Fonctions de Walsh pour le benchmarking des métaheuristiques

Sara Tari, Mahmoud Omidvar, and Sébastien Verel

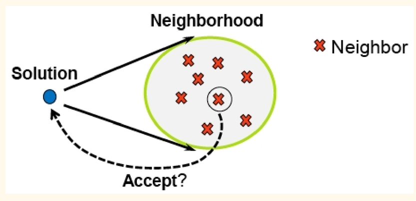
Université du Littoral Côte d'Opale, LISIC, France

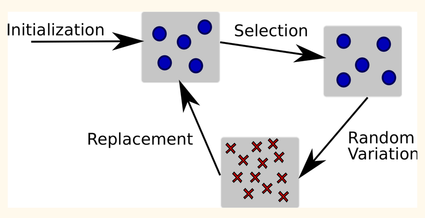
"gradient free" optimization algorithms

Principle: enumeration of a subset of the search space

- Many ways to enumerate the search space
  - Monte Carlo, bayesian optimization, ...
  - Local search, and evolutionary algorithms

Local search / Evolutionary algorithms





Question and method

- A lot of problem instances
- A lot of optimization algorithms (and parameters)

How to tune, or select an efficient algorithm according to the problem instance ?

Problem —Extract→ Features —Learn→ Algo.
- algebric
- geometric (Fitness landscape)

Benchmark

Pseudo-boolean functions: $f : \{0,1\}^n \rightarrow \mathbb{R}$

Use a benchmark to test algo, and learn pb. *vs.* algo.

Ex. in quantum computing, operational research, etc.

PHYSICAL REVIEW E **101**, 023316 (2020)

Computational hardness of spin-glass problems with tile-planted solutions

Dilina Perera,[1,*] Firas Hamze,[2] Jack Raymond,[2] Martin Weigel,[3] and Helmut G. Katzgraber[4,1,5]

[1]*Department of Physics and Astronomy, Texas A&M University, College Station, Texas 77843-4242, USA*
[2]*D-Wave Systems, Inc., 3033 Beta Avenue, Burnaby, British Columbia, Canada V5G 4M9*
[3]*Centre for Fluid and Complex Systems, Coventry University, Coventry, CV1 5FB, England*
[4]*Microsoft Quantum, Microsoft, Redmond, Washington 98052, USA*
[5]*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501, USA*

(Received 24 July 2019; revised manuscript received 16 January 2020; accepted 5 February 2020; published 28 February 2020)

Walsh functions

Definition [Bethke:1980]
For any $k \in [0, 2^n - 1]$, Walsh function

$$\varphi_k : \{0,1\}^n \rightarrow \{-1,1\}$$

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

$(\varphi_0, \ldots, \varphi_{2^n-1})$ is an orthogonal basis:

x	φ_0	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7
$0 = 000$	1	1	1	1	1	1	1	1
$1 = 001$	1	−1	1	−1	1	−1	1	−1
$2 = 010$	1	1	−1	−1	1	1	−1	−1
$3 = 011$	1	−1	−1	1	1	−1	−1	1
$4 = 100$	1	1	1	1	−1	−1	−1	−1
$5 = 101$	1	−1	1	−1	−1	1	−1	1
$6 = 110$	1	1	−1	−1	−1	−1	1	1
$7 = 111$	1	−1	−1	1	−1	1	1	−1

Decomposition of pseudo-bool. func.

$$\forall x \in \{0,1\}^n, \quad f(x) = \sum_{k=0}^{2^n-1} w_k.\varphi_k(x)$$

$\forall k \in [0, 2^n - 1]$,

$$w_k = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x).\varphi_k(x)$$

$f(x) = w_4(-1)^{x_2} +$
$w_3(-1)^{x_0+x_1} + w_6(-1)^{x_1+x_2} + w_{10}(-1)^{x_1+x_3} + w_{12}(-1)^{x_2+x_3}$

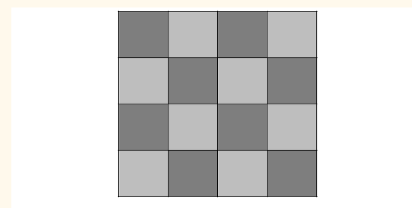
Applications

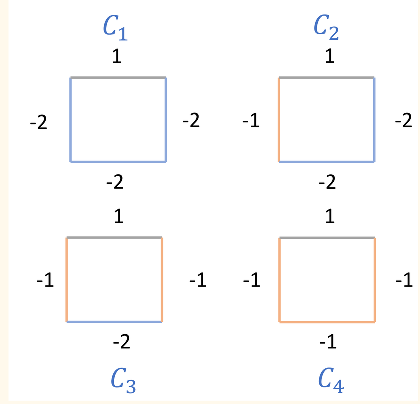
- Design deceptive functions
- Grey-box optimization : use linear decomposition for smart computation (see F. Chicano, D. Withley, etc.)
- Optimization based on digital twins (surrogate model)

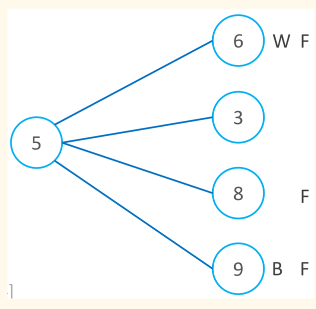
First experiments

$$\sum_k C_k(x) = \sum_{i,j} J_{ij}\sigma_i\sigma_j$$

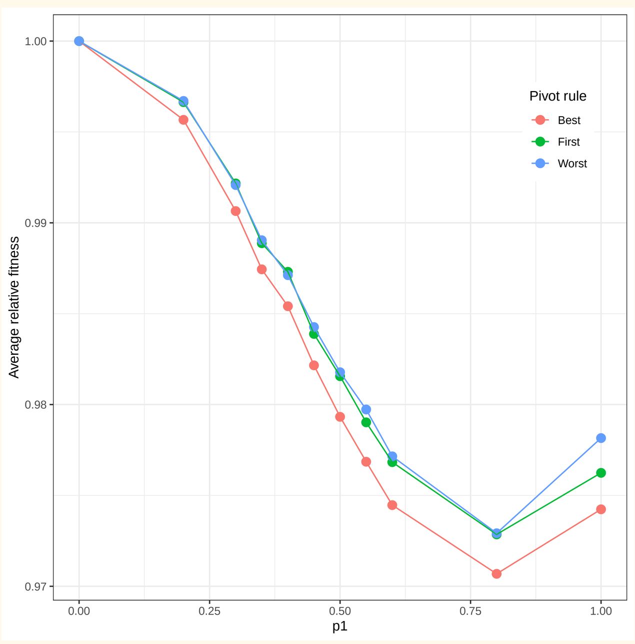
with $\sigma_i \in \{-1,1\}$



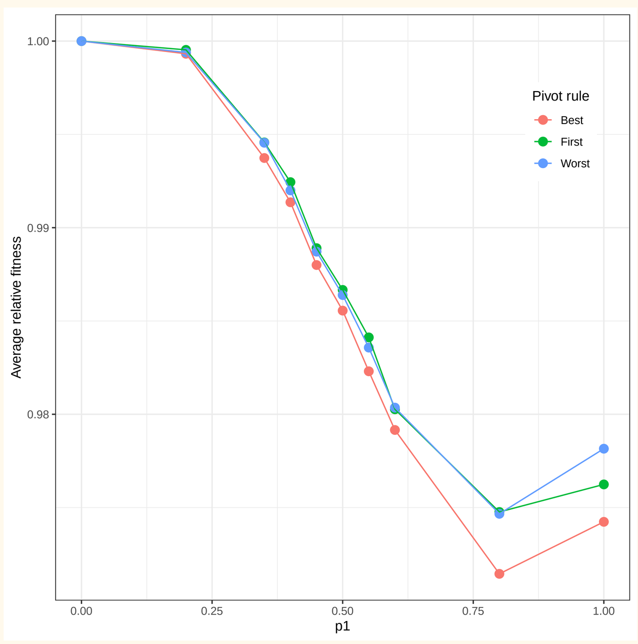




First results with Tile-Planting benchmark



Parameters p_1 *vs.* p_3



Parameters p_1 *vs.* p_4

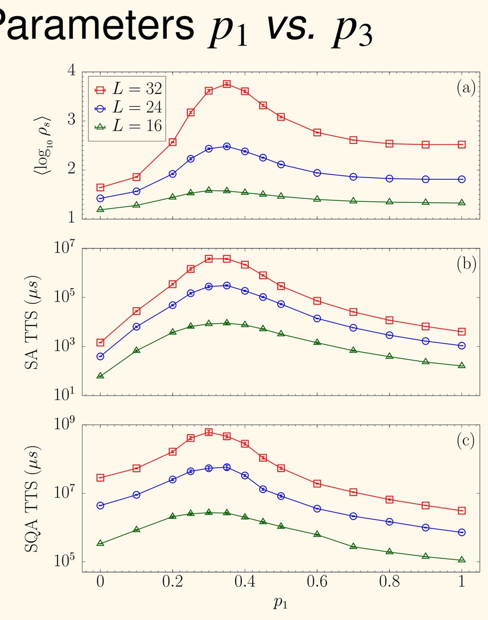


FIG. 12: (a) Population annealing $\langle \log_{10} \rho_s \rangle$, (b) simulated annealing optimal median TTS, and (c) simulated quantum annealing optimal median TTS for instance classes comprised of mixtures of C_1 and C_3 subproblem types. The results are plotted against p_1, which is the probability of choosing subproblems from class C_1. The curves are for three system sizes, $L = 16$, $L = 24$, and $L = 32$. $\langle \log_{10} \rho_s \rangle$ and median TTS values are estimated using 200 problems per instance class, per system size. All panels have the same horizontal scale.

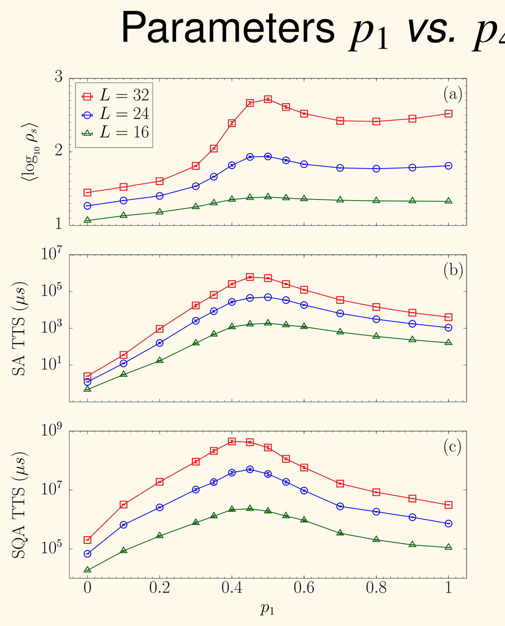


FIG. 13: (a) Population annealing $\langle \log_{10} \rho_s \rangle$, (b) simulated annealing optimal median TTS, and (c) simulated quantum annealing optimal median TTS for instance classes comprised of mixtures of C_1 and C_4 subproblem types. The results are plotted against p_1, which is the probability of choosing subproblems from class C_1. The curves are for three system sizes, $L = 16$, $L = 24$, and $L = 32$. $\langle \log_{10} \rho_s \rangle$ and median TTS values are estimated using 200 problems per instance class, per system size. All panels have the same horizontal scale.

Toward benchmark with importance

The function is defined by:

$$f(x) = \sum_{i=1}^{m} \alpha_i C_{k_i}(x_{i_1}, \ldots, x_{i_{a_i}})$$

where $\forall j \in 1,k, \; C_j : \{0,1\}^{a_j} \rightarrow \mathbb{R}$ are pseudo-boolean functions of arity a_j. $\forall i\{1,\ldots,m\}$, $\alpha_i \in \mathbb{R}$, and k_i is the index of the clause.

Importance
The variables $X = \{x_1, \ldots, x_n\}$ are split into k classes of importance: $c_i \subset X$ such that $\cup_k c_k = X$, and $c_i \cap c_j = \emptyset$.

Each class c_i of importance has a degree of importance d_i.
The probability that a variable of class c_i to appear in a clause is $p_i = \frac{d_i}{\sum_j d_j}$.

A parameter (factor) defined the probability that the same class of importance appear in the same clause:

$$Pr(cl(x_{i_1}) = c_1, cl(x_{i_2}) = c_2, \ldots cl(x_{i_a}) = c_a)$$

When the variable importances are independent:

$$Pr(cl(x_{i_1}) = c_1, cl(x_{i_2}) = c_2, \ldots cl(x_{i_a}) = c_a) = p_{c_1}p_{c_2}\ldots p_{c_a}$$

$\alpha_i = 1$, or $\alpha_i = (\Pi_i d_i)^{1/a_i}$

Discussion

- Design domain "independent" benchmark based on Walsh functions
- Compare benchmark, and algorithme with quantum computing
- Representation of real-world problems in Walsh basis of function

Perspectives

- Build a large set of instances, and test a large class of algorithms
- Apply such techniques on expensive multiobjective optimization problems: Combinatorial problems ou mixed optimization problems based numerical simulation