

Présentation de la recherche

~

Séminaire scientifique

Arnaud Liefoghe
arnaud.liefoghe@univ-littoral.fr

Organisation

- **Initiation aux fonctionnements de la recherche**

(2 séances)



- **Séminaire scientifique** (8 séances)

Thème scientifique lié à la spécialité et découverte pratique d'une problématique

Organisation

- Supports + rendus : **Moodle**

<https://portail.eilco.fr:28/>

- Poser des **questions** durant les séances
- Beaucoup de problèmes ouverts, ne pas hésiter à **chercher par vous même et à proposer des solutions !**

Evaluation

Contrôle continu

Projet (par binôme, un seul rendu final sur Moodle)

- > compte-rendu sous forme d'un **rapport scientifique**
(10-15 pages)
- > **annexe** avec résultats détaillés reproductibles
- > développement : code, scripts... reproductibles

Environment scientifique local

A2U



7 grandes thématiques

1. Intelligence artificielle et optimisation
2. Énergie
3. Mer et Littoral
4. Environnement et ressources naturelles
5. Santé
6. Adaptation sociétale aux mutations
7. Cultures, Territoires & Patrimoines

ULCO



LISIC

Le LISIC est composé de 4 équipes de recherche

IC

Ingénierie
des Connaissances

OSMOSE

Optimisation Simulation
MOdeliSation Evolutionnaire

IMAP

Images
et Apprentissage

SPECIFI

Systèmes de Perception
et Fusion d'Informations

et de 3 groupes projet

LEC

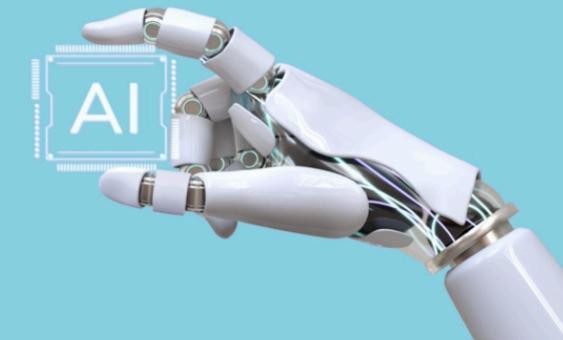
LANDSCAPE-AWARE
EVOLUTIONARY COMPUTATION

HYPERBDD

CRÉATION DE BD
D'IMAGES HYPERSPECTRALES

ARTISIS

ARTIFICIAL INTELLIGENCE
FOR IMAGE SYNTHESIS



<https://lisic-prod.univ-littoral.fr/>

LISIC : Laboratoire Informatique, Signal, Image de la Côte d'Opale

Création en 2010 (LIL + LASL)

- 44 enseignants-chercheurs permanents
- 23 doctorants
- 2 post-docs
- 1 ingénieur de recherche
- 4 équipes, 3 groupes projet

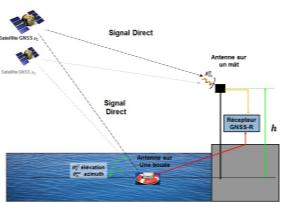


Calais Saint Omer

Mots clés :
Traitement du signal, Fusion d'informations, Perception multi-capteurs,
Synthèse d'images, Modélisation/Simulation/Optimisation, Apprentissage automatique

SPECIFI

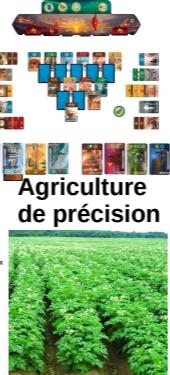
Systèmes de
Perception et Fusion
d'Informations



Décision dans l'incertain



Agriculture de précision



OSMOSE

Optimisation
Simulation
MOdellisation
Evolutionnaire

Transition énergétique

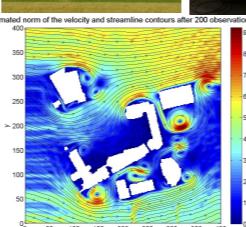
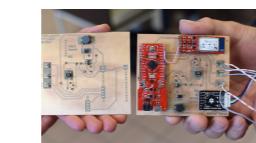


Thèmes :
- Modélisation, simulation
- Optimisation
- Apprentissage par renforcement, aide décision

Méthodes :
- automates à évé. discrets
- Algo. Évolutionnaires
- Optimisation Multi-objective
- GAN, Multi-Armed bandit, DeepL

Thèmes :
- Fusion multicapteurs : Lidar, GNSS, acoustique sous marine, Imagerie hyperspectrale, etc.

Méthodes :
Estimation circulaire,
Estimation dynamique,
Factorisation matricielle

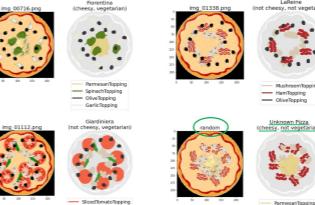


Agriculture de précision

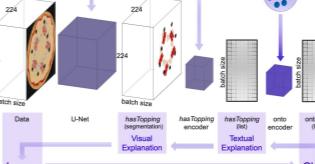
Jumeau Numérique



Rendu Image (HDR)



Explicabilité



IMAP

Images et Apprentissage



Thèmes :
- Classification de données
- Synthèse d'images

Méthodes :
- HMM, classification Spectrale
- Deep ML, optimisation
- Monte Carlo



IC

Ingénierie des Connaissances

Thèmes :
- Explicabilité (IA)
- systèmes réflexifs,
- industrie 4.0

Méthodes :
- intégration d'ontologiques
- Deep ML

thématique du séminaire

Analyse expérimentale et benchmarking de modèles et algorithmes d'IA

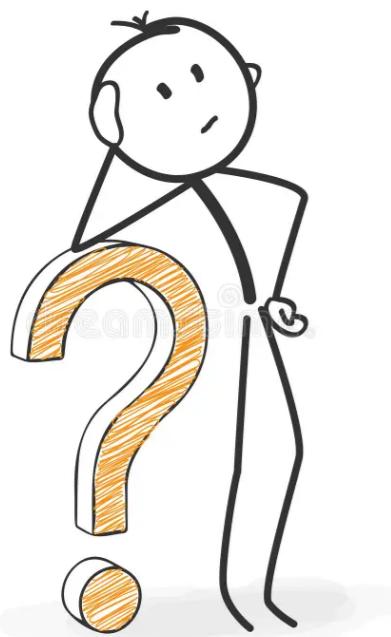
thématique du séminaire

Analyse expérimentale et benchmarking de **modèles et algorithmes d'IA**



thématique du séminaire

Analyse expérimentale et benchmarking de modèles et algorithmes d'IA



Objectifs du cours

- Comprendre la **démarche scientifique**
- Sensibilisation à / découverte de la problématique d'**analyse expérimentale** en IA
- Focus sur le **benchmarking** en **optimisation** boite-noire
- Savoir analyser des **expériences** (informatiques), présenter et interpréter les **résultats** de façon rigoureuse
- **Reproductibilité** du code et des données (artifacts)
- **Recommendation** interprétable et **hyper-paramétrage**

Démarche scientifique (en informatique)

Sébastien Verel, Cours de démarche scientifique

Démarche scientifique

Recherche vs. Ingénierie

- Rôle de la bibliographie et de l'état de l'art
- Publications et relectures par les pairs
- Production = connaissances vs. solutions

La démarche scientifique

Observations

J'observe un phénomène et me pose une question



Conclusion

j'analyse mes résultats et je conclue en validant ou non mes hypothèses



J'observe, prends des notes, schématisé. etc.

Hypothèses

J'émet des hypothèse : je pense ..car ..



Expérimentations

Je réalise des expériences afin de valider mes hypothèses



Informatique expérimentale

- Situation typique 1 : comparaison de méthodes
 - **Problématique** : reconnaissance de visage, représentation d'une scène 3d, architecture d'une application, détection d'une substance, communication de données, etc.
 - **Hypothèse** : « L'algorithme SN est meilleur que algorithme OP »
 - **Méthodologie** : Définition benchmark + mesures de performance
 - ⇒ Comparaison des algorithmes SN et OP selon les mesures
- L'hypothèse est réfutable, donc **hypothèse scientifique**
- Hypothèse **corroboree** ou **réfutée** : SN meilleur ou non que OP

Informatique expérimentale

- Situation typique 2 : acquisition de nouvelles connaissances
 - **Problématique** : reconnaissance de visage, représentation d'une scène 3d, architecture d'une application, détection d'une substance, communication de données, etc.
 - **Hypothèse** : « L'algo SN meilleur car il profite de nombreux 01 »
 - **Méthodologie** : Définition benchmark avec et sans la propriété + mesures relatives à l'hypothèse
 - ⇒ Comparaison / corrélation des mesures
- L'hypothèse est réfutable, donc **hypothèse scientifique**
- Hypothèse **corroboree** ou **réfutée** : l'implication est vraie ou fausse

Informatique théorique

- En informatique « théorique », on démontre mathématiquement certaines propriétés des algorithmes. Ce n'est plus une démarche purement expérimentale même si on peut comparer les algorithmes de cette façon.
- Il est parfois difficile (voire impossible) de définir une mesure de comparaison « objective » : Pas de « vérité » absolue, relative aux tests.

Modélisation et simulation informatique

- Buts de la modélisation et de la simulation
 - Modèle pour **Comprendre** un phénomène, système, etc.
 - Modèle pour **Prévoir** l'évolution d'un système
 - La simulation informatique permet l'**exploration**, l'**expérimentation**
- Modèle
 - Simplification de la réalité
 - Représentation abstraite, formelle : « La carte et le territoire »
- Simulation
 - Mise en oeuvre d'un modèle, exécution, calcul d'un modèle

Démarche à partir de modèles

1. Formulation d'une question scientifique (problème)
2. Conception d'un modèle (expression d'une hypothèse)
3. Production de données à partir de la simulation du modèle
4. Validation du modèle par rapport au réel en fonction de la question posée (réfutabilité)
5. Ajustement ou non du modèle (retour à l'étape 2)

Pourquoi la simulation ?

La simulation permet l'exploration, l'expérimentation :

- A moindre coût (argent, temps, etc.)
- Manipuler de « l'inaccessible » (dont les conséquences sont réfutables)
- Tester des hypothèses

Conséquence

- Une fois un modèle validé, la simulation se substitue au phénomène
- ... sur lequel on peut formuler des questions (dans le cadre de valider du modèle)

Démarche scientifique

Science

- Corpus de connaissances organisées par une démarche intellectuelle particulière

Proposition scientifique

- Proposition réfutable dont on ne peut affirmer qu'elle ne sera jamais réfutée

Analyse expérimentale (a.k.a. analyse empirique)

H. Hoos & T. Stützle, Stochastic Local Search: Foundations and Applications (2004)

Performance

qualité



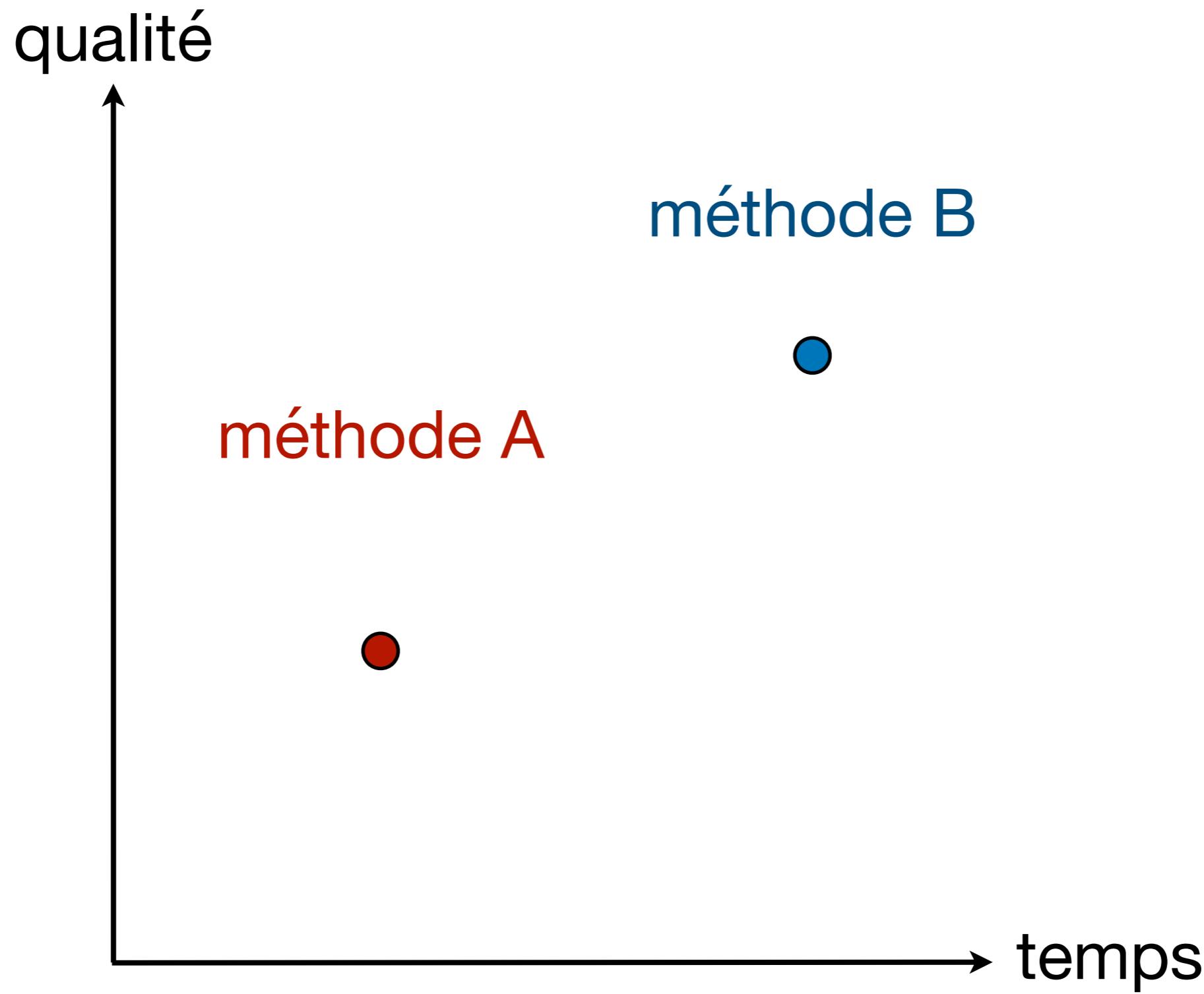
méthode A



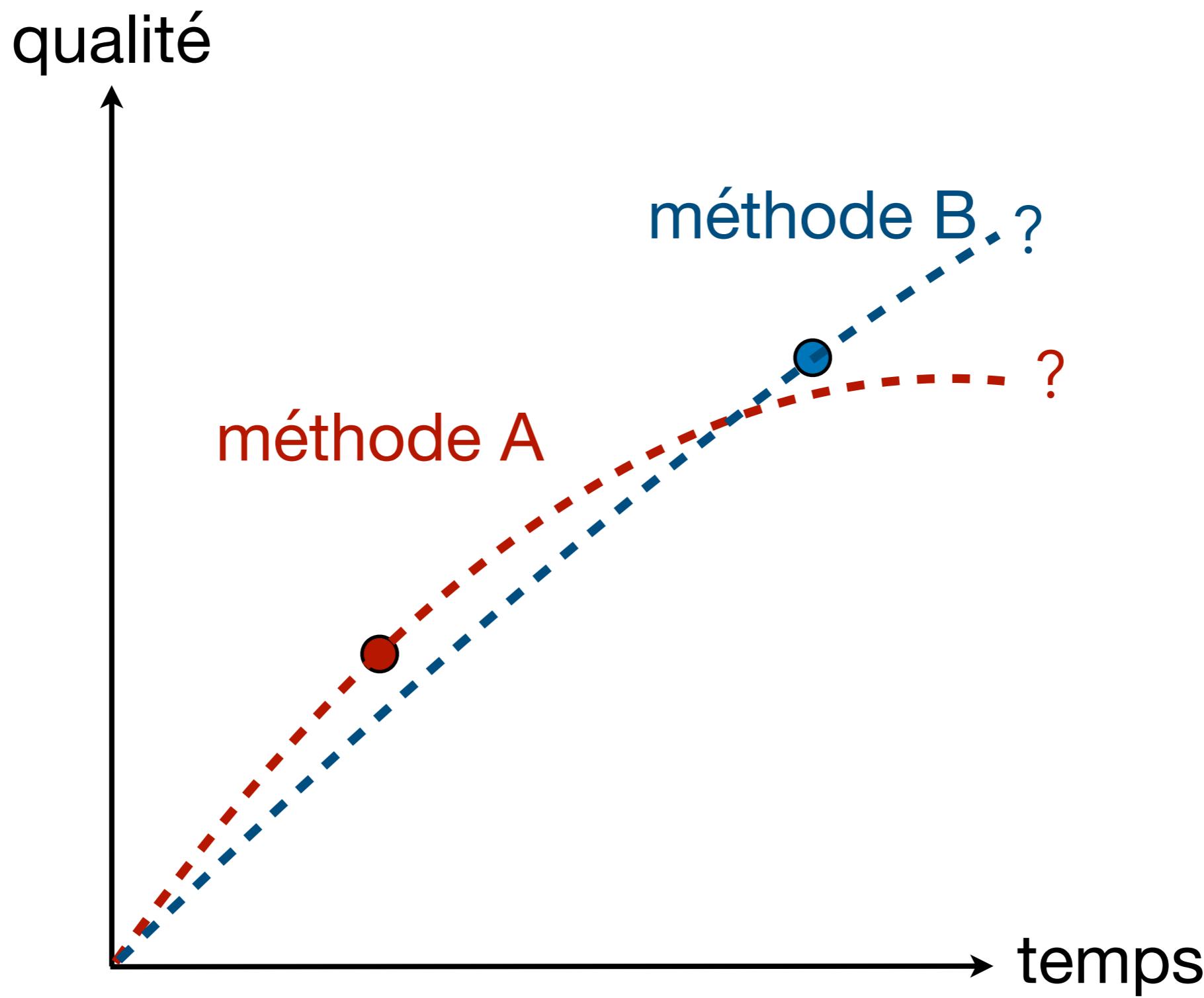
méthode B



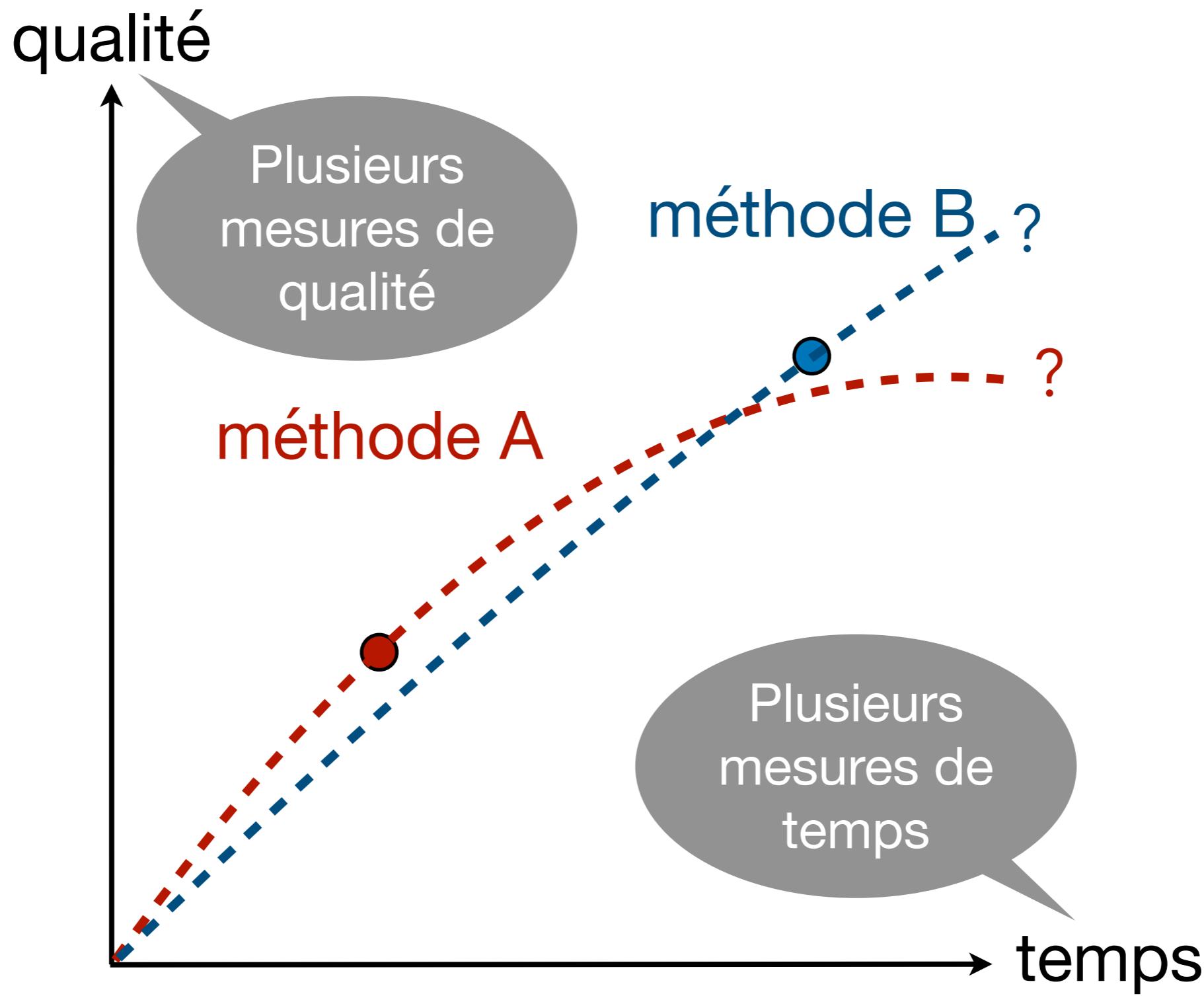
Performance x 2



Performance x n

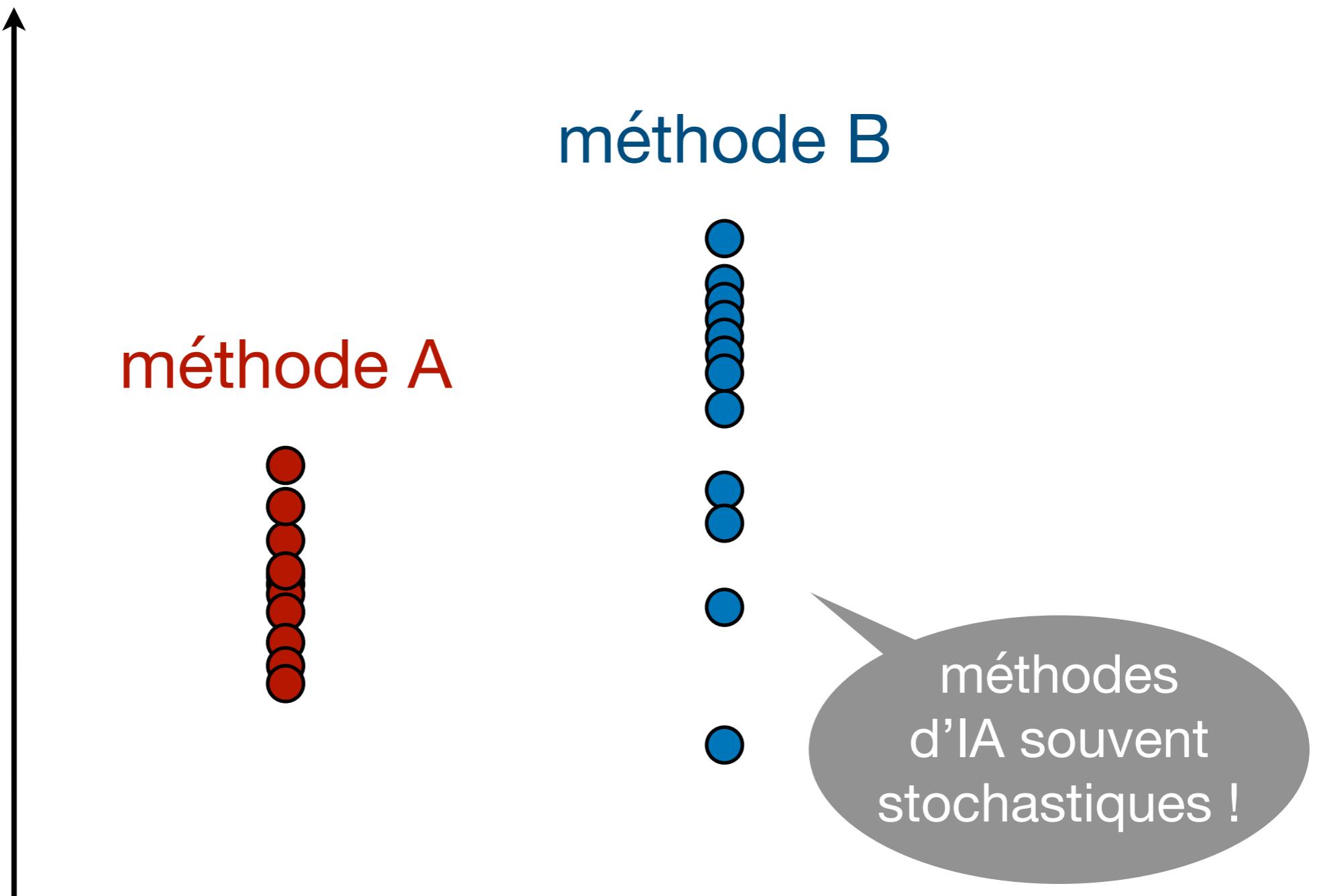


Performance x *



Performance

mesure de perf.



Performance x

Il existe plusieurs mesures de performance !

Quel but pour l'analyse ?

- Combien de temps pour identifier une solution réalisable ? Un modèle ?
- Quelle est la qualité de la solution / du modèle trouvés ?
- Quel méthode / opérateur / paramètre dois-je utiliser ?
- A quel point mon algorithme sait résoudre différents problèmes ? De différentes tailles ?
- Quel est la « meilleure » méthode ?

Analyse théorique ?

- Algorithmes stochastiques
- Complexité des problèmes traités
- Beaucoup de degrés de liberté

Analyse expérimentale

- Observations
- Hypothèse(s) (modèles)
- Concevoir des expériences
- Réaliser les expériences
- Analyser les résultats
- Et... recommencer !

Buts de l'analyse

- Performance de l'**algorithme** / du modèle
 - Adéquation envers les applications
 - Bases de **comparaison**
 - Caractéristiques du comportement
 - Facilité d'**amélioration**

Buts de l'analyse

- Performance des aspects **stochastiques**
 - Variabilité due à la **randomisation**
 - Robustesse par rapport au **paramétrage**
 - Robustesse par rapport aux **instances / données** de différents types
 - ... et de différentes **tailles**

Approche typique (1)

- Exécuter n fois avec un budget max
- Mesurer la qualité des solutions obtenues
 - ex. valeur objectif, déviation par rapport à l'optimum / meilleure solution connue, ...
- ... ou la précision du modèle obtenu
 - ex. R², (R)MSE, MAE, F1, ...
- ... par instance / jeu de données !

Approche typique (2)

- Exécuter n fois avec une qualité /précision minimale
- Mesurer les ressources requises
 - ex. temps CPU, nombre d'évaluations ou autres unités de complexité
 - ... par instance / jeu de données !

Approche typique (3)

- Exécuter n fois avec une qualité min ET un temps max
- Mesurer le taux de succès
- ... par instance / jeu de données !

Généralisation

- **Biais** des conclusions par rapport aux instances / données considérées ?
- Problématique du **sur-apprentissage** (overfitting),
... **sur-interprétation**
 - Données d'entraînement vs. données de test
 - Validation croisée (k-plis, etc.)
 - ...

Statistiques descriptives

- Moyenne
- Médianes (q0.5) et autres quartiles
(par exemple : q0.25, q0.75)
- Écart-type, variance, coefficient de variation (écart-type / moyenne)
- En raison de la stochasticité des algorithmes, les mesures de variabilité sont importantes !

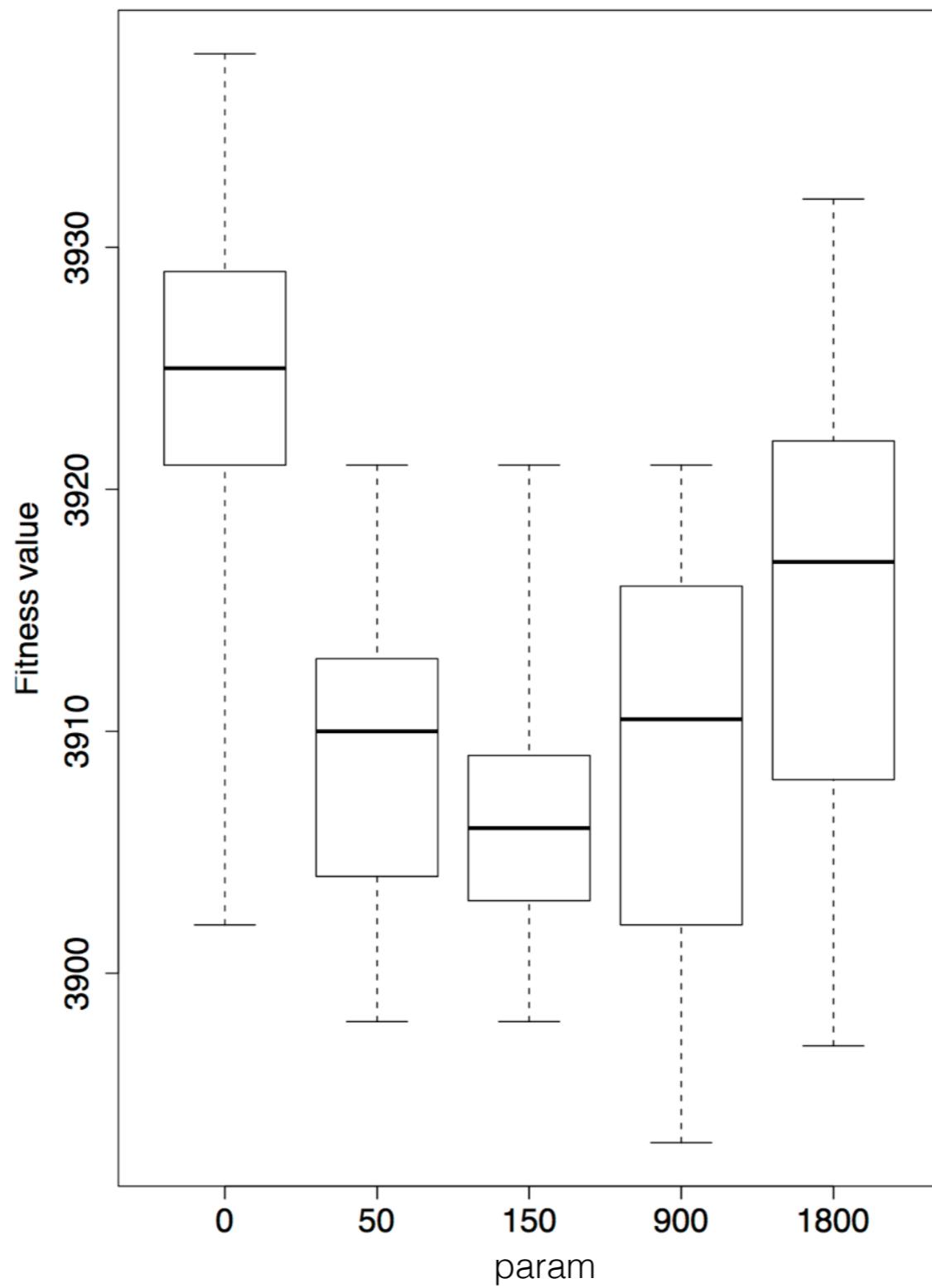
Exemple (1)

Performance d'un algo sur une instance SAT

mean	57 606.23	median	38 911
min	107	$q_{0.25}$; $q_{0.1}$	16 762; 5 332
max	443 496	$q_{0.75}$; $q_{0.9}$	80 709; 137 863
stddev	58 953.60	$q_{0.75}/q_{0.25}$	4.81
vc	1.02	$q_{0.9}/q_{0.1}$	25.86

stop = budget
mesure = fitness

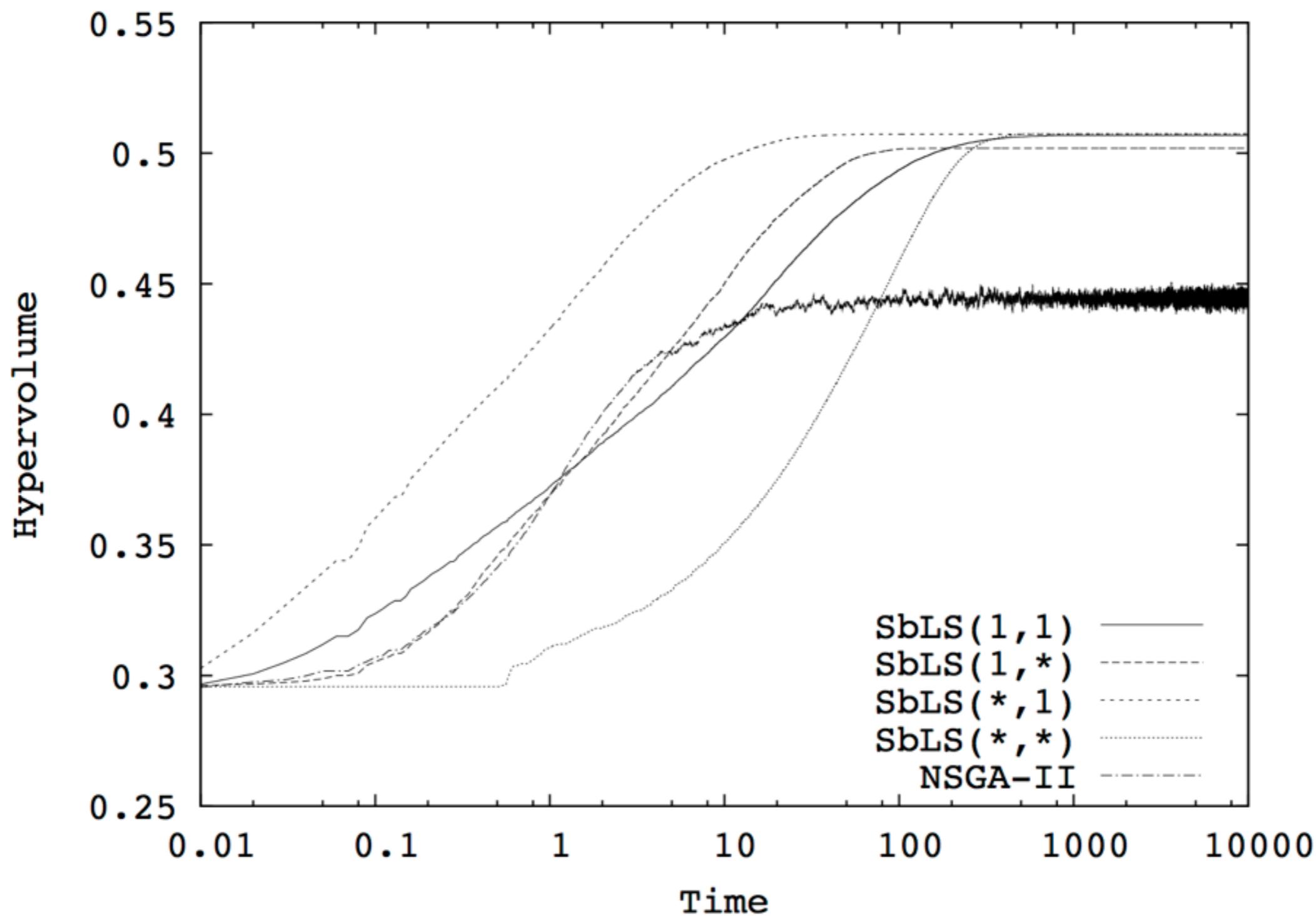
Exemple (2)



stop = budget
mesure = qualité

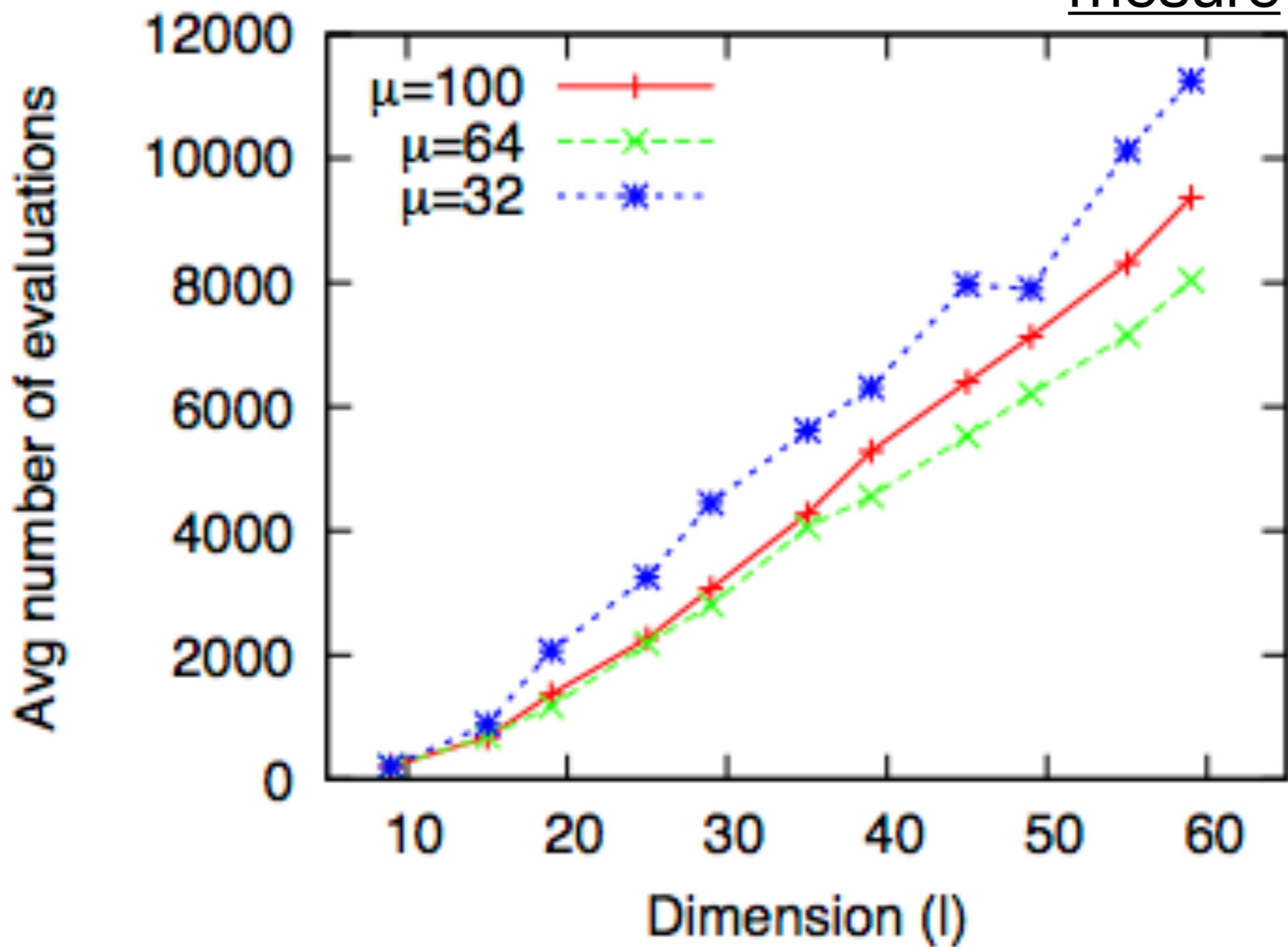
Exemple (3)

stop = budgets
mesure = qualité



Exemple (4)

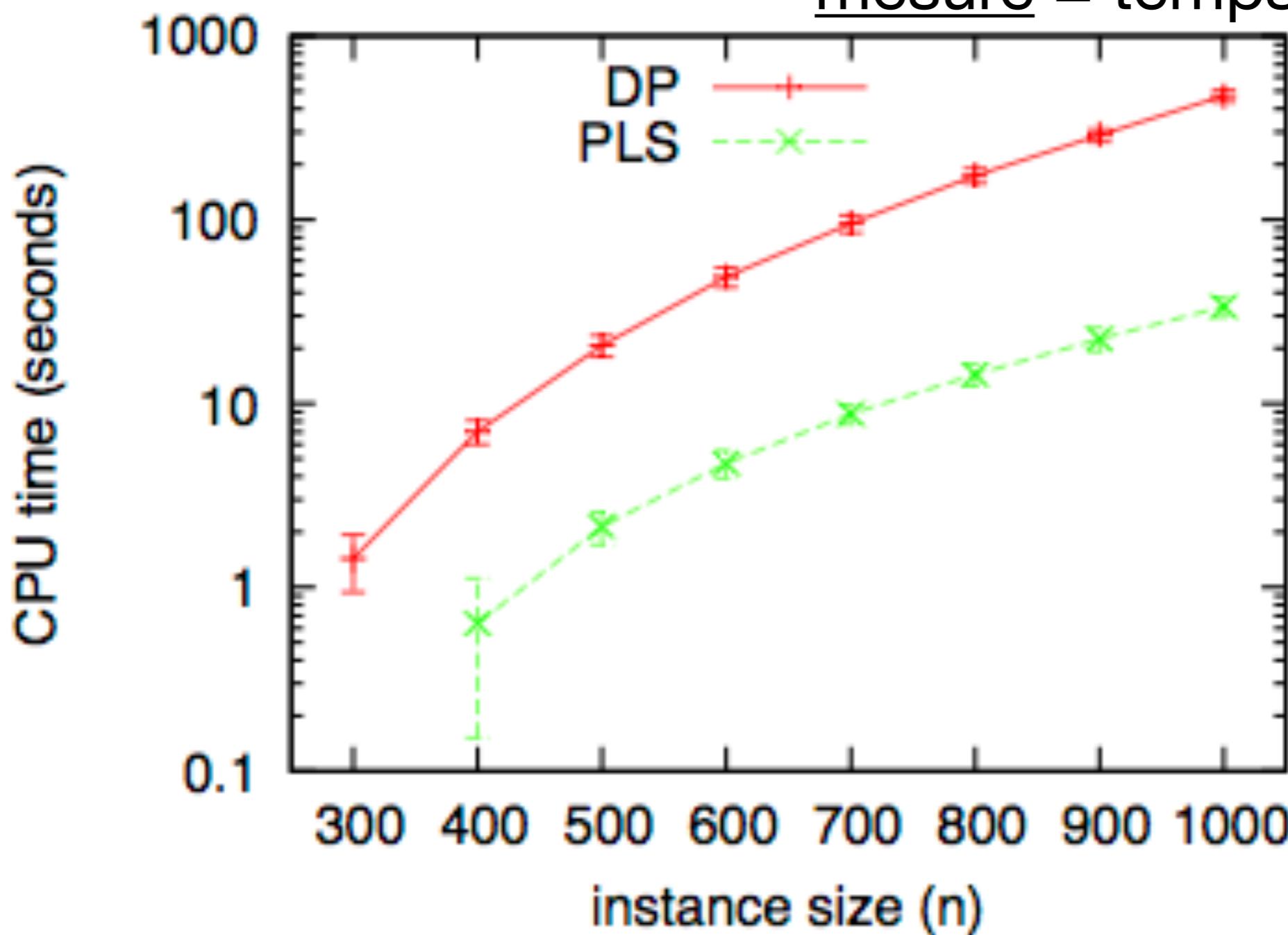
stop = %qualité
mesure = nb eval



Exemple (5)

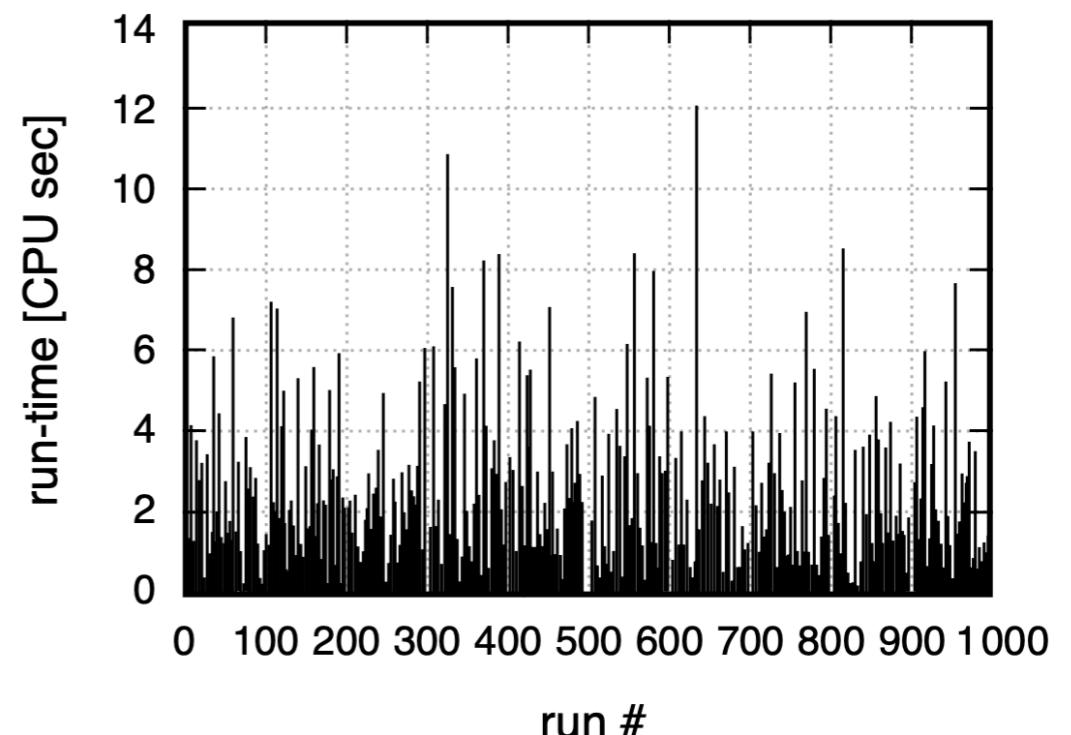
stop = %qualité

mesure = temps + variation



Tests statistiques

- La mesure (qualité, temps, taux de succès) d'un algorithme sur une instance est une **variable aléatoire**
 - Comparer 2 algorithmes
= comparer 2 distributions
- ⇒ Test d'hypothèse statistique



Test d'hypothèse

- Évaluer la validité des déclarations sur les propriétés ou sur les relations entre des ensembles de données statistiques
- Déclaration à tester (ou sa négation) = **hypothèse nulle (H_0)** du test
 - Exemple: Pour un test de Wilcoxon, l'hypothèse nulle est « les distributions de deux échantillons ont la même médiane »

Test d'hypothèse

- Risque de première espèce **alpha** = risque de rejeter à tort l'hypothèse nulle (faux positif)
 - Valeurs typiques de alpha = 0,05 ou 0,01
- L'application du test fournit une **p-valeur**
 - Probabilité que l'échantillon soit correct sous l'hypothèse nulle
 - $p\text{-value} \leq \alpha \Rightarrow H_0$ acceptée
- Risque de deuxième espèce (**puissance**) = ne pas rejeter une hypothèse fausse (faux négatif)

Test de rangs de Wilcoxon

- Test non-paramétrique
 - Aucune hypothèse sur la distribution sous-jacente des données (~~loi normale~~)
- Échantillons appariés
 - Test des rangs signés de Wilcoxon
- Échantillons non-appariés
 - Test de la somme des rangs de Wilcoxon

Exemple de Script R

```
# read solution quality (to be minimized) from 30 runs of algo1 on instance1
data_instance1_algo1 = scan(file = "instance1_algo1_min.30") →
# descriptive statistics
mean(data_instance1_algo1)
median(data_instance1_algo1)
min(data_instance1_algo1)
max(data_instance1_algo1)

# read solution quality (to be minimized) from 30 runs of algo2 on instance1
data_instance1_algo2 = scan(file = "instance1_algo2_min.30") →
# plot solution quality for both algorithms
boxplot(data_instance1_algo1, data_instance1_algo2, ylab = "solution quality (lower is better)", names = c("algo 1", "algo 2"))

# Wilcoxon signed-rank test for paired samples, alpha = 0.01
wilcox.test(data_instance1_algo1, data_instance1_algo2, paired = TRUE, conf.level = 0.99)
```

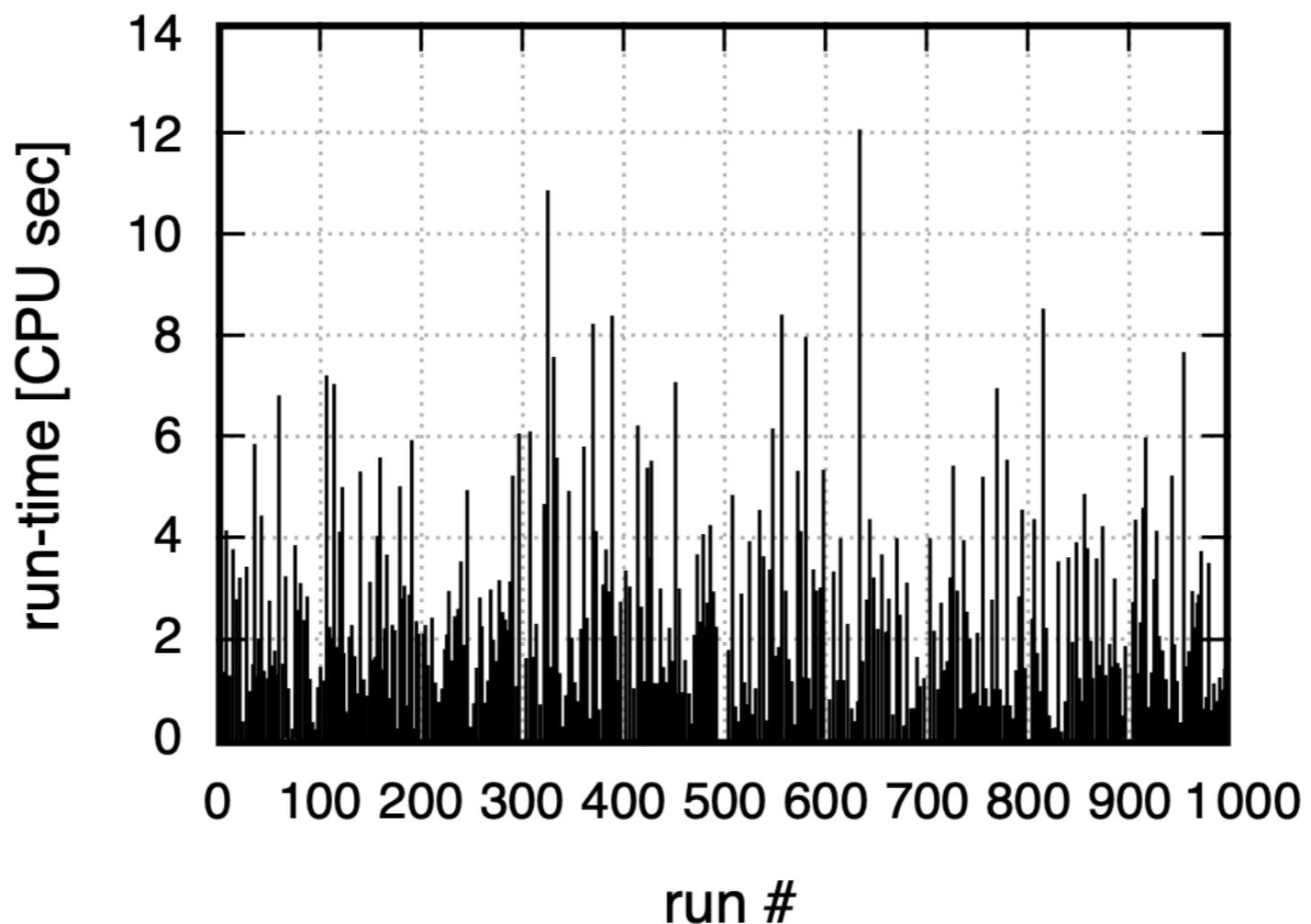
instance1_algo1_min.30

1.988591890e-02
1.829774749e-02
1.603592717e-02
...
1.762107468e-02

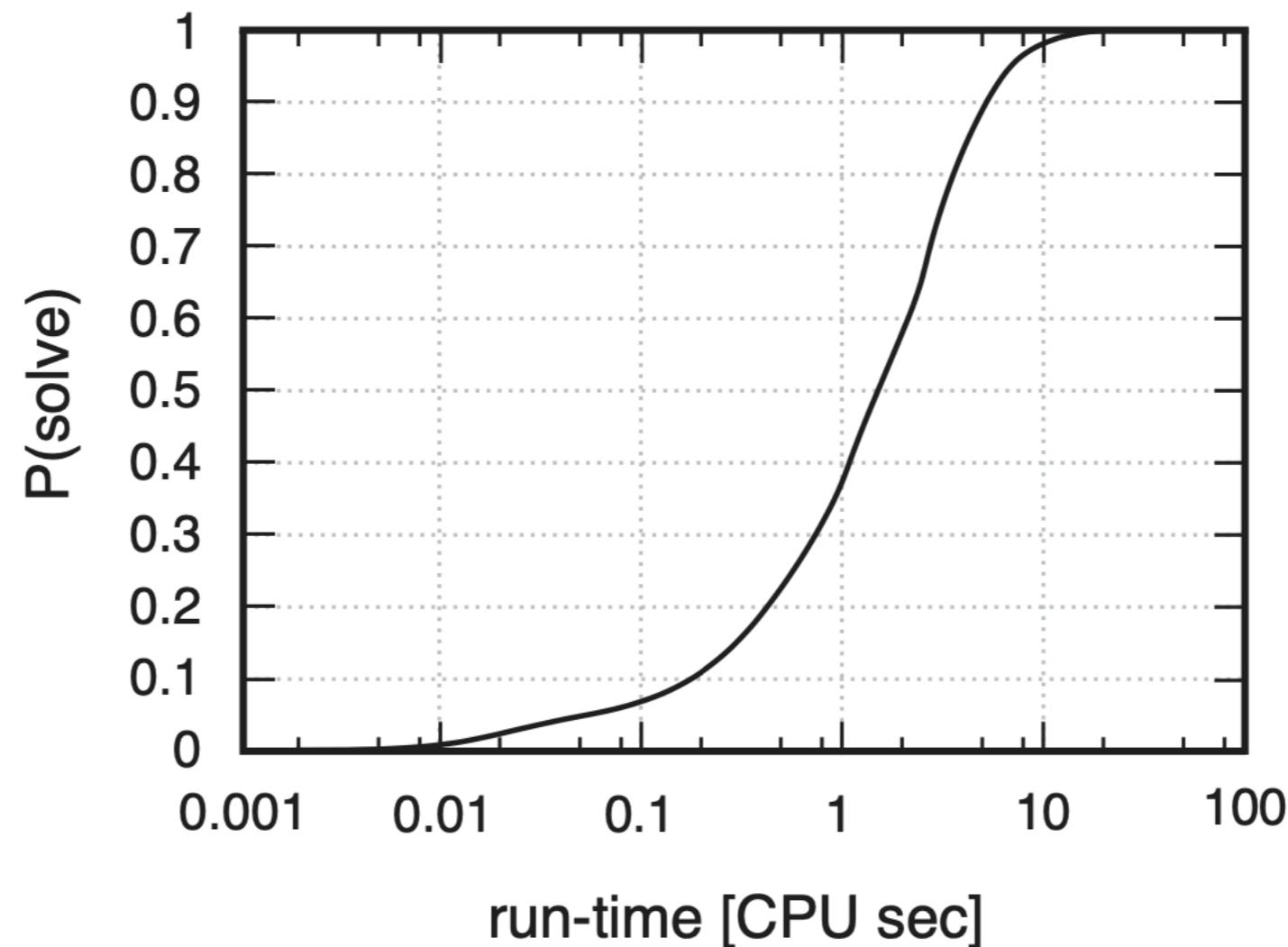
instance1_algo2_min.30

2.862727033e-02
2.611966079e-02
2.565107111e-02
...
2.870988739e-02

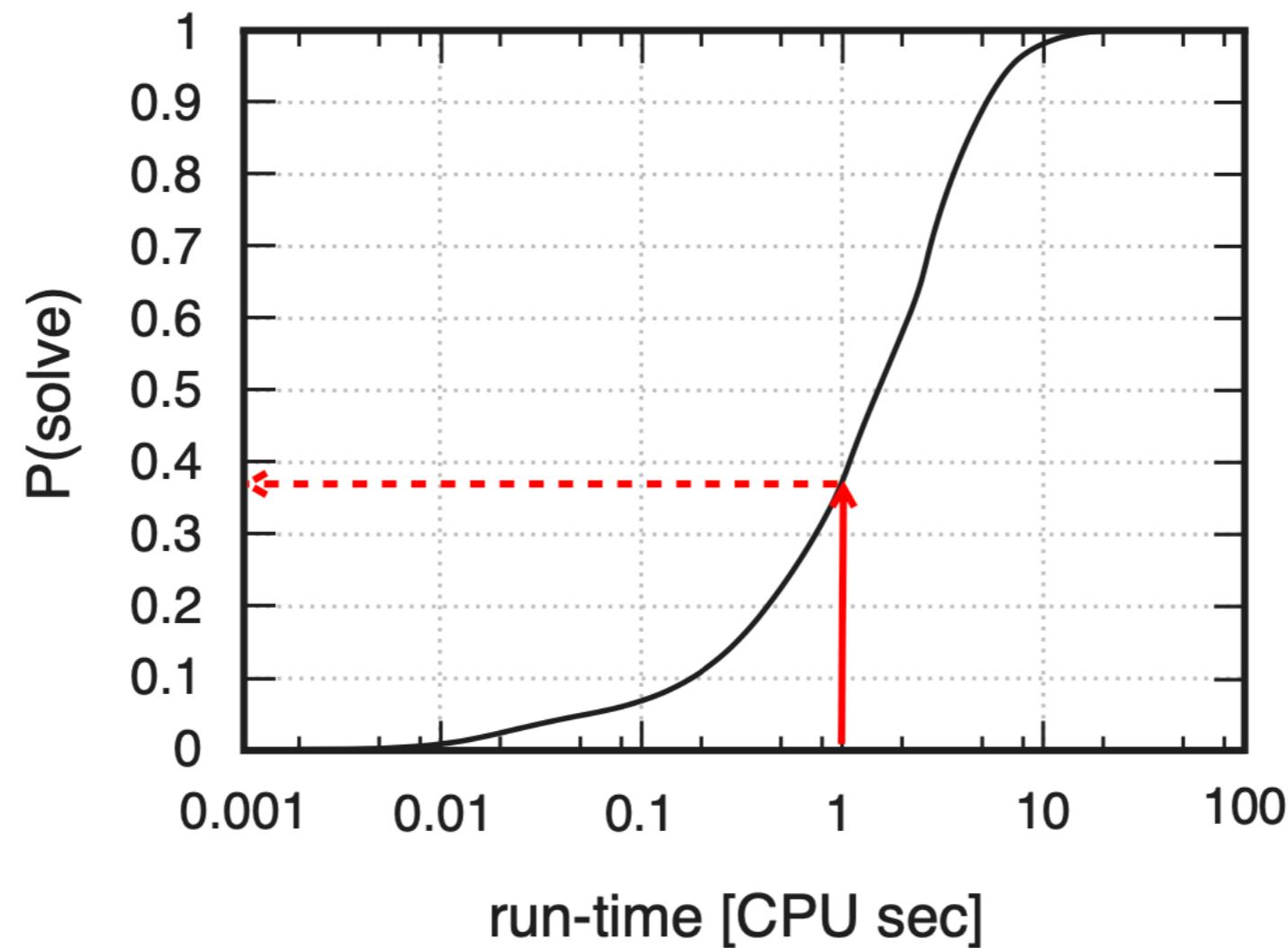
Distribution du runtime ?



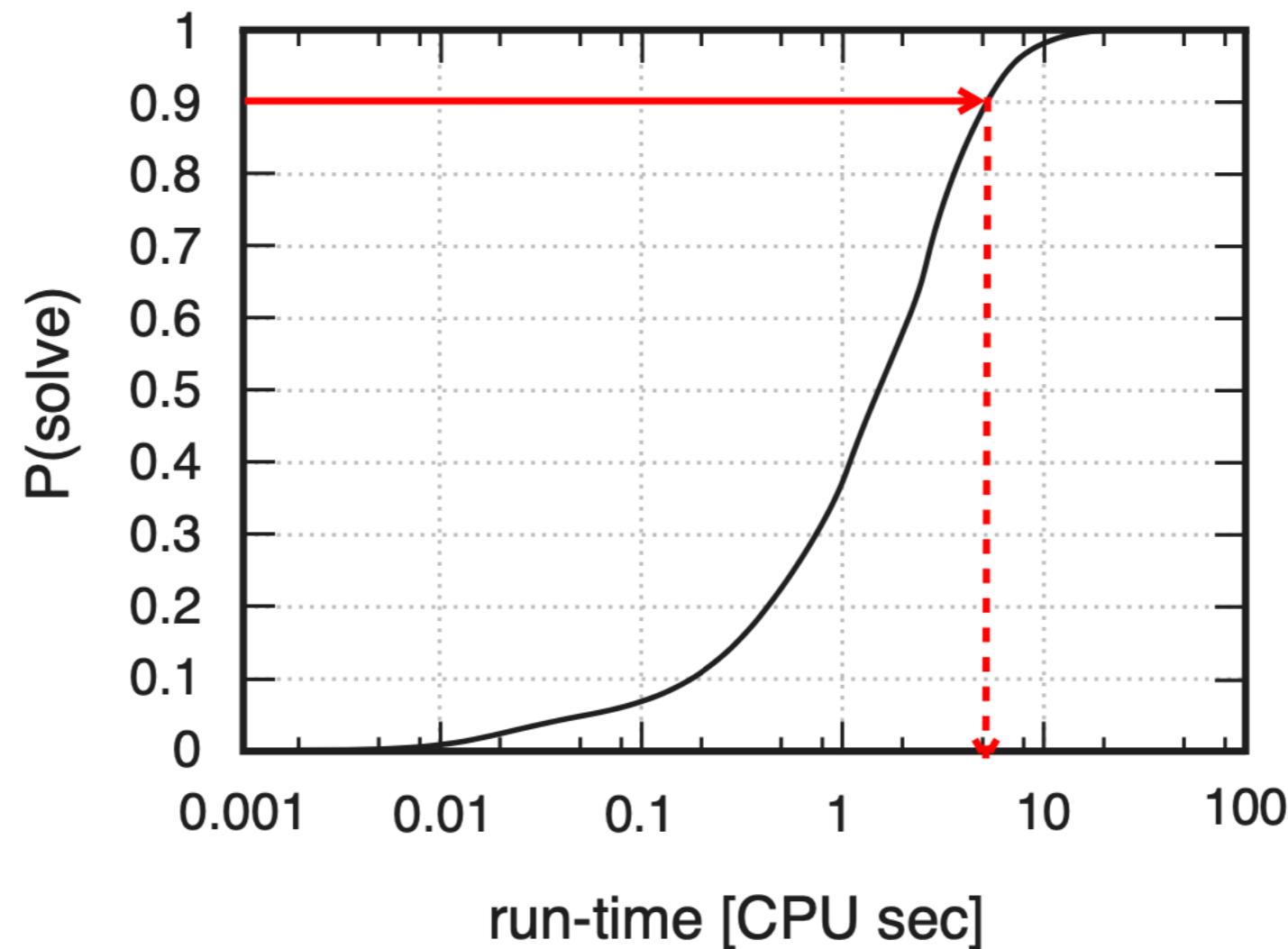
Distribution du runtime



Distribution du runtime



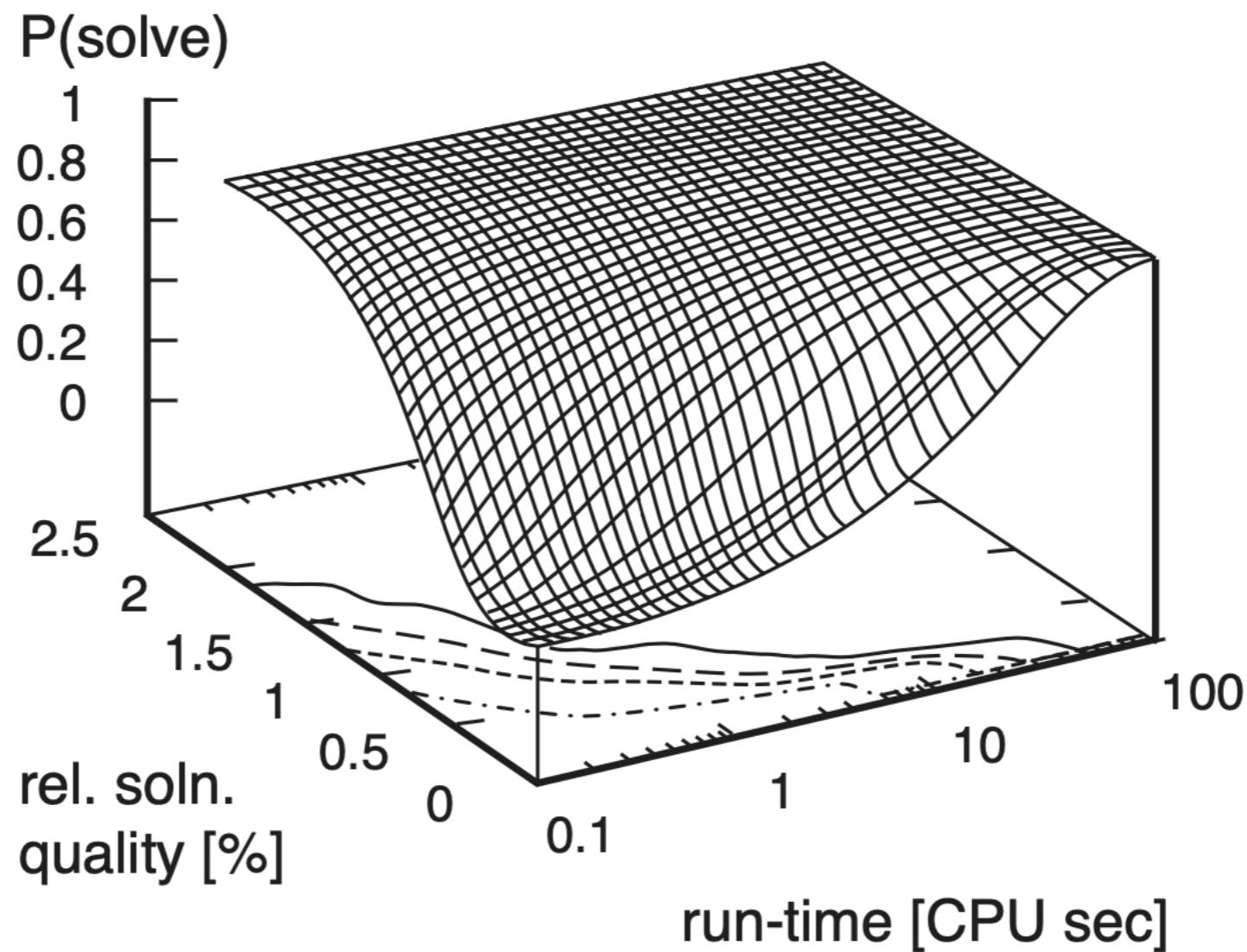
Distribution du runtime



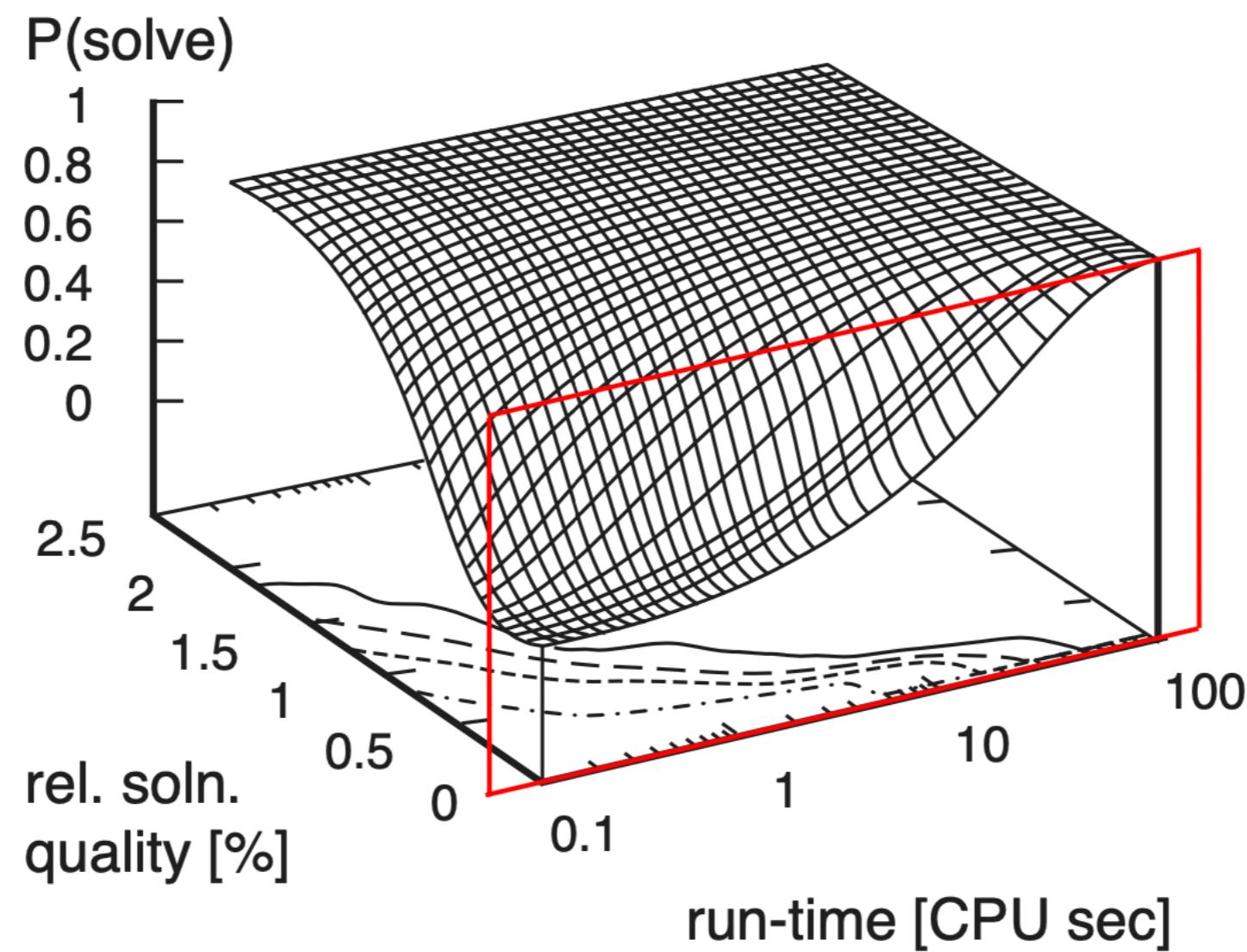
Distribution du runtime

- Qualité / précision vs Runtime
- Les deux sont des variables aléatoires
- Distributions bivariées

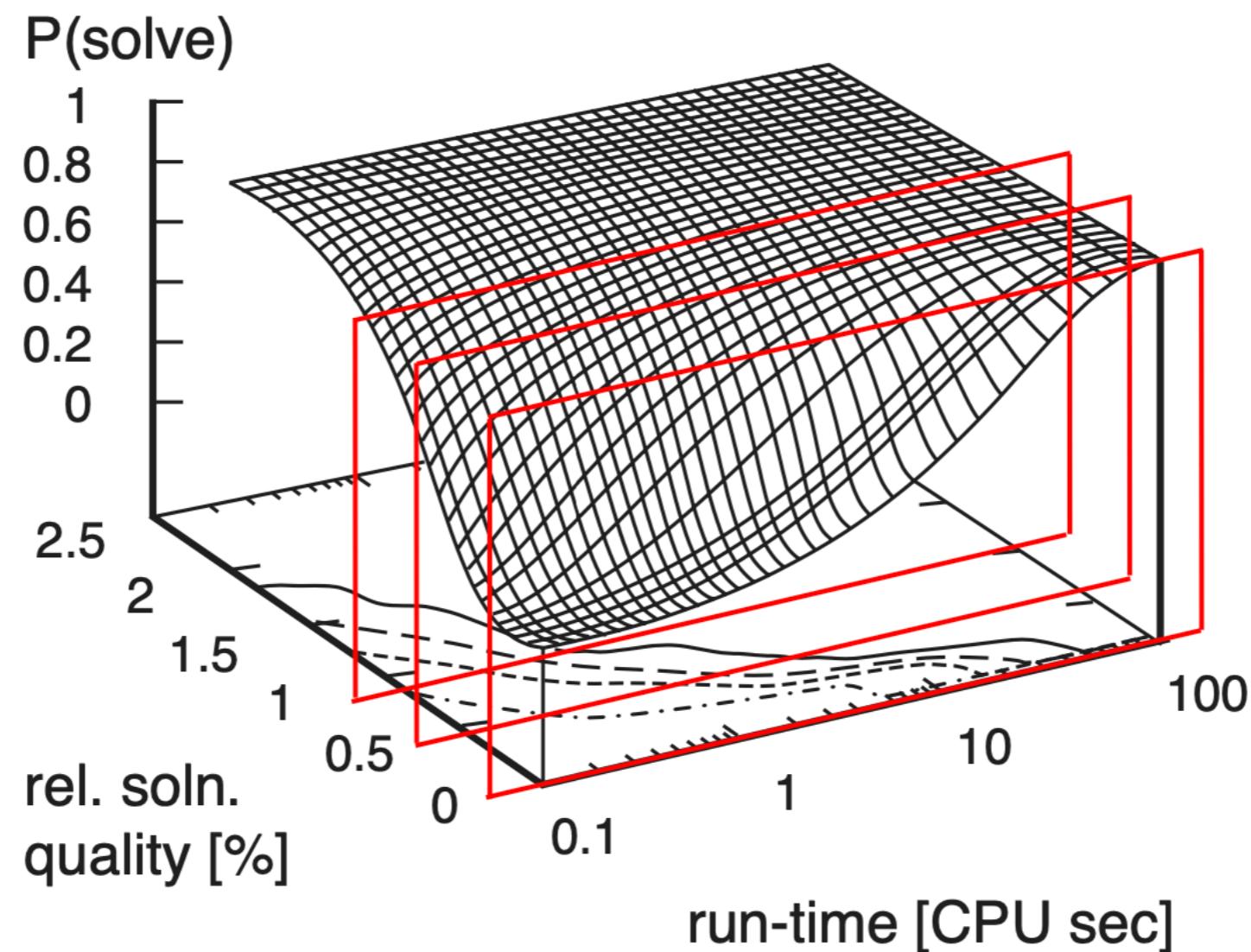
Distribution du runtime



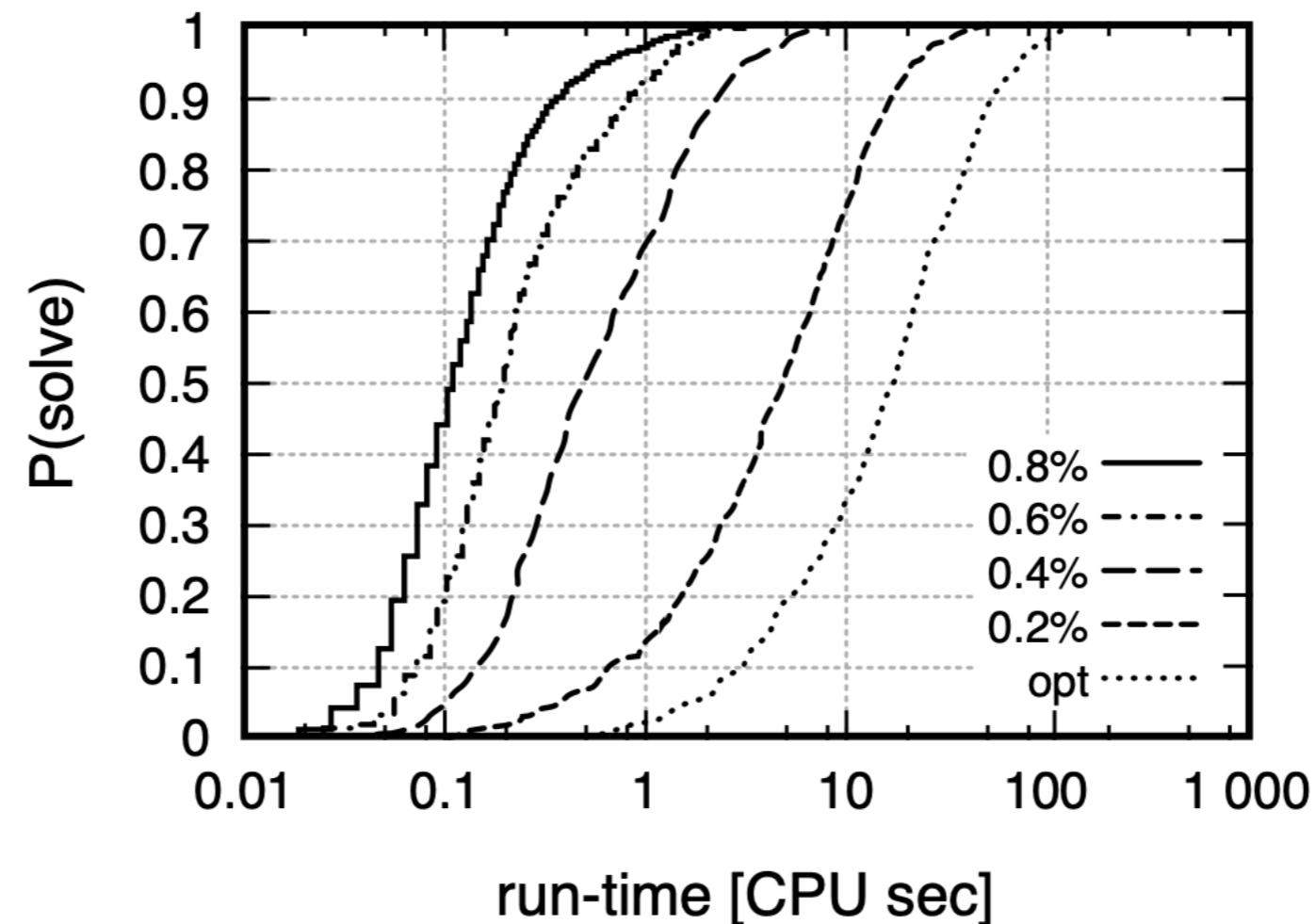
Distribution du runtime



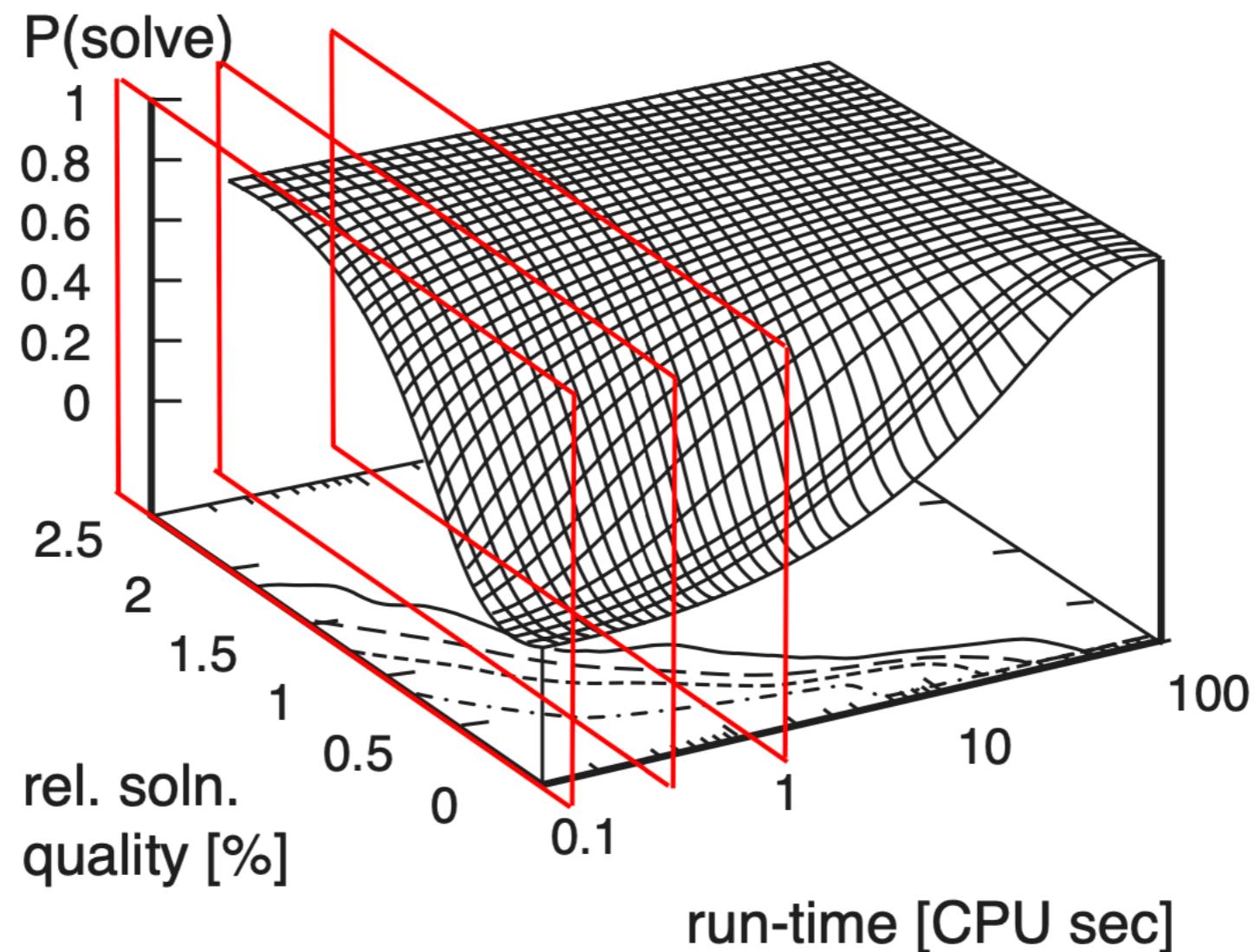
Distribution du runtime



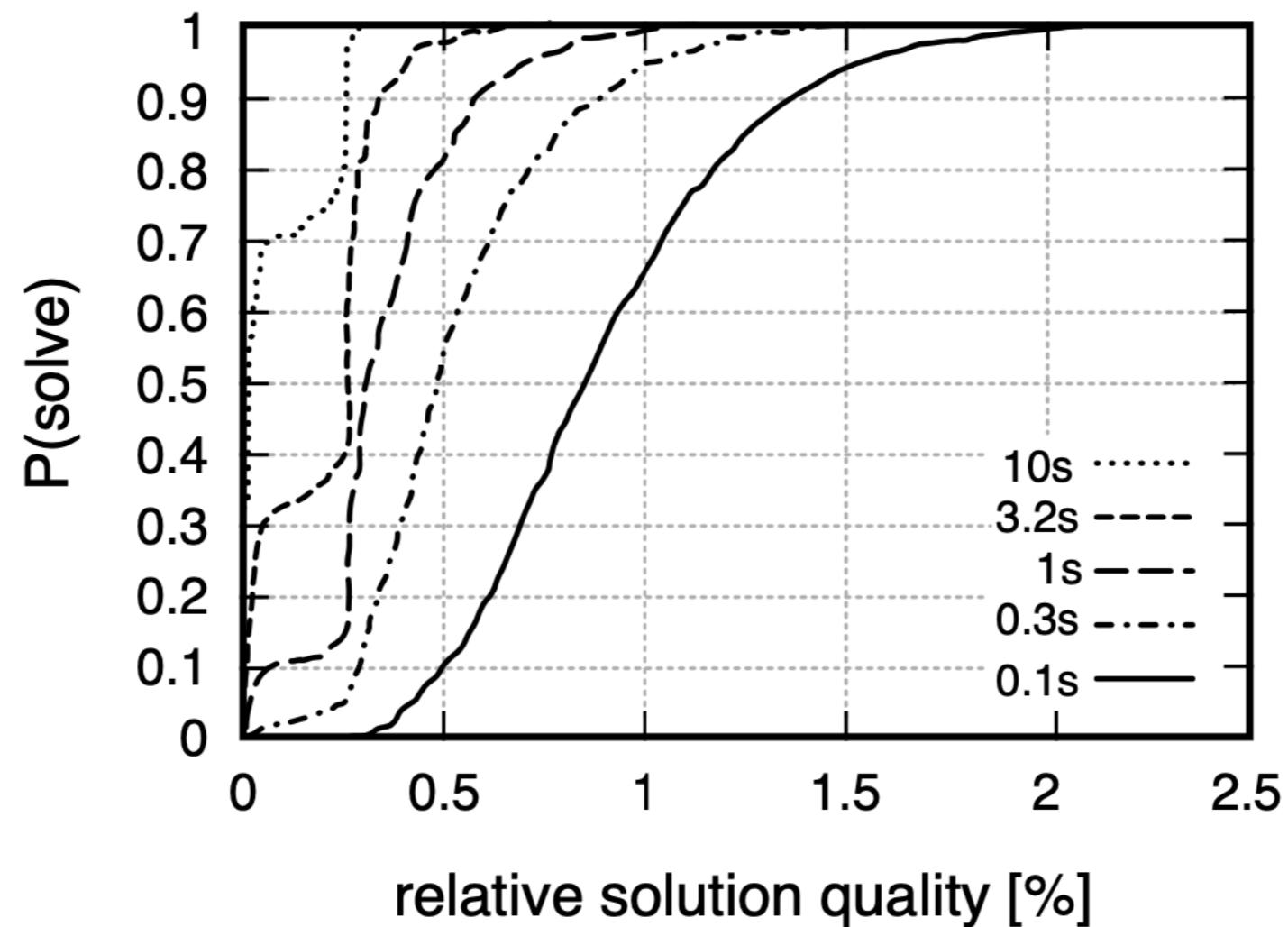
Distribution qualifiée



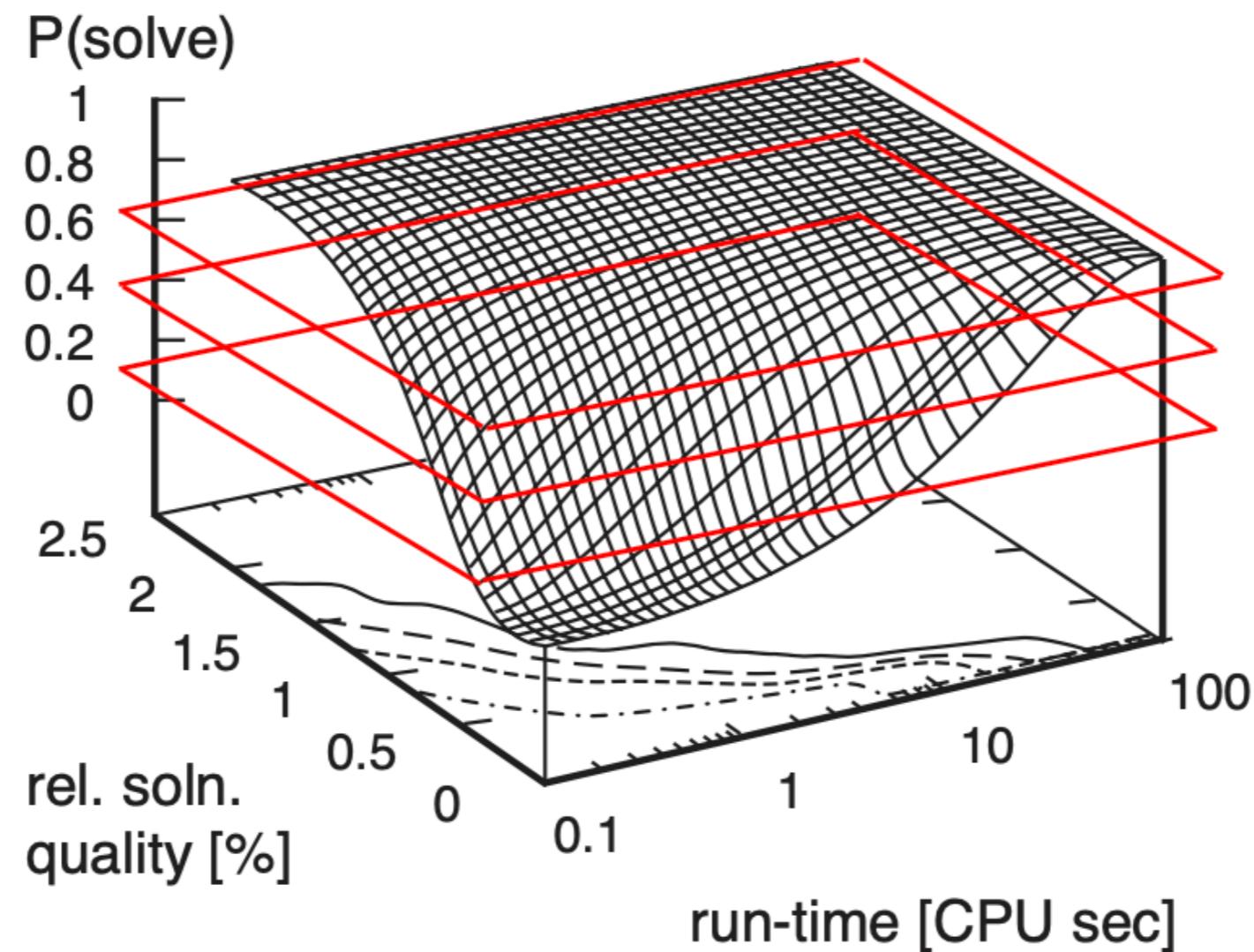
Distribution du runtime



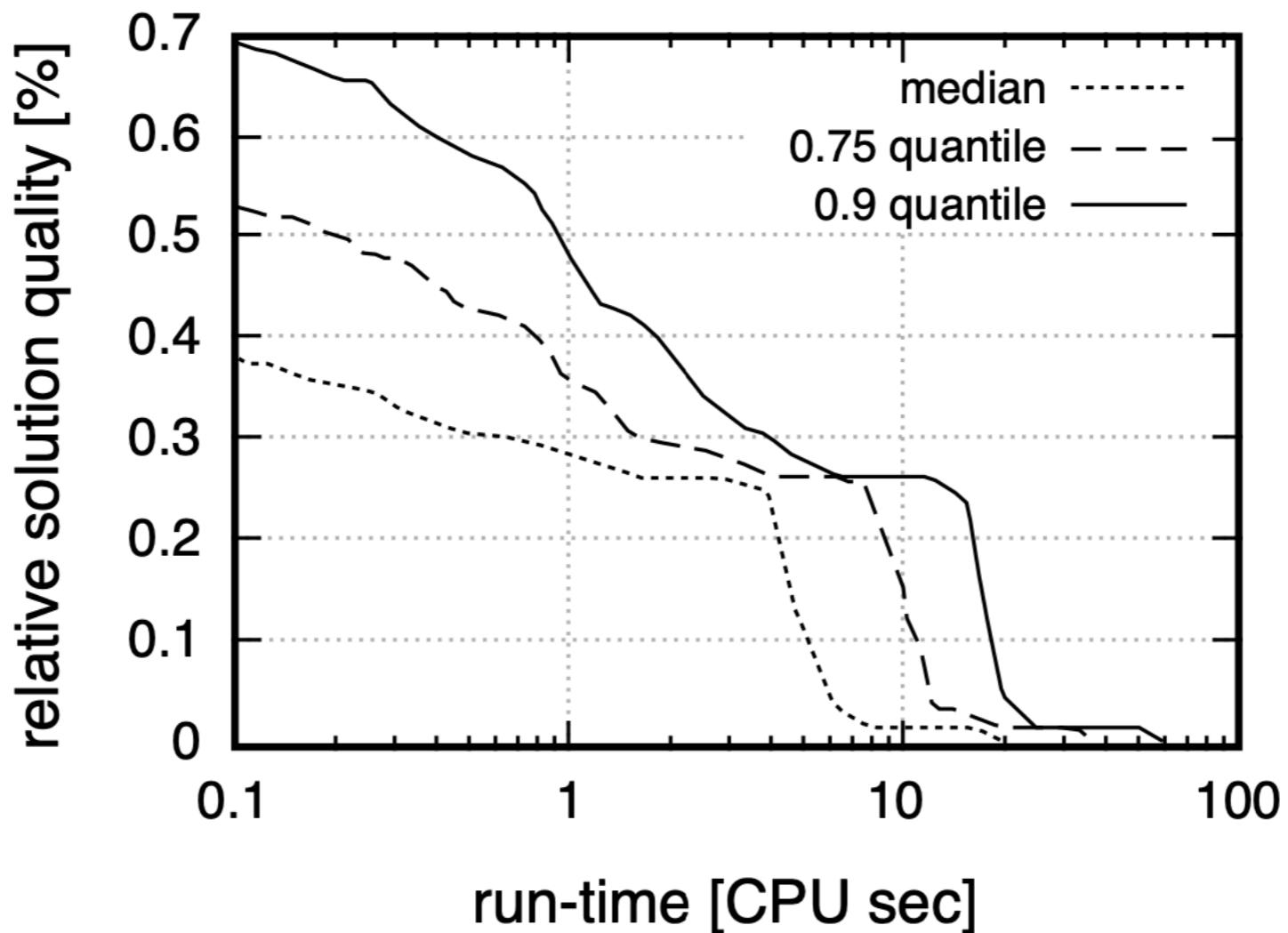
Distribution de la qualité



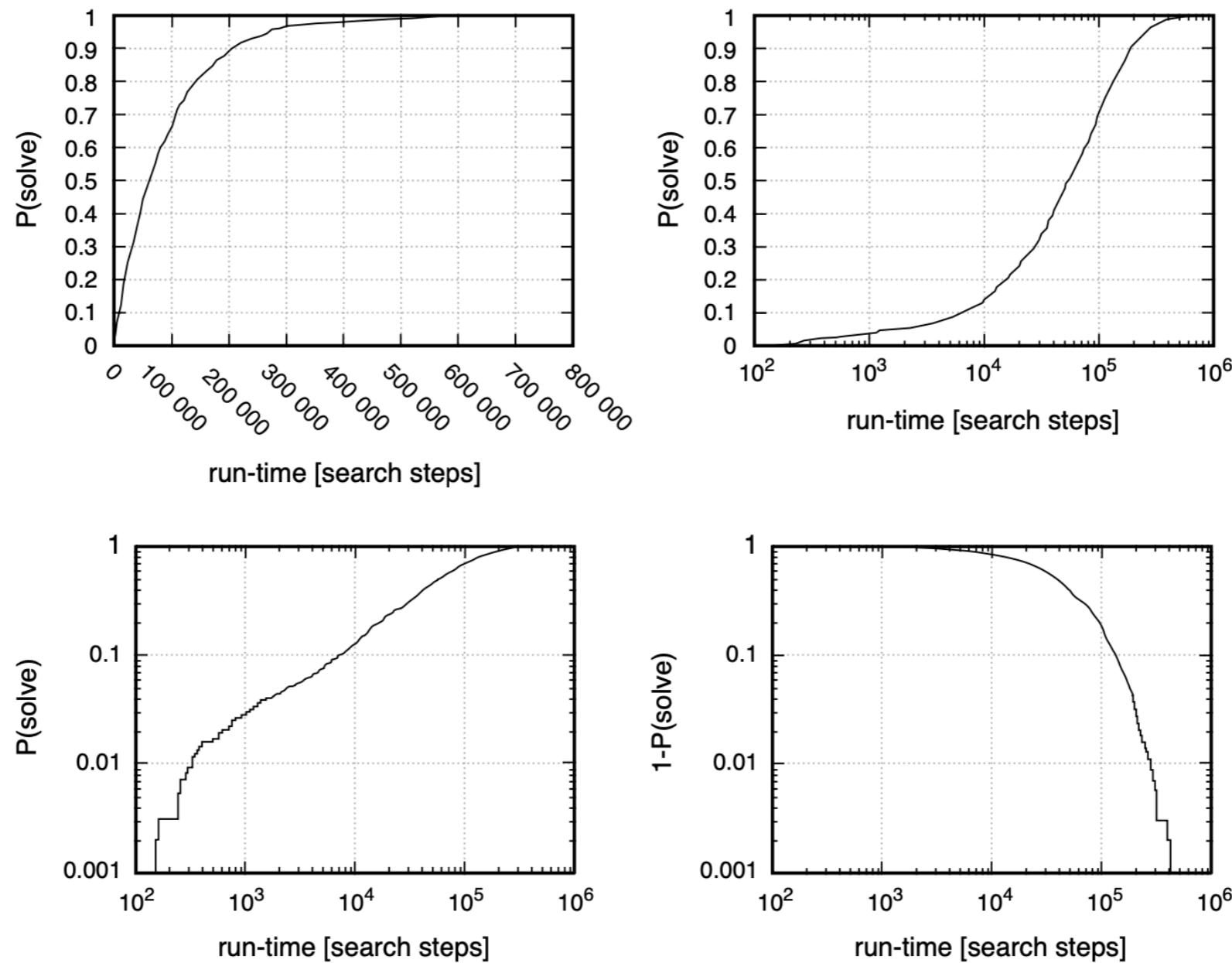
Distribution du runtime



Profile « Anytime »



Différentes représentations



Remarques supplémentaires

- Instances / données
 - Type, taille, plus ou moins difficiles, ...
 - Choix du **benchmark** influent fortement sur les résultats
- Environnement **informatique**
 - Machine, système d'exploitation, compilateur
 - Implémentation, structure des données, langage
 - ... **Reproductibilité ?**

An Introduction to Scientific Experimentation and Benchmarking

A. Auger & N. Hansen, GECCO (2023)



An Introduction to Scientific Experimentation and Benchmarking

Anne Auger and Nikolaus Hansen
Inria and CMAP, Ecole Polytechnique, IP Paris

Full set of slides: <http://www.cmap.polytechnique.fr/~nikolaus.hansen/gecco-2023-benchmarking-tutorial.pdf>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

© 2023 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0120-7/23/07...\$15.00

<https://doi.org/10.1145/3583133.3595064>



An Introduction to Scientific Experimentation and Benchmarking

[http://www.cmap.polytechnique.fr/~nikolaus.hansen/
gecco-2023-benchmarking-tutorial.pdf](http://www.cmap.polytechnique.fr/~nikolaus.hansen/gecco-2023-benchmarking-tutorial.pdf)

Benchmarking Multiobjective Optimizers 2.0

D. Brockhoff & T. Tušar, GECCO (2023)



GECCO 2023 Tutorial on Benchmarking Multiobjective Optimizers 2.0

Dimo Brockhoff

dimo.brockhoff@inria.fr



Tea Tušar

tea.tusar@ijs.si



The final slides will be made available at
<http://www.cmap.polytechnique.fr/~dimo.brockhoff/>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

© 2023 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0120-7/23/07\$15.00

<https://doi.org/10.1145/3583133.3595060>

Benchmarking Multiobjective Optimizers 2.0

[http://www.cmap.polytechnique.fr/~dimo.brockhoff/
publicationListFiles/bt2023a.pdf](http://www.cmap.polytechnique.fr/~dimo.brockhoff/publicationListFiles/bt2023a.pdf)

Buts du benchmarking

T. Bartz-Beielstein et. al (2020)

Common Goals of Benchmarking Studies

Visualization and Basic Assessment

- G1.1 Basic Assessment of Performance and Search Behavior
- G1.2 Algorithm Comparison
- G1.3 Competition
- G1.4 Assessment of the Optimization Problem
- G1.5 Illustrating Algorithms' Search Behavior

Sensitivity of Performance

- G2.1 Testing Invariances
- G2.2 Algorithm Tuning
- G2.3 Understanding the Influence of Parameters and Algorithmic Components
- G2.4 Characterizing Algorithms' Performance by Problem Features

Performance Extrapolation

- G3.1 Performance Regression
- G3.2 Automated Algorithm Design, Selection, and Configuration

Theory-Oriented Goals

- G4.1 Cross-Validation and Complementation of Theoretical Results
- G4.2 Source of Inspiration for Theoretical Studies
- G4.3 Benchmarking as Intermediary between Theory and Practice

Benchmarking in Algorithm Development

- G5.1 Code Validation
- G5.2 Algorithm Development

Projet

But du projet

Proposer une analyse fine et rigoureuse de la performance d'algorithmes d'optimisation

Contexte du projet

- Optimisation ...
... boite-noire, multi-objectif, continue, et sans contrainte



Contexte du projet

- Optimisation ...
... boite-noire, multi-objectif, continue, et sans contrainte
 - ⇒ **qualité** = hypervolume (à minimiser)
 - ⇒ **temps** = nombre de solutions évaluées (à minimiser)



Evaluation

Contrôle continu

Projet (par binôme, un seul rendu final sur Moodle)

- > compte-rendu sous forme d'un **rapport scientifique**
(10-15 pages)
- > **annexe** avec résultats détaillés reproductibles
- > développement : code, scripts... reproductibles

Première étape

Analyse détaillée des données d'execution (brutes) d'un algorithme sur un problème (et ≠ runs / instances)

1. Voir <https://numbbo.github.io/data-archive/bbob-biobj/>
2. Choisir un algorithme (\neq de ses voisin.e.s)
3. Télécharger et comprendre les données (colonne « data »)
4. Visualiser et analyser l'exécution d'un run (une instance)
5. Visualiser et analyser l'exécution de plusieurs runs (instances)
6. Choisir une agrégation (des runs/instances), la visualiser et l'analyser

Première étape

Analyse détaillée des données d'execution (brutes) d'un algorithme sur un problème (et \neq runs / instances)

1. Voir <https://numbbo.github.io/data-archive/bbob-biobj/>
2. Choisir un algorithme
3. Télécharger et comprendre le code (donne « data »)
4. Visualiser et analyser une instance (une instance)
5. Visualiser et analyser l'exécution de plusieurs runs (instances)
6. Choisir une agrégation (des runs/instances), la visualiser et l'analyser



Deuxième étape

Analyse détaillée des données d'execution (brutes) d'un algorithme sur plusieurs problèmes (et \neq runs / instances)

- Répéter par problème et analyser
- Interpréter la performance de l'algorithme en fonction du type et de la dimension du problème
- (sauver et expliquer les scripts et données intermédiaires)

Troisième étape

Analyse détaillée des données d'execution (brutes) de plusieurs algorithme sur plusieurs problèmes

- Choisir au moins 3 algorithmes (dont RANDOMSEARCH-5)
- Répéter par algorithme et analyser
- Interpréter la performance (relative) des algos en fonction du type et de la dimension du problème
- (sauver et expliquer les scripts et données intermédiaires)

Prochaines séances

- ~~Analyse expérimentale / empirique~~
- **Présentation** des résultats et reproductibilité
- **Recommendation** interprétable et **hyper-paramétrage**