Surrogate assisted optimization

Sébastien Verel

LISIC
Université du Littoral Côte d'Opale
Equipe OSMOSE

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

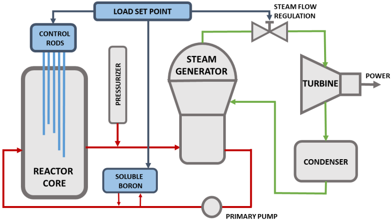
Master informatique WeDSci, ULCO,

2022, version 1.0

Context

Mobility system

Nuclear energy system



A priori, each domain is very different
But, share :

- Design problems for new perspectives,
- Inaccessible (cost) quantities, scales, etc.

Solving design problems, etc.

Solving optimization problems (mono- or multi-objective)

- Using the cognitive, and social abilities of humans :
  expert knowledges, evaluation of risk, uncertainties, divide into sub-problems, complex reasoning, etc.
- Using the computational, and memorization abilities of machines :
  automatic, data, formal language, speed, multi-scale, etc.

Main AI approaches for automatic solving

- Algebraic approach : algebraic, or formel model
- Digital twin approach :
  numerical model, and numerical simulation

Comparaison of approaches

Algebraic approach

- Formal model
- Aggregated variables, noice (demand, incertainties), contraints,...
- Artificial or real-like problem instances
- Offline

Tools :
cplex, gurobi, constr. prog., local search, ea, etc.

Digital twin approach

- Low level model
- Complex interactions
- Flow of data : sensor, etc.
- Offline, Online

Tools :
simcore, simio, matsim, devs,...

$$\max \sum_{t\in T}(pf_t) - \sum_{t\in T}\sum_{o\in O}(g_{t,o}u_{t,o} + 5s_{t,o})$$

$$s_{t-1,o} + b_{t,o} = u_{t,o} + s_{t,o} \quad \forall t \in T \setminus t_0,\ \forall o \in O$$

$$i + b_{t_0,o} = u_{t_0,o} + s_{t_0,o} \quad \forall o \in O$$

$$s_{t_e,o} = i \quad \forall o \in O$$

$$\sum_{o\in V} u_{t,o} \leq 200 \quad \forall t \in T$$

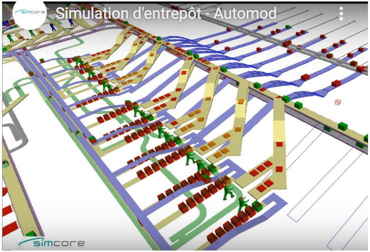
$$\sum_{o\in N} u_{t,o} \leq 250 \quad \forall t \in T$$

$$3f_t \leq \sum_{o\in V} h_o u_{t,o} \leq 6f_t \quad \forall t \in T$$

$$\sum_{o\in V} u_{t,o} = f_t \quad \forall t \in T$$

$$b_{t,o}, u_{t,o}, s_{t,o} \geq 0 \quad \forall t \in T,\ \forall o \in O$$

$$f_t \geq 0 \quad \forall t \in T$$



Consequences for automatic solving

$$\max \sum_{t\in T}(pf_t) - \sum_{t\in T}\sum_{o\in O}(g_{t,o}u_{t,o} + 5s_{t,o})$$

$$s_{t-1,o} + b_{t,o} = u_{t,o} + s_{t,o} \quad \forall t \in T \setminus t_0,\ \forall o \in O$$

$$i + b_{t_0,o} = u_{t_0,o} + s_{t_0,o} \quad \forall o \in O$$

$$s_{t_e,o} = i \quad \forall o \in O$$

$$\sum_{o\in V} u_{t,o} \leq 200 \quad \forall t \in T$$

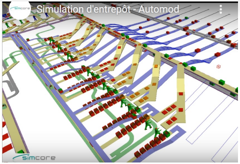
$$\sum_{o\in N} u_{t,o} \leq 250 \quad \forall t \in T$$

$$3f_t \leq \sum_{o\in V} h_o u_{t,o} \leq 6f_t \quad \forall t \in T$$

$$\sum_{o\in V} u_{t,o} = f_t \quad \forall t \in T$$

$$b_{t,o}, u_{t,o}, s_{t,o} \geq 0 \quad \forall t \in T,\ \forall o \in O$$

$$f_t \geq 0 \quad \forall t \in T$$



Algebraic approach

Pros :

- Exploitation of the algebraic properties (fast to compute)
- Explicit, and synthetic model

Difficulties :

- Design of the model :
  creation of languages, etc.

Digital twin approach

Pros :

- Low level description
- Tests, visualization

Difficulties :

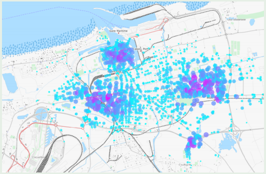
- $\approx$ Black-box : $(x, f(x))$
- Costly simulation
  (time, energy)

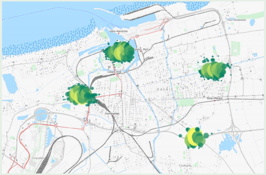
Indeed, not only "solving", but also support of decision making : before, during, and after the optimization process

Digital twin for mobility system

F. Leprêtre, V. Marion, C. Fonlupt, S. Verel (LISIC) - thesis 2017 - 2020.
H. Aguirre, R. Armas, K. Tanaka (Shinshu Univ., Nagano, jp)
Partner : Calais City, Marie Capon, (expertise, and funding)

SIALAC benchmark of mobility





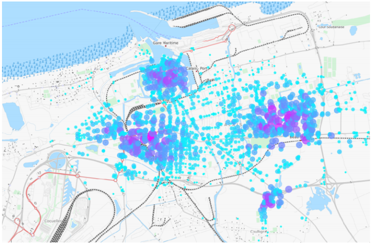
Different and, futur scenario : home, agents, activity

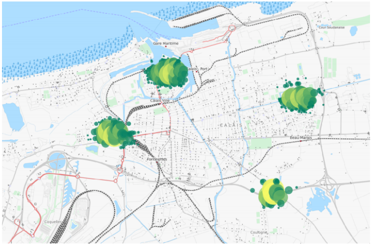
Two problems

- Tuning of traffic light
- Bus stop position

SIALAC benchmark of mobility

Leprêtre, F., et al. Applied Soft Computing, 2019 [12]





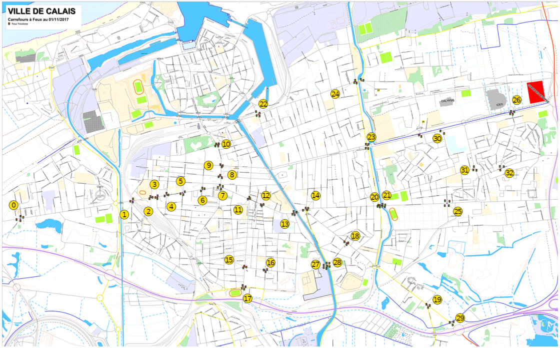
Number of agents	$\{5, 10, 15, 20\} \times 10^3$
Home	$\{1 \text{ cluster}, 4 \text{ clusters}, \text{ uniform}\}$
Activity	$\{1 \text{ cluster}, 4 \text{ clusters}\}$
Signals systems	$\{50\%, 75\%, 100\%\}$

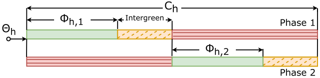
72 scenario using MatSim (Multi-Agent Transport Simulation)

Goal

- Show to the partner what it is possible with such tools
- Design **robust** optimization algorithms for mobility problems

Traffic light problem for Calais, and Quito cities





- Space : 33 (Calais, France), 70 (Quito, Equator) intersections
  search space dim. $\times 4$ integer variables
- Criteria : minimize average travel time (black-box problem)
- Computational time per simulation $\approx$ 1 minute

Gradient free optimization algorithms

Stochastic Hill Climber

$x \leftarrow$ initialize random solution
repeat

$x' \leftarrow$ mutate x
$x \leftarrow x'$ **if** $f(x') < f(x)$
until stopping criterion met

Evolutionary Algorithm

$P = \{x_1, \ldots, x_\mu\} \leftarrow$ rnd. init.
repeat
$P_{genitor} \leftarrow$ selection from P
$P_{children} \leftarrow$ breed $P_{genitor}$
$P \leftarrow$ replace $P \cup P_{children}$
until stopping criterion met

- `mutate` : random variation of candidate solution
- Tradeoff exploration / exploitation : `mutate` / `selection`

How to tune the mutation operator ?
i.e. Where to explore ?

Surrogate, and model-based approaches

According to the context, the search strategy can be different.

When the evaluation time of a single candidate solution is :

- short : try, and test strategy (local search, EA, etc.)
  a test is fast, so multiple tests are possible.
  Memory "less" strategy.
  ex. : re-computation of a solution
- long : model based strategy
  spend more time to design a new candidate solution,
  aggregation of information on the problem (model), and test
  ex. : 200 evaluations available on problem of dimension 100

Structure of real-world problems

Intuitively

Real-world problem instances are often "structured" :

- Local sub-problems are not random,
- Interdependency between sub-problems are not random.

Importance of variables

Consequence : some variables are more impactful than others.

Examples

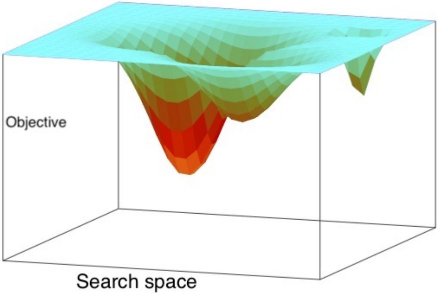
Isolated traffic lights are less impact on travel time
than central traffic lights

How to detect important variable to design a model of problem ?
Expert knowledge, or more automatically....

Fitness landscape : a model of the search space

Fitness landscape (Wright 1920)

- $\mathcal{S}$: set of candidate solutions, search space
- $f : \mathcal{S} \rightarrow \mathbb{R}$: objective function
- $\mathcal{N} : \mathcal{S} \rightarrow 2^{\mathcal{S}}$, neighborhood relation between solutions



- Geometry of the fitness landscape :
  Features/metrics
  are correlated to
  algorithm performance

$\Rightarrow$ Toward automatic design (tuning/control) of algorithms

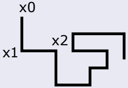
Offline model of problem

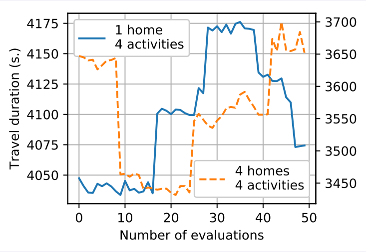
Importance degree of variable i

$$\delta_i = |f(mutate_i(x)) - f(x)|$$

Estimation : Random walk on fitness landscape

Sequence of neighboring solutions : $(x_0, x_1, x_2, \ldots)$

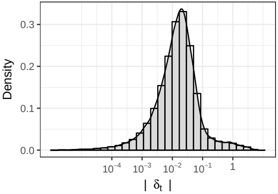


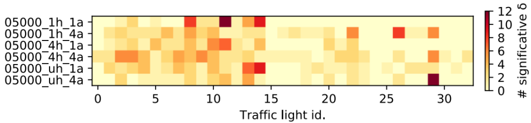


Offline model, before running the algorithm (expert knowledge ?) :

Importance δ_i associated to each var. i

Add explainable model to optimization





⇒ Explainable model (cf. XAI) defined with "score",
Improve the communication with the partners

Adaptive algorithm based on offline model

Backbone in combinatorial problems :
"good" solutions have some specific variables value

Design of mutation operator

- **Hypothesis** : modify in priority important variables
- **Goal** : automatic learning of expert knowledge

Method

- Divide the set of variables into 3 groups according to importance
- Use reinforcement machine learning technique to select the group to mutate.

Adaptive bandit descent

Multi-armed bandit problem (reinforcement learning)

UCB strategy to select to relevant arm :

$$\hat{r}_i + C\sqrt{\frac{S}{s_i}}$$

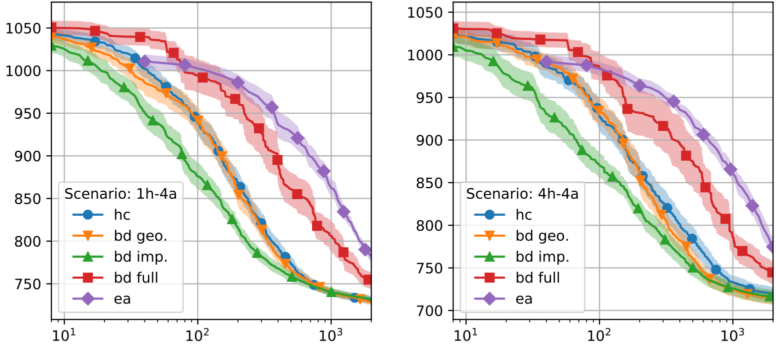
$\hat{r}_i$: reward, s_i : nb. of selection of arm i, and S : total nb. of selection, C : tradeoff parameter

Adaptive algorithm

$G \leftarrow$ split var. into groups
$x \leftarrow$ initialize random solution
repeat
  $g \leftarrow$ select group in G using UCB rule
  $x' \leftarrow$ mutate a variable from g of x
  $x \leftarrow x'$ **if** $f(x') < f(x)$
  Update rewards
until stopping criterion met

Some results

Quality *vs.* number of evaluations :



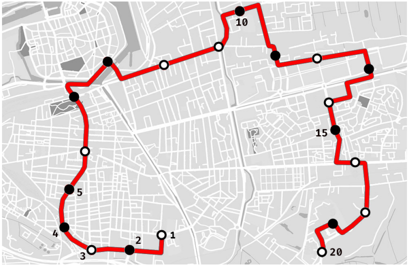
- Speed up the convergence
- Better than "hand made" groups,
  or previous Evolutionary Algorithm
- Robust on different scenario (also for Quito city)

Optimisation with time expensive simulation

- Parallel computation : distribute computation on machines
- **Surrogate model** : online substitution of the original function with an (approximated) function fast to compute

Surrogate model

- A lot of works on numerical optimization :
  $f : \mathbb{R}^d \to \mathbb{R}$
- Few works on discrete/combinatorial optimization :
  $f : \{0,1\}^d \to \mathbb{R}$, or $f : \mathcal{S}_n \to \mathbb{R}$



Bus stop position problem

- Space : $\{0,1\}^d$ open or close possible stops
- Criterium : min. travel time

Surrogate-assisted opt. of pseudo-boolean problems

Florain Leprêtre, Virginie Marion, Cyril Fonlupt (LISIC),
K. Tanaka, H. Aguirre (Univ. Shinshu), A. Liefooghe, B. Derbel (univ. Lille)

Surrogate-Assisted Optimization

$X \leftarrow$ initial sample
repeat
  $M \leftarrow$ Build model of f from X
  $x^\star \leftarrow$ Optimize *w.r.t.* an acquisition function based on M
  $y^\star \leftarrow f(x^\star)$ using the numerical simulation
  $X \leftarrow X \cup \{(x^\star, y^\star)\}$
until time limit

In numerical optimization [18] :

- Models :
  Gaussian Process, polynomial chaos, NN, RBF, RF, deep$*$, etc.
- Acquisition function :
  M, Expected improvement, probability impr., UCB, etc.

In discrete optimization [2] :

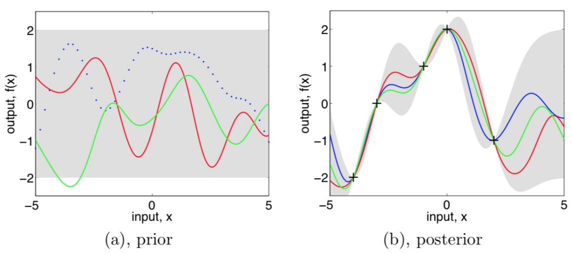
- Use discrete distance, or numerical variable

Example : Efficient Global Optimizer [9] [20]

- Model : Gaussian Process $M(x) \approx \mathcal{N}(m(x), K(x, x'))$
- Acquisition function : Expected Improvement

GP : Random variables which have joint Gaussian distribution.

mean : $m(y(x)) = \mu$; covariance : $cov(y(x), y(x')) = exp(-\theta \text{ dist}(x, x')^p)$



from : Rasmussen, Williams, GP for ML, MIT Press, 2006.

pros : estimation of incertainty (expected improvement etc.)
cons : estimation is costly, and distance in high distance is not informative

Polynomial regression model

Polynomial chaos regression (PRC)

Model

A basis of functions $\{\varphi_j : j \in \{1, \ldots, p\}\}$

$$M(x) = \sum_{j=1}^{p} \beta_j \varphi_j(x)$$

Regression using least square method, or bayesian approach

Example : second-order polynomial

$$M_2(x) = \beta_0 + \sum_{i=1}^{d} \beta_i \; x_i + \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \beta_{ij} \; x_i x_j$$

Pros :

Easy interpretation (XAI), fast to compute, polynomial regression

Cons :

Use a relevant basis of functions (Fourier transform, etc.)

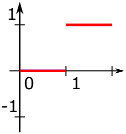
Number of terms increases exponentially with order (sparse methods)

Pseudo-boolean surrogate : Walsh functions

- Space pseudo-boolean function is a vector space
- Basis : multi-linear functions, $x_{k_1} \dots x_{k_\ell}$ [Baptista, Poloczek, BOCS, ICML 2018][1]

Multi-linear :

$d = 1$, $\psi_1(x) = x$

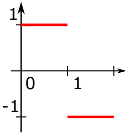


Orthogonal : No

x	ψ_0	ψ_1
0	1	0
1	1	1

Walsh :

$d = 1$, $\varphi_1(x) = (-1)^x$



Orthogonal : Yes

x	φ_0	φ_1
0	1	1
1	1	-1

Extension to dimension d using tensorial product :

$$\psi_{k_1 \dots k_\ell}(x) = x_{k_1} \dots x_{k_\ell}$$

$$\varphi_{k_1 \dots k_\ell}(x) = (-1)^{x_{k_1}} \dots (-1)^{x_{k_\ell}}$$

Surrogate model for pseudo-boolean functions

Walsh functions

$$\forall x \in \{0,1\}^d, \quad \varphi_k(x) = (-1)^{\sum_{j=0}^{d-1} k_j x_j}$$

Normal, and orthogonal basis

Any function can be written as :

$$f(x) = \sum_{k=0}^{2^d-1} \beta_k . \varphi_k(x)$$

with : $\beta_k = \frac{1}{2^d} \sum_{x \in \{0,1\}^d} f(x) . \varphi_k(x)$

Example with order 2, model limited to quadratic interactions :

$$f(x) = \beta_0 + \sum_{i=1}^{d} \beta_i . \sigma_i + \sum_{i<j=1}^{d} \beta_{ij} . \sigma_i \sigma_j \text{ with } \sigma_i = (-1)^{x_i}$$

Why Walsh functions ?

$$f(x) = \beta_0 + \sum_{i=1}^{d} \beta_i.\sigma_i + \sum_{i<j} \beta_{ij}.\sigma_i\sigma_j \text{ with } \sigma_i = (-1)^{x_i}$$

- Explicit algebraic model (not black-box) : easy to interpret
  Interaction between variables, intensity of interaction $|\beta_{i,j}|$
- Efficient algorithms to optimize such problems
- Model of function used in quantum computing
  Also know as Spin-Glasses, or QUBO / UBQP problems [8]

Surrogate model based on Walsh fonctions

Expansion to order ℓ (cf. polynomial chaos, sparse grid, etc.)

$$M(x) = \sum_{k \ : \ \text{ord}(\varphi_k) \leqslant \ell} \widehat{\beta}_k . \varphi_k(x)$$

- Pros :
  See previous slides
- Cons :
  model dimension (quadratic, cubic, etc.)
  No uncertainty estimation

Estimation of coefficients :
linear regression using sparse techniques : LARS/LASSO, etc.
LASSO : $\hat{\beta} = \text{argmin}((M(x_i) - y_i)^2 + \alpha||\beta||_1\ )$

Walsh Surrogate-assisted Optimizer (WSaO)

Surrogate-Assisted Optimization

$X \leftarrow$ initial sample
repeat
  $M \leftarrow$ Build Walsh model of f from X
  $x^\star \leftarrow$ Optimize M using Eff. Hill-Climber
  $y^\star \leftarrow f(x^\star)$ using the numerical simulation
  $X \leftarrow X \cup \{(x^\star, y^\star)\}$
until time limit

Efficient optimization algorithm for Walsh functions

using the additive property :

$$\delta_i(x) = M(x \bigoplus i) - M(x) = -2 \sum_{k \supset i} \beta_k \varphi_k(x)$$
$$\delta_{ij}(x) = \delta_i(x \bigoplus j) - \delta_i(x) = 4 \sum_{k \supset i \& k \supset j} \beta_k \varphi_k(x)$$

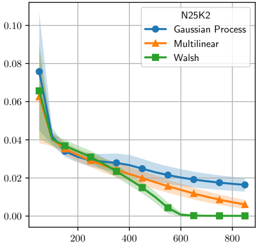
Find best improving move in $O(\ell)$ at each step of the search.
Partition crossover to combine 2 solutions

Chicano, Whitley, Ochoa, and Tinós. "Optimizing one million variable NK landscapes by hybridizing deterministic recombination and local search." In Genetic and Evolutionary Computation Conference, 2017. [3]

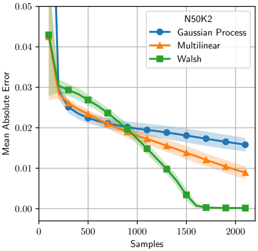
Quality of Walsh regression on academic benchmarks

Mean abs. error on NK-landscapes benchmark

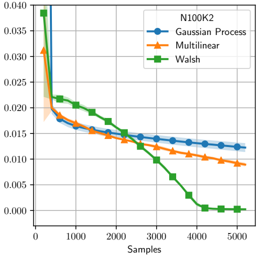
$d = 10$



$d = 25$

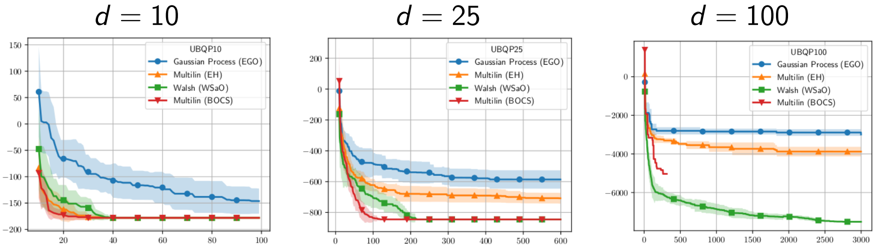


$d = 100$



Walsh Surrogate-assisted Optimizer (WSaO)

Performance on UBQP benchmark

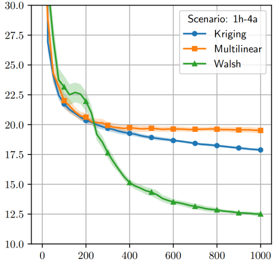


- Krigging : information of distance decreases with dimension
- BOCS : bayesian estimation of multilinear basis, SA opt. alg. (very expensive to compute)

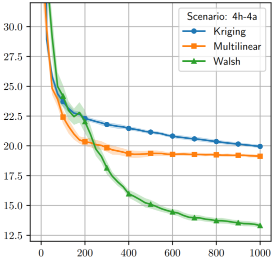
Preliminary results on bus stop problem

Mean abs. error on instances with $d = 20$, 4 activity centers

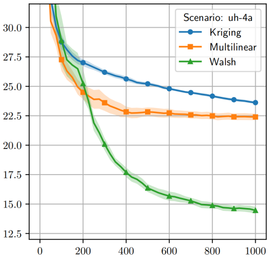
1 home



4 home

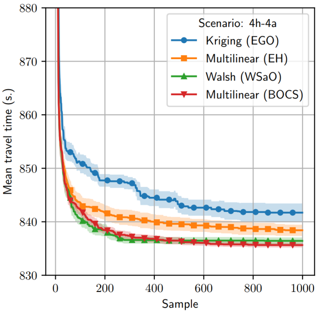


uniform



WSaO on bus stop problem

Preliminary results for small dimension $d = 20$ problem



The work is progressing on real data :
Valentin Vendi, PhD student, 2021-2024, "Design of decision-making tools for sustainable mobility in the Hauts-De-France region", co-direction with C. Fonlupt.

Others master student positions, and possible PhD position coming soon, please contact me.

Comments with surrogate models

- Result with surrogate assisted optimization :
  Near optimal solution, and an explicit model of your problem
  Use non black-box machine learning model are useful !
- Open issues :
  Tradeoff between quality of the model (uncertainty),
  and optimization effort
- Perspectives :
  multi-objective optimization, uncertainty,
  permutation space, numerical & discret, large scale, etc.

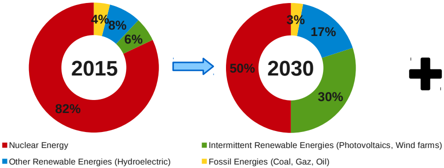
Context

Joined work

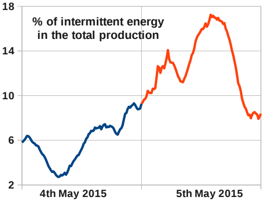
Jean-Michel Do, Jean-Charles Le Pallec, Cheikh Diop, CEA Saclay,
PhD : Mathieu Muniglia (2014 - 2017), Valentin Drouet (2017 - 2020), Baptiste Gasse (2020 - 2023)

Multi-objective optimization of nuclear power plant control for load following in the context of energy transition using evolutionary algorithms

Context



(Source : ADEME)



(Source : RTE eco2mix)

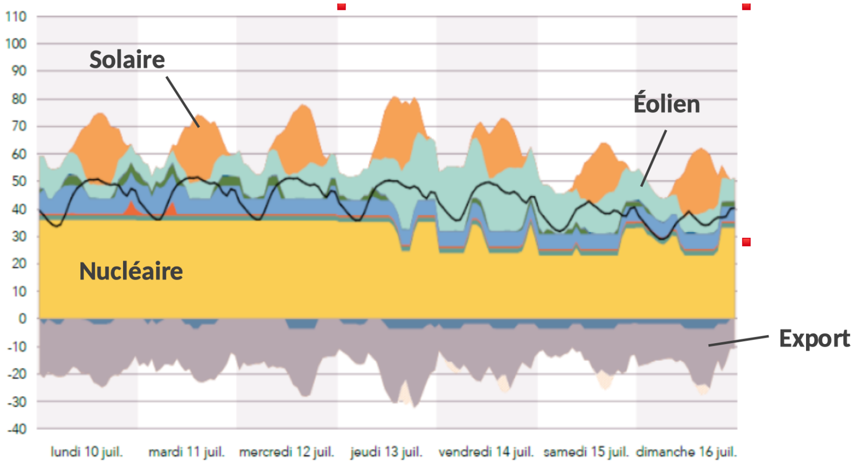
Large scale deployment of **intermittent renewable energies** in France

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

Possible solutions of intermittency :

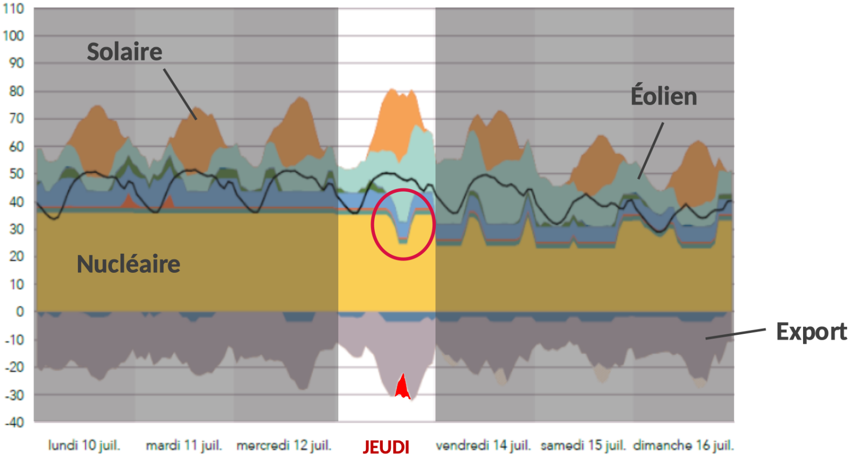
- Flexibility (on demand)
- Smart grid
- Storage
- Manageability of Pressurized Water Reactors

Scenario of energetic transition in France



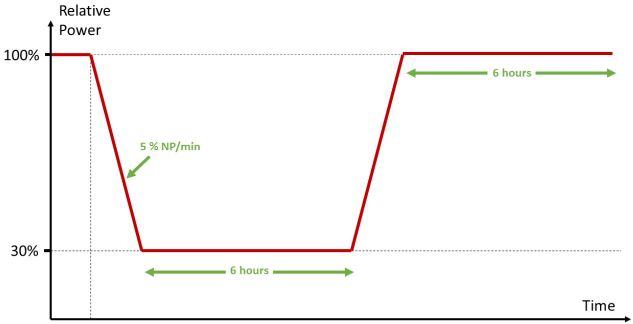
RTE (french electricity transport compagny) prediction for a typical week in 2035 (VOLT scenario)

Scenario of energetic transition in France



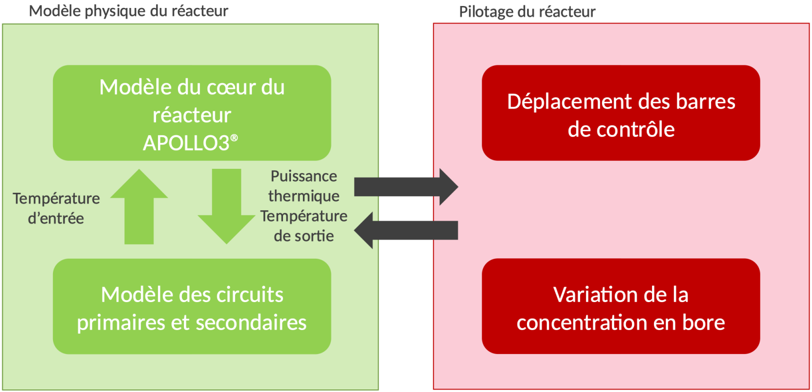
RTE (french electricity transport compagny) prediction for a typical week in 2035 (VOLT scenario)

Target production transient



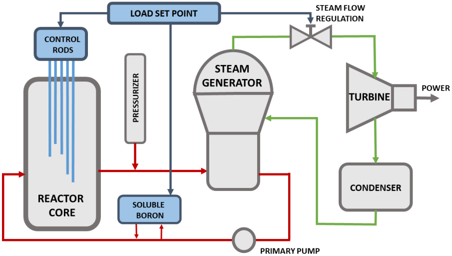
24h of production : most penalizing possible transient

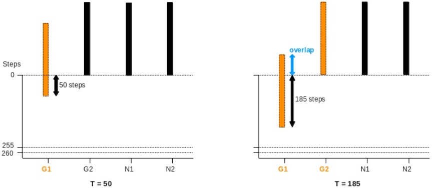
Multi-physic simulator



Around 10 minutes for the simulation of one transient
(Now in 2022, 4 reactors, and potentially 40 min of simulation...)

Optimization problem





Possible criteria

More than 7 criteria can be used :

related to cost, safety, and stability

Available control parameters

- Power Shimming Rods :
  - Overlap (x3)
  - Speed control (x4)
- Temperature Regulation Rods :
  - maneuvering band (x1)

Search space size $\approx 10^{12}$

Fitness landscape analysis : offline model

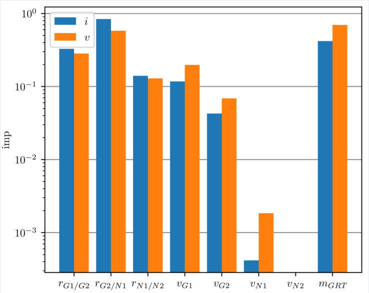
Using random walk sampling

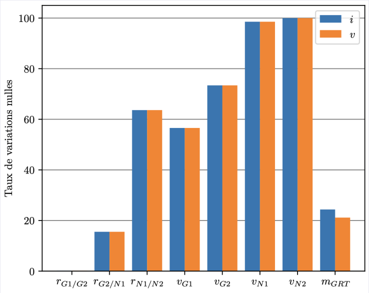
Objective correlation

	v	N_R	C_{IPG}	ΔT	f_T	i	F_v
v	1.0	-0.75	-0.91	-0.14	0.12	-0.06	-0.01
N_R	-0.75	1.0	0.7	0.68	0.17	0.14	0.4
C_{IPG}	-0.91	0.7	1.0	0.12	-0.17	-0.02	-0.1
ΔT	-0.14	0.68	0.12	1.0	0.7	0.51	0.79
f_T	0.12	0.17	-0.17	0.7	1.0	0.83	0.72
i	-0.06	0.14	-0.02	0.51	0.83	1.0	0.58
F_v	-0.01	0.4	-0.1	0.79	0.72	0.58	1.0

2 groups are highly correlated

Variable importance

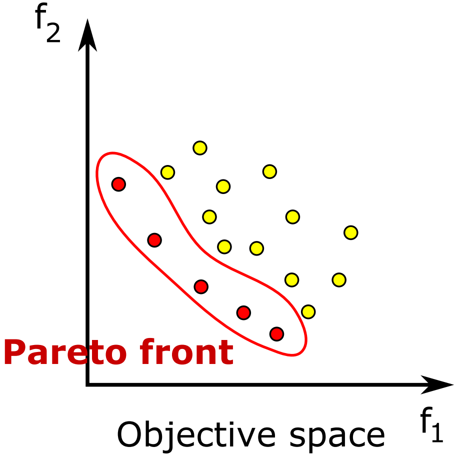




Reduce the dimension to 6, and better understanding of the system

Allow to tune the mutation parameters

Multiobjective optimization

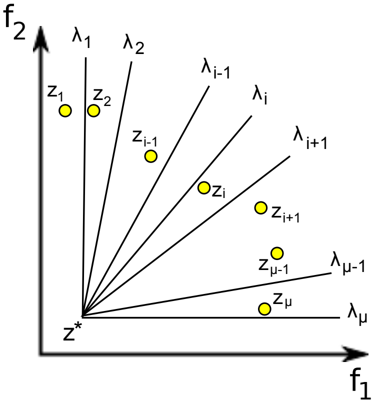


Goal

No a priori on the order/importance of the objectives,
Decision a posteriori based on the optimal Pareto solutions.

MOEA/D : Multi-Obj. Evo. Algo. based on Decomposition

A lot of MO algo. :
Pareto based (NSGAII,...), indicator based (IBEA,...), and

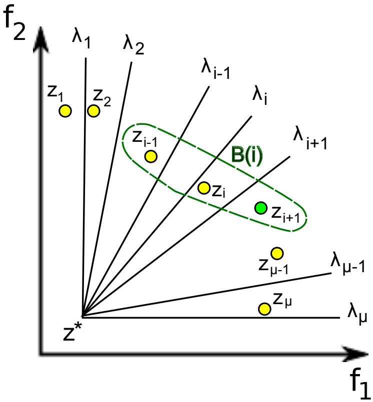


Population at iteration t

- One solution x_i for each sub pb. i of direction λ_i
- Scalar function g :
  Weighted Tchebycheff
- Representation of solutions in objective space : $z_i = g(x_i | \lambda_i, z_i^\star)$
- Same reference point for all sub-pb. $z^\star = z_1^\star = \ldots = z_\mu^\star$
- Neighborhood size $\sharp B(i) = T = 3$

MOEA/D : Multi-Obj. Evo. Algo. based on Decomposition

A lot of MO algo. :
Pareto based (NSGAII,...), indicator based (IBEA,...), and

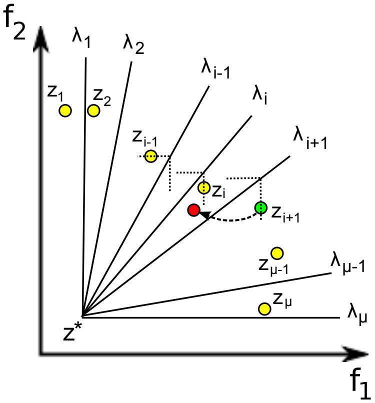


From the neigh. $B(i)$ of sub-pb. i, x_{i+1} is selected

- One solution x_i for each sub pb. i of direction λ_i
- Scalar function g :
  Weighted Tchebycheff
- Representation of solutions in objective space : $z_i = g(x_i | \lambda_i, z_i^\star)$
- Same reference point for all sub-pb. $z^\star = z_1^\star = \ldots = z_\mu^\star$
- Neighborhood size $\sharp B(i) = T = 3$

MOEA/D : Multi-Obj. Evo. Algo. based on Decomposition

A lot of MO algo. :
Pareto based (NSGAII,...), indicator based (IBEA,...), and

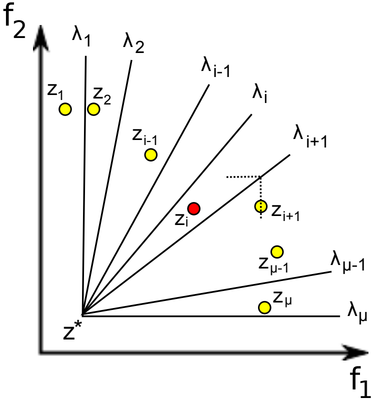


The mutated solution y is created

- One solution x_i for each sub pb. i of direction λ_i
- Scalar function g :
  Weighted Tchebycheff
- Representation of solutions in objective space : $z_i = g(x_i|\lambda_i, z_i^\star)$
- Same reference point for all sub-pb. $z^\star = z_1^\star = \ldots = z_\mu^\star$
- Neighborhood size $\sharp B(i) = T = 3$

MOEA/D : Multi-Obj. Evo. Algo. based on Decomposition

A lot of MO algo. :
Pareto based (NSGAII,...), indicator based (IBEA,...), and

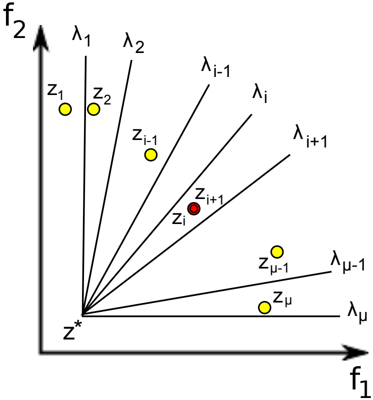


According to scalar fonction,
y is worst than x_{i-1},
y is better than x_i and replaces it.

- One solution x_i for each sub pb. i of direction λ_i
- Scalar function g :
  Weighted Tchebycheff
- Representation of solutions in objective space : $z_i = g(x_i|\lambda_i, z_i^\star)$
- Same reference point for all sub-pb. $z^\star = z_1^\star = \ldots = z_\mu^\star$
- Neighborhood size $\sharp B(i) = T = 3$

MOEA/D : Multi-Obj. Evo. Algo. based on Decomposition

A lot of MO algo. :
Pareto based (NSGAII,...), indicator based (IBEA,...), and



According to scalar fonction,
y is also better than x_{i+1}
and replaces it for the next iteration.

- One solution x_i for each sub pb. i of direction λ_i
- Scalar function g :
  Weighted Tchebycheff
- Representation of solutions in objective space : $z_i = g(x_i | \lambda_i, z_i^\star)$
- Same reference point for all sub-pb. $z^\star = z_1^\star = \ldots = z_\mu^\star$
- Neighborhood size $\sharp B(i) = T = 3$

Massively parallel algorithm

Optimization of problems based on expensive simulation

- Relevant tuning of parameters of the algorithm
- Surrogate model
- Parallel computing

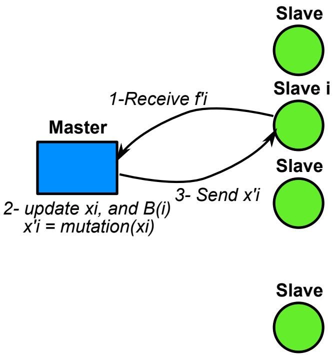
Here,
Simulation for one burnup : 10 min
Simulation of 4 burnups (life cycle) : 40min

Massive parallel system (HPC)

Algorithms for the TGCC (GENCI Projet)
2 500 000 hours of available computation
1008 cores for 24h of computation.

Asynchronous MOEA/D

Master-slaves architecture



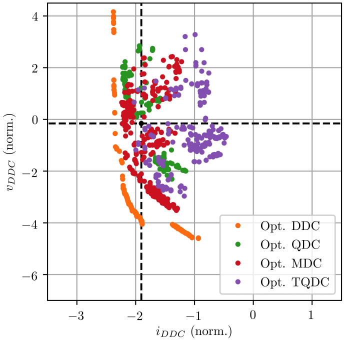
Algorithm on Master

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

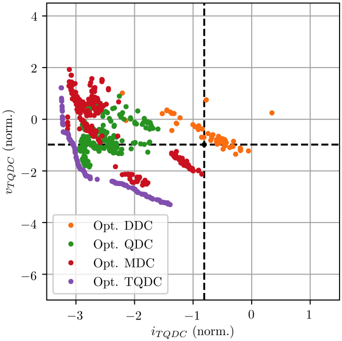
V. Drouet, S. Verel, and J-M. Do. "Surrogate-assisted asynchronous multiobjective algorithm for nuclear power plant operations.", Gecco 2020. [4]

Results at different burnups

At the beginning of exploitation



At the end of exploitation



Optimization on the whole cycle is necessary

Asynchronous MOEA/D with surrogate model

Algorithm on Master

$\{x_1, \dots, x_\lambda\} \leftarrow$ **Initialization**()
for $i = 1..\lambda$ **do**
  Send (Non-blocking) x_i to slave S_i
end for
repeat
  if there is a pending mess. from S_i **then**
    Receive fitness f_i' of x_i' from S_i
    Add (x_i', f_i') to sample S
    Update x_i, and $x_j \in B(i)$ with (x_i', f_i')
    Train model M with Sample S
    if $|S| < N_{start}$ **then**
      $x_i' \leftarrow \texttt{mutation}(x_i)$
    else
      Select x_i' using model M
    end if
    Send (Non-blocking) x_i' to slave S_i
  end if
until time limit

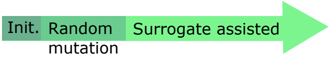
Surrogate model

Random forest

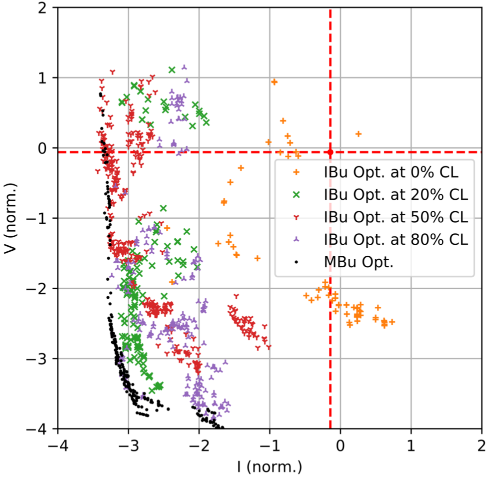
Offline tuning using data

Acceleration of convergence :
Double the prob. of improv.

Surrogate model can be misleading
(poor accuracy at the begin.) :

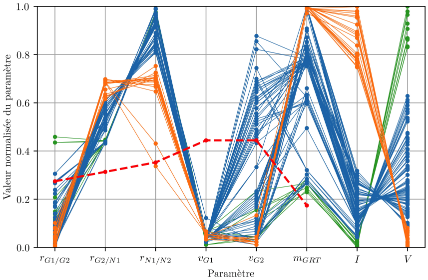


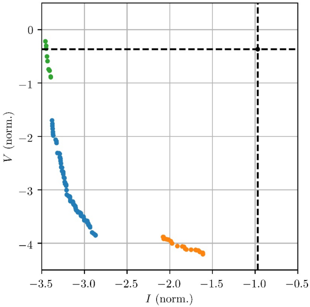
Results on whole exploitation cycle



Reduce Volume of effluent, and Instability (axial offset) from current setting

Analysis of decision variables





A posterio interpretation of the candidate solutions on Pareto front

Conclusion

Optimization, machine learning techniques to solve design problems with digital twins

- Main tools :
  Analysis fitness landscape to understand pb., and tune algo.
  Use surrogate models, to accelerate the search, and bring an algebraic model
  Parallel, and distributed computation
- A good algorithm is a tradeoff between :
  Final decision making
  Search space dimension, and its properties
  Computation time, and power.
- Digital twins, and AI offers a lot of perspectives
  How to combine different methods ?
  How to better understand systems ? ...

Ricardo Baptista and Matthias Poloczek.
Bayesian optimization of combinatorial structures.
In *International Conference on Machine Learning*, pages 462–471. PMLR, 2018.

Thomas Bartz-Beielstein and Martin Zaefferer.
Model-based methods for continuous and discrete global optimization.
Applied Soft Computing, 55 :154–167, 2017.

Francisco Chicano, Darrell Whitley, Gabriela Ochoa, and Renato Tinós.
Optimizing one million variable nk landscapes by hybridizing deterministic recombination and local search.
In *Proceedings of the genetic and evolutionary computation conference*, pages 753–760, 2017.

V Drouet, Sébastien Verel, and J-M Do.
Surrogate-assisted asynchronous multiobjective algorithm for nuclear power plant operations.

In *Proceedings of the 2020 genetic and evolutionary computation conference*, pages 1073–1081, 2020.

V. Drouet, Sébastien Verel, and J.-M. Do.
Surrogate-assisted asynchronous multiobjective algorithm for nuclear power plant operations.
In *2020 Genetic and Evolutionary Computation Conference (GECCO '20)*, GECCO '20 : Proceedings of the 2020 Genetic and Evolutionary Computation Conference, pages 1073–1081, Cancún, Mexico, July 2020. ACM.

Valentin Drouet, Jean-Michel Do, and Sébastien Verel.
Optimization of load-follow operations of a 1300MW pressurized water reactor using evolutionnary algorithms.
In M. Margulis and P. Blaise, editors, *International Conference on Physics of Reactors : Transition to a Scalable Nuclear Future (PHYSOR 2020)*, volume 247, page 11001, Cambridge, United Kingdom, March 2020.

Valentin Drouet, Jean-Michel Do, Sébastien Verel, and Jean-Charles Le Pallec.
Design of a simulator oriented PWR model and optimization of load-follow operations.
In *International Congress on Advances in nuclear Power Plants (ICAPP)*, Juan-les-pins, France, May 2019.

Firas Hamze, Darryl C Jacob, Andrew J Ochoa, Dilina Perera, Wenlong Wang, and Helmut G Katzgraber.
From near to eternity : spin-glass planting, tiling puzzles, and constraint-satisfaction problems.
Physical Review E, 97(4) :043303, 2018.

Donald R Jones, Matthias Schonlau, and William J Welch.
Efficient global optimization of expensive black-box functions.
Journal of Global optimization, 13(4) :455–492, 1998.

Florian Leprêtre, Cyril Fonlupt, Sébastien Verel, and Virginie Marion.

SIALAC Benchmark : On the design of adaptive algorithms for traffic lights problems.
GECCO 2018, July 2018.
Poster.

Florian Leprêtre, Cyril Fonlupt, Sébastien Verel, and Virginie Marion.
Combinatorial Surrogate-Assisted Optimization for Bus Stops Spacing Problem.
In *Biennial International Conference on Artificial Evolution (EA 2019)*, Mulhouse, France, 2019.

Florian Leprêtre, Cyril Fonlupt, Sébastien Verel, Virginie Marion, Rolando Armas, Hernán Aguirre, and Kiyoshi Tanaka.
Fitness landscapes analysis and adaptive algorithms design for traffic lights optimization on sialac benchmark.
Applied Soft Computing, 85 :105869, 2019.

Florian Leprêtre, Sébastien Verel, Cyril Fonlupt, and Virginie Marion.

Walsh functions as surrogate model for pseudo-boolean optimization problems.
In *The Genetic and Evolutionary Computation Conference (GECCO 2019)*, Proceedings of the Genetic and Evolutionary Computation Conference, pages 303–311, Prague, Czech Republic, July 2019. ACM.

Mathieu Muniglia, Jean-Michel Do, Le Pallec Jean-Charles, Hubert Grard, Sébastien Verel, and S. David.
A Multi-Physics PWR Model for the Load Following.
In *International Congress on Advances in Nuclear Power Plants (ICAPP)*, San Francisco, United States, April 2016.

Mathieu Muniglia, Le Pallec Jean-Charles, Jean-Michel Do, and Sebastien Verel.
Design of a load following management for a PWR reactor using an optimization method.
In *MC 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science &*

Engineering, International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, South Korea, April 2017.

Mathieu Muniglia, Sébastien Verel, Le Pallec Jean-Charles, and Jean-Michel Do.
Massive asynchronous master-worker EA for nuclear reactor optimization : a fitness landscape perspective.
In *Genetic and Evolutionary Computation Conference (GECCO 2017)*, GECCO '17 Proceedings of the Genetic and Evolutionary Computation Conference Companion, Berlin, Germany, July 2017.

Mathieu Muniglia, Sébastien Verel, Jean-Charles Le Pallec, and Jean-Michel Do.
A Fitness Landscape View on the Tuning of an Asynchronous Master-Worker EA for Nuclear Reactor Design.
In *International Conference on Artificial Evolution (Evolution Artificielle)*, pages 30–46, Paris, France, October 2017.

Nestor V Queipo, Raphael T Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P Kevin Tucker.
Surrogate-based analysis and optimization.
Progress in aerospace sciences, 41(1) :1–28, 2005.

Sébastien Verel, Bilel Derbel, Arnaud Liefooghe, Hernan Aguirre, and Kiyoshi Tanaka.
A surrogate model based on Walsh decomposition for pseudo-boolean functions.
In *International Conference on Parallel Problem Solving from Nature (PPSN 2018)*, volume 11102 of *Lecture Notes in Computer Science*, pages 181–193, Coimbra, Portugal, September 2018.

Martin Zaefferer, Jörg Stork, Martina Friese, Andreas Fischbach, Boris Naujoks, and Thomas Bartz-Beielstein.
Efficient global optimization for combinatorial problems.
In *Proceedings of the 2014 annual conference on genetic and evolutionary computation*, pages 871–878, 2014.