_i **then**
    Receive fitness f_i of x_i from S_i
    if $f_i \leqslant f_{best}$ **then**
      $x_{best} \leftarrow x_i$, and $f_{best} \leftarrow f_i$
    end if
    $x_i \leftarrow \texttt{mutation}(x_{best})$
    Send (Non-blocking) x_i to slave S_i
  end if
until time limit

3072 computation nodes during 24h max.

Parameter tuning : Mutation

Parametric program defined by $n = 11$ integers :

	PSR Overlaps			PSR Velocities				TRR V.			
	o_1	o_2	o_3	v_1	v_2	v_3	v_4	V	v	mb	db
lower b.	0	0	0	10	10	10	10	3	3	7	8
upper b.	255	255	255	110	110	110	110	13	13	117	16
ref. val.	185	175	160	60	60	60	60	72	8	27	8

Mutation operator

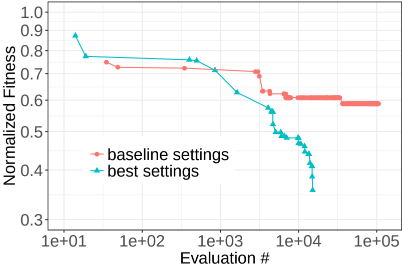
Parameters :

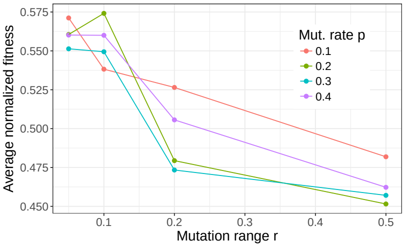
- For each integer, a mutation rate is defined by p (default $p = 1/n$)
- For each integer, a mutation range is defined by δ (default $5\%(\overline{x_i} - \underline{x_i})$)

Procedure :

- For each integer, the value is changed according to the rate of mutation (Bernoulli law of parameter p)
- When an integer is modified, a random value (uniform distribution) is pick from : $[x_i - \delta, x_i + \delta] \cap [\underline{x_i}, \overline{x_i}] \setminus \{x_i\}$

Performance *vs.* mutation parameters values





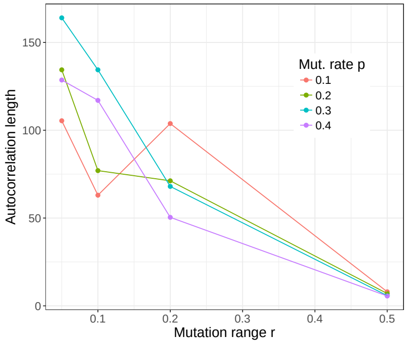
- Baseline settings :
  punctuated equilibrium dynamics

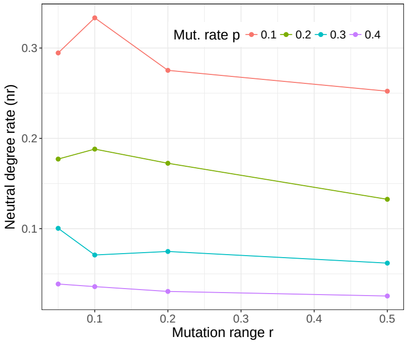
- Impact of parameters values
- Best settings :
  $p = 0.2,\ r = 0.5$

Fitness landscape analysis

One random walk of length 10^3 for each mutation operator values

- Autocorrelation length k : $|\rho(k)| \leqslant 4/\sqrt{\ell}$
- Neutral degree rate : estimated with $\ell - 1$ solutions

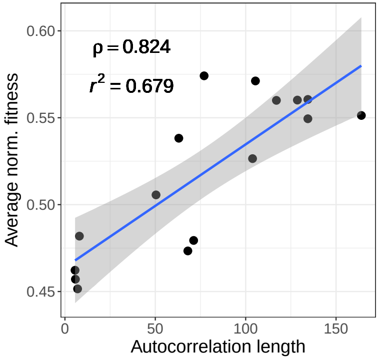


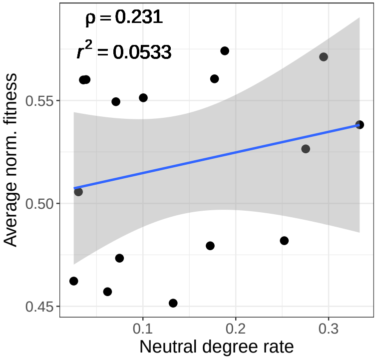


In short : r impacts the ruggedness, p impacts the neutrality

Fitness landscape features *vs.* performance

Features of fitness landscape related to the performance ?



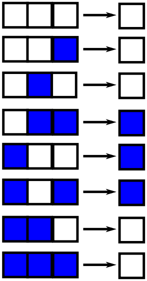


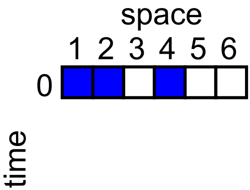
In short : possible to tune mutation parameters based on the fitness landscape analysis (but the more rugged, the better !)
Future work : bi-objective optimization...

Cellular automata

- Discrete dynamical system
- Set of finite state machines
- Program : transition function
- Model of decentralized computation

Transition function :

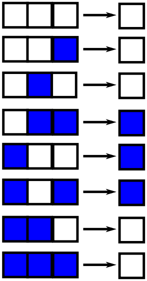


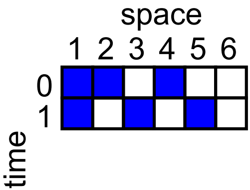


Cellular automata

- Discrete dynamical system
- Set of finite state machines
- Program : transition function
- Model of decentralized computation

Transition function :

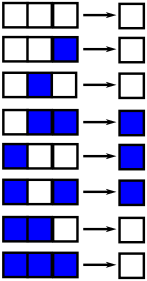


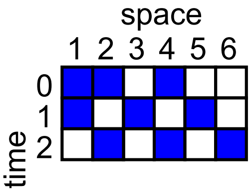


Cellular automata

- Discrete dynamical system
- Set of finite state machines
- Program : transition function
- Model of decentralized computation

Transition function :

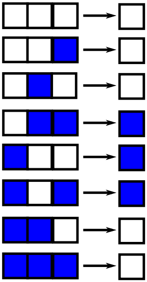


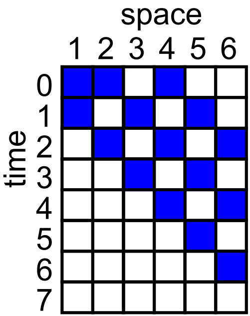


Cellular automata

- Discrete dynamical system
- Set of finite state machines
- Program : transition function
- Model of decentralized computation

Transition function :





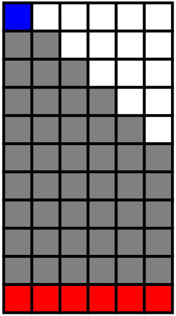
Firing Squad Synchronization Problem

John Myhill, 1957, published by Edward Moore, 1964 [Moo64]

Synchronization problem of decentralized computation nodes without global coordinator and bounded communication

The problem

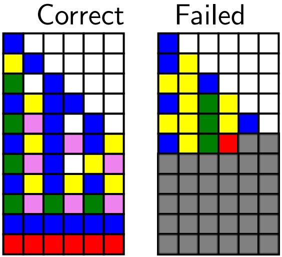
- One-dimensional cellular automata
- Communication with left and right hands
- Initial configuration :
  one cell in state "general",
  the others cells are in "rest" state
- All cells in "rest" state remains "rest"
- **Goal** : Find the set of rules such that all cells reach for the first time the "firing" state at the same time



Firing Squad Synchronization Problem

Precisions

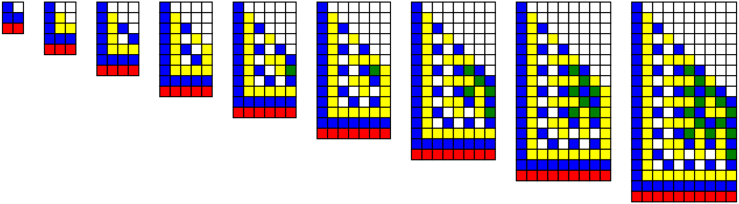
- Neighborhood of a cell : left, middle and, right cells
- Special rules for left and right boundaries
  number of rules for k states : $nr_k = (k-1)^3 + 2(k-1)^2 - 3$
  number of CA for k states : k^{nr_k}.
- Same rules (CA) for all lengths n : $\forall n \geq 2$, $\text{fssp}_n(CA)$ true.
- Minimal time : $2n - 2$ time steps for n cells



Firing Squad Synchronization Problem

Precisions

- Neighborhood of a cell : left, middle and, right cells
- Special rules for left and right boundaries
  number of rules for k states : $nr_k = (k-1)^3 + 2(k-1)^2 - 3$
  number of CA for k states : k^{nr_k}.
- Same rules (CA) for all lengths n : $\forall n \geq 2,\ \text{fssp}_n(CA)$ true.
- Minimal time : $2n-2$ time steps for n cells



History of the problem [Yun07]

Non-minimal time :

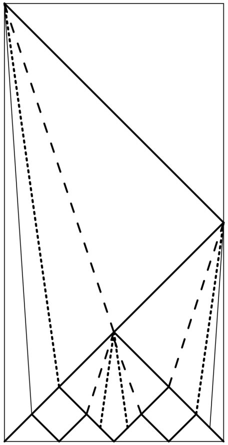
- 1967, Minsky & McMarthy [Min67]
  15 states, $3n + O(log(n))$ time steps

Minimal time :

- 1962, E. Goto [Got62] : $\approx 10^6$ states
- 1967, Waksman : 16 states,
  Balzer : 8 states [Bal67]
- 1987, Mazoyer [Maz87], **6** states

Minimal time with 4 states :

- 1967, No solution [Bal67] [San94]



Waksmann / Balzer

History of the problem [Yun07]

Non-minimal time :

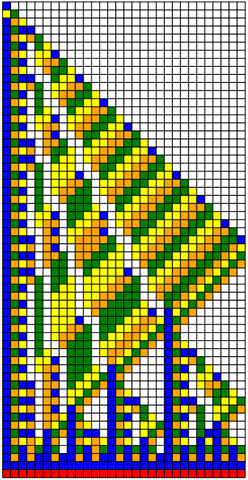
- 1967, Minsky & McMarthy [Min67]
  15 states, $3n + O(log(n))$ time steps

Minimal time :

- 1962, E. Goto [Got62] : $\approx 10^6$ states
- 1967, Waksman : 16 states,
  Balzer : 8 states [Bal67]
- 1987, Mazoyer [Maz87], **6** states

Minimal time with 4 states :

- 1967, No solution [Bal67] [San94]



Mazoyer

History of the problem [Yun07]

Non-minimal time :

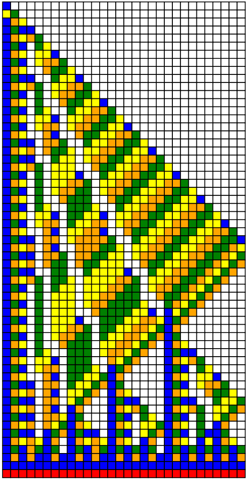
- 1967, Minsky & McMarthy [Min67]
  15 states, $3n + O(log(n))$ time steps

Minimal time :

- 1962, E. Goto [Got62] : $\approx 10^6$ states
- 1967, Waksman : 16 states,
  Balzer : 8 states [Bal67]
- 1987, Mazoyer [Maz87], **6** states

Minimal time with 4 states :

- 1967, No solution [Bal67] [San94]



Mazoyer

"But I lost another solution..."

FSSP as a black-box combinatorial optimization problem

Associated optimization problem

- Search space :
  Set of all CA with 6 states,
- Objective function :
  largest length of synchronized cells

$f(x) = n$ iff $\forall i \in [2, n], \text{fssp}_i(x) = \text{true}$ and $\text{fssp}_{n+1}(x) = \text{false}$

Huge search space : Brut force fails

For $k = 6$ states :

number of rules : $nr_k = (k-1)^3 + 2(k-1)^2 - 3 = 172$

number of CA : $\sharp X = k^{nr_k} \approx 10^{133}$

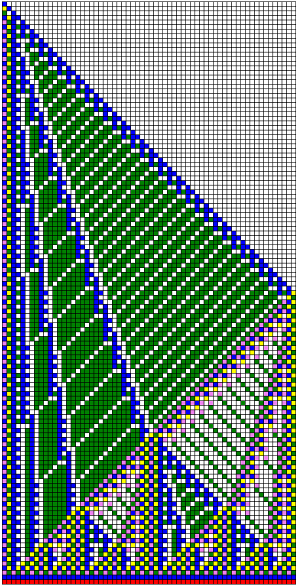
Goal

Find one maximum of f which synchronizes the largest length.

Iterated Local Search

Choose randomly initial solution $x \in X$
$x \leftarrow \text{hc}(x, e_{hc})$
Initialize the number of evaluation e_{tot}
repeat
  $y \leftarrow \text{perturbation}_k(x)$
  $z \leftarrow \text{hc}(y, e_{hc})$, and update the number of evaluation e_{tot}
  if $f(x) \leq f(z)$ **then**
    $x \leftarrow z$
  end if
until $e_{tot} \geq e_{max}$

- first-improvement hill-climbing with $\leqslant$ acceptance criterion
- neighborhood relation : modification of 1 rule
- Perturbation : randomly modify k rules
- Number of evaluations : 100×10^9



A minimal Kolmogorov complexity solution
80 (human +LS) < 119 rules (human+paper)

Problem solved

2665 different solutions found
(synchronization until $n = 10^3$)

Number of successful runs (over 200)

hc eval e_{hc} ($\times 10^6$)	Perturbation k 3	4	5	6	cumu.
0.5	**7**	**4**	**5**	**5**	**21**
0.7	**8**	**8**	**8**	1	**25**
0.9	**9**	**8**	**5**	**2**	**24**
1.1	**8**	**6**	**4**	**7**	**25**
1.5	**9**	**3**	**5**	**5**	**22**
5.0	**3**	**5**	1	0	**9**
cumu.	**44**	**34**	**28**	20	126

Why it works ?

Citation from an CA expert

"Local search can't work for solving a CA problem because when you change one rule, everything change"

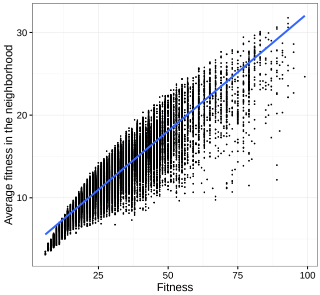
But...

From the point of view of Local Search :

- A lot a one-rule modifications do not change the space-time diagram
- The rules which are not used for length n could be benefit for length $n + 1$
- With "high" probability, it exists some modifications which can improve the CA

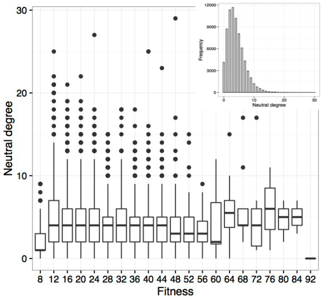
Fitness landscape analysis

Fitness cloud :



Average fitness in the neighborhood

Neutrality :



Neutral degree

- Performance of neighboring solution is correlated
- Same performance for ≈ 4 neighbors (from used rules) + equal fitness neighbors from unused rules.

Surprisingly, some local modifications of program are useful

Discussions

Fitness landscape analysis

- Helps to understand the structure of real-world problems
- Possible way to tune the parameters of local search heuristics.

References I

Robert Balzer.
An 8-state minimal time solution to the firing squad synchronization problem.
Information and Control, 10(1) :22 – 42, 1967.

E. Goto.
A minimum time solution of the firing squad problem.
Course Notes for Applied Mathematics, 298, 1962.

Jacques Mazoyer.
A six-state minimal time solution to the firing squad synchronization problem.
Theoretical Computer Science, 50(2) :183 – 238, 1987.

References II

Mathieu Muniglia, Jean-Michel Do, Hubert Grard, Sébastien Verel, S David, et al.
A multi-physics pwr model for the load following.
In *International Congress on Advances in Nuclear Power Plants (ICAPP)*, 2016.

Marvin L. Minsky.
Computation : Finite and Infinite Machines.
Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.

E.F. Moore.
The Firing Squad Synchronization Problem in Sequential Machines, volume Sequential Machines. Selected Papers., pages 213–214.
Addison-Wesley, Reading MA, 1964.

References III

P. Sanders.
Massively parallel search for transition-tables of polyautomata.
In W. Wilhelmi C. Jesshope, V. Jossifov, editor, *Proc. of the VI Int. Workshop on Parallel Processing by Cellular Automata and Arrays*, pages 99–108. Akademie, Berlin, 1994.

J.-B. Yunes.
A propos d'automates cellulaires, suivi par des fonctions booléennes.
HDR, 2007.