



HAL
open science

Plans d'expériences pour l'évaluation d'algorithmes évolutionnaires et la constitution de classes de référence

Olivier François, Christian Lavergne

► **To cite this version:**

Olivier François, Christian Lavergne. Plans d'expériences pour l'évaluation d'algorithmes évolutionnaires et la constitution de classes de référence. [Rapport de recherche] RR-3601, INRIA. 1999. inria-00073078

HAL Id: inria-00073078

<https://inria.hal.science/inria-00073078>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*Plans d'expériences pour l'évaluation
d'algorithmes évolutionnaires et la constitution
de classes de référence*

Olivier François, Christian Lavergne

No 3601

Janvier 1999

_____ THÈME 4 _____



*Rapport
de recherche*



Plans d'expériences pour l'évaluation d'algorithmes évolutionnaires et la constitution de classes de référence

Olivier François, Christian Lavergne

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet IS2

Rapport de recherche n° 3601 — Janvier 1999 — 34 pages

Abstract: Cet article décrit une méthodologie statistique qui permet à l'utilisateur d'algorithmes évolutionnaires de configurer correctement les paramètres de ces algorithmes. La méthode se fonde sur un modèle statistique appelé modèle linéaire généralisé. Un tel modèle permet de piloter les données aléatoires produites par les algorithmes. En particulier, les expériences croisées et passées peuvent être prises en compte efficacement. Les procédures d'estimation et de test permettent d'évaluer l'effet de la variation des paramètres sur la solution proposée par l'algorithme. À l'aide de cette méthode, le nombre d'essais de simulation correspondant à chaque combinaison de paramètres peut être considérablement réduit. La méthode permet aussi de créer des classes de problèmes pour lesquels le comportement final de l'algorithme est invariant. La constitution de telles classes permet d'augmenter la qualité des estimations à un coût de calcul identique.

Key-words: algorithmes évolutionnaires, modèle linéaire généralisé, plan d'expérience, distribution gamma.

(Résumé : tsvp)

Experimental Design for Evolutionary Algorithms : a Statistical Perspective

Résumé : This article describes a statistical method which helps a user of evolutionary algorithms to configurate the algorithm's parameters. The method is based upon a statistical model called generalized linear model. Such a model allows to manage the data which are produced by the algorithm. Actually, crossed and past simulation experiments can be efficiently accounted for. Estimation and test procedures permit to assess the effect the variation of the parameters has on the solution proposed by the algorithm. As a result of the method, the number of trials corresponding to each combination of the parameters can often be drastically reduced. The method also enables the creation of classes of problems for which the final behavior of the algorithm is invariant. Creating such classes enhances the quality of the estimations at a computational cost which remains unchanged.

Mots-clé : evolutionary algorithm, generalized linear model, experimental design, gamma distribution.

1 Introduction

Les algorithmes évolutionnaires (algorithmes génétiques, stratégies d'évolution, etc.) (Fogel, 1995) et de recuit simulé (Aarts et Korst, 1988) forment une famille d'algorithmes stochastiques de recherche d'extrema dont l'efficacité est reconnue mais dont les réglages relèvent la plupart du temps de l'expérience personnelle de l'utilisateur. De tels algorithmes sont souvent utilisés lorsque le problème de recherche est très complexe et que peu d'information n'est analytiquement accessible. Dans de telles situations, les règles fondées sur des grandeurs analytiques du problème envisagé sont proscrites. Ces grandeurs sont en effet trop souvent difficiles à estimer en pratique.

D'après le théorème NFL (Wolpert et MacReady, 1995), un algorithme donné ne peut être plus performant en moyenne sur un problème choisi au hasard que n'importe quel autre algorithme tant qu'aucune connaissance n'y est injectée. Il est évidemment essentiel de choisir un algorithme de résolution en relation avec le problème que l'on souhaite résoudre. Cela nécessite de pouvoir prédire le comportement de l'algorithme et de pouvoir le paramétrer correctement.

Face à un problème réel, tout praticien passe par une phase de réglage de ses algorithmes. Cette phase de réglage comporte de nombreuses simulations. Le meilleur moyen de quantifier les performances de divers algorithmes pour un problème donné est de répéter des essais de simulation dans des contextes différents, en changeant d'algorithme le plus souvent possible. Dans cette démarche, il est essentiel de limiter le nombre d'essais (parfois coûteux) et de tirer parti de l'information contenue dans les essais croisés. Un plan d'expériences adéquat permet de diminuer le nombre d'essais. Il est alors nécessaire d'utiliser de manière globale l'information donnée par les simulations et de comprendre quels essais sont nécessaires. Il s'agit aussi d'intégrer efficacement les expériences passées.

Dans ce travail, nous abordons cette question à l'aide de modèles statistiques de type *modèles linéaires généralisés*. L'introduction d'un modèle permet d'exploiter efficacement les simulations afin de déterminer les configurations optimales des paramètres. Elle permet aussi de réduire le nombre des simulations à venir. La méthodologie statistique se structure en trois étapes. La première étape concerne l'identification de la structure statistique correspondant à l'évaluation de la solution produite par un algorithme pour un problème donné. La seconde étape correspond à la construction d'un modèle au niveau *individuel*. Il s'agit de modéliser cette évaluation pour un problème et une famille d'algorithmes donnés. La troisième étape correspond à la construction d'un modèle *global* de l'évaluation pour la même famille d'algorithmes et un ensemble de problèmes. Le stade ultime de la démarche d'intégration de l'information est la constitution d'ensembles de problèmes de référence pour la famille d'algorithmes en question.

Cette démarche sera illustrée à l'aide d'une famille paramétrique d'algorithmes stochastiques de recherche d'extremum. La famille, notée $\{a_\psi\}$, dépend d'un paramètre vectoriel ψ . Différents comportements sont attendus pour des valeurs distinctes du paramètre. Nous souhaitons déterminer, pour un problème de minimisation donné, une configuration du paramètre qui garantit de bonnes performances. Pour cela, nous souhaitons utiliser un plan d'expériences de taille réduite. Nous souhaitons ensuite nous reporter à une classe de

problèmes préalablement étudiés pour lesquels la paramétrisation optimale est connue. Si le problème appartient à la classe en question, nous utiliserons la paramétrisation donnée par la classe de référence. Bien entendu, cette démarche suscite deux questions fondamentales. La première question concerne la classe. Selon quels critères et comment peut-elle être fabriquée ? La seconde question concerne le problème. Comment tester son appartenance à une classe de référence ?

Cet article est structuré de la manière suivante. Dans la section 2, nous présentons le contexte d'application de notre démarche. En particulier, nous décrivons les problématiques et les algorithmes choisis pour tester cette démarche. Dans la section 3, nous présentons les trois niveaux de notre démarche : identification de la structure, modèle individuel et modèle global. Un paragraphe important sera consacré à la description des modèles linéaires généralisés (McCullagh et Nelder, 1989). Dans la section 4, nous présentons les résultats obtenus pour les problématiques de test introduites dans la section 2. Toutes les procédures d'estimation ont été réalisées à partir du logiciel *Splus* (Chambers et Hastie, 1991 ou Venables et Ripley, 1997).

2 Contexte

Dans cette section, nous introduisons le contexte d'application choisi pour tester la méthodologie statistique. De manière générale, il s'agit de minimiser une fonction définie sur un sous-ensemble compact de R^D , $D \geq 1$, à l'aide d'algorithmes évolutionnaires spécifiques. Nous présentons deux types de problématique et une famille d'algorithmes précise. Ce contexte nous servira à présenter les résultats concrets de la section 4. Les choix effectués sont arbitraires. Bien entendu, la méthode que nous allons présenter se généralise à de nombreux autres types de problématiques et d'algorithmes.

2.1 Choix de problématiques

2.1.1 Fonctions test usuelles

Nous étudions 5 problèmes de minimisation fréquemment utilisés pour tester les algorithmes évolutionnaires. La dimension du problème varie d'une situation à l'autre. Pour toutes les fonctions à minimiser, les coordonnées sont définies dans l'intervalle $I = [-5.12, +5.12]$. La fonction f_1 (*Sphère*, dimension 8) est définie de la manière suivante

$$\forall x \in [-5.12, +5.12]^8, \quad f_1(x) = f_1(x_1, \dots, x_8) = \sum_{i=1}^8 x_i^2.$$

La fonction f_2 (dimension 20) est définie de la manière suivante

$$\forall x \in [-5.12, +5.12]^{20}, \quad f_2(x) = f_2(x_1, \dots, x_{20}) = \sum_{i=1}^{20} x_i^2 - \cos(\pi x_i/2) + 1.$$

La fonction f_3 (dimension 20) est définie de la manière suivante

$$\forall x \in [-5.12, +5.12]^{20}, \quad f_3(x) = \sum_{i=1}^{20} [(20x_i + 0.5)]^2$$

où $[a]$ désigne la partie entière du scalaire a . La fonction *aléatoire* f_4 (dimension 20) est définie de la manière suivante

$$\forall x \in [-5.12, +5.12]^{20}, \quad f_4(x) = \sum_{i=1}^{20} x_i^2 + 0.5U_i$$

où U_i est une variable aléatoire de loi uniforme sur $(0, 1)$ (renouvelée de manière indépendante à chaque évaluation). La fonction f_5 , appelée fonction de *Rastrigin* (dimension 4), est définie de la manière suivante

$$\forall x \in [-5.12, +5.12]^4, \quad f_5(x) = f_5(x_1, \dots, x_4) = \sum_{i=1}^4 x_i^2 - 10 \cos(2\pi x_i) + 10.$$

Note. Le problème f_1 est considéré comme facile. Le choix d'une dimension faible contribue à ce caractère. Pour ce problème, on attend de bonnes performances des algorithmes. À l'opposé, les problèmes f_3, f_4 sont considérés comme délicats (voire mal défini en ce qui concerne f_4). En dimension 20, la fonction de Rastrigin offre un problème de complexité plus élevée que les autres. Nous souhaitons conserver le même nombre d'évaluations des solutions pour tous les problèmes. Pour cette raison, nous étudions le problème f_5 (dimension 4) qui peut être résolu dans le même ordre de temps de calcul que f_4 .

2.1.2 Approximation par réseaux de neurones

La seconde problématique utilisée pour illustrer la méthode statistique est celle de l'approximation d'une fonction réelle f , définie sur l'intervalle $[-1, +1]$, par un réseau de neurones artificiels de type perceptron. La structure de réseau choisie est une structure fixe. La "sortie" de ce réseau est donnée par la formule suivante

$$\forall u \in [-1, +1], \quad g(u) = w_i + \sum_{j=1}^4 w_{ji} s(w_{ij}u + w_{jj})$$

où s est la fonction logistique

$$\forall u \in [-1, +1], \quad s(u) = \frac{1}{1 + \exp(-u)}$$

et les coefficients w sont inconnus. L'indice i est introduit par convention de notation. Cette structure définit un perceptron à trois couches, 4 cellules cachées et une sortie linéaire.

Etant donné n points, $\{u_k, 1 \leq k \leq n\}$, contenus dans l'intervalle $[-1, +1]$, nous définissons l'erreur d'approximation de la manière suivante

$$Err(w) = \frac{1}{n} \sum_{k=1}^n (g(u_k) - f(u_k))^2.$$

La problématique choisie consiste à minimiser l'erreur $Err(w)$ en w , en supposant

$$\forall j \leq 4, \quad w_{ij}, w_{ji}, w_{jj} \in I = [-5.0, +5.0], \quad w_i \in I.$$

Les problèmes d'approximation considérés dans la section 4 seront associés aux fonctions $f = f_6, f_7, f_8, f_9$ dont les graphes sont définis dans la figure 1. Le nombre de points dans l'intervalle $[-1, +1]$ est égal à $n = 100$. Il s'agit de problèmes de minimisation en dimension égale à 13. Les erreurs correspondantes sont notées $Err = E_6, E_7, E_8, E_9$. Un perceptron à trois couches peut définir un approximateur universel (voir par exemple Ripley (1996)). Malheureusement, lorsque le rang des coefficients w est borné et le nombre de cellules cachées est limité, cette affirmation n'est pas valide. Nous ne connaissons alors aucune borne inférieure pratiquement utilisable pour l'erreur d'approximation E_i . Cet état de fait augmente la difficulté du problème de minimisation.

Note. Les problématiques considérées dans les deux sections peuvent être considérées comme "jouets", c'est-à-dire peu réalistes. Toutefois, il paraît difficile de couvrir l'universalité des problèmes abordés par les algorithmes évolutionnaires. Les problèmes d'approximations sont susceptibles de constituer une "classe" de référence pour l'algorithme que nous étudierons. Nous verrons dans la section 4 à quel niveau de significativité cette affirmation pourra être considérée comme exacte. Nous verrons aussi comment tester qu'une fonction n'appartient pas à cette "classe". Notons que les problèmes de réseaux de neurones ont des caractéristiques bien reconnues. Les minima locaux nombreux et l'existence de plateaux font que les méthodes numériques locales (*backpropagation* et variantes) convergent souvent lentement ou vers des solutions peu intéressantes (Ripley, 1996). Les méthodes évolutionnaires ont souvent été appliquées à des problèmes du même type (Xin Yao, 1993), (Hui, Lam et Chea, 1997). Toutefois, il est rarement fait mention d'une méthode de configuration des algorithmes utilisés.

2.2 Algorithmes

Nous avons choisi d'illustrer notre méthode à l'aide d'un algorithme évolutionnaire paramétré par deux composantes. Il s'agit de l'algorithme MOSES : Mutation Or Selection Evolutionary Strategy (François, 1998). Ce choix est motivé par la bonne connaissance que nous avons de cet algorithme et de son implantation. Bien entendu, ce choix est arbitraire. Il est possible d'étudier des algorithmes génétiques (AG), des stratégies d'évolution (ES) ou des méthodes hybrides couplant procédures numériques et évolutionnaires, du recuit simulé, etc., issus de bibliothèques standardisées. Ces familles d'algorithmes dépendent d'un

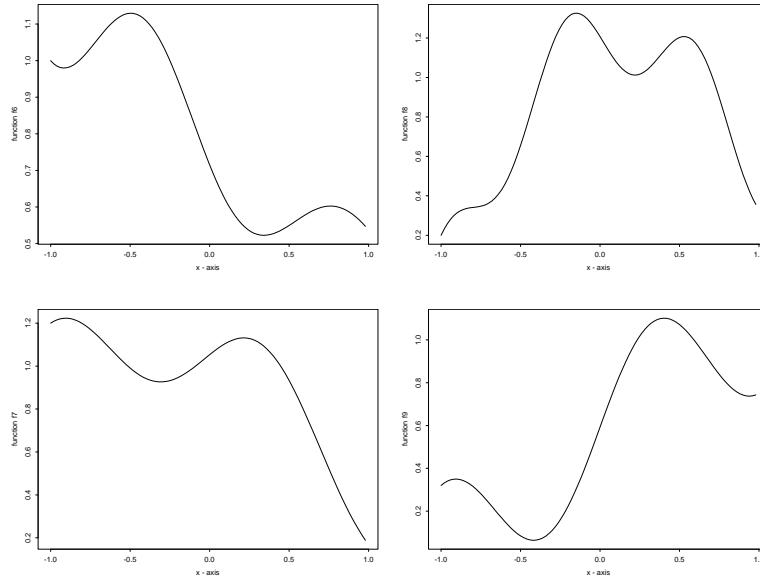


FIG. 1: Graphes des fonctions f_6 , f_7 , f_8 , f_9 . Les expressions analytiques des fonctions sont $f_6(u) = e^{-v}(1 + 0.5 \sin^2(5v))$, $f_7(u) = \sin(\pi v) + 0.2 \cos(5\pi v) + 0.3v + 0.23v^2$, $f_8(u) = 1 - v^2 + 0.2 \cos(3\pi v) + ve^{-v}$ et $f_9(u) = 0.32 \cos(3\pi v) - 0.23v^2 + 1.3v$ où $v = (1 + u)/2$, $u \in [-1, +1]$.

paramètre vectoriel. Le choix du paramètre se pose de manière identique pour MOSES ou pour les autres méthodes.

Notons que la méthodologie définie dans ce travail est fondée sur la comparaison de performances au sein d'une famille paramétrique d'algorithmes. La même technique permet aussi de comparer des algorithmes issus de familles distinctes (AG et ES par exemple). Nous discuterons ce point dans la conclusion.

L'algorithme MOSES décrit l'évolution d'une population de solutions de taille fixe. Son objectif est de trouver le minimum absolu d'une fonction définie sur un sous-ensemble compact de R^D . Dans ce travail, nous supposons que la taille est une constante (égale à $m = 10 + 1$) et non un paramètre. Ce choix est motivé par la clarté de la discussion. Toutefois, nous avons vérifié qu'augmenter la taille de $m = 11$ à $m = 31$ par exemple n'a pas d'effet significatif sur la performance de l'algorithme (pour les 9 problèmes considérés dans cet article).

Les deux composantes du paramètre de l'algorithme MOSES sont appelés *rayon* et *probabilité de mutation*. Ces composantes sont notées r et p . Nous avons donc $\psi = (r, p)$.

L'algorithme est défini de la manière suivante. La population à la génération t est représentée par la matrice $A = (a_{ij})$, où a_{ij} désigne la j ème coordonnée de la solution a_i , $1 \leq i \leq m$. La population est initialisée de manière arbitraire dans l'espace de recherche. Pour construire la population à la génération $t + 1$, on tire au hasard un nombre N selon la loi binomiale $B(m - 1, p)$. Puis on perturbe les N premiers vecteurs colonnes de la matrice A en ajoutant à une coordonnée de a_i ($i \in [1, \dots, N]$), disons a_{ij} , une variable aléatoire U de loi uniforme sur l'intervalle $[-r, r]$

$$a'_{ij} = a_{ij} + U .$$

Bien entendu, on contraint la nouvelle coordonnée a'_{ij} à rester dans son intervalle de définition I . On remplace ensuite les individus restants par le meilleur individu obtenu à la génération t . Cette procédure est itérée jusqu'à épuisement de la ressource de calcul.

Il s'agit en fait d'une version élitiste de l'algorithme MOSES. Parmi les m individus de la population, seuls $m - 1$ sont susceptibles de subir une mutation. Le dernier individu est un témoin de la meilleure solution trouvée depuis la génération initiale. En moyenne, à chaque génération, $(m - 1)p$ comparaisons sont effectuées pour déterminer le meilleur individu.

Lorsque le paramètre p est petit, il est possible de quantifier le comportement de MOSES en horizon de temps lointain (*i.e.*, $t \rightarrow \infty$). La probabilité pour que la solution proposée par l'algorithme MOSES se trouve dans un voisinage d'un minimum absolu converge vers 1. La vitesse de convergence est géométrique, de raison $O(p^{1/d_*})$. L'indice de convergence d_* représente le nombre minimal de mutations nécessaires pour trouver une solution d'évaluation meilleure partant d'un point différent d'un minimum absolu de la fonction objectif (François, 1998). Ce résultat montre l'importance du choix du paramètre r pour l'algorithme. Son influence est déterminante sur la grandeur d_* . Bien entendu, la grandeur analytique d_* n'est pas accessible en pratique pour des problèmes complexes.

2.3 Travaux antérieurs

De nombreux articles font état de travaux de comparaison d'algorithmes et d'évaluation de performances. Il s'agit bien entendu d'une thématique majeure dans la communauté des algorithmes stochastiques. Il est impossible de citer tous les travaux (voir, par exemple, Ingber et Rosen (1992), Park (1995), Keane (1997)). Les statistiques sont souvent présentes mais elles sont, en général, utilisées de manière très élémentaire. La plupart des auteurs se contentent de répéter de nombreuses simulations des algorithmes avec une variabilité très faible sur les paramètres de ces algorithmes. En général, aucun modèle statistique de l'évaluation de la solution n'est introduit pour traiter globalement les données. De rares travaux (Bergeret et Besse, 1997) mentionnent les méthodes statistiques de plans d'expériences (plans factoriels). Enfin, le travail effectué dans cet article peut servir à évaluer les problèmes test eux-mêmes (Whitley, Mathias, Rana et Dzuber, 1995).

3 Méthode

Il existe différentes manières d'évaluer la performance d'un algorithme de recherche sur un problème donné. Il est naturel de fixer le nombre d'évaluations de la fonction objectif et d'observer la valeur atteinte par l'algorithme. Il est aussi possible de fixer un seuil de performance à atteindre et d'observer le nombre d'évaluations nécessaires pour dépasser ce seuil. Toutefois, cette dernière approche nécessite de connaître la valeur minimale. Dans de nombreuses situations pratiques, cette valeur est inconnue et la première approche est fréquemment adoptée.

3.1 Identification du modèle

Un modèle naïf utiliserait la loi gaussienne pour modéliser la valeur de la solution proposée par l'algorithme. Nous allons faire un choix différent. Nous observons une variable positive, dissymétrique, prenant fréquemment des valeurs élevées, avec un mode souvent proche des valeurs d'évaluation minimales (ici zéro qui correspond à la solution exacte du problème de minimisation dans la plupart des cas test). De plus, cette variable prend fréquemment des valeurs élevées. La loi normale ne satisfait pas à ces contraintes. À l'opposé, la loi gamma paraît mieux adaptée pour modéliser une telle variable.

Pour valider cette argumentation, nous avons effectué 1000 simulations de l'algorithme pour les problèmes f_i et E_i pour une valeur fixe du paramètre ψ

$$r = 1.2, \quad p = 0.3.$$

Les histogrammes des valeurs de la solution produite par l'algorithme MOSES ressemblent à des histogrammes de loi gamma (figure 2). Les deux premiers moments de la variable étudiée (tableau 1) montrent que le comportement de l'algorithme est très différent pour chacune des fonctions f_1, \dots, f_5 .

	f_1	f_2	f_3	f_4	f_5
moyenne	0.000562	34.73	0.285	4.098	1.83
écart-type	0.000432	21.22	0.129	0.6358	1.985

TAB. 1: Moyenne et écart-type de l'évaluation de la solution produite par MOSES, $r = 1.2$, $p = 0.3$ (1000 répétitions).

Les histogrammes qui correspondent à l'évaluation de MOSES pour f_1, f_2, f_3 (figure 2) font apparaître une dissymétrie dans la distribution de la variable. Cette dissymétrie est prise en compte lors de la modélisation par la loi gamma. L'histogramme qui correspond à la fonction f_4 (figure 3) est proche de celui d'une variable gaussienne. Nous verrons par la suite qu'une modélisation par la loi gamma n'est pas un inconvénient. Pour la fonction de Rastrigin (fonction f_5), l'aspect dentelé de l'histogramme (figure 3) s'explique par la présence de minima locaux très proches de la solution optimale. Cet histogramme ne correspond à aucune loi de probabilité classique. Le choix d'une distribution gamma est donc arbitraire.

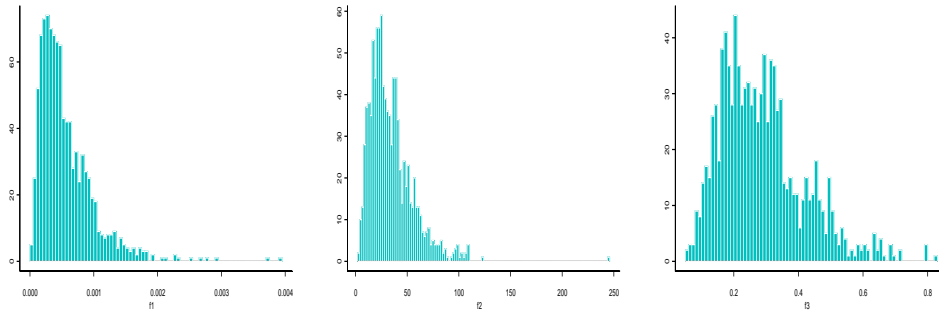


FIG. 2: Histogrammes de l'évaluation de la solution produite par MOSES ($r = 1.2$, $p = 0.3$) pour les fonctions f_1 , f_2 , f_3 .

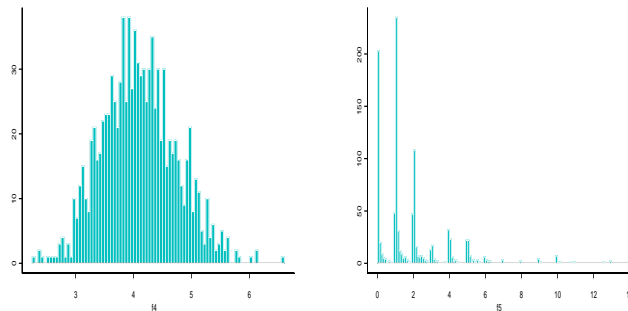


FIG. 3: Histogramme de l'évaluation de la solution produite par MOSES ($r = 1.2$, $p = 0.3$) pour les fonctions f_4 , f_5 .

Nous utilisons pour finir une méthode de validation graphique fondée sur le *qqplot*. Il s'agit de tracer les quantiles de l'échantillon empirique en fonction de ceux d'une loi de référence. La loi de référence est soit gaussienne (dans ce cas, on fait appel à la procédure *qqnorm* sous *Splus*) soit gamma (procédure *qqplot* sous *Splus*). L'hypothèse d'adéquation à une distribution est réaliste lorsque les points obtenus par les méthodes graphiques sont alignés. Pour chacune des 5 fonctions, l'adéquation à la structure gaussienne et à la structure gamma est testée (figures 4 et 5). L'hypothèse gamma est validée pour f_1 , f_2 , f_3 , f_4 par la méthode graphique et par un test de Kolmogorov-Smirnov au seuil 0.01. Pour la fonction de Rastrigin, l'adéquation à la loi gamma n'est pas validée par un test de Kolmogorov-Smirnov.

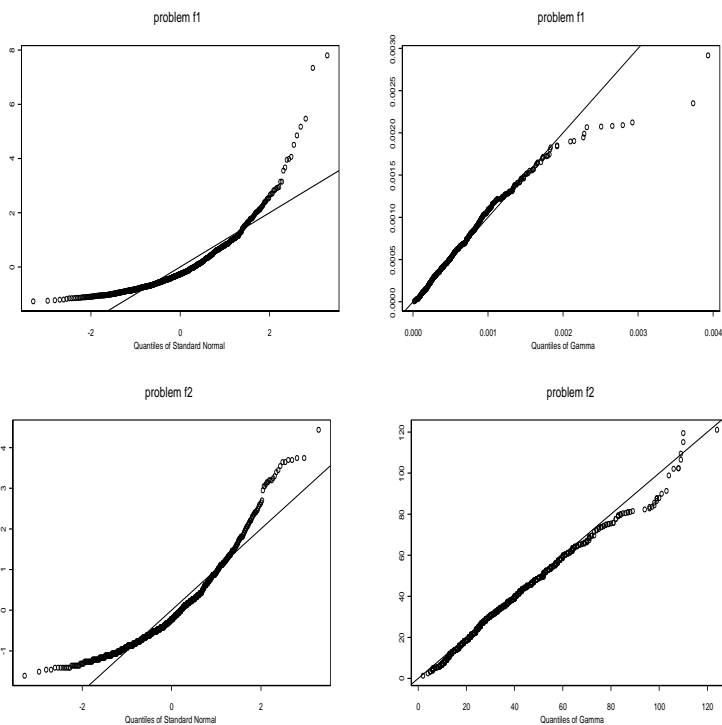


FIG. 4: *qqnorm* et *qqplot* pour les problèmes de minimisation associés à f_1 et f_2

Toutefois, le test graphique montre (figure 6) que la structure gamma est plus adaptée à l'analyse des données que la structure gaussienne.

3.2 Modèles linéaires généralisés

Constituer une classe de référence nécessite une connaissance approfondie du comportement des algorithmes sur les problèmes de la classe. De nombreuses simulations sont donc obligatoires afin de couvrir les différentes valeurs du paramètre ψ pour chaque problème.

Nous proposons une modélisation statistique pour décrire les résultats produits par les algorithmes. Utiliser un tel modèle permet de réduire sensiblement le nombre d'expériences en utilisant l'information contenue dans les observations qui correspondent à des valeurs croisées des paramètres. La modélisation choisie est du type GLM (*Generalized Linear Model*) dans le cas particulier d'une loi gamma. Pour estimer les paramètres introduits par la modélisation, il est nécessaire de se replacer dans le contexte mathématique associé à ce type de modèle. Trois hypothèses permettent de caractériser un GLM : la distribution de la variable

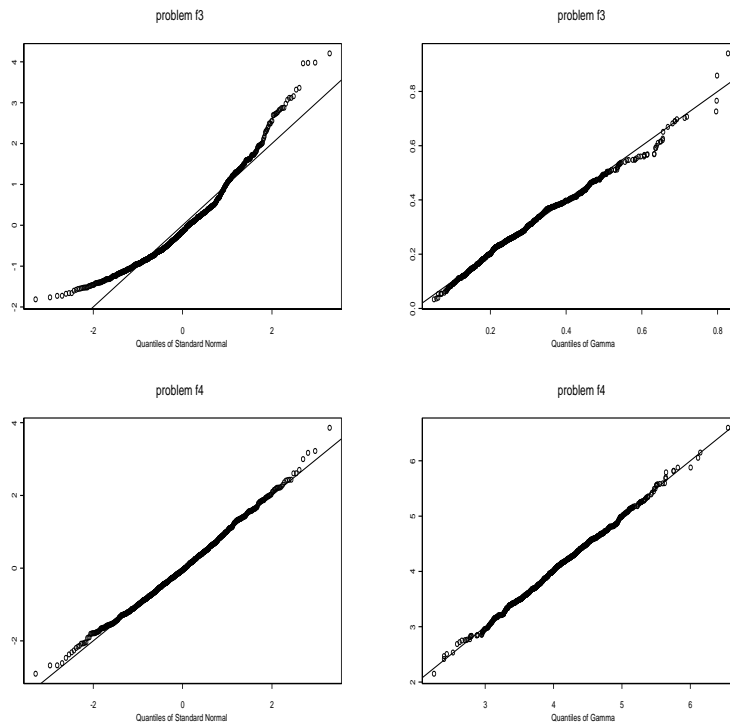


FIG. 5: *qqnorm* et *qqplot* pour les problèmes de minimisation associés à f_3 et f_4

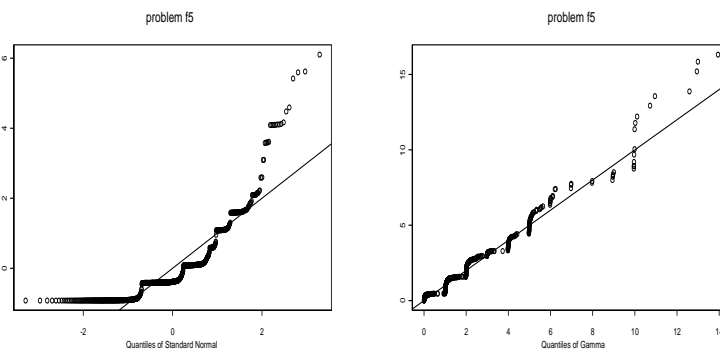


FIG. 6: *qqnorm* et *qqplot* pour la fonction de Rastrigin

à expliquer, l'expression de la linéarité mettant en jeu les variables explicatives (prédicteur

linéaire) et la fonction qui relie le prédicteur à l'espérance mathématique de la variable à expliquer (fonction de lien).

On note y le vecteur de taille N des observations, réalisation du vecteur aléatoire Y (variable à expliquer). On suppose que les composantes Y_i ($i = 1, \dots, N$) de Y sont indépendantes et distribuées selon une loi appartenant à la famille exponentielle au sens de Nelder (McCullagh et Nelder, 1989). La fonction de densité de la variable Y_i s'écrit

$$f(y_i, \theta_i) = \exp \left(\frac{y_i \theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi) \right),$$

où θ_i est un paramètre *canonique* et ϕ un paramètre de *dispersion* > 0 . Les fonctions b et c sont spécifiques à chaque distribution. Cette famille de lois regroupe un certain nombre de lois classiques : binomiale, Poisson, normale, gamma, etc. En particulier la loi gamma, admet pour densité

$$\forall y \in R_+, \quad G(\alpha, \lambda)(y) = \frac{\lambda^\alpha}{\Gamma(\alpha)} y^{\alpha-1} e^{-\lambda y}, \quad \lambda > 0, \alpha > 0.$$

Son espérance est égale à α/λ et sa variance est égale à α/λ^2 . Les expressions du paramètre canonique θ , de ϕ et de a , b en fonction des paramètres naturels de la loi gamma sont

$$\theta = \frac{\lambda}{\alpha}, \quad b(\theta) = \log(\theta), \quad \phi = \frac{1}{\alpha}, \quad a(\phi) = -\phi.$$

Comme dans les modèles linéaires standards, les variables explicatives interviennent linéairement dans la modélisation. On définit le prédicteur linéaire par

$$\eta = X\beta,$$

où β est un vecteur de paramètres inconnus de taille d , et X une matrice $N \times d$ connue, fixée par l'expérience. La relation liant β et θ est décrite à l'aide la *fonction de lien* g :

$$X\beta = g(b'(\theta)).$$

Les coefficients β sont estimés par maximum de vraisemblance. La mise en œuvre de cette méthode est décrite ci-dessous. D'après l'hypothèse d'indépendance des coordonnées de Y , la log-vraisemblance du vecteur des paramètres canoniques θ , au vu du vecteur d'observations y , est

$$\log L(\theta; y) = \sum_{i=1}^N \left[\frac{y_i \theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi) \right] = \sum_{i=1}^N \log L_i(\theta_i; y_i).$$

Ainsi, la fonction $\log L$, vue comme log-vraisemblance du vecteur de paramètres β , peut être dérivée par rapport à ses diverses composantes. Notons

$$\mu_i = E[Y_i] = b'(\theta_i).$$

Les dérivées sont alors, pour tout $i \in \{1, \dots, N\}$, $j \in \{1, \dots, d\}$,

$$\begin{aligned} \frac{\partial \log L_i}{\partial \beta_j} &= \frac{\partial \eta_i}{\partial \beta_j} \frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \log L_i}{\partial \theta_i} = X_{ij} \frac{1}{g'(\mu_i)} \frac{1}{b''(\theta_i)} \frac{y_i - \mu_i}{a(\phi)}, \\ \frac{\partial \log L}{\partial \beta_j} &= \sum_{i=1}^N X_{ij} \frac{1}{g'(\mu_i)^2 \text{Var}(Y_i)} g'(\mu_i) (y_i - \mu_i). \end{aligned}$$

Ainsi, en considérant la matrice diagonale

$$W_\beta = (\text{Var}(Y_i) g'(\mu_i)^2 \delta_{ij})_{i,j=1,\dots,N}$$

les équations du maximum de vraisemblance pour β s'écrivent :

$$X'W_\beta^{-1} \frac{d\eta}{d\mu}(y - \mu) = 0 \quad (1)$$

où

$$\frac{d\eta}{d\mu} = \left(\frac{d\eta_i}{d\mu_i} \delta_{ij} \right)_{i,j=1,\dots,N} = (g'(\mu_i) \delta_{ij})_{i,j=1,\dots,N}.$$

L'algorithme itératif mis en place pour résoudre ces équations est l'algorithme des scores de Fisher. Il procède aux itérations suivantes ($t \geq 0$)

$$\begin{aligned} \beta^{[t+1]} &= \beta^{[t]} - \left(E \left[\left\{ \frac{\partial^2 \log L}{\partial \beta \partial \beta'} \right\} \right]^{[t]} \right)^{-1} \frac{\partial \log L^{[t]}}{\partial \beta} \\ &= \beta^{[t]} + \left(X'W_{\beta^{[t]}}^{-1} X \right)^{-1} X'W_{\beta^{[t]}}^{-1} \frac{d\eta^{[t]}}{d\mu} (y - \mu^{[t]}). \end{aligned}$$

La fonction *déviante* est définie par :

$$D(y; \hat{\beta}) = 2 \left[\log L(y; y) - \log L(y; \hat{\beta}) \right]$$

où $L(y; \hat{\beta})$ est la valeur de la log-vraisemblance obtenue à partir du paramètre estimé $\hat{\beta}$. La fonction *déviante résiduelle* est définie comme la généralisation de la somme des carrés des résidus du cas gaussien :

$$D^*(y; \hat{\beta}) = \phi D(y; \hat{\beta})$$

Sous des conditions de régularité assez générales pour la fonction de vraisemblance et vérifiées dans le cas de la famille exponentielle, la déviante suit asymptotiquement une loi du χ^2 à $N - d$ degrés de liberté. Pour plus de détails sur les procédures d'estimation utilisées dans les GLM et leurs propriétés, on pourra se référer au livre de McCullagh et Nelder (1989).

Cette propriété de convergence en loi de la déviante permet de construire des tests de comparaison de modèles. On désire tester le modèle M_0 contre le modèle M_1 , avec $M_0 \subset$

M_1 . Notons respectivement D_0^* , D_1^* et d_0 , d_1 les déviations résiduelles et les degrés de liberté associés aux modèles M_0 et M_1 . Sous des hypothèses de régularité classiques, le test de niveau asymptotique α est défini par la région critique :

$$R_\alpha = \{\xi > \chi_{1-\alpha}^2(d_0 - d_1)\},$$

avec

$$\xi = \frac{D_0^* - D_1^*}{\phi},$$

où $\chi_{1-\alpha}^2(d_0 - d_1)$ est le quantile $1 - \alpha$ de la loi du χ^2 à $(d_0 - d_1)$ degrés de liberté. On rejette le modèle M_0 contre M_1 si $\xi > \chi_{1-\alpha}^2(d_0 - d_1)$.

Note. Les procédures que nous adoptons par la suite sont analogues à celles de l'analyse de la variance (*anova*). La distinction provient du fait que nous supposons que les variables observées sont distribuées selon la loi gamma plutôt que la loi normale. Les procédures gaussiennes de type *anova* peuvent aussi être utilisées à titre indicatif mais il est plus difficile de justifier les décisions prises.

3.3 Méthodologie

3.3.1 Modèle individuel

Nous souhaitons quantifier l'effet de chaque variable intervenant dans la solution produite par la famille d'algorithmes MOSES, notée $\{a_\psi\}$, sur un problème donné. Pour cela, nous considérons toutes les coordonnées du paramètre vectoriel, ainsi que leurs puissances.

Nous supposons que des simulations ont été effectuées pour une famille $\{x_\psi\}$ de valeurs du paramètre vectoriel ψ . La donnée de la famille $\{x_\psi\}$ constitue une partie du plan d'expériences. Pour caractériser le plan en entier, il est nécessaire de préciser le nombre de répétitions correspondant à chaque "cellule" x_ψ . Un prédicteur est alors construit. Les variables explicatives sont choisies comme les puissances des coordonnées de x_ψ

$$\eta_\psi = \beta_0 + p_\psi(x_\psi). \quad (2)$$

Nous supposons que l'observation y_ψ de la valeur donnée par l'algorithme correspondant au paramètre x_ψ est distribuée selon une loi gamma d'espérance mathématique

$$E[y_\psi] = \exp(\eta_\psi). \quad (3)$$

Le coefficient β_0 intervenant dans l'expression du prédicteur η_ψ est appelé *intercept*. Le polynôme p_ψ s'annule au point zéro. Dans la situation qui nous concerne, le paramètre ψ possède deux composantes : $\psi = (r, p)$. Ainsi, nous avons

$$p_\psi(x_{(r,p)}) = \beta_r^1 x_r + \beta_p^1 x_p + \beta_r^2 x_r^2 + \beta_p^2 x_p^2 + \dots = \varphi_r(x_r) + \varphi_p(x_p)$$

où φ_r et φ_p sont deux polynômes unidimensionnels. Les valeurs des coefficients du polynôme p_ψ traduisent l'effet du paramètre ψ sur la variable observée y_ψ . Le modèle est déterminé par le choix des monômes de p_ψ correspondant à des coefficients non nuls.

Note. Dans l'équation 3, nous utilisons un lien logarithmique. Cela garantit la positivité de la moyenne. D'autre part, nous travaillons avec des valeurs proches de zéro. Le lien log assure un meilleur comportement numérique des estimateurs de maximum de vraisemblance.

3.3.2 Modèle global

La section précédente concerne un ensemble de données issues d'un unique problème de recherche d'extremum. Il est intéressant de construire un modèle englobant les données issues de plans d'expériences qui correspondent à plusieurs problèmes distincts. Nous souhaitons non seulement estimer l'effet de chaque composantes du paramètre ψ sur la solution produite par les algorithmes, mais aussi l'effet de chaque problème. Pour cela, nous considérons qu'à chaque problème correspond un niveau f de facteur. Typiquement, s'il y a trois problèmes différents, f prend les valeurs 1, 2, 3. Le modèle doit prendre en compte les effets d'interaction entre les niveaux du facteur et les variables x_ψ . Le prédicteur est alors donné par la relation

$$\eta_{\psi f} = \beta_0 + p_\psi(x_\psi) + \alpha_f + p_{\psi f}(x_\psi) \quad (4)$$

Le coefficient α_f mesure l'effet propre au problème f sur la solution produite par l'algorithme a_ψ . Le polynôme $p_{\psi f}$ s'annule en zéro. Nous avons

$$p_{\psi f}(x_{(r,p)}) = \beta_{rf}^1 x_r + \beta_{pf}^1 x_p + \beta_{rf}^2 x_r^2 + \beta_{pf}^2 x_p^2 + \dots$$

Si l'on peut décider que ce polynôme est identiquement nul alors il est possible de conclure que le comportement final de l'algorithme est le même pour tous les problèmes pris en compte par le modèle, au facteur f près. En particulier, nous pouvons déterminer un paramètre optimal ψ_{opt} commun aux problèmes en question.

3.3.3 Plan d'expériences complet

Dans l'objectif d'identifier des modèles individuels ou globaux, nous introduisons le plan d'expériences suivant. Les données correspondent à l'évaluation finale de l'algorithme MOSES après 2000 évaluations dans le cas des fonctions test usuelles et 5000 évaluations dans le cas des problèmes d'approximation. Les paramètres r (*rayon*) et p (*probabilité de mutation*) sont discrétisés de la manière suivante :

$$\begin{aligned} x_{ri} &= 0.2i, & i &= 1, \dots, 20, \\ x_{pi} &= 0.1i, & i &= 1, \dots, 10. \end{aligned}$$

Pour les problèmes f_i , $i = 1, \dots, 5$, chaque cellule (correspondant à la valeur croisée des paramètres) comporte 10 répétitions. Le plan complet comporte donc 2000 données. Pour les problèmes d'approximation E_i , $i = 6, \dots, 9$, chaque cellule comporte 5 répétitions. Le plan comporte donc 1000 données. L'obtention de ces données nécessite quelques minutes de calcul (station IBM) pour les problèmes f_i et 4 à 5 heures pour les problèmes d'approximations.

3.3.4 Démarche

L'objectif du modèle individuel est d'identifier les polynômes φ_r et φ_p qui interviennent dans la relation

$$E[y] = \exp(\beta_0 + \varphi_r(r) + \varphi_p(p)) . \quad (5)$$

En d'autres termes, il s'agit de caractériser la dépendance entre l'évaluation moyenne de la solution produite par MOSES et les paramètres r et p . Cela permet d'estimer la valeur optimale de ces paramètres. L'identification des polynômes intervenant dans la relation (5) se fait par estimation et tests statistiques. Ayant à disposition les observations issues du plan d'expériences complet pour la fonction f_1 , nous cherchons un "bon" modèle correspondant à ces observations. Pour cela, nous partons d'un modèle surparamétré, c'est-à-dire construit à partir de polynômes de degrés élevés. Le paramètre statistique $\hat{\beta}$ est estimé et les t -values sont calculées (les valeurs des t -values donnent une indication de la significativité des composantes de $\hat{\beta}$). La significativité est testée grâce à la statistique de différence de déviance entre le modèle complet et le sous-modèle dans lequel les coefficients qui correspondent à une t -value faible sont éliminés. Un modèle individuel "optimal" est ainsi identifié. Ensuite, on cherche à évaluer la robustesse du modèle optimal à une réduction des données. Pour cela, un plan réduit est constitué en supprimant des données. Le modèle est "robuste" à une réduction du plan si les estimations obtenues avec le jeu de données restreint conduisent aux mêmes conclusions. Pour les problèmes suivants, la même démarche est mise en place. Dans la section 4, nous chercherons à valider le plan réduit construit à partir des observations obtenues à partir de f_1 en l'utilisant pour les autres problèmes usuels.

L'intérêt du modèle global est d'utiliser, en plus du plan courant, l'information contenue dans des plans d'expériences concernant des problèmes passés. Un objectif est de pouvoir constituer des classes de problèmes pour lesquelles l'algorithme MOSES se comporte au final de manière statistiquement identique. La constitution d'une classe se fait selon le critère suivant. On dit que K problèmes de minimisation appartiennent à une même classe si les effets des variables (composantes de ψ) et l'effet du problème lui-même n'interagissent pas sur les solutions produites en moyenne par l'algorithme. En particulier, les valeurs optimales des paramètres sont identiques pour les K problèmes. Du point de vue statistique, cela revient à tester la nullité du polynôme $p_{\psi f}$. Lorsque l'on crée une classe de cette manière, il est à nouveau essentiel de déterminer le nombre minimal de données qui permette de conclure à la même classe (c'est-à-dire au même modèle global). Constituer un plan complet est coûteux. Nous testons donc la robustesse du modèle global à la réduction du nombre de données.

On suppose maintenant qu'une classe contienne K problèmes. Nous souhaitons tester l'appartenance d'un nouveau problème à une telle classe. Nous utilisons alors le plan réduit pour ce problème. On choisit ensuite L problèmes au hasard dans la classe ($L \leq K$). Puis on construit un modèle englobant les L problèmes de la classe et le nouveau problème. Si le modèle optimal obtenu pour cet ensemble est différent du modèle de référence de la classe alors nous rejetons l'appartenance à la classe (en général, la nullité du polynôme $p_{\psi f}$ ne sera pas vérifiée). Cette démarche sera illustrée dans la section 4 à l'aide des problèmes $E_6 - E_9$.

Note. La notion de classe reposant sur un modèle global est une notion forte. Deux problèmes sont dans la même classe si le comportement de l’algorithme est similaire pour les deux problèmes. L’estimation du paramètre optimal par un modèle global sera alors meilleure que celle obtenue par la comparaison des modèles individuels car elle repose sur un nombre plus grand de données. On pourrait exiger d’une classe que seule la valeur du paramètre optimal soit invariante sur la classe. Malheureusement, le paramètre optimal ne peut être vu comme une variable ou un facteur car il ne peut être déterminé qu’à l’issue du plan.

3.4 Niveau de significativité des tests

Le seuil de confiance des tests est choisi en sorte de ne garder que les traits les plus importants du modèle, c’est-à-dire les coefficients du modèle les plus significatifs. Le seuil est donc choisi petit : typiquement de 0.01 à 0.001. Il y a plusieurs raisons qui justifient ce choix. La première raison est que le plan complet est un plan riche en données (plusieurs centaines). Ceci permet d’être sévère sur les t -values correspondant aux estimations des coefficients du modèle. De plus, nous souhaitons constituer des classes. Cette démarche est antagoniste de celle qui consiste à détecter les comportements fins des algorithmes. De tels comportements sont particuliers à chaque problème. Nous ne devons pas nous placer à un niveau de détail qui serait inférieur à celui donné par le plan d’expériences et la discrétisation du paramètre ψ . Cette condition permet en pratique de fixer précisément le niveau des tests (comme nous le verrons dans la section 4.2.3).

4 Résultats

Nos calculs ont été effectués à l’aide du logiciel *Splus* qui permet un traitement systématique de ce type de plans d’expériences à l’aide de modèles linéaires généralisés. La fonction *glm* de *Splus* donne, en particulier, les statistiques de déviance, l’estimation des paramètres statistiques $\hat{\beta}$ et les t -values. Il est facile de comparer deux modèles grâce au test de rapport de vraisemblance et à la fonction *anova* qui permet de présenter les résultats sous la forme standard de l’analyse de la variance. Les sorties graphiques et les intervalles de confiance sur les estimations sont aussi générés par ce logiciel.

4.1 Fonctions tests classiques

La démarche détaillée dans la section 3.3.4 est présentée de manière complète pour le problème de minimisation correspondant à la fonction test f_1 . Pour les autres problèmes tests, seules les conclusions sont présentées.

4.1.1 Problème test f_1 .

La démarche mise en œuvre dans ce paragraphe repose sur la régression polynomiale (à travers un lien logarithmique) de l’évaluation de la solution produite par l’algorithme MOSES par le paramètre vectoriel $\psi = (r, p)$ (équation 5).

La première étape de cette démarche consiste à déterminer le degré des polynômes φ_r et φ_p qui interviennent dans l'équation 5. Le plan d'expériences comporte 20 valeurs différentes pour le paramètre r et 10 valeurs pour le paramètre p . Un modèle reposant sur des polynômes φ_r et φ_p de degrés respectifs 10 et 8 sera certainement un modèle surparamétré. La statistique de déviance et le test associé (χ^2) montrent que l'on peut diminuer significativement la complexité du modèle. Voici la sortie du logiciel *Splus* sous la forme d'une table d'analyse de la déviance (2000 données).

Response: f1

	Terms	Resid. Df	Resid. Dev	Test	Df	Deviance	Pr(Chi)
1	poly(r, 7) + poly(p, 5)	1987	2109.216				
2	poly(r, 10) + poly(p, 8)	1981	2100.998	1 vs. 2	6	8.218626	0.37478

La déviance résiduelle associée au modèle $poly(r, 10) + poly(p, 8)$ est égale à 2100.998 pour 1981 degrés de liberté alors que la déviance résiduelle associée au modèle $poly(r, 7) + poly(p, 5)$ est égale à 2109.216 pour 6 degrés de liberté supplémentaires. La p -value, égale à 0.37478, est bien supérieure à 0.01 (voir section 3.4). Le modèle $poly(r, 7) + poly(p, 5)$ est donc retenu. La deuxième étape consiste à donner les estimations des coefficients des polynômes φ_r et φ_p dans le modèle pertinent $poly(r, 7) + poly(p, 5)$. Les estimations (*Value*) sont données avec leurs erreurs (*Std. Error*) et la t -value correspondante dans le tableau suivant.

Coefficients:

	Value	Std. Error	t value
(Intercept)	2.3001496	0.72489423	3.173083
r	-49.1513310	3.85526093	-12.749158
r^2	86.8236412	9.34600858	9.289917
r^3	-78.5897177	10.66460646	-7.369209
r^4	40.1207706	6.44432207	6.225755
r^5	-11.6122036	2.11935306	-5.479127
r^6	1.7741428	0.35803999	4.955153
r^7	-0.1109959	0.02430691	-4.566434
p	28.7288287	7.28032113	3.946094
p^2	-127.3098370	36.41776025	-3.495817
p^3	310.7372037	79.34864456	3.916100
p^4	-353.1474993	77.94627695	-4.530653
p^5	151.5188363	28.25578461	5.362401

Le paramètre de dispersion pour la structure gamma a été estimé à $\phi = 1.245$. Les t -values associées aux estimations sont élevées (> 3). Ceci indique que tous les coefficients sont significatifs à un niveau faible (inférieur à 10^{-3}). Il n'est donc plus possible de diminuer l'ordre du modèle. Les graphes des polynômes φ_r et φ_p sont représentés dans la figure 7. L'intervalle de confiance (niveau 0.05) sur ces courbes est aussi représenté. Il est possible déterminer graphiquement ou numériquement les valeurs des paramètres r_{opt} et p_{opt} pour lesquels les performances de l'algorithme MOSES sont en moyenne optimales. Pour le

problème de minimisation lié à la fonction f_1 , nous obtenons

$$r_{opt} \approx 0.8 \quad p_{opt} \approx 0.1.$$

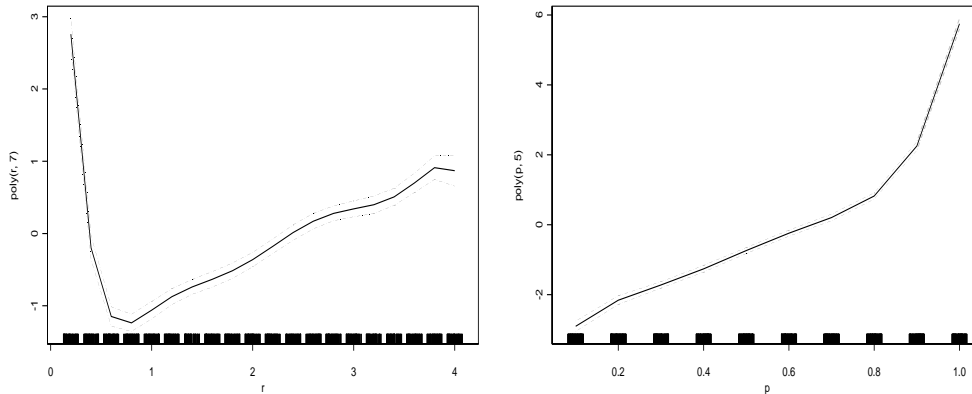


FIG. 7: Graphes des polynômes φ_r et φ_p pour la fonction f_1 obtenus avec 2000 observations

Les intervalles de confiance (figure 7) de largeur réduite témoignent de la richesse du plan d'expériences (le plan contient 10 répétitions de l'algorithme par cellule). Il est sans doute possible de réduire le nombre de données par cellule sans que cela n'affecte significativement les estimations précédentes. Nous reproduisons la démarche d'estimation et de tests en ne gardant qu'une unique répétition par cellule. Le nouveau jeu de données est donc de taille 200. Afin d'obtenir des conclusions robustes, nous itérons cette procédure pour 4 jeux indépendants de taille 200. Dans chacun des quatre cas, on compare le modèle optimal au modèle $poly(r, 7) + poly(p, 5)$.

Response: f1 avec 200 observations

	Terms	Resid. Df	Resid. Dev	Test	Df	Deviance	Pr(Chi)
Jeu de données 1							
1	poly(r, 6) + poly(p, 4)	189	217.3262				
2	poly(r, 7) + poly(p, 5)	187	212.0745	1 vs. 2	2	5.251704	0.09076
Jeu de données 2							
1	poly(r, 6) + poly(p, 5)	188	186.6887				
2	poly(r, 7) + poly(p, 5)	187	183.6248	1 vs. 2	1	3.063939	0.05276

Jeu de données 3

1	$\text{poly}(r, 5) + \text{poly}(p, 4)$	190	202.5935				
2	$\text{poly}(r, 7) + \text{poly}(p, 5)$	187	194.6927	1 vs. 2	3	7.900798	0.04811

Jeu de données 4

1	$\text{poly}(r, 5) + \text{poly}(p, 4)$	190	213.1554				
2	$\text{poly}(r, 7) + \text{poly}(p, 5)$	187	204.9935	1 vs. 2	3	8.161901	0.05300

La table d'analyse de la déviance permet de conclure de la manière suivante. Le modèle sélectionné dans chaque cas est différent du modèle construit avec toutes les observations. Toutefois, l'ordre de chaque modèle sélectionné est très proche de celui du modèle d'origine. De plus, les graphes des polynômes obtenus sont similaires (figure 8). Les estimations des paramètres optimaux sont inchangées. Les conclusions sont donc robustes à la réduction d'une observation par cellule.

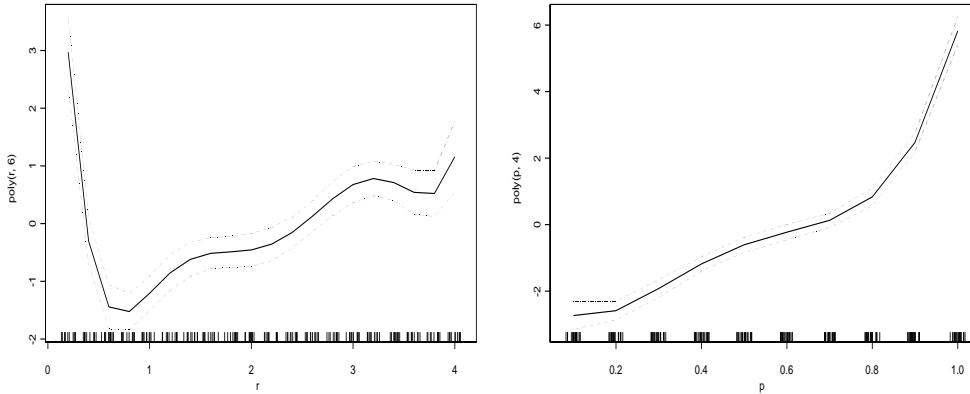


FIG. 8: Graphes des polynômes φ_r et φ_p pour la fonction f_1 obtenus avec 200 observations

4.1.2 Fonctions f_4 et f_5

Nous reprenons dans cette section la même démarche que dans la section 4.1.1. Comme précédemment, nous vérifierons que les conclusions obtenues avec 2000 observations sont robustes à une réduction du nombre des données à 200.

Pour la fonction f_4 , le modèle choisi pour démarrer est $\text{poly}(r, 8) + \text{poly}(p, 6)$. Le modèle finalement retenu est $\text{poly}(r, 5) + \text{poly}(p, 6)$.

	Terms	Resid. Df	Resid. Dev	Test Df	Deviance	Pr(Chi)
1	poly(r, 5) + poly(p, 6)	1988	121.0741			
2	poly(r, 8) + poly(p, 6)	1985	120.7934	1 vs. 2	3	0.2806839

Les coefficients estimés du modèle $poly(r, 5) + poly(p, 6)$ sont :

	Value	Std. Error	t value
(Intercept)	5.59899051	0.225008061	24.883511
r	-7.01525888	0.237330766	-29.558995
r^2	5.93559892	0.329557359	18.010822
r^3	-2.42257912	0.193426739	-12.524531
r^4	0.47968278	0.050267215	9.542657
r^5	-0.03686341	0.004766925	-7.733164
p	-9.64405819	4.101209679	-2.351516
p^2	72.03904082	26.997095874	2.668400
p^3	-250.35209975	83.356674681	-3.003384
p^4	449.78967036	131.074098961	3.431568
p^5	-401.04442152	101.536040549	-3.949774
p^6	141.02452540	30.711751545	4.591875

(Dispersion Parameter for Gamma family taken to be 0.059268).

Les t -values élevées montrent qu'il n'est pas possible de diminuer l'ordre du modèle $poly(r, 5) + poly(p, 6)$. Les graphes des polynômes φ_r et φ_p sont représentés dans la figure 9. À l'opposé du problème f_1 , on ne peut pas expliciter de valeur optimale précise des paramètres. On détecte en fait une plage optimale pour chaque paramètre. La longueur de cette plage est fonction de la zone de confiance associée aux polynômes φ_r et φ_p . Le paramètre p peut être choisi inférieur à 0.3 et le paramètre r peut être choisi entre 1.4 et 2.5.

Avec 200 observations, les conclusions sont identiques. Le modèle optimal s'écrit $poly(r, 5) + poly(p, 5)$. Pour ce modèle, les graphes des polynômes φ_r et φ_p (figure 10) sont similaires à ceux obtenus pour 2000 observations (figure 9).

L'étude du problème f_5 conduit, avec 2000 observations, à retenir le modèle $poly(r, 7) + poly(p, 5)$. Avec 200 observations, le modèle retenu est $poly(r, 5) + poly(p, 4)$. Les graphes des polynômes φ_r et φ_p obtenus avec 2000 observations et 200 observations sont représentés dans les figures 11 et 12. Pour le problème de minimisation lié à la fonction f_5 , les paramètres optimaux de l'algorithme MOSES sont

$$r_{opt} \approx 1.5 \quad p_{opt} \approx 0.1.$$

Note. Pour le problème f_5 , les intervalles de confiance sont plus étalés que pour les autres problèmes. Ceci est dû au choix de la distribution gamma qui, dans ce cas, est plus arbitraire.

4.1.3 Fonctions f_2 et f_3 .

Pour ces deux fonctions, les conclusions sont identiques à celles des sections 4.1.1 et 4.1.2. Il est possible de réduire le plan d'expérience à une simulation par cellule en conservant les

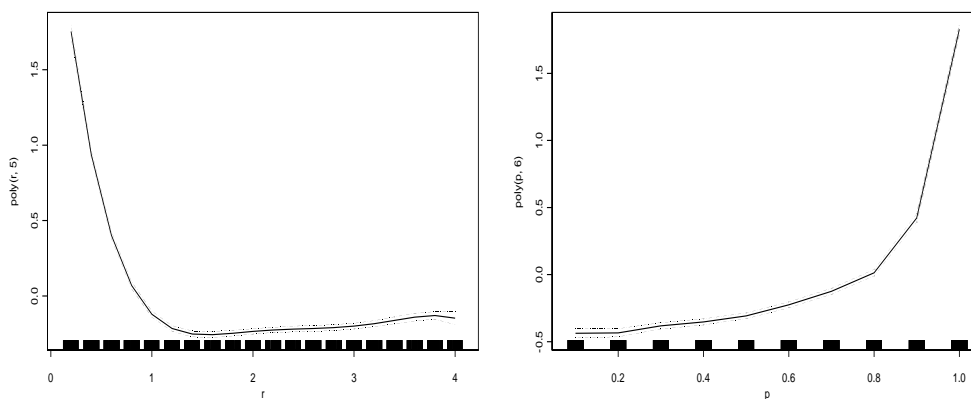


FIG. 9: Graphes des polynômes φ_r et φ_p pour la fonction f_4 obtenus avec 2000 observations

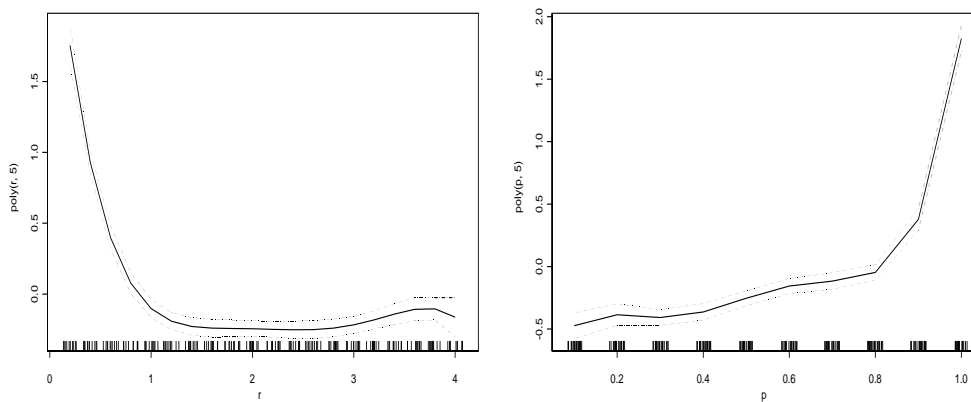


FIG. 10: Graphes des polynômes φ_r et φ_p pour la fonction f_4 obtenus avec 200 observations

mêmes estimations. Nous ne donnons dans cette section que les 2 graphes obtenus par modélisation sur un jeu de données de 200 observations (figure 13 et 14).

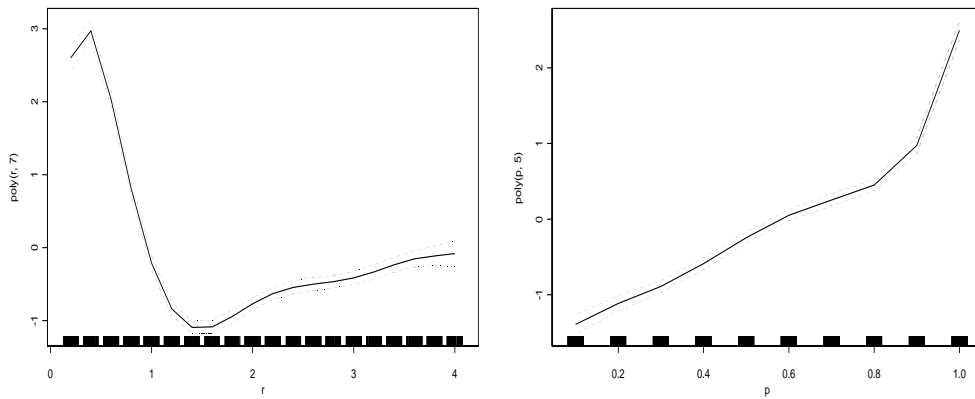


FIG. 11: Graphes des polynômes φ_r et φ_p pour la fonction f_5 obtenus avec 2000 observations

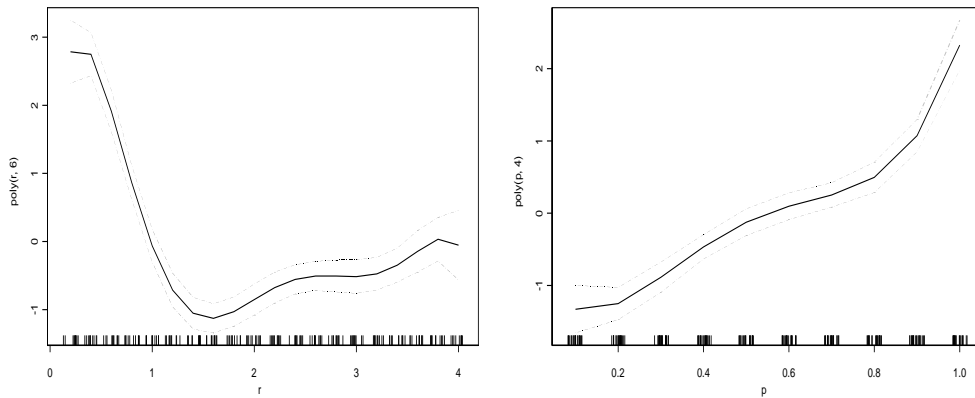


FIG. 12: Graphes des polynômes φ_r et φ_p pour la fonction f_5 obtenus avec 200 observations.

Pour le problème de minimisation lié à la fonction f_2 , les paramètres optimaux de l'algorithme MOSES sont

$$r_{opt} \approx 1.4 \quad p_{opt} \approx 0.1 .$$

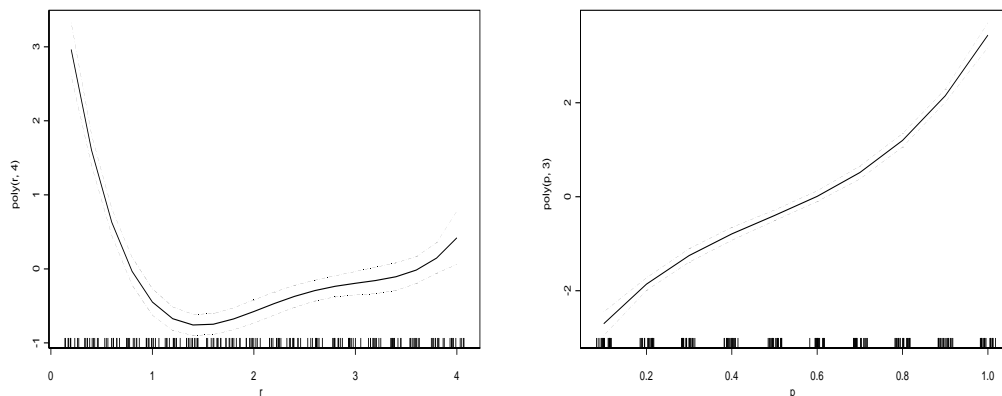


FIG. 13: Graphes des polynômes φ_r et φ_p pour f_2 obtenus avec 200 observations.

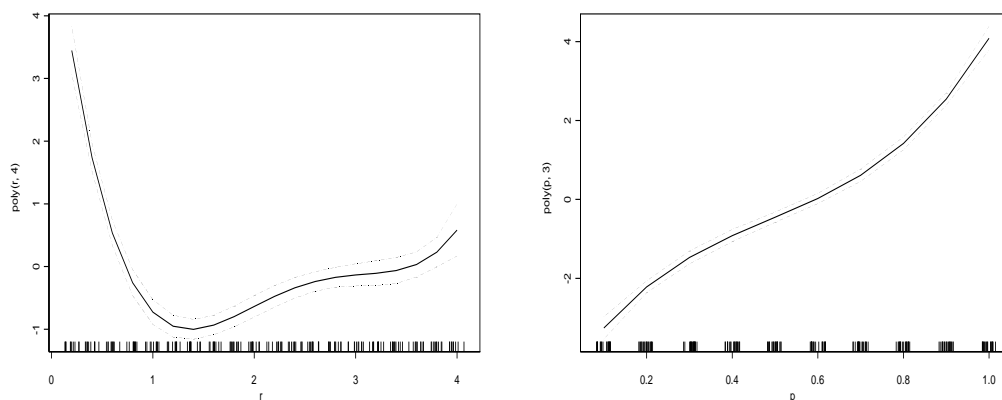


FIG. 14: Graphes des polynômes φ_r et φ_p pour f_3 obtenus avec 200 observations.

Pour le problème de minimisation lié à la fonction f_3 , les paramètres optimaux de l'algorithme MOSES sont

$$r_{opt} \approx 1.3 \quad p_{opt} \approx 0.1 .$$

4.1.4 Conclusion

L'outil statistique est utile pour estimer les paramètres optimaux de l'algorithme MOSES lorsque l'on dispose d'un nombre important de données de simulations croisées. L'étude menée sur cinq problèmes usuels montre qu'il n'est pas nécessaire de répéter un grand nombre de simulations par cellule. Cela permet de choisir la discrétisation des paramètres de la manière la plus fine possible (en fonction de la ressource de calcul disponible). Nous avons montré qu'un nombre réduit de simulations permet de quantifier correctement l'évaluation de la solution produite par l'algorithme. L'incertitude est donnée par les intervalles de confiance associés à chaque graphe des polynômes φ_r et φ_p .

4.1.5 Globalisation

Dans cette section, nous discutons de l'existence de classes de problèmes de minimisation pris dans l'ensemble $\{f_1, \dots, f_5\}$. L'existence d'une classe permet d'affiner l'estimation des paramètres optimaux en réduisant la taille des intervalles de confiance. Afin de constituer de telles classes, il est nécessaire d'examiner toutes les paires $\{f_i, f_j\}$. Pour fixer les idées, considérons la paire $\{f_2, f_3\}$. D'après les résultats de la section 4.1.3, ces deux fonctions semblent posséder des caractéristiques similaires. Ainsi, la paire $\{f_2, f_3\}$ est un candidat idéal pour constituer une classe. Nous construisons des modèles sur le jeu de données qui correspond à $\{f_2, f_3\}$ (4000 données). Pour cela, nous introduisons un facteur *fonction* f à 2 niveaux. Les degrés des polynômes φ_r et de φ_p sont égaux à 5. Voici les différentes statistiques de déviance résiduelle pour 4 modèles différents. Les trois premiers modèles comportent une interaction entre les effets des paramètres r et p et celui du facteur f . Le dernier modèle (modèle 4) ne contient pas d'interaction.

	Terms	Resid.	Df	Resid.	Dev
1	$\text{poly}(r,5) + \text{poly}(p,5)* f$	3978		2350.368	
2	$\text{poly}(r,5) + \text{poly}(p,5)* f$	3983		2366.463	
3	$\text{poly}(r,5)* f + \text{poly}(p,5)$	3983		2417.777	
4	$\text{poly}(r,5) + \text{poly}(p,5)+ f$	3988		2430.092	

Dans le modèle 2, on cherche à tester l'interaction entre le facteur f et le polynôme φ_r . Dans ce cas, la statistique de test du modèle 2 contre le modèle 1 vaut 31.6898 pour une différence de degrés de liberté égale à 5. Or, la probabilité pour qu'une variable de χ^2 à 5 degrés de liberté dépasse la valeur 31.69 est égale à 6.84e-006. Compte tenu du seuil de test égal à 10^{-3} , il y a une interaction significative entre le facteur f et l'effet des variables r et p . Il est donc impossible de considérer que les 2 problèmes appartiennent à une même classe. Dans les autres modèles (modèles 3,4), la statistique de test prend des valeurs encore plus élevées, et les conclusions sont identiques.

La même démarche a été menée pour chaque paire $\{f_i, f_j\}$. Les résultats sont résumés dans le tableau 2. Deux modèles sont considérés. Le modèle M_0 s'écrit $\text{poly}(r, 7) + \text{poly}(p, 6) + f$ (3972 degrés de liberté). Il ne prend pas en compte l'interaction de f avec les paramètres

r et p . Le modèle M_1 s'écrit $(poly(r, 7) + poly(p, 6)) * f$ (3985 degrés de liberté). Il s'agit d'un modèle qui contient M_0 et dans lequel toutes les interactions sont a priori présentes.

	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
modèle M_0	3890	4201	4734	6209	2426	2610	3718	3537	4393	2640
modèle M_1	3106	3448	2225	3985	2346	1123	2882	1465	3225	2001

TAB. 2: Déviations résiduelles des modèles M_0 et M_1 . Le test de différence de déviations permet de rejeter le modèle M_0 dans chaque cas.

Le tableau 2 permet de construire le test de significativité de l'interaction (appartenance à une même classe). Les 2 fonctions les plus susceptibles d'appartenir à une même classe sont en fait les fonctions f_2 et f_3 . La différence de déviations entre les modèles M_0 et M_1 est dans ce cas la plus petite possible (égale à 80). Le test de différence de déviations permet de rejeter le modèle M_0 dans chaque cas. En conclusion, nous ne pouvons considérer aucune classe de problèmes parmi $\{f_1, \dots, f_5\}$. Pour les problèmes tests, l'analyse de l'algorithme MOSES ne peut se faire que de manière individuelle. En ce sens, ces problèmes constituent des benchmarks significatifs pour cette famille d'algorithme.

4.2 Problèmes d'approximation

Il n'a pas été possible de constituer de classe à partir des problèmes f_1, \dots, f_5 . Cela témoigne d'un fonctionnement significativement différent de l'algorithme MOSES pour ces 5 problèmes. Nous allons voir dans cette section que la situation est différente en ce qui concerne les problèmes d'approximation. Il sera possible de constituer une classe de référence à partir des problèmes $E_6 - E_9$ à un niveau de significativité raisonnable. L'estimation du paramètre optimal ψ_{opt} aura de fait un intervalle de confiance plus étroit que si l'on utilisait l'information sur chaque problème séparément. Nous ne décrivons pas de modèle individuel dans cette section. Cette démarche est inutile dès lors qu'il est possible d'utiliser des modèles globaux. Nous regroupons en premier lieu les problèmes E_6 et E_7 pour lesquels l'algorithme ne fait aucune différence (pas d'effet du choix du problème). Nous regroupons ensuite les problèmes E_6, E_7 et E_8 . Pour ceux-ci, l'effet du facteur "problème" est non nul mais il n'y a pas d'interaction entre cet effet et celui des paramètres r et p . Finalement, nous discuterons de la classe $E_6 - E_9$ et du test d'appartenance à cette classe.

Pour chaque problème d'approximation, le plan d'expériences est identique. Il comporte 5 répétitions pour chaque croisement des variables x_r et x_p . Les jeux de données sont de taille 1000.

4.2.1 Regroupement des problèmes d'approximation E_6 et E_7

Dans cette section, nous testons l'existence de la classe $\{E_6, E_7\}$. Nous introduisons un facteur f à deux niveaux correspondant aux problèmes E_6 et E_7 . Le modèle choisi pour

initialiser la démarche s'écrit $(poly(r, 8) + poly(p, 6)) * f$. Il inclut toutes les interactions entre le facteur f et les composantes des 2 polynômes. Dans un premier temps, toutes les interactions avec le facteur f sont supprimées. Le modèle correspondant s'écrit donc $poly(r, 8) + poly(p, 6) + f$.

	Terms	Resid. Df	Resid. Dev	Df	Deviance	Pr(Chi)
1	poly(r,8)+poly(p,6)+f	1984	572.7175			
2	(poly(r,8)+poly(p,6))*f	1970	568.4354	14	4.28211	0.58970

La p -value est élevée. Cela signifie que le polynôme $p_{\psi f}$ est significativement nul. D'après la section 3.3.4, les problèmes E_6 et E_7 sont dans la même classe pour l'algorithme MOSES. De plus, la déviance résiduelle du modèle $poly(r, 8) + poly(p, 6)$ est égale à 572.72 pour 1985 degrés de liberté. Il n'y a pas d'effet significatif du facteur f ($\alpha_f = 0$). L'algorithme se comporte de manière identique pour les 2 problèmes. Lorsque l'on cherche à réduire l'ordre du modèle, on obtient les résultats suivants.

	Terms	Resid. Df	Resid. Dev	Df	Deviance	Pr(Chi)
1	poly(r,3)+poly(p,4)	1992	578.0422			
2	(poly(r,8)+poly(p,6))*f	1970	568.4354	22	9.60678	0.19694

Les t -values du modèle $poly(r, 3) + poly(p, 4)$ sont toutes supérieures à 3.0. Il s'agit du modèle optimal pour la classe $\{E_6, E_7\}$.

4.2.2 Regroupement des problèmes E_6, E_7 et E_8

Nous cherchons à constituer une classe contenant les problèmes $\{E_6, E_7, E_8\}$. Le jeu de données est de taille 3000. Nous utilisons un facteur f à deux niveaux. Les problèmes E_6 et E_7 sont regroupés dans le premier niveau. Le second niveau correspond au problème E_8 . La table ci dessous montre que, dans le modèle $(poly(r, 8) + poly(p, 6)) * f$, aucune interaction n'est significative.

	Terms	Resid. Df	Resid. Dev	Df	Deviance	Pr(Chi)
1	poly(r,8) + poly(p,6) + f	2984	772.8264			
2	(poly(r,8) + poly(p,6)) * f	2970	767.3329	14	5.493463	0.23860

Le problème E_8 appartient à la même classe que les problèmes E_6 et E_7 . En revanche, le facteur f possède maintenant un effet très significatif. Le modèle retenu est $poly(r, 3) + poly(p, 5) + f$. Il possède une déviance résiduelle égale à 775.69 pour 2990 degrés de liberté. L'évaluation de la solution produite par MOSES dépend de chaque problème. Toutefois, le comportement final de l'algorithme en fonction des paramètres r et p est invariant sur la classe.

4.2.3 Globalisation $E_6 - E_9$

Nous cherchons à constituer une classe contenant les problèmes $\{E_6, E_7, E_8, E_9\}$. Le jeu de données est de taille 4000. Nous introduisons un facteur f à 3 niveaux. Dans un premier temps, nous testons la significativité des interactions entre le facteur f et le polynôme φ_p .

	Terms	Resid.	Df	Resid.	Dev	Df	Deviance	Pr(Chi)
1	poly(r,8)* f + poly(p,6)	3967		1080.759				
2	(poly(r,8) + poly(p,6))* f	3955		1076.708	12	4.050462	0.35662	

Les interactions entre le facteur f et le paramètre p ne sont pas significatives. On cherche maintenant à supprimer les interactions entre le facteur f et le paramètre r .

	Terms	Resid.	Df	Resid.	Dev	Df	Deviance	Pr(Chi)
1	poly(r,8) + poly(p,6) + f	3983		1092.291				
2	poly(r,8) * f + poly(p,6)	3967		1080.759	16	11.5322	0.001962	

Dans ce cas, la p -value est peu élevée. Pour un niveau de test (par exemple) égal à 0.05, les interactions entre le facteur f et le paramètre r sont significatives. Toutefois, la discussion menée à la section 3.4 conclut en faveur d'un niveau de test plus faible. Il est donc raisonnable d'accepter l'hypothèse d'une classe $E_6 - E_9$ pour laquelle le comportement final de l'algorithme est globalement identique. Le modèle correspondant à cette classe s'écrit $poly(r,3) + poly(p,4) + f$ (la déviance résiduelle est égale à 1099.39 pour 3990 degrés de liberté).

4.2.4 Réduction du plan

Comme cela a été fait dans la section 4.1.1, nous proposons de réduire le nombre d'observations et d'analyser la robustesse des conclusions précédentes (existence de classe). Nous réduisons le plan en diminuant le nombre de répétitions pour chaque croisement des variables x_r et x_p . Pour la classe de problèmes d'approximation $E_6 - E_9$, lorsque l'on réduit à une observation par croisement, les modèles proposés sur des jeux de données distincts deviennent significativement différents. Cela peut s'expliquer par la complexité des problèmes d'approximation. Cette complexité est certainement plus élevée que pour les fonctions introduites dans la section 2.1.1.

En revanche, si l'on utilise 2 observations par croisement, les conclusions obtenues à la section précédentes sont "robustes". Nous donnons les résultats (test et estimation des paramètres dans le modèle $poly(r,3) + poly(p,4) + f$ et les graphes correspondants (figure 15) pour un jeu de données de taille $4 \times 2 \times 200 = 1600$.

	Terms	Resid.	Df	Resid.	Dev	Df	Deviance	Pr(Chi)
1	poly(r,3) + poly(p,4) + f	1590		441.1227				
2	(poly(r,8) + poly(p,6)) * f	1555		425.9549	35	15.16779	0.10334	

	Value	Std. Error	t value
(Intercept)	-3.79652211	0.1755074	-21.631694
r	-0.81133955	0.1495469	-5.425317
r^2	0.34370327	0.0816882	4.207502
r^3	-0.04256202	0.0128052	-3.323806
p	-3.50869706	1.7919751	-1.958005
p^2	15.46275968	6.2063203	2.491454
p^3	-24.05242920	8.2933123	-2.900220

p^{-4}	12.61449758	3.7490879	3.364684
f_2 (E8)	0.95154617	0.0372731	25.529033
f_3 (E9)	0.77552739	0.0372731	20.806625

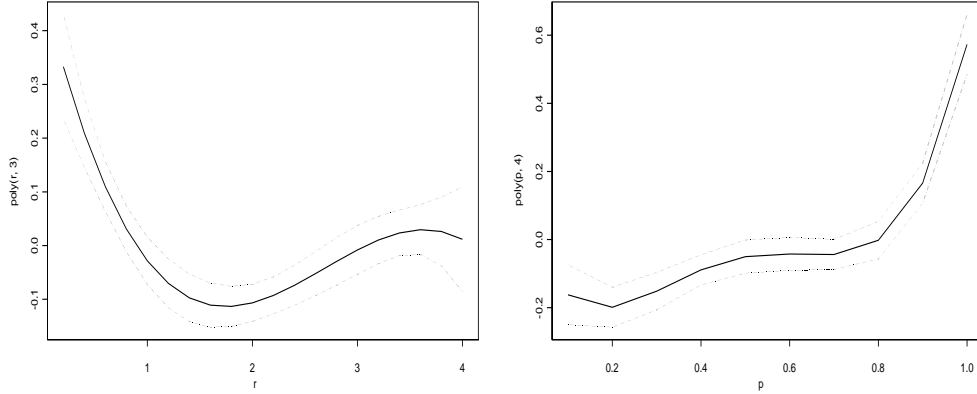


FIG. 15: Graphes des polynômes φ_r et φ_p pour la classe des problèmes d'approximation obtenus avec $4 * 400$ observations

L'estimation des paramètres optimaux se fait de la manière suivante. Compte tenu des intervalles de confiance associés à chaque graphe (figure 15), le paramètre p_{opt} se situe entre 0.1 et 0.4. Le paramètre r_{opt} se situe entre 1.3 et 2.2. Lorsque l'on ne globalise pas, l'intervalle de confiance sur les paramètres optimaux est environ deux fois plus large.

Conclusion. Nous avons abouti à la constitution d'une classe de problèmes pour lesquels l'algorithme MOSES se comporte de manière identique. Pour cela, nous avons utilisé un niveau de test relativement bas (10^{-3}). À titre de comparaison, il faut un niveau de test de l'ordre de 10^{-6} pour regrouper f_2 et f_3 . Nous avons de plus montré que cette classe est robuste lorsque l'on réduit le plan à 2 répétitions par cellule. Ce plan réduit sera systématiquement utilisé pour tester l'appartenance à cette classe de nouveaux problèmes.

4.2.5 Test d'appartenance à la classe $E_6 - E_9$

Afin de tester si un nouveau problème appartient à la classe $E_6 - E_9$, nous utilisons le plan d'expériences réduit. Les nouvelles données sont superposées aux jeux de données précédents (1600 + 400 données en tout). Un facteur *classe* c à deux niveaux est alors introduit. Le premier niveau est associé à la classe $E_6 - E_9$. Le deuxième niveau est associé aux nouvelles

données. Un niveau supplémentaire est ajouté au facteur f . Un nouveau problème appartient à la classe $E_6 - E_9$ si les interactions du facteur c dans le modèle $(poly(r, 8) + poly(p, 6)) * c + f$ ne sont pas significatives.

Le test d'appartenance est appliqué aux fonctions f_j , $j = 1, \dots, 5$. Les résultats de déviance résiduelle des modèles $M_1 = (poly(r, 8) + poly(p, 6)) * c + f$ (1968 degrés de liberté) et $M_0 = poly(r, 8) + poly(p, 6) + f$ (1982 degrés de liberté) sont présentés dans le tableau 3.

	f_1	f_2	f_3	f_4	f_5
modèle M_0	2285.97	1460.78	1754.22	588.25	1295.51
modèle M_1	835.76	638.97	702.35	461.52	823.81

TAB. 3: Déviances résiduelles des modèles M_0 et M_1 . La différence la plus faible est obtenue pour f_4 .

La différence la plus faible est obtenue pour f_4 . Dans ce cas, le test de différence des déviances permet de rejeter le modèle M_0 (l'interaction du facteur c est significative). Ces résultats montrent qu'aucune des 5 fonctions ne peut être associée à la classe $E_6 - E_9$.

4.2.6 Validation

Afin de valider les résultats expérimentaux précédents, l'algorithme a été utilisé sur les problèmes $E_6 - E_9$ avec deux configurations différentes de paramètres. La première configuration est $p = 0.2$, $r = 0.6$. Dans ce cas, le nombre d'évaluations est égal à 80,000 (plutôt que 5000). La seconde configuration est $p = 0.2$, $r = 1.6$. Le nombre d'évaluations est égal à 50,000. Les fonctions sont correctement approchées par le réseau de neurones. Dans tous les cas, les simulations montrent que la meilleure évaluation est donnée par la seconde configuration qui se situe dans la plage optimale (figure 16). Avec une configuration sous-optimale, l'approximation est plus grossière, bien que le nombre d'évaluations soit plus élevé.

Il est difficile d'extrapoler ces résultats. Le paramétrage optimal a été déterminé pour une valeur fixée du nombre d'évaluations (5000). En général, augmenter le nombre d'évaluations a pour effet de diminuer le rayon optimal. Toutefois, l'exemple des problèmes d'approximation montre que l'extrapolation à de plus grands nombres d'évaluations est raisonnable.

5 Conclusion

Dans cet article, nous avons présenté une méthodologie statistique destinée à aider l'utilisateur d'algorithmes évolutionnaires à configurer ses algorithmes. L'outil statistique permet de piloter de manière pertinente et efficace le plan de simulations inhérent à toute phase de réglage.

Ce travail ne contient ni recette immédiatement applicable, ni conclusion directement répliquative. Toutefois, la méthodologie présentée permet de prendre en compte efficacement

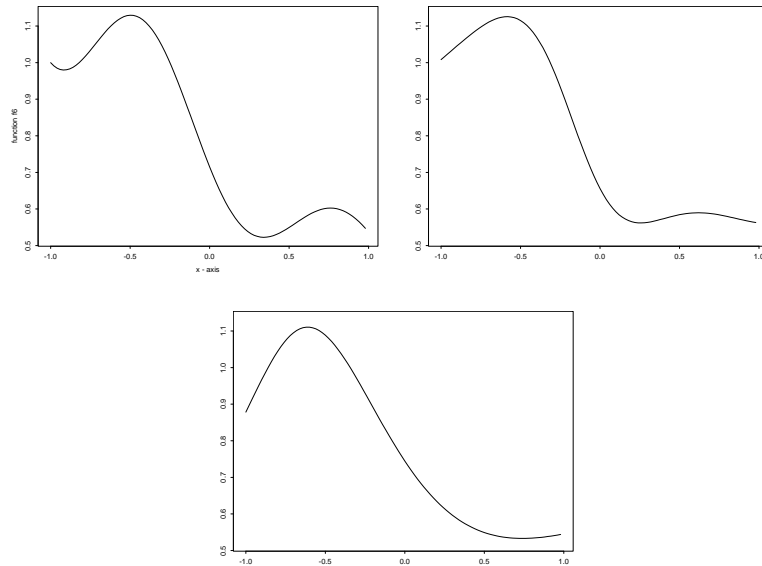


FIG. 16: Graphes de la fonction f_6 , d'une solution trouvée avec un paramètre optimal (50,000 évaluations), d'une solution trouvée avec un paramètre non-optimal (80,000 évaluations).

les expériences croisées ou passées. En utilisant l'information contenue dans toutes les simulations, un modèle statistique permet de rationaliser les plans d'expériences. En particulier, pour de nombreux problèmes, il est possible de gérer un très petit nombre de simulations pour chaque combinaison de paramètres.

Les procédures d'estimation et de tests permettent d'identifier les effets de la variation des paramètres de l'algorithme sur la solution produite. La possibilité de constituer des classes de problèmes permet d'améliorer la qualité des estimations au même coût de simulation. Le modèle statistique utilisé dans ce travail est un modèle de type "boîte noire". Il ne prend en compte aucune propriété géométrique du problème à minimiser. Il modélise la qualité des solutions produites par l'algorithme et non le fonctionnement de l'algorithme lui-même. Dans la plupart des problèmes réalistes, aucune propriété analytique du problème à résoudre n'est accessible. Un modèle de type "boîte noire" est naturel et la démarche proposée s'impose.

Cette méthode semble être dédiée à un contexte d'application spécifique, dans lequel on souhaite une évaluation correcte et rapide dans un environnement qui ne varie pas trop. Bien qu'il s'agisse d'une problématique académique, l'exemple de l'approximation par réseaux de neurones relève parfaitement de ce contexte.

Finalement, il est possible d'utiliser les mêmes techniques pour comparer différentes familles d'algorithmes. On répond à cette question en introduisant un nouveau facteur. À

chaque niveau d'un tel facteur correspond une famille d'algorithmes. Lorsque l'on souhaite tester l'effet de ce facteur, des procédures similaires à celles de cet article peuvent être utilisées.

Références

- [1] E.H.L Aarts, J.H.M Korst. Simulated annealing and Boltzmann machines, Wiley, New York, 1988.
- [2] F. Bergeret, P. Besse. Simulated annealing, weighted simulated annealing and genetic algorithms at work, *Computational Statistics*, 12, (1997), 447-465.
- [3] J.M. Chambers, T.J. Hastie. Statistical Models in S, Chapman & Hall 1991.
- [4] B.D. Fogel. Evolutionary Computation : toward a new philosophy of machine intelligence, IEEE Press, New-York, 1995.
- [5] O. François. An evolutionary strategy for global minimization and its Markov chain analysis, *IEEE transactions on Evolutionary Computation*, 2, (1998), To appear.
- [6] O. François. Convergence in simulated evolution algorithms, *Complex Systems*, 2, (1996), 311-325.
- [7] L. Ingber, B. Rosen. Genetic algorithms and very fast simulated reannealing : a comparison, *Mathematical and Computer Modelling*, 16, (1992), 87-100.
- [8] A.J. Keane. A brief comparison of some evolutionary optimization methods, *Research Report, Oxford University*, 1997.
- [9] L.C. Kwong Hui, K.Y. Lam, C.W. Chea. Global optimization in neural network training, *Neural Computing and Applications*, 5, (1997), 58-64.
- [10] P. McCullagh, J.A Nelder. Generalized linear models, London, Chapman & hall (2^e edition) 1989.
- [11] K. Park. A comparative study of genetic search, *Proceedings ICGA*, (1995), 512-519, Morgan-Kauffmann.
- [12] B.D. Ripley. Pattern recognition and neural networks, Cambridge University Press, 1996.
- [13] W.N. Venables, B.D. Ripley. Modern Applied Statistics with Splus, Springer-Verlag, New-York, (2^e edition) 1997.
- [14] D. Whitley, K. Mathias, S. Rana, J. Dzubera. Building better test functions, *Proceedings ICGA*, (1995), 512-519, Morgan-Kauffmann.
- [15] D.H. Wolpert, W.G. MacReady. No free lunch theorem for search, *Research Report, Sante Fe Institute*, 1995.
- [16] X. Yao. Evolutionary artificial neural networks, *International Journal of Neural Systems*, 4, (1993), 203-222.

Table des matières

1	Introduction	3
2	Contexte	4
2.1	Choix de problématiques	4
2.1.1	Fonctions test usuelles	4
2.1.2	Approximation par réseaux de neurones	5
2.2	Algorithmes	6
2.3	Travaux antérieurs	8
3	Méthode	9
3.1	Identification du modèle	9
3.2	Modèles linéaires généralisés	11
3.3	Méthodologie	15
3.3.1	Modèle individuel	15
3.3.2	Modèle global	16
3.3.3	Plan d'expériences complet	16
3.3.4	Démarche	17
3.4	Niveau de significativité des tests	18
4	Résultats	18
4.1	Fonctions tests classiques	18
4.1.1	Problème test f_1	18
4.1.2	Fonctions f_4 et f_5	21
4.1.3	Fonctions f_2 et f_3	22
4.1.4	Conclusion	26
4.1.5	Globalisation	26
4.2	Problèmes d'approximation	27
4.2.1	Regroupement des problèmes d'approximation E_6 et E_7	27
4.2.2	Regroupement des problèmes E_6 , E_7 et E_8	28
4.2.3	Globalisation $E_6 - E_9$	28
4.2.4	Réduction du plan	29
4.2.5	Test d'appartenance à la classe $E_6 - E_9$	30
4.2.6	Validation	31
5	Conclusion	31



Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399