



HAL
open science

Well placement optimization

Zyed Bouzarkouna

► **To cite this version:**

Zyed Bouzarkouna. Well placement optimization. Other [cs.OH]. Université Paris Sud - Paris XI, 2012. English. NNT: 2012PA112061 . tel-00690456

HAL Id: tel-00690456

<https://theses.hal.science/tel-00690456v1>

Submitted on 23 Apr 2012

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.

UNIVERSITE PARIS-SUD

ÉCOLE DOCTORALE : Informatique
Laboratoire de Recherche en Informatique

DISCIPLINE : INFORMATIQUE

THÈSE DE DOCTORAT

soutenue le 03/04/2012

par

Zyed BOUZARKOUNA

Optimisation de placement des puits

Directeur de thèse :
Co-directeur de thèse :

Marc SCHOENAUER
Anne AUGER

INRIA Saclay - Île-de-France, France
INRIA Saclay - Île-de-France, France

Composition du jury :

Président du jury :
Rapporteurs :

Jan Dirk JANSEN
Christian IGEL
Pierre THORE
Didier Yu DING

TU Delft, Pays-Bas
University of Copenhagen, Danemark
Total, Royaume-Uni
IFP Energies nouvelles, France

Examineurs :



Comprendre le monde,
construire l'avenir®

UNIVERSITY PARIS-SUD

ÉCOLE DOCTORALE : Informatique
Laboratoire de Recherche en Informatique

DISCIPLINE: COMPUTER SCIENCE

Ph.D. THESIS

defended on 03/04/2012

by

Zyed BOUZARKOUNA

Well placement optimization

Thesis advisor:
Thesis co-advisor:

Marc SCHOENAUER
Anne AUGER

INRIA Saclay - Île-de-France, France
INRIA Saclay - Île-de-France, France

Jury members:

President of the jury:
Reviewers:

Jan Dirk JANSEN
Christian IGEL
Pierre THORE
Didier Yu DING

TU Delft, Netherlands
University of Copenhagen, Denmark
Total, United Kingdom
IFP Energies nouvelles, France

Examiners:

To my Mother.

Acknowledgements

It would not have been possible to write this doctoral thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

First and foremost I want to thank my thesis advisor Marc Schoenauer. It has been an honor to be one of his Ph.D. students. I appreciate all his contributions of time, and funding to make my Ph.D. experience productive and stimulating.

This thesis would not have been possible without the help, support and patience of my thesis co-advisor, Anne Auger, not to mention her advice and unsurpassed knowledge of her research area: the Optimization. Her good advice, support and friendship has been invaluable on both an academic and a personal level, for which I am extremely grateful. The joy and enthusiasm she has for her research was contagious and motivational for me! I would like as well to express my deep and sincere gratitude to Didier Yu Ding, my advisor in IFP Energies nouvelles. I thank Didier for his strategic insight that made this research project both interesting and doable. He has spent a huge amount of time, energy and dedication into this research: thank you Didier!

The jury for my thesis has consisted of Christian Igel, Pierre Thore and Jan Dirk Jansen. I am grateful and honored that you have accepted to be part of the Jury of my PhD defense. In particular, I thank Christian Igel and Pierre Thore for taking the time to read and review the thesis.

I would like also to thank all the “Simulation of Flows and Transfers in Porous Media” department in the IFP Energies nouvelles, for having created such a nice and ideal environment. Special thanks to Benoit Noetinger, Mickaele Le Ravalec, Sébastien Da Veiga, Frédéric Roggero, Sylvie Hoguet, and all the others whose names were not mentioned here. It was an honor for me to be part of your team.

I would like also to thank all the “TAO team, INRIA Saclay - Île de France”, for their help, guidance and enjoying environment. Special and particular thanks to Nikolaus Hansen, whose scientific insight and experience was invaluable, for his helpful discussions and insightful comments. I thank also Mohamed, Mouadh, Ilya, Hassen, Raymond, and all the other TAO members.

Also, a lot of thanks to all the PhD students and all the friends I had the chance to know during my stay in IFP Energies nouvelles. I spent three very pleasant years in your company, I will surely miss it. In particular, I am thankful for Salem, Samir, Ekaterina, Franck, François, Romain, Ratiba, Marius, Fakher, Höel, and all the other nice persons from Rueil-Malmaison and Solaize I have known. We will keep in touch and we will surely meet again many times. “It’s A Small World!”, well, the world of Geosciences is even smaller!

I also thank all the friends I had, beginning from Gabés, to Tunis, then to Paris and now in Pau! They all know who they are.

Finally, I want to thank my family: my mother, my brother, my sister, my brother-in-law and my nephew.

This thesis is in particular dedicated to my mother: words can not express how grateful I am for all of the sacrifices that you have made on my behalf.

Abstract

The amount of hydrocarbon recovered can be considerably increased by finding optimal placement of non-conventional wells. For that purpose, the use of optimization algorithms, where the objective function is evaluated using a reservoir simulator, is needed. Furthermore, for complex reservoir geologies with high heterogeneities, the optimization problem requires algorithms able to cope with the non-regularity of the objective function. The goal of this thesis was to develop an efficient methodology for determining optimal well locations and trajectories, that offers the maximum asset value using a technically feasible number of reservoir simulations.

In this thesis, we show a successful application of the Covariance Matrix Adaptation - Evolution Strategy (CMA-ES) which is recognized as one of the most powerful derivative-free optimizers for continuous optimization. Furthermore, in order to reduce the number of reservoir simulations (objective function evaluations), we design two new algorithms. First, we propose a new variant of CMA-ES with meta-models, called the new-local-meta-model CMA-ES (nlmm-CMA), improving over the already existing variant of the local-meta-model CMA-ES (lmm-CMA) on most benchmark functions, in particular for population sizes larger than the default one. Then, we propose to exploit the partial separability of the objective function in the optimization process to define a new algorithm called the partially separable local-meta-model CMA-ES (p-sep lmm-CMA), leading to an important speedup compared to the standard CMA-ES.

In this thesis, we apply also the developed algorithms (nlmm-CMA and p-sep lmm-CMA) on the well placement problem to show, through several examples, a significant reduction of the number of reservoir simulations needed to find optimal well configurations. The proposed approaches are shown to be promising when considering a restricted budget of reservoir simulations, which is the imposed context in practice.

Finally, we propose a new approach to handle geological uncertainty for the well placement optimization problem. The proposed approach uses only one realization together with the neighborhood of each well configuration in order to estimate its objective function instead of using multiple realizations. The approach is illustrated on a synthetic benchmark reservoir case, and is shown to be able to capture the geological uncertainty using a reduced number of reservoir simulations.

Contents

1	Introduction	1
1.1	Problem statement	1
1.2	Literature review	5
1.2.1	Optimization algorithms	5
1.2.1.1	Deterministic methods	5
1.2.1.2	Stochastic methods	7
1.2.1.3	Search algorithms using surrogates	8
1.2.1.4	Hybrid methods	9
1.2.2	Well placement optimization	9
1.3	Thesis objectives and methodology	10
1.4	Summary of contributions	11
1.5	Dissertation road-map	13
2	CMA-ES and CMA-ES with meta-models	14
2.1	Covariance Matrix Adaptation - Evolution Strategy	14
2.2	Handling constraints with CMA-ES	20
2.3	CMA-ES with local meta-models	22
2.3.1	The local-meta-model CMA-ES (lmm-CMA)	23
2.3.1.1	Locally weighted regression	23
2.3.1.2	Approximate ranking procedure	24
2.3.2	Evaluating lmm-CMA on increasing population size	27
2.3.2.1	Experimental procedure	27
2.3.2.2	Performances of lmm-CMA with increasing population size	28
2.3.3	A new variant of lmm-CMA	28
2.3.3.1	A new meta-model acceptance criteria	28
2.3.3.2	Evaluation of nlmm-CMA	29
2.3.3.3	Impact of the recombination type	30
2.3.3.4	Impact of initial parameters	30
2.4	Summary and discussions	33

3	Well placement optimization with CMA-ES and CMA-ES with meta-models	35
3.1	The well placement optimization problem formulation	35
3.1.1	Objective function	36
3.1.2	Well parameterization	37
3.2	CMA-ES and a real-coded GA for the well placement problem	39
3.2.1	Well placement using CMA-ES	39
3.2.2	Well placement using GA	39
3.2.3	Well placement performance	41
3.3	Application of CMA-ES with meta-models on the PUNQ-S3 case	45
3.3.1	Placement of one unilateral producer and one unilateral injector	46
3.3.2	Placement of one multi-segment producer	50
3.4	Summary and discussions	52
4	Local-meta-model CMA-ES for partially separable functions	54
4.1	Partial separability and problem modeling	54
4.2	Imm-CMA for partially separable functions	56
4.3	Evaluation of p-sep Imm-CMA	59
4.3.1	Test functions	59
4.3.2	Performance of p-sep Imm-CMA	59
4.3.3	Optimal bandwidth for building partially separated meta-models	63
4.3.4	Computational cost	64
4.4	Summary and discussions	64
5	Partially separated meta-models with CMA-ES for well placement optimization	66
5.1	p-sep Imm-CMA for well placement optimization	66
5.2	Application of p-sep Imm-CMA on the PUNQ-S3 case	68
5.3	Summary and discussions	73
6	Well placement optimization under geological uncertainty	75
6.1	Optimization under uncertainty: a literature review	75
6.1.1	Optimization community	76
6.1.1.1	Explicit Averaging	77
6.1.1.2	Implicit Averaging	78
6.1.2	Petroleum community	78
6.2	Well placement under uncertainty with CMA-ES using the neighborhood	80
6.3	Application of CMA-ES using the neighborhood approach on the PUNQ-S3 case	82

6.4 Summary and discussions	89
7 Conclusions and perspectives	90
7.1 Conclusions	90
7.2 Perspectives	91
Nomenclature	93
Résumé en Français	95
References	105

List of Figures

1.1	Brent crude oil price (in US dollar), Oct 2007 - Sep 2011	2
2.1	Geometrical representation of a 2-dimensional multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$	16
2.2	Speedup of nlmm-CMA and lmm-CMA on $f_{\text{Schw}^{1/4}}$, f_{Rosen} and f_{Rast} for dimension $n = 5$	27
3.1	An example of a single multilateral well parameterization with two segments ($\mathcal{N}_s = 2$) and one branch ($\mathcal{N}_b = 1$)	38
3.2	Elevation (in meters) and geometry of the PUNQ-S3 test case	41
3.3	The mean value of NPV (in US dollar) and its corresponding standard deviation for well placement optimization using CMA-ES and GA. Fourteen runs are performed for each algorithm. Constraints are handled using an adaptive penalization with rejection technique for CMA-ES and using Genocop III for GA	42
3.4	The mean number of unfeasible individuals per generation and its corresponding standard deviation using CMA-ES with an adaptive penalization with rejection technique. Here, we consider only unfeasible individuals far from the feasible domain, i.e., resampled individuals	42
3.5	The positions of solution wells found by 14 runs of CMA-ES projected on the top face of the reservoir	43
3.6	The positions of solution wells found by 14 runs of the GA projected on the top face of the reservoir	44
3.7	The mean value of NPV (in US dollar) and its corresponding standard deviation for well placement optimization using CMA-ES with meta-models and CMA-ES. Ten runs are performed for each algorithm. Constraints are handled using an adaptive penalization with rejection technique	46
3.8	The mean number of reservoir simulations needed to reach a given NPV value using CMA-ES with meta-models and CMA-ES. Ten runs are performed for each algorithm	47

3.9	The SoPhiH map, with solution well configuration obtained using CMA-ES with meta-models and two engineer’s proposed well configurations	48
3.10	Production curves for an optimized solution using CMA-ES with meta-models and two engineer’s proposed configurations	49
3.11	The mean value of NPV (in US dollar) and its corresponding standard deviation for well placement optimization using CMA-ES with meta-models of one multi-segment well. Ten runs are performed	51
3.12	The positions of solution multi-segment producers found by 10 runs of CMA-ES with meta-models. A zoom on the region containing the solution wells is performed	51
3.13	The positions of solution multi-segment producers found by 10 runs of CMA-ES with meta-models projected on the top face of the reservoir. A zoom on the region containing the solution wells is performed	52
4.1	Average number of evaluations of the p-sep lmm-CMA on f_{Rosen}^{100} to reach f_{stop} for varying population sizes $\lambda = \gamma \times \lambda_{\text{default}}$, and average number of evaluations per generation of the p-sep lmm-CMA on f_{Rosen}^{100} for varying population sizes $\lambda = \gamma \times \lambda_{\text{default}}$	58
4.2	Success performance SP1 over the dimension of the problem on $f_{\text{Rosen}}^{\alpha}$, with $\alpha = 1, 10^2$ and 10^4 for dimensions in between 4 and 40. The dimension of the sub-functions n_M equals 2	61
4.3	Average speedup with respect to CMA-ES to reach f_{stop} with a varying number of points used to build the meta-model $k_i = \beta \times k_{i,\text{min}}$ where $k_{i,\text{min}} = \frac{n_i(n_i+3)}{2} + 1$. Each point corresponds to 20 runs performed	63
5.1	The SoPhiH map with the location of the injector already drilled	69
5.2	The mean value of NPV (in US dollars) for well placement optimization using CMA-ES with partially separated meta-models denoted p-sep lmm-CMA, CMA-ES with meta-models denoted lmm-CMA and CMA-ES. Ten runs are performed for each method	70
5.3	The SoPhiH map with the location of the injector already drilled, and the solution producers	72

5.4	The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with partially separated meta-model, i.e., p-sep lmmCMA. The three figures depict one of the ten performed runs of p-sep lmm-CMA. In (a), the evolution of the best overall NPV value and the best NPV obtained each generation is depicted. In (b), the evolution of the well trajectory parameters, where each well is plotted using a different color representing three group of parameters is depicted. The group of angles encoding each well is shown in the lower part of the figure (values below 10). The group of well lengths is shown in the intermediate part of the figure (the three curves with values around 500). The group of Cartesian coordinates of the wells is shown in the upper part of the figure. In (c) the evolution of the well trajectory parameters on the log-scale is depicted . . .	74
6.1	The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the mean of samples” approach. The best mean value of the NPV over the 20 possible realizations, i.e., NPV^R is shown. Three runs are performed	83
6.2	The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the neighborhood” approach, for three independent runs in (a), (b) and (c). The evolutions of the best estimated objective function, i.e., NPV^E are drawn with green lines. The evaluations on the true objective function over the 20 possible realizations, i.e., NPV^R are depicted with red crosses	84
6.3	The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the neighborhood” approach for eight independent runs. (a) shows the evolution of the evaluations on NPV^R . (b) shows the evolution of the best found evaluation on NPV^R . The maximum distance of selection for the neighborhood d_{max} is equal to 4000	84
6.4	The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the mean of samples” approach and the “using the neighborhood” approach. The evolution of the best found evaluation on NPV^R for the “using the neighborhood” approach is drawn with red lines. The evolution of the best found evaluation on NPV^R for the “using the mean of samples” approach is drawn with blue lines. Three independent runs are performed for each approach. For the “using the neighborhood” approach, the maximum distance of selection for the neighborhood d_{max} is equal to 4000	86

6.5	The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the neighborhood” approach for four independent runs. (a) shows the evolution of the evaluations on NPV^R . (b) shows the evolution of the best found evaluation on NPV^R . The maximum distance of selection for the neighborhood d_{\max} is equal to 3000	86
6.6	The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the neighborhood” approach for four independent runs. (a) shows the evolution of the evaluations on NPV^R . (b) shows the evolution of the best found evaluation on NPV^R . The maximum distance of selection for the neighborhood d_{\max} is equal to 6000	87
6.7	The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using one realization” approach, for three independent runs in (a), (b) and (c). The evolutions of the best estimated objective function (equal to an evaluation on a randomly chosen realization) are drawn with green lines. The evaluations on the true objective function over the 20 possible realizations, i.e., NPV^R are depicted with blue crosses	88
6.8	The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using one realization” approach. The best mean value of the NPV over the 20 possible realizations, i.e., NPV^R is shown. Three runs are performed	89
.1	Un exemple de paramétrage d’un puits multilatéral ayant deux segments et une branche	96
.2	Valeur moyenne du NPV (en dollars) et l’écart type correspondant pour un problème de placement des puits utilisant CMA-ES et AG. Quatorze tests sont effectués pour chaque algorithme	99
.3	Valeur moyenne du NPV (en dollars) et l’écart type correspondant pour un problème de placement des puits utilisant CMA-ES avec des méta-modèles et CMA-ES. Dix tests sont effectués pour chaque algorithme	100
.4	Valeur moyenne du NPV (en dollars) pour un problème de placement des puits utilisant CMA-ES avec des méta-modèles partiellement séparés, i.e., p-sep lmm-CMA, CMA-ES avec des méta-modèles, i.e., lmm-CMA et CMA-ES. Dix tests sont effectués pour chaque algorithme	101
.5	L’évolution du meilleur NPV (en dollars) obtenu pour un problème de placement des puits utilisant CMA-ES avec une approche utilisant la moyenne des échantillons, et avec l’approche proposée utilisant le voisinage. Trois tests sont démontrés pour chaque algorithme	102

Chapter 1

Introduction

“Drill for oil? You mean drill into the ground to try and find oil? You’re crazy.” – this was what drillers who Edwin L. Drake¹ tried to enlist to his project to drill for oil in 1859, said.

“If you can draw it (the well), I can drill it !” – this becomes the modern refrain of a driller.

1.1 Problem statement

The state of the art in reservoir management has been recently greatly influenced by technologies. Nowadays, drilling technologies have made great strides with the advances achieved in directional drilling capabilities. Hence, reservoir engineers can take advantage from the use of different well architectures such as vertical, horizontal and more complex configurations to enhance reservoir productivity, especially given the present price of oil which is although continuing to fluctuate in recent years, still above the US\$40/barrel (Fig. 1.1).

Environments, work areas and conditions in which oil and gas fields are now being discovered are much more complex and challenging. The existing fields are becoming more depleted and, therefore, are more marginal. Unless there is a way to optimize their productivity and to take corrective actions, it would be hard to justify to continue

¹Edwin L. Drake (1819 - 1880) was an American oil driller, popularly credited with being the first to drill for oil in the United States.

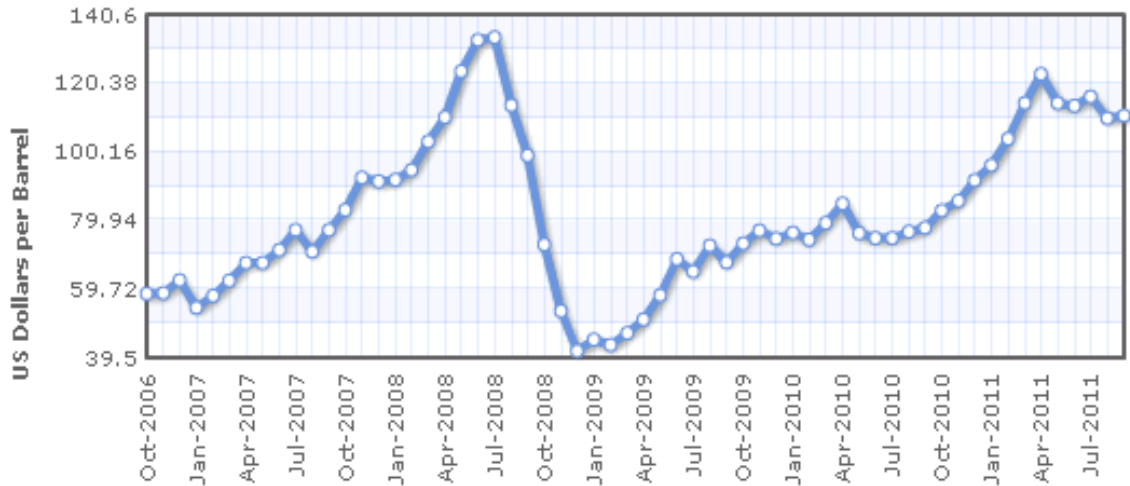


Figure 1.1: Brent crude oil price (in US dollar), Oct 2007 - Sep 2011. Reprinted from Index Mundi website, November 9, 2011.

[<http://www.indexmundi.com/commodities/?commodity=crude-oil-brent&months=60>]

investing to produce these existing fields for economic reasons [14]. On the other hand, new discoveries also need an optimal production scheme to be economically viable.

One of the most important issues that must be addressed to maximize a given project's asset value is to optimally decide where to drill wells. A well placement decision affects the hydrocarbon recovery and thus the asset value of a project. In general, such a decision is difficult to make since an optimal placement depends on a large number of parameters such as reservoir heterogeneities, faults and fluids in place. Moreover, dealing with complex well configurations, e.g., non-conventional wells, implies additional challenges such as the concentration of investment and the well intervention difficulty¹.

The current approach, mostly used in the industry, is based on the so-called professional judgment made by reservoir engineers –requiring the understanding of the impact of different influencing engineering and geological parameters– and confirmed by a number of reservoir simulation trials. However, the reservoir performance is influenced by non-linearly correlated parameters, which may also evolve with time. Hence, the professional judgment approach, in general, fails to predict the best well configurations.

Recently, many efforts were made to formulate the well placement decision as an optimization problem: the objective function optimized, which is evaluated using a reservoir simulator, evaluates the economics of the project; the parameters thought encode the position of the different wells (that include locations and trajectories). We define the location of a given well as the position of the starting point of the well, and we define the trajectory of a given well as the positions of the mainbore and the laterals (if any). If the number of wells to be placed and their type (injector or producer) is fixed, the parameters encoding

¹Drilling a well costs in general from US\$1 million to US\$30 million.

the well positions are real numbers and the objective function f maps a subset of \mathbb{R}^n where n , the number of parameters, equals the sum of the number of parameters needed to encode each well position that need to be placed. Formally we want to find a vector of parameter $\mathbf{p}_{\max} \in \mathbb{R}^n$ such that:

$$f(\mathbf{p}_{\max}) = \max_{\mathbf{p}} \{f(\mathbf{p})\} \quad , \quad (1.1)$$

where \mathbf{p} denotes the vector of parameters to be optimized encoding the positions and trajectories of the well configuration. The vector \mathbf{p}_{\max} must be found using a technically feasible number of reservoir simulations.

The well placement optimization problem is challenging as:

- The objective function, e.g., the net present value (NPV) is difficult to optimize. In particular, it is *multi-modal*, i.e., with multiple local optima, *non-convex* and *non-smooth*. An illustration can be found in [103] where the NPV of a single vertical well placement is sampled to construct the objective function surface. The surface is shown to be highly non-smooth and to contain several local optima. In this illustration, the problem dimension equals two and it has thus been possible to sample all the points from a fine grid spanning regularly the search space. However, this becomes impossible for problem dimensions larger than 3 as the number of points, to keep a fine discretization, would need to grow exponentially in the search space dimension (this is referred as *curse of dimensionality*) rendering the search task difficult.
- The problem is *costly*: a single function evaluation requires one reservoir simulation which is often very demanding in CPU time (several minutes to several hours). The affordable number of reservoir simulations is often then restricted.
- The problem involves in general optimizing under *geological uncertainty*: the problem assumes that we have already defined a (or a number of) realistic geological model(s). Each model is obtained using history matching which consists in the adjustment of the reservoir model until it closely reproduces the past behavior of the reservoir (historical production and pressures). However, history matching problem is a mathematically ill-posed with non-unique solutions, i.e., several possible (generally equiprobable) geological models. Thus, taking into account several geological models introduces the problem of handling geological uncertainty which adds an other challenge to the optimization of the objective function, in particular it leads to a large increase of the number of performed reservoir simulations. In the context of geological uncertainty which will be addressed in Chapter 6, we will denote by

f the objective function to optimize, and let us consider a number \mathcal{N}_r of geological realizations denoted by $(R_i)_{i=1,\dots,\mathcal{N}_r}$. We denote by $f(\mathbf{p}, R_i)$ the objective function value on the well configuration \mathbf{p} on the realization R_i . Thus, we want to find a vector of parameter $\mathbf{p}_{\max,R} \in \mathbb{R}^n$ such that:

$$f^R(\mathbf{p}_{\max,R}) = \max_{\mathbf{p}} \{f^R(\mathbf{p})\} \quad , \quad (1.2)$$

where f^R is in general an averaged sum of the objective function evaluations on the well configuration \mathbf{p} over all the realizations:

$$f^R(\mathbf{p}) = \frac{1}{\mathcal{N}_r} \sum_{i=1}^{\mathcal{N}_r} f(\mathbf{p}, R_i) \quad . \quad (1.3)$$

Furthermore, constraints are imposed to guarantee the physical feasibility of the solution wells, and thus to avoid very long wells or wells that violate common engineering practices (e.g., wells outside the reservoir). Therefore, a constraint optimization problem needs to be handled. Formally, when dealing with constraints, we want to find a vector of parameter $\mathbf{p}_{\max} \in \mathbb{R}^n$ such that:

$$\begin{cases} f(\mathbf{p}_{\max}) = \max f(\mathbf{p}) \\ \text{s.t. } h_i(\mathbf{p}) \leq d_i \quad \forall i = 1, \dots, m \end{cases} \quad , \quad (1.4)$$

where m is the number of constraints, d_i are real numbers and $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are the constraint functions that need to be satisfied.

The main objective of this thesis is to propose a procedure for solving the well placement optimization problem, in particular the well locations and trajectories optimization problem. The proposed procedure must offer the maximum asset value using a technically feasible number of reservoir simulations. This implies to address the challenges explained above namely:

- (I) The non-smoothness, the multi-modality, the non-convexity and the high dimensionality of the objective function;
- (II) The expensive cost of the objective function;
- (III) The geological uncertainty handling problem.

In this thesis, we will consider the well placement optimization problem as a *black-box* optimization (also known as derivative-free optimization) problem. The black-box optimization means that only the inputs and outputs of the objective function are observed, and not its internal operations and processes. The black-box context is natural in our

context since an objective function evaluation involves a reservoir simulation which corresponds in general to a commercial software, in which the internal structure and code are often unavailable.

We now review the critical points of current knowledge and methodological approaches related to the well placement optimization.

1.2 Literature review

Many optimization algorithms exist to address the continuous optimization problem formulated in Eq. (1.1). In this section, we give a survey of the existing continuous optimization algorithms. Only some of these algorithms will be detailed depending on their importance for this thesis. Other algorithms will be briefly mentioned with their corresponding references for more details. Then, a survey of studies describing existing approaches used for the well placement optimization problem will be given. A detailed literature review for well placement optimization under geological uncertainty formulated in Eq. (1.2) will be provided in Chapter 6.

1.2.1 Optimization algorithms

Optimization algorithms for non-linear continuous optimization can be divided depending on the method they use to explore the search space. In the following, we enumerate a number of selected representative algorithms divided into four categories: deterministic algorithms, stochastic algorithms, search algorithms using surrogates and hybrid algorithms.

1.2.1.1 Deterministic methods

Deterministic algorithms include descent methods which use the explicit value of the gradient or higher order derivatives of the objective function. If this information is not available, i.e., in case of black-box optimization, it can be approximated. Other deterministic optimization techniques include trust region methods (e.g., [107]), direct pattern search methods [80] and simplex methods [101]. A major drawback of deterministic optimization methods is that they can easily get stuck in a local optimum.

- **Descent methods:** Descent methods are defined as iterative methods that need the gradient of the objective function to search for a minimum of a given objective function f . After fixing an initial point \mathbf{x}_k at iteration k , a new point is calculated as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{1.5}$$

where \mathbf{p}_k is the search direction at iteration k and α_k denotes the step width. The optimization process continues until reaching the convergence criterion. The search direction can be calculated using a linear approximation (first order) of the target function, i.e., $\mathbf{p}_k = -\nabla f(\mathbf{x}_k)$. In this case, the method is called the steepest descent method. A second order approach uses a quadratic approximation and leads to methods referred to as Newton methods. Quasi-Newton methods are based on Newton methods, but without computing the Hessian matrix. In this case, the search direction $\mathbf{p}_k = -\tilde{H}_k^{-1}\nabla f(\mathbf{x}_k)$, where \tilde{H}_k is an approximation of the Hessian matrix in the current solution. The most popular quasi-Newton method is the Broyden-Fletcher-Goldfarb-Shanno algorithm (BFGS) [31, 53, 61, 118].

If no explicit formula of the objective function is available, derivatives are in general approximated using methods such as finite difference methods. An other way to compute the gradients is by using adjoint methods. In contrast to finite difference methods, where the number of objective function evaluations required to estimate the gradients grows linearly with the number of the parameters of the problem, adjoint methods provide the gradients in a fraction of the computational time of objective function evaluation. However, implementing adjoint methods requires a deep understanding of the so-called simulation code (corresponding to the objective function evaluation) which is not usually trivial for real-world problems. It also requires having access to the simulation code, which is not usually available for real-world problems. Adjoint methods are widely used in aerodynamics [81]. In the oil and gas industry, it is still difficult to apply adjoint method approaches, although some research has already been performed in particular in the reservoir simulation community [89].

- **Trust region methods:** Trust region methods, called also quadratic approximation methods rely on an approximation of the objective function f with a quadratic function which is supposed to be a reasonable approximation of f in a neighborhood of the the current estimate. This neighborhood is called the *trust region*. A state-of-the-art trust region method is the NEW Unconstrained Optimization Algorithm (NEWUOA) [107] which is a derivative-free optimization method. At each iteration, NEWUOA creates a quadratic model that interpolates the objective function f at m points (usually $m = 2n + 1$, where n is the number of parameters to be optimized). The quadratic model is then updated by minimizing it inside the trust region. A more detailed presentation of trust region methods can be found in [91].

1.2.1.2 Stochastic methods

Stochastic methods have been employed to mitigate the defect of deterministic methods for difficult functions to solve (e.g., non-smooth and multi-modal). In particular, stochastic optimization algorithms aim at being more robust when dealing with multi-modal objective functions. These methods include methods such as simulated annealing (SA) [88, 124], particle swarm optimization (PSO) [86], simultaneous perturbation stochastic algorithm (SPSA) [119] and evolutionary algorithms (EA). EAs which have received an increasing interest has mainly three origins: genetic algorithms (GA) [78, 79], evolutionary programming (EP) [56, 56] and evolution strategies (ES) [108, 117].

- **Evolutionary algorithms (EA):** An overview of evolutionary algorithms is presented in [15]. EAs are stochastic optimization algorithms inspired by biological evolution. Starting with an initial population of points called individuals and at each iteration, candidate solutions evolve by *selection*, *mutation* and *recombination* until reaching the stopping criteria with a satisfactory solution. This process is used by the three origins of EAs, i.e., GA, EP and ES. Only two of them will be detailed in this section: genetic algorithms and evolution strategies.

Genetic algorithms (GA) [78, 79] are stochastic search algorithms designed initially to deal with binary encoded individuals. For continuous optimization, problem variables can either be mapped to binary strings or other encoding can be adopted such as real encoding. However, representing real vectors as bit strings leads to poor performance [122].

Evolution strategies (ES) [108, 117]: besides the common principles shared with other EAs, i.e., mutation, recombination and selection, during the optimization process, ESs sample new individuals according to a multivariate normal distribution, and use a self-learning mechanism to adapt its parameters called adaptive search. The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [74] is the state-of-the-art Evolution Strategy where the multivariate normal distribution has a mean and a covariance matrix continually updated during the optimization process. Intensive benchmarking of several derivative-free algorithms have established that CMA-ES is one of the most efficient method for dealing with difficult numerical optimization problems [70]. CMA-ES has also been applied to real-world problems [18, 42, 92, 94]. More details about CMA-ES are provided in Chapter 2.

- **Simulated annealing (SA)** [88, 124]: The name and the inspiration of simulated annealing comes from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The algorithm avoids getting trapped in local optima by allowing moves

that may lead to a deterioration in the objective function values. The SA algorithm is outlined as follows. Given a candidate solution \mathbf{s} , a neighbor random solution \mathbf{s}' is accepted¹ if (1) \mathbf{s}' is better than \mathbf{s} with respect to the objective function or (2) with a probability that depends on the change of the corresponding objective function values and a control parameter T , called the temperature. Otherwise, if none of the above conditions are met, the current solution remains unchanged. The parameter T is gradually decreased to zero in the course of the optimization according to a deterministic “cooling schedule”. The performance of the simulated annealing algorithm is very sensitive to the choice of the cooling schedule.

- **Particle swarm optimization (PSO)** [86]: PSO is an iterative population based algorithm, inspired from movement of swarms of birds or insects searching for food or protection. Each particle movement is influenced by its own experience (its best found locality) and by the experience of the others (the best found locality of all the particles). Based on these best found localities, the localities of the members of the swarm and their velocities are adjusted. The performance of PSO are not invariant with respect to rotations of the coordinate system, i.e., the performance of PSO on non-separable, ill-conditioned functions declines dramatically with increasing condition numbers [75].
- **Simultaneous perturbation stochastic algorithm (SPSA)** [119]: SPSA is a stochastic gradient approximation method, in which at each iteration the parameters are randomly perturbed, and the objective function is evaluated at the perturbed points to estimate the gradient.

1.2.1.3 Search algorithms using surrogates

Search algorithms using surrogates, called proxy-modeling or meta-modeling in the literature, are based on approximating the objective function by a an approximate model (called also surrogate, proxy-model or meta-model). In the context of costly objective functions, a surrogate can be considered as a computationally cheaper replacement of the objective function. Thus, during the optimization process the surrogate is constructed and the objective function evaluations are replaced by evaluations on the surrogate [20, 83]. Search algorithms using surrogates needs to consider the so-called exploration-exploitation trade-off [58], i.e., evaluating more (respectively, less) candidate solutions using the “true” objective function implies a better (respectively, worst) quality of the surrogate but on the other hand a higher (respectively, reduced) computational cost of the optimization.

¹If a candidate solution is accepted, it replaces the current solution

The most popular surrogate models include polynomial response surfaces, Kriging [90, 48], support vector machines [40] and artificial neural networks [110].

1.2.1.4 Hybrid methods

Several algorithms (two or more) from different classes can be combined in order to form the so-called hybrid methods. Hybridization aims at having a resulting algorithm which contains the positive features of the combined algorithms. Several hybridizations have been proposed in the literature in order to tackle specific applications. For instance, a review of hybridization of genetic algorithms can be found in [46]. Also, a review of hybridization of the particle swarm optimization can be found in [123].

1.2.2 Well placement optimization

Well placement optimization is a recent area of research that is gaining growing interest. Different methodologies have been used in the literature to tackle the well placement problem.

On the one hand, approaches based on stochastic search algorithms were used, where minimal assumptions on the problem are needed and that are thus more robust than deterministic methods when dealing with rugged problems such as the well placement problem. Simulated annealing (SA) was used in [19] for well placement and scheduling, and in [85] for well placement. Particle swarm optimization (PSO) was applied in [103] for the determination of optima well type and position. Genetic algorithm (GA) was applied in [98, 47, 99, 33]. Simultaneous perturbation stochastic algorithm (SPSA) was used in [16, 17]. In particular, in [17], a comparison between three optimization algorithms is performed: the SPSA algorithm, the very fast simulated annealing (VFSA) and the finite difference gradient (FDG).

On the other hand, deterministic optimization methods were also used. Descent algorithms were mostly used, in which adjoint methods were used for computing the gradients [67, 114, 57, 125, 130]. Using descent methods implies that the underlying model of the function needs to be smooth enough. In [67], the adjoint method is used to place an injector in a 2D oil-water reservoir with 4 producers already fixed in each of the four corners grid blocks. Results show that the algorithm, as expected due to its deterministic aspect, converges to a different local optimum for every initial well location. In [114], the wells are defined by continuous variables and the adjoint method is tested on a few synthetic waterflood optimization problems.

Search algorithms using surrogates, or proxy-modeling were also used in the literature. In proxy-modeling the true objective function is replaced by a proxy-model, and different optimization techniques are applied to the proxy. Proxy-models include least squares and

kriging [105], radial basis functions [50], quality maps [41, 100], and multiple regression techniques (including kriging) [1]. Although proxy-modeling is an efficient way to have an approach with a reduced number of reservoir simulations, its application, with increasing complexity of the solution space, is not recommended [132].

Stochastic algorithms have been combined with search algorithms using surrogates and deterministic approaches to form hybrid algorithms: GA with a polytope algorithm and kriging [63, 64], GA with a polytope algorithm, kriging and neural networks [65], GA with neural networks, a hill climber and a near-well upscaling technique [129]. Results show that a hybrid stochastic algorithm converges in general to a reasonable solution with a reduced number of evaluations compared to a pure stochastic algorithm. The approaches in [63, 64, 65, 129] build at each iteration a proxy-model, determine its maximum and include the location of this maximum in the population (replacing the worst individual) if it is better than the best individual of the current population. In [10], a GA is defined, in which at each iteration, only a predefined percentage of the individuals, chosen according to a set of scenario attributes, is simulated. The objective function of the non-simulated points is estimated using a statistical proxy based on cluster analysis.

1.3 Thesis objectives and methodology

In this thesis the objective is to address the previously mentioned challenges (I), (II) and (III) in Section 1.1, namely:

- (I) The non-smoothness, the multi-modality, the non-convexity and the high dimensionality of the objective function;
- (II) The expensive cost of the objective function;
- (III) The geological uncertainty handling problem.

Considering the state of the art in optimization, the choice of the CMA-ES algorithm [74] seems a priori natural to address problem (I). Indeed, CMA-ES is recognized as one of the most powerful derivative-free optimizers for continuous optimization [70]. CMA-ES is both a fast and robust local search algorithm, exhibiting linear convergence on wide classes of functions and a global search algorithm when playing with restart and increase of population size. CMA-ES, in contrast to most other evolutionary algorithms, is a quasi parameter-free algorithm¹.

In the petroleum industry, CMA-ES have been applied only in two studies, to the best of our knowledge, previous to this work: a characterization of fracture conductivities

¹Only the population size is suggested to be adjusted by the user in order to account for the ruggedness of the objective function landscape.

from well tests inversion [32], a well placement optimization but with respect to simple attributes (e.g., productivity indexes) [43]. A more recent application on the well placement optimization was shown in [116, 115].

To tackle problem (II), we propose to investigate coupling the CMA-ES optimizer with surrogates (or meta-models). In this context, we aim at defining an efficient variant of CMA-ES coupled with meta-models able to reduce significantly the number of the reservoir simulations. Furthermore, we aim at exploiting the knowledge about the optimization problem, in particular the so-called partial separability of the objective function in order to reduce more the number of reservoir simulations.

Finally, to tackle problem (III), we aim at defining an approach (for CMA-ES) able to capture the geological uncertainty with a significantly reduced cost of reservoir simulations. In this context, we aim at defining an approach that performs a small number of reservoir simulations (typically one) for each well configuration instead of performing reservoir simulations on all possible geological realizations.

1.4 Summary of contributions

The following presents a summary of the contributions of this thesis.

We have tackled the problem (I) related to the non-smoothness, the multi-modality, the non-convexity and the high dimensionality of the objective function in the well placement problem, and we have shown:

A first successful application of CMA-ES on the well placement problem. (Results published in [26, 24]) We propose a new methodology for well location and trajectory optimization based on the population based stochastic search algorithm called Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [74]. We propose to use a new adaptive penalization with rejection technique to handle constraints. Because genetic algorithms are quite often the method of choice in petroleum industry, we show the improvement of applying CMA-ES over a GA on the synthetic benchmark reservoir case PUNQ-S3 [54]. To allow a fair comparison, both algorithms are used without parameter tuning on the problem, standard settings are used for the GA and default settings for CMA-ES. It is shown that our new approach outperforms the genetic algorithm: it leads in general to both a higher net present value and a significant reduction in the number of reservoir simulations needed to reach a good well configuration.

After this application of CMA-ES on the well placement problem, we have tackled the problem (II) related to the expensive cost of the objective function, and we have proposed two new algorithms:

A new variant of CMA-ES with local meta-models. (Results published in [22])

The local-meta-model CMA-ES (lmm-CMA) [87] coupling local quadratic meta-models with the Covariance Matrix Adaptation Evolution Strategy is investigated. The scaling of the algorithm with respect to the population size is analyzed and limitations of the approach for population sizes larger than the default one are shown. A new variant for deciding when the meta-model is accepted is proposed –called the new-local-meta-model CMA-ES (nlmm-CMA).

A new variant of CMA-ES with local meta-models for partially separable functions. (Results published in [23])

We propose a new variant of the covariance matrix adaptation evolution strategy with local meta-models for optimizing partially separable functions –called the partially separable local-meta-model CMA-ES (p-sep lmm-CMA). We propose to exploit partial separability by building at each iteration a meta-model for each element function (or *sub-function*) using a full quadratic local model. Our results demonstrate that exploiting partial separability leads to an important speedup compared to the standard CMA-ES. We show on the tested functions that the speedup increases with increasing dimensions for a fixed dimension of the element function. On the standard Rosenbrock function the maximum speedup of λ is reached in dimension 40 using element functions of dimension 2, where λ is the population size. We show also that higher speedups can be achieved by increasing the population size.

Now, we have applied the two new proposed algorithms on the well placement problem to achieve:

A significant reduction of the number of reservoir simulations for the well placement problem. (Results published in [26, 24, 25])

We propose to apply CMA-ES with local meta-models (nlmm-CMA) on the well placement problem, where for each well configuration in the population, an approximate convex quadratic model is built using true objective function evaluations collected during the optimization process. Coupling CMA-ES with a meta-model leads to a significant improvement, which was around 20% for the synthetic benchmark reservoir case PUNQ-S3.

Moreover, we propose also to apply p-sep lmm-CMA on the well placement problem, by building partially separated meta-models for each well or set of wells, which results in a more accurate modeling. Results show that taking advantage of the partial separability of

the objective function leads to a significant decrease in the number of reservoir simulations needed to find the “optimal” well configuration, given a restricted budget of reservoir simulations.

We have also tackled the problem (III) related to the geological uncertainty handling, and we have proposed:

A new approach to handle geological uncertainty for the well placement problem. We propose a new approach to handle geological uncertainty for the well placement problem with a reduced number of reservoir simulations. We propose to use only one realization together with the neighborhood of each well configuration in order to estimate its objective function instead of using multiple realizations. The approach is applied on the synthetic benchmark reservoir case PUNQ-S3 and shown to be able to capture the geological uncertainty using a reduced number of reservoir simulations.

1.5 Dissertation road-map

This thesis is structured as follows. Chapter 2 gives a “theoretical” overview of the optimization method used in this thesis: the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). An adaptive penalization technique to handle the optimization constraints is also introduced and a combination of CMA-ES with meta-models is investigated to propose a new variant of CMA-ES with local-meta-models, called the new-local-meta-model CMA-ES (nlmm-CMA).

In Chapter 3, the CMA-ES optimizer is applied on the well placement problem. The improvement of applying CMA-ES over a GA on a synthetic benchmark reservoir case is shown. In addition, the contribution of the CMA-ES with meta-models in reducing the number of reservoir simulations is demonstrated on a number of examples.

In Chapter 4, we propose a new variant of CMA-ES with local meta-models for optimizing partially separable functions, called the partially separable local-meta-model CMA-ES (p-sep lmm-CMA).

In Chapter 5, the resulting approach (p-sep lmm-CMA) is applied on the well placement problem.

Finally, in Chapter 6, the problem of dealing with uncertainty in well placement is tackled. A new approach using the neighborhood of each well configuration is proposed and demonstrated on a synthetic benchmark reservoir case.

The thesis closes with the conclusions and a number of suggestions for future work.

Chapter 2

CMA-ES and CMA-ES with meta-models

This chapter is based on the paper [22]. It gives a detailed overview of the optimization methods applied in Chapter 3 to the well placement problem. We present the CMA-ES algorithm, a constraint handling needed for well placement and a new surrogate approach that couples CMA-ES with meta-models. This latter approach mitigate some defects of the local-meta-model CMA-ES (Imm-CMA). The different defined methodologies are tested and validated on some mathematical test functions.

This chapter is structured as follows. Section 2.1 gives an overview of the optimization algorithm CMA-ES. In Section 2.2, we propose an adaptive penalization and rejection technique in order to handle optimization constraints. Finally in Section 2.3, the reduction of the number of evaluations is addressed by coupling CMA-ES with meta-models.

In the following, we denote the objective function to be optimized by $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

2.1 Covariance Matrix Adaptation - Evolution Strategy

The Covariance Matrix Adaptation - Evolution Strategy (CMA-ES) [74, 71] is an iterative stochastic optimization algorithm where at each iteration, a population of candidate solutions is sampled. In contrast to the classical presentation of population based stochastic search algorithms (like genetic algorithms [78, 79]) where the different steps of the algorithms are described in terms of operators acting on the population (crossover, mutation), the natural algorithm template for CMA-ES translates the evolution of the probability distribution used to sample points at each iteration. Indeed, the algorithm loops over the following steps:

1. sample a population of λ candidate solutions (points of \mathbb{R}^n)
2. evaluate the λ candidate solutions on f

3. adapt the sampling distribution (using the feedback from f obtained at step 2.)

We see that this general template depends on a probability distribution (sampling distribution) and on the update of this probability distribution. The sampling distribution in CMA-ES is a *multivariate normal distribution*. In the next paragraphs we will give more insights on multivariate normal distributions and their geometrical interpretation and then explain how its update is performed at each iteration within CMA-ES.

Multivariate normal distributions A random vector of \mathbb{R}^n distributed according to a multivariate normal distribution is usually denoted by $\mathcal{N}(\mathbf{m}, \mathbf{C})$ where \mathbf{m} is a vector of \mathbb{R}^n and \mathbf{C} an $n \times n$ symmetric positive definite matrix corresponding to the covariance matrix of the random vector. The set of parameters (\mathbf{m}, \mathbf{C}) entirely determines the random vector. Fig. 2.1 gives the geometric interpretation of a random vector $\mathcal{N}(\mathbf{m}, \mathbf{C})$ in two dimensions. We visualize that \mathbf{m} is the symmetry center of the distribution and that isodensity lines are ellipsoid centered in \mathbf{m} with main axes corresponding to eigenvectors of \mathbf{C} and lengths determined by the square roots of the eigenvalues of \mathbf{C} . Fig. 2.1 depicts also points sampled according to a multivariate normal distribution. As expected, the spread of the points follows the isodensity lines. A useful relation is $\mathbf{m} + \mathcal{N}(\mathbf{0}, \mathbf{C}) = \mathcal{N}(\mathbf{m}, \mathbf{C})$ that interprets \mathbf{m} as the displacement from the origin $\mathbf{0}$.

In CMA-ES, the mean vector represents the favorite solution or best estimate of the optimum, and the covariance matrix \mathbf{C} characterizing the geometric shape of the distribution defines where new solutions are sampled. Furthermore, an additional parameter is added, which is the step-size σ used as a global scaling factor for the covariance matrix. Overall, in step 1. for CMA-ES, points are sampled according to:

$$\mathbf{m} + \sigma \mathcal{N}(\mathbf{0}, \mathbf{C}) . \tag{2.1}$$

The adaptation of \mathbf{m} targets to find the best estimate of the optimum, the adaptation of \mathbf{C} aims at learning the right coordinate system of the problem (rotation and scaling of the main axes) and the adaptation of σ aims at achieving fast convergence to an optimum and preventing premature convergence. We will now describe how the distribution is updated, that is how the parameters \mathbf{m} , σ and \mathbf{C} are updated in step 3. of the template.

Update of mean vector, covariance matrix and step-size We adopt here some time-dependent notations. The iteration index is denoted g . Let $(\mathbf{m}^{(g)}, g \in \mathbb{N})$ be the sequence of mean vectors of the multivariate normal distribution generated by CMA-ES and let $(\sigma^{(g)}, g \in \mathbb{N})$ and $(\mathbf{C}^{(g)}, g \in \mathbb{N})$ be respectively the sequences of step-sizes and

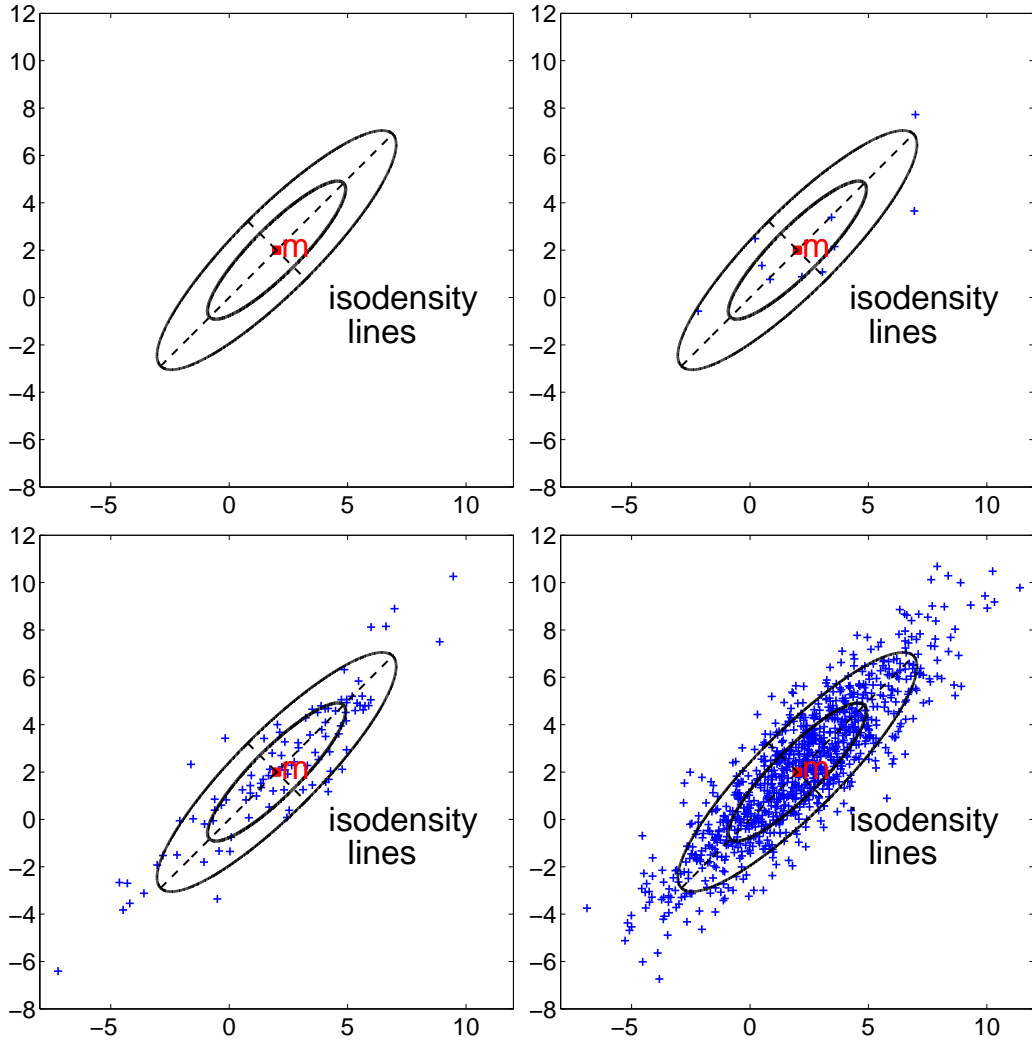


Figure 2.1: Geometrical representation of a 2-dimensional multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$ where $\mathbf{m} = (2, 2)^T$ and the covariance matrix \mathbf{C} admits $\frac{1}{\sqrt{2}}(1, 1)$ and $\frac{1}{\sqrt{2}}(-1, 1)$ as normalized eigenvectors with respective eigenvalues 16 and 1. Depicted on each plot is the mean vector \mathbf{m} and the ellipsoid isodensity lines defined as $(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1} (\mathbf{x} - \mathbf{m}) = c$ where the constant c equals 1 (inner line) and 3 (outer line). The main axes of the (isodensity) ellipsoid are carried by eigenvectors of \mathbf{C} . The half lengths of the axis of the unit isodensity lines ($(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1} (\mathbf{x} - \mathbf{m}) = 1$) are the square roots of the eigenvalues of \mathbf{C} . Depicted on the 2nd, 3rd and 4th plots are samples among 10 (resp. 100 and 1000) samples from $\mathcal{N}(\mathbf{m}, \mathbf{C})$ falling into the box plot $[-8, 12] \times [-8, 12]$.

2.1 Covariance Matrix Adaptation - Evolution Strategy

covariance matrices. Assume that $\mathbf{m}^{(g)}$, $\sigma^{(g)}$, $\mathbf{C}^{(g)}$ are given, the λ new points or *individuals* are sampled in step 1. according to:

$$\mathbf{x}_i^{(g)} = \mathbf{m}^{(g)} + \sigma^{(g)} \underbrace{\mathcal{N}_i(0, \mathbf{C}^{(g)})}_{=\mathbf{y}_i}, \quad \text{for } i = 1, \dots, \lambda. \quad (2.2)$$

Those λ individuals are evaluated in step 2. and ranked according to f :

$$f(\mathbf{x}_{1:\lambda}^{(g)}) \leq \dots \leq f(\mathbf{x}_{\mu:\lambda}^{(g)}) \leq \dots \leq f(\mathbf{x}_{\lambda:\lambda}^{(g)}), \quad (2.3)$$

where we use the notation $\mathbf{x}_{i:\lambda}^{(g)}$ for i^{th} best individual.

The mean $\mathbf{m}^{(g)}$ is then updated by taking the weighted mean of the best μ individuals:

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} \omega_i \mathbf{x}_{i:\lambda}^{(g)} = \mathbf{m}^{(g)} + \sigma^{(g)} \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda}, \quad (2.4)$$

where $\mathbf{y}_{i:\lambda} = (\mathbf{x}_{i:\lambda}^{(g)} - \mathbf{m}^{(g)})/\sigma^{(g)}$. In general $\mu = \frac{\lambda}{2}$ and $(\omega_i)_{1 \leq i \leq \mu}$ are strictly positive and normalized weights, i.e., satisfying $\sum_{i=1}^{\mu} \omega_i = 1$. This update displaces the mean vector toward the best solutions. The increment $\sigma^{(g)} \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda}$ has an interpretation in terms of (stochastic) approximation of the gradient with respect to \mathbf{m} of a joint criterion \mathcal{J} mapping $(\mathbf{m}, \sigma, \mathbf{C})$ to \mathbb{R} and depending on quantiles of the objective function f [9].

A measure characterizing the recombination used is called *the variance effective selection mass* and defined by $\mu_{\text{eff}} = \left(\sum_{i=1}^{\mu} \omega_i^2 \right)^{-1}$. The choice of the recombination type has an important impact on the efficiency of the algorithm [6]. The default weights are equal to:

$$\omega_i = \frac{\ln(\mu + 1) - \ln(i)}{\mu \ln(\mu + 1) - \ln(\mu!)}, \quad \text{for } i = 1, \dots, \mu. \quad (2.5)$$

The update of the covariance matrix $\mathbf{C}^{(g)}$ uses two mechanisms. First of all the *rank-one update* [74] using the so called evolution path $\mathbf{p}_c^{(g)} \in \mathbb{R}^n$ whose update is given by:

$$\mathbf{p}_c^{(g+1)} = (1 - c_c) \mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c) \mu_{\text{eff}}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}, \quad (2.6)$$

where $c_c \in]0, 1]$. For the constant $c_c = 1$, the evolution path points toward the descent direction $\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$ and for $c_c \neq 1$, the vector $\mathbf{p}_c^{(g)}$ adds the steps followed by the mean vector over the iterations using some normalization to dampen previous steps, so as not to rely too much on old information. The vector $\mathbf{p}_c^{(g+1)}$ gives a direction where we expect to see good solutions. From the evolution path, the rank-one matrix $\mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)T}$ is built and added to the covariance matrix (see Eq. (2.7)). Geometrically it deforms the ellipsoid-density in the direction $\mathbf{p}_c^{(g+1)}$, i.e., the rank-one update increases the probability to sample in the next iteration in the direction $\mathbf{p}_c^{(g+1)}$.

2.1 Covariance Matrix Adaptation - Evolution Strategy

The second mechanism is the *rank-mu update* [72] where the rank-mu matrix $\sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^T$ is added to the covariance matrix. This rank-mu matrix is also the stochastic approximation of the gradient of the joint criterion \mathcal{J} with respect to \mathbf{C} [9]. The update of the covariance matrix combines rank-one and rank-mu update and reads:

$$\mathbf{C}^{(g+1)} = (1 - c_{\text{cov}}) \mathbf{C}^{(g)} + \underbrace{\frac{c_{\text{cov}}}{\mu_{\text{cov}}} \mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)T}}_{\text{rank-one update}} + c_{\text{cov}} \underbrace{\left(1 - \frac{1}{\mu_{\text{cov}}}\right) \times \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^T}_{\text{rank-mu update}}. \quad (2.7)$$

The initial evolution path $\mathbf{p}_c^{(0)}$, c_c , c_{cov} and μ_{cov} are parameters of the algorithm. Default values can be found in [71].

In addition to the covariance matrix adaptation, the step-size $\sigma^{(g)}$ is controlled after every iteration. To perform the adaptation, a conjugate evolution path $\mathbf{p}_\sigma^{(g)} \in \mathbb{R}^n$ at generation g is updated according to:

$$\begin{aligned} \mathbf{p}_\sigma^{(g+1)} &= (1 - c_\sigma) \mathbf{p}_\sigma^{(g)} \\ &\quad + \sqrt{c_\sigma (2 - c_\sigma) \mu_{\text{eff}}} \mathbf{C}^{(g) - \frac{1}{2}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}. \end{aligned} \quad (2.8)$$

The conjugate path differs from the evolution path in the direction of the steps added, as in the conjugate path the normalized step $\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$ is multiplied by the matrix $\mathbf{C}^{(g) - \frac{1}{2}} \mathbf{1}$.

The step-size is adapted according to:

$$\sigma^{(g+1)} = \sigma^{(g)} \exp \left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma^{(g+1)}\|}{\mathbb{E}\|\mathcal{N}(0, \mathbf{I})\|} - 1 \right) \right), \quad (2.9)$$

where $\mathbf{p}_\sigma^{(0)}$, c_σ and d_σ are parameters of the algorithm with default values defined in [71]. This update rule implements to increase the step-size when the length of the conjugate evolution path is larger than the length it would have if selection would be random (this length will then be equal to $\|\mathcal{N}(0, \mathbf{I})\|$) and decrease it otherwise.

All the updates rely on the ranking determined by Eq. (2.3) only and not on the exact value of the objective functions making the algorithm invariant to monotonic transformations of the objective functions that preserve the ranking of solutions.

On the class of functions $\mathbf{x} \mapsto g_M \circ f_{\text{cq}}(\mathbf{x})$ where f_{cq} is a convex quadratic function and $g_M : \mathbb{R} \rightarrow \mathbb{R}$ a monotonically increasing function, the covariance matrix sequence $\mathbf{C}^{(g)}$ becomes proportional to the inverse Hessian of the function $f_{\text{cq}}(\mathbf{x})$, i.e., the algorithm is able to learn second order information without using any derivatives.

¹This difference is mainly technical in order to be able to compare the length of the conjugate path at different iterations though the steps have been sampled with different covariance matrices [74]

Step-size adaptation is important to achieve fast convergence corresponding to linear convergence with rates close to optimal rates that can be achieved by evolution strategies algorithms. In combination with covariance matrix adaptation, step-size adaptation allows to achieve linear convergence on a wide range of functions including *ill-conditioned problems*.

CMA-ES and EnOpt The ensemble-based optimization (EnOpt) [37, 36, 131] shares similarities with CMA-ES. In the following, we briefly present the main idea of EnOpt as well as the similarities and differences with CMA-ES. Original notations defined in [131] have been changed in order to be in accordance with the notations used for CMA-ES.

In EnOpt, for every iteration, an ensemble of λ points is sampled according to:

$$\mathbf{x}_i^{(g+1)} = \mathbf{m}^{(g)} + \mathcal{N}_i(\mathbf{0}, \mathbf{C}_X) \quad \text{for } i = 1, \dots, \lambda, \quad (2.10)$$

where $\mathcal{N}_i(\mathbf{0}, \mathbf{C}_X)_{1 \leq i \leq \lambda}$ are λ independent multivariate normal distributions with zero mean vector and covariance matrix \mathbf{C}_X . \mathbf{C}_X is a user specified matrix, which remains constant during the whole optimization process. Therefore, EnOpt adapts only the mean $\mathbf{m}^{(g)}$ of the distribution according to:

$$\mathbf{m}^{(g+1)} = \mathbf{m}^{(g)} + \alpha^{(g)} \mathbf{C}_X \mathbf{C}_{X,J}^{(g)}, \quad (2.11)$$

where $\alpha^{(g)}$ is the step-size and $\mathbf{C}_{X,J}^{(g)}$ is the cross-covariance between the population and the approximate gradient of the objective function.

Hence, while EnOpt and CMA-ES shares some similarities, CMA-ES presents three important advantages:

- CMA-ES adapts the covariance matrix used to sample its population to the landscape of the objective function as shown above. However, EnOpt uses the same covariance matrix during the whole optimization process which may lead to difficulties in refining the search at the end of the optimization;
- CMA-ES uses a step-size adaptation mechanism where the step-size is increased or decreased depending on the situation which is crucial to obtain linear convergence. However, in EnOpt, the step-size is always decreased and thus too small values at the beginning will be very detrimental for the convergence rate. Situations where step-size should be increased (linear environment) are also sub-optimally handled;
- CMA-ES is invariant to monotonic transformations of the objective functions that preserve the ranking of solutions, which represents a source of robustness of the algorithm [59]. More particularly, this invariance of CMA-ES removes the need to

tune the parameters of the algorithm according to the scale of the objective function, which is in general a challenging task. However, EnOpt uses the exact values of the objective function to update the mean of its search distribution which leads to breaking the invariance that comparison-based algorithms, such as CMA-ES, have.

2.2 Handling constraints with CMA-ES

Several methods are used, in the literature, to handle constraints in stochastic optimization algorithms. In general, unfeasible individuals can be rejected, penalized or repaired. In the following, we briefly discuss these alternatives. A more detailed study and comparison can be found in [96].

- Rejection of unfeasible individuals: Besides its simplicity and ease of implementation, rejecting the unfeasible individuals, also called “death penalty” does not require any parameter to be tuned. However, ignoring unfeasible individuals can prevent the algorithm from finding the region containing the optimum solution if it is close to the feasible domain boundaries [95];
- Penalizing unfeasible individuals: Penalization is the most widespread approach used to handle constraints. This method corresponds to a transformation of the optimization problem:

$$\begin{cases} \min f(\mathbf{x}) \\ \text{s.t. } h_i(\mathbf{x}) \leq d_i \quad \forall i = 1, \dots, m \end{cases} \quad (2.12)$$

$$\Rightarrow \min f(\mathbf{x}) + \sum_{i=1}^m g(h_i(\mathbf{x}) - d_i) \quad ,$$

where m is the number of constraints and $g(\cdot)$ is the penalty function which is non-negative, equal to zero in \mathbb{R}_- and increasing in \mathbb{R}_+ . In general, $g(\cdot)$ contains parameters to be tuned. These parameters depend on the problem to be optimized. A solution to avoid the difficulty of tuning those parameters consists in using an adaptive penalization which does not require any user specified constant. However, penalizing all unfeasible individuals implies evaluating all unfeasible individuals which can be costly;

- Repairing unfeasible individuals: Another popular solution to handle constraints is to repair each unfeasible individual before evaluating it. An important parameter to be specified is the probability of replacement of the unfeasible individual by the repaired new feasible individual. Moreover, repairing introduces a new individual in the population which may not obey to the adapted distribution, and hence may hold up the optimization process of CMA-ES.

Knowing the limitations of each of the constraint-handling approaches, the approach used in the present work is a mixture between two approaches: *adaptive penalization of the marginally unfeasible individuals* and *rejection of only the unfeasible individuals far from the boundaries of the feasible domain*. Using this approach, rejecting only individuals far from the feasible domain does not prevent the algorithm from finding a solution near the feasible domain boundaries, and by using adaptive penalization, the critical penalization coefficients are adapted automatically during the course of the search¹.

A box constraint handling is presented in [73] in which the feasible space is a hypercube defined by lower and upper boundary values for each parameter. In the following, this approach is generalized in order to handle feasible spaces defined by lower and upper boundary values for a sum of some of the parameters (e.g., to constrain the length of multilateral wells).

Given an optimization problem with a dimension n , let us suppose we have $m \in \mathbb{N}$ constraints denoted by S_j , $\forall j = 1, \dots, m$. For each constraint S_j , we define $P_j \subset \{1, \dots, n\}$ such that a vector $\mathbf{x} = (x_i)_{1 \leq i \leq n}$ is feasible with respect to the constraint S_j if:

$$v_{(j,-)} < q_j = \sum_{p \in P_j} x_p < v_{(j,+)} \quad , \quad (2.13)$$

where $v_{(j,-)}$ and $v_{(j,+)}$ are the lower and upper boundaries defining S_j . Constraints are then handled as follows, when evaluating an individual \mathbf{x} :

- *Initializing weights*: In the first generation, boundary weights γ_j are initialized to $\gamma_j = 0$, $\forall j = 1, \dots, m$;
- *Setting weights*: From the second generation upwards, if the distribution mean is unfeasible and weights are not set yet

$$\gamma_j \leftarrow \frac{2\delta_{\text{fit}}}{\sigma^2 \frac{1}{n} \sum_{i=1}^n \mathbf{C}_{ii}} \quad , \quad \forall j = 1, \dots, m \quad , \quad (2.14)$$

where δ_{fit} is the median from the last $(20 + \frac{3n}{\lambda})$ generations of the interquartile range of the unpenalized objective function evaluations and \mathbf{C}_{ii} is the i^{th} diagonal element of the covariance matrix. The term $\left(\sigma^2 \frac{1}{n} \sum_{i=1}^n \mathbf{C}_{ii}\right)$ represents the mean of $(\sigma^2 \mathbf{C}_{ii})_{i=1, \dots, n}$ which will be used in Eq. (2.16) in order to normalize the square of the distance which is $(q_j^{\text{feas}} - q_j)^2$ with respect to the covariance matrix adapted by CMA-ES ;

- *Increasing weights*: For each constraint S_j , if the distribution mean M_j , i.e., the mean of q_j for the λ individuals of the current generation, is out-of-bounds and the distance from M_j to the feasible domain, i.e., $\max(0, M_j - v_{(j,+)}) + \max(0, v_{(j,-)} - M_j)$ is larger than

¹The penalization method depends in general on other parameters which are on the other hand much less critical and which are tuned beforehand to be suitable for a wide range of problems [73].

$$\sigma \times \sqrt{\frac{1}{\text{card}(P_j)} \sum_{p \in P_j} \mathbf{C}_{pp}} \times \max\left(1, \frac{\sqrt{n}}{\mu_{\text{eff}}}\right) \text{ then}$$

$$\gamma_j \leftarrow \gamma_j \times 1.1^{\max(1, \frac{\mu_{\text{eff}}}{10n})}, \quad \forall j = 1, \dots, m, \quad (2.15)$$

where $\text{card}(P_j)$ denotes the cardinality of the set P_j ;

- *Evaluating the individual:*

$$f(\mathbf{x}) \leftarrow f(\mathbf{x}) + \frac{1}{m} \sum_{j=1}^m \gamma_j \frac{(q_j^{\text{feas}} - q_j)^2}{\xi_j}, \quad (2.16)$$

where q_j^{feas} is the projection of q_j on the feasible domain and $\xi_j = \exp\left(0.9 \left(\frac{1}{\text{card}(P_j)} \sum_{p \in P_j} \log(\mathbf{C}_{pp}) - \frac{1}{n} \times \sum_{i=1}^n \log(\mathbf{C}_{ii})\right)\right)$.

An individual \mathbf{x} , in the following, will be rejected and resampled if $|q_j^{\text{feas}} - q_j| > p\% \times |v_{(j,+)} - v_{(j,-)}|$, where $p\%$ is a parameter to be chosen. In all runs presented in the sequel, $p\%$ is chosen to be equal to 20%.

2.3 CMA-ES with local meta-models

Many real-world optimization problems are formulated in a black-box scenario where the objective function to optimize may have noise, multiple optima and can be computationally expensive. For expensive objective functions—several minutes to several hours for one evaluation—a strategy is to couple evolutionary algorithms with meta-models or surrogates: a model of f is built, based on “true” evaluations of f , and used during the optimization process to save evaluations of the expensive objective function [83]. One key issue when coupling EAs and meta-models is to decide when the quality of the model is good enough to continue exploiting this model and when new evaluations on the “true” objective functions should be performed, i.e., the exploration-exploitation trade-off defined in Section 1.2.1.3. Indeed, performing too few evaluations on the original objective function can result in suboptimal solutions whereas performing too many of them can lead to a non efficient approach.

CMA-ES was coupled with local meta-models to define the local-meta-model CMA-ES (lmm-CMA) [87]. In the proposed algorithm, the quality of the meta-model is appraised by tracking the change in the *exact ranking* of the best individuals. The lmm-CMA algorithm has been evaluated on test functions using the default population size of CMA-ES for unimodal functions and for some multi-modal functions and has been shown to improve CMA-ES [87].

In this section, we review the lmm-CMA algorithm as defined in [87] in Section 2.3.1 and then we analyze the performance of lmm-CMA when using population sizes larger than

the default one in Section 2.3.2. We show that tracking the exact rank-change of the best solutions to determine when to re-evaluate new solutions is a too conservative criterion and leads to a decrease of the speedup with respect to CMA-ES when the population size is increased. Instead we propose in Section 2.3.3 a less conservative criterion that we evaluate on test functions to define a new variant of CMA-ES with meta-models that we call the new-local-meta-model CMA-ES (nlmm-CMA).

2.3.1 The local-meta-model CMA-ES (lmm-CMA)

The lmm-CMA algorithm [87] combines the CMA-ES with local meta-models by exploiting the fact that the updates of CMA-ES only rely on the ranking of the μ best solutions. An iteration of lmm-CMA consists of one iteration of CMA-ES where the evaluation step on the (true) objective function that usually determines the ranking of the μ best solutions is replaced by the approximate ranking procedure that outputs an approximate ranking of the candidate solutions and that costs maximally λ function evaluations on the (true) objective function (the benefit of the approach comes of course when it costs less than λ). The mean value, covariance matrix and step-size of CMA-ES are then updated according to the update equations defined by the standard CMA-ES.

2.3.1.1 Locally weighted regression

To build an approximate model of the objective function f , denoted by \hat{f} , we use a locally weighted regression. During the optimization process, a database, i.e., a training set is built by storing, after every evaluation on the true objective function, points together with their objective function values $(\mathbf{x}, y = f(\mathbf{x}))$. Assuming that the training set contains a sufficient number m of couples $(\mathbf{x}, f(\mathbf{x}))$, let us consider an individual denoted $\mathbf{q} \in \mathbb{R}^n$ to be evaluated with the approximate model, where n is the dimension of the problem. We begin by selecting the k nearest points $(\mathbf{x}_j)_{1 \leq j \leq k}$ from the training set. The distance used for this purpose exploits the natural metric defined by the covariance matrix of CMA, namely the Mahalanobis distance with respect to the current covariance matrix \mathbf{C} defined for two given points $\mathbf{z}_1 \in \mathbb{R}^n$ and $\mathbf{z}_2 \in \mathbb{R}^n$ by $d_{\mathbf{C}}(\mathbf{z}_1, \mathbf{z}_2) = \sqrt{(\mathbf{z}_1 - \mathbf{z}_2)^T \mathbf{C}^{-1} (\mathbf{z}_1 - \mathbf{z}_2)}$. We build with locally weighted regression an approximate objective function using (true) evaluations $(y_j)_{1 \leq j \leq k}$ corresponding to the k selected nearest points to \mathbf{q} .

The use of a full quadratic meta-model is suggested in [87]. Hence, using a vector $\beta \in \mathbb{R}^{\frac{n(n+3)}{2}+1}$, we define \hat{f} as follows:

$$\hat{f}(\mathbf{x}, \beta) = \beta^T \begin{pmatrix} x_1^2, \dots, x_n^2, \dots, x_1 x_2, \dots, \\ x_{n-1} x_n, x_1, \dots, x_n, 1 \end{pmatrix}^T. \quad (2.17)$$

The full quadratic meta-model is built based on minimizing the following criterion with respect to the vector of parameters β of the meta-model at \mathbf{q} :

$$A(\mathbf{q}) = \sum_{j=1}^k \left[\left(\hat{f}(\mathbf{x}_j, \beta) - y_j \right)^2 K \left(\frac{d_{\mathbf{C}}(\mathbf{x}_j, \mathbf{q})}{h} \right) \right]. \quad (2.18)$$

The kernel weighting function $K(\cdot)$ is defined by $K(\zeta) = (1 - \zeta^2)^2$, and h is the bandwidth defined by the distance of the k^{th} nearest neighbor data point to \mathbf{q} where k must be greater or equal to $\frac{n(n+3)}{2} + 1$ for a full quadratic meta-model.

2.3.1.2 Approximate ranking procedure

To incorporate the approximate model built using the locally weighted regression, we use the approximate ranking procedure [111]. This procedure decides whether the quality of the model is good enough in order to continue exploiting this model or new true objective function evaluations should be performed. The resulting method is called the local-meta-model CMA-ES (lmm-CMA) [87] and is defined as follows. For a given generation, let us denote individuals of the current population of CMA-ES by $(\mathbf{x}_i)_{1 \leq i \leq \lambda}$, where λ is the population size. The following procedure is then performed:

1. *build* $\hat{f}(\mathbf{x}_i)$ for all individuals of the current population $(\mathbf{x}_i)_{1 \leq i \leq \lambda}$.
2. *rank* individuals according to their approximated value $\hat{f}(\mathbf{x}_i)$: ranking_0 .
3. *evaluate* the best n_{init} individuals with the true objective function and *add* their evaluations to the training set.
4. for n_{ic} from 1 to $\left(\frac{\lambda - n_{\text{init}}}{n_{\text{b}}} \right)$, we:
 - (a) *build* $\hat{f}(\mathbf{x}_i)_{1 \leq i \leq \lambda}$.
 - (b) *rank* individuals according to their approximated value $\hat{f}(\mathbf{x}_i)$ ¹: $\text{ranking}_{n_{\text{ic}}}$.
 - (c) if $(\text{ranking}_{n_{\text{ic}}} = \text{ranking}_{n_{\text{ic}}-1})$, the meta-model is accepted.
 - (d) if the meta-model is accepted, we break. If not, we *evaluate* the best n_{b} unevaluated individuals with the true objective function, *add* their evaluations to the training set, and *loop* to step 4, until reaching the acceptance criterion of the meta-model.
5. if $(n_{\text{ic}} > 2)$, $n_{\text{init}} = \min(n_{\text{init}} + n_{\text{b}}, \lambda - n_{\text{b}})$.
6. if $(n_{\text{ic}} < 2)$, $n_{\text{init}} = \max(n_{\text{b}}, n_{\text{init}} - n_{\text{b}})$.

¹Or true objective function if the individuals have been evaluated on it.

Table 2.1: Test functions and their corresponding initial intervals and standard deviations. The starting point is uniformly drawn from the initialized interval.

Name	Function	Init.	σ^0
Noisy Sphere	$f_{\text{NSphere}}(\mathbf{x}) = \left(\sum_{i=1}^n x_i^2\right) \exp(\epsilon \mathcal{N}(0, 1))$	$[-3, 7]^n$	5
Schwefel	$f_{\text{Schw}}(\mathbf{x}) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)^2$	$[-10, 10]^n$	10
Schwefel ^{1/4}	$f_{\text{Schw}^{1/4}}(\mathbf{x}) = \left(f_{\text{Schwefel}}(x)\right)^{\frac{1}{4}}$	$[-10, 10]^n$	10
Rosenbrock	$f_{\text{Rosen}}(\mathbf{x}) = \sum_{i=1}^{n-1} \left(100 \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right)$	$[-5, 5]^n$	5
Ackley	$f_{\text{Ack}}(\mathbf{x}) = 20 - 20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) + e - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right)$	$[1, 30]^n$	14.5
Rastrigin	$f_{\text{Rast}}(\mathbf{x}) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cdot \cos(2\pi x_i))$	$[1, 5]^n$	2

This procedure heavily exploits the rank-based property of the CMA-ES algorithm. Initially, a number n_{init} of best individuals based on the meta-model is evaluated using the true objective function and then added to the training set. A batch of n_{b} individuals is evaluated until satisfying the meta-model acceptance criterion: *keeping the ranking of each of the μ best individuals based on the meta-model unchanged for two iteration cycles*. Hence, $(n_{\text{init}} + n_{\text{b}} * n_{\text{ic}})$ individuals are evaluated every generation where n_{ic} represents the number of iteration cycles needed to satisfy the meta-model acceptance criterion. The integer n_{b} is chosen to be equal to $\max[1, (\frac{\lambda}{10})]$ and n_{init} is initialized to λ and adapted after every generation. The minimum number of evaluations performed for a given generation, which corresponds to the minimum value that n_{init} can reach, is then equal to n_{b} .

2.3 CMA-ES with local meta-models

Table 2.2: Success performance SP1, i.e., the average number of function evaluations for successful runs divided by the ratio of successful runs, standard deviations of the number of function evaluations for successful runs and speedup performance spu, to reach $f_{\text{stop}} = 10^{-10}$ of lmm-CMA and nlmm-CMA. The ratio of successful runs is denoted between brackets if it is < 1.0 . Results with a constant dimension $n = 5$ and an increasing λ are highlighted in grey.

Function	n	λ	ϵ	lmm-CMA		spu	nlmm-CMA		spu	CMA-ES		
f_{Rosen}	2	6		291	± 59	2.7	252	± 52	3.1	779	± 236	
	4	8		776	± 102	[0.95] 2.8	719	± 54	[0.85] 3.0	2185	± 359	[0.95]
	5	8		1131	± 143	2.7	1014	± 94	[0.90] 3.0	3012	± 394	[0.90]
	5	16		1703	± 230	[0.95] 2.0	901	± 64	3.7	3319	± 409	
	5	24		2784	± 263	1.4	1272	± 90	[0.95] 3.0	3840	± 256	
	5	32		3364	± 221	1.3	1567	± 159	2.9	4515	± 275	
	5	48		4339	± 223	1.3	1973	± 144	2.9	5714	± 297	
5	96		6923	± 322	1.2	3218	± 132	2.5	7992	± 428		
f_{Schw}	8	10		2545	± 233	[0.95] 2.1	2234	± 202	[0.95] 2.4	5245	± 644	
	2	6		89	± 9	4.3	87	± 7	4.4	385	± 35	
	4	8		166	± 8	5.4	166	± 6	5.4	897	± 51	
	8	10		334	± 9	6.2	333	± 9	6.2	2078	± 138	
	16	12		899	± 40	5.9	855	± 30	6.2	5305	± 166	
$f_{\text{Schw}^{1/4}}$	2	6		556	± 25	2.4	413	± 25	3.3	1343	± 72	
	4	8		1715	± 87	1.7	971	± 36	2.9	2856	± 135	
	5	8		2145	± 69	1.6	1302	± 31	2.7	3522	± 136	
	5	16		3775	± 137	1.3	1446	± 31	3.4	4841	± 127	
	5	24		5034	± 142	1.2	1825	± 45	3.4	6151	± 252	
	5	32		6397	± 174	1.2	2461	± 43	3.2	7765	± 227	
	5	48		8233	± 190	1.2	3150	± 58	3.2	10178	± 202	
5	96		11810	± 177	1.2	4930	± 94	2.9	14290	± 252		
f_{NSphere}	8	10		4046	± 127	1.5	2714	± 41	2.2	5943	± 133	
	2	6	0.35	124	± 14	2.7	109	± 12	3.1	337	± 34	
	4	8	0.25	316	± 45	2.3	236	± 19	3.1	739	± 30	
	8	10	0.18	842	± 77	1.8	636	± 33	2.4	1539	± 69	
f_{Ack}	16	12	0.13	2125	± 72	1.3	2156	± 216	1.3	2856	± 88	
	2	5		302	± 43	[0.90] 2.6	227	± 23	3.5	782	± 114	[0.95]
	5	7		1036	± 620	2.0	704	± 23	[0.90] 3.0	2104	± 117	[0.85]
f_{Rast}	10	10		2642	± 93	[0.90] 1.4	2066	± 119	[0.95] 1.8	3787	± 151	[0.95]
	2	50		898	± 160	[0.95] 2.7	524	± 48	[0.95] 4.7	2440	± 294	[0.75]
	5	70		19911	± 599	[0.15] 0.6	9131	± 135	[0.15] 1.3	11676	± 711	[0.50]
	5	140		6543	± 569	[0.80] 1.6	4037	± 209	[0.60] 2.6	10338	± 1254	[0.85]
	5	280		10851	± 1008	[0.85] 1.3	4949	± 425	[0.85] 2.9	14266	± 1069	

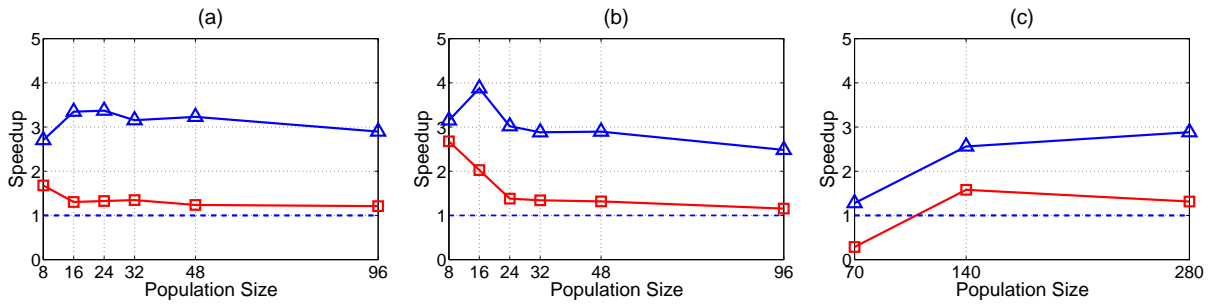


Figure 2.2: Speedup of nlmm-CMA (\triangle) and lmm-CMA (\square) on (a) $f_{\text{Schw}^{1/4}}$, (b) f_{Rosen} and (c) f_{Rast} for dimension $n = 5$.

2.3.2 Evaluating lmm-CMA on increasing population size

2.3.2.1 Experimental procedure

The lmm-CMA and the other variants tested in this chapter are evaluated on the objective functions presented in Table 2.1 corresponding to the functions used in [87] except two functions: (1) the function $f_{\text{Schw}^{1/4}}$ where we compose the convex quadratic function f_{Schw} by a strictly increasing mapping $g : x \in \mathbb{R} \mapsto x^{1/4}$, introduced because we suspect that the results on f_{Schw} are artificial and only reflect the fact that the model used in lmm-CMA is quadratic and (2) the noisy sphere function f_{NSphere} whose definition has been modified following the recommendations of [82]. We have followed the experimental procedure in [87] and performed for each test function 20 independent runs using an implementation of lmm-CMA based on a java code of CMA-ES¹ randomly initialized from initial intervals defined in Table 2.1 and with initial standard deviations σ_0 in Table 2.1 and other standard parameter settings in [71]. The algorithm performance is measured using the success performance SP1 used in [11]. SP1 is defined as the average number of evaluations for successful runs divided by the ratio of successful runs, where a run is considered as successful if it succeeds in reaching $f_{\text{stop}} = 10^{-10}$. Another performance measure that might be used was the expected running time ERT [69] which is defined as the number of function evaluations conducted in all runs (successful and unsuccessful runs) divided by the ratio of successful runs. In this chapter, we opt for SP1 since the stopping criteria for unsuccessful runs were not properly tuned which can affect the performance comparison. We have reproduced the results for the lmm-CMA presented in [87, Table 3]. Those results are presented in Table 2.2².

¹See http://www.lri.fr/~hansen/cmaes_inmatlab.html.

²Experiments have been performed with $k = n(n+3) + 2$ indicated in [87]. However we observed some differences on f_{Rosen} and f_{Schw} with this value of k and found out that $k = \frac{n(n+3)}{2} + 1$ allows to obtain the results presented in [87, Table 3]. We did backup this finding by using the Matlab code provided by Stefan Kern.

2.3.2.2 Performances of Imm-CMA with increasing population size

In Imm-CMA, a meta-model is accepted if the exact ranking of the μ best individuals remains unchanged. However, this criterion is more and more difficult to satisfy when the population size λ and thus $\mu (= \lambda/2)$ increases. We suspect that this can have drastic consequences on the performances of Imm-CMA. To test our hypothesis we perform tests for $n = 5$ on f_{Rosen} , $f_{\text{Schw}^{1/4}}$ with $\lambda = 8, 16, 24, 32, 48, 96$ and for f_{Rast} for $\lambda = 70, 140, 280$. The results are presented in Fig. 2.2 and in Table 2.2 (rows highlighted in grey). On f_{Rosen} and $f_{\text{Schw}^{1/4}}$, we observe, as expected that the speedup with respect to CMA-ES drops with increasing λ and is approaching 1. On f_{Rast} , we observe that the speedup for $\lambda = 140$ is larger than for $\lambda = 280$ (respectively equal to 1.6 and 1.3).

2.3.3 A new variant of Imm-CMA

We propose now a new variant of Imm-CMA, the new-local-meta-model CMA-ES (nlmm-CMA) that tackles the problem detected in the previous section.

2.3.3.1 A new meta-model acceptance criteria

We have seen that requiring the preservation of the exact ranking of the μ best individuals is a too conservative criterion for population sizes larger than the default one to measure the quality of meta-models. We therefore propose to replace this criterion by the following one: after building the model and ranking it, a meta-model is accepted if it succeeds in keeping, both the ensemble of μ individuals and the best individual unchanged. In this case, we ignore any change in the rank of each individual from the best μ individuals, except for the best individual which must be the same, as long as this individual is still an element of the μ best ensemble. Another criterion is added to the acceptance of the meta-model: once more than one fourth of the population is evaluated, the model is accepted if it succeeds to keep the best individual unchanged. The proposed procedure is then defined as follows. For a given generation, let us denote individuals of the current population of CMA-ES by $(\mathbf{x}_i)_{1 \leq i \leq \lambda}$, where λ is the population size. The following new approximate ranking procedure is then performed:

1. *build* $\hat{f}(\mathbf{x}_i)$ for all individuals of the current population $(\mathbf{x}_i)_{1 \leq i \leq \lambda}$.
2. *rank* individuals according to their approximated value $\hat{f}(\mathbf{x}_i)$ and *determine* the μ best individuals set and the best individual.
3. *evaluate* the n_{init} best individuals with the true objective function and *add* their evaluations to the training set.
4. for n_{ic} from 1 to $\left(\frac{\lambda - n_{\text{init}}}{n_{\text{b}}}\right)$, we:

-
- (a) *build* $\hat{f}(\mathbf{x}_i)_{1 \leq i \leq \lambda}$.
 - (b) *rank* individuals according to their approximated value $\hat{f}(\mathbf{x}_i)$ ¹ and *determine* the μ best individuals set and the best individual.
 - (c) if less than one fourth of the population is evaluated, the meta-model is accepted if it succeeds in keeping both **the best individual** and **the ensemble of μ best individuals** unchanged.
 - (d) if more than one fourth of the population is evaluated, the meta-model is accepted if it succeeds in keeping **the best individual** unchanged.
 - (e) if the meta-model is accepted, we break. If not, we *evaluate* the n_b best unevaluated individuals with the true objective function, *add* their evaluations to the training set, and *loop* to step 4, until reaching the acceptance criterion of the meta-model.
5. if $(n_{ic} > 2)$, $n_{init} = \min(n_{init} + n_b, \lambda - n_b)$.
 6. if $(n_{ic} < 2)$, $n_{init} = \max(n_b, n_{init} - n_b)$.

Considering only changes in the whole parent set, without taking into account the exact rank of each individual, and setting an upper limit on the number of true objective function evaluations was first proposed in [13]. The new variant is called nlmm-CMA in the sequel.

2.3.3.2 Evaluation of nlmm-CMA

The performance results of nlmm-CMA are presented in Table 2.2 together with the ones of lmm-CMA. Table 2.2 shows that on f_{Rast} , the nlmm-CMA speedup is in between 2.5 and 5 instead of 1.5 and 3 for lmm-CMA, and on f_{Ack} nlmm-CMA outperforms lmm-CMA with speedups between 1.5 and 3.5 for nlmm-CMA and between 1.4 and 3 for lmm-CMA. On these functions, nlmm-CMA is significantly more efficient. For the other tested functions f_{Rast} , f_{Schw} and $f_{Schw^{1/4}}$, nlmm-CMA is marginally more efficient than the standard lmm-CMA. In Fig. 2.2 and in Table 2.2 (highlighted rows), we evaluate the effect of increasing λ on nlmm-CMA using the same setting as in Section 2.3.2.2. Using population sizes larger than the default one, nlmm-CMA improves CMA-ES by a factor between 2.5 and 3.5 for all tested functions f_{Rosen} , $f_{Schw^{1/4}}$ and f_{Rast} . Therefore, nlmm-CMA maintains a significant speedup for λ larger than the default one contrary to lmm-CMA which offers a speedup approaching to 1 for f_{Rosen} and $f_{Schw^{1/4}}$ and a decreasing speedup (from 1.6 to 1.3) when λ increases (from 140 to 280) for f_{Rast} .

¹Or true objective function if the individuals have been evaluated on it.

2.3.3.3 Impact of the recombination type

The choice of the recombination type has an important impact on the efficiency of evolution strategies in general [6] and CMA-ES in particular [74, 71]. In the previous section, all the runs performed use the default weighted recombination type defined by Eq. (2.5). In the new variant of lmm-CMA, the meta-model acceptance criterion does not take into account the exact rank of each individual except the best one. By modifying the meta-model acceptance criteria of lmm-CMA, a possible accepted meta-model may be a meta-model that preserves the μ best individuals set and the best individual but generates a ranking far from the “true” ranking, i.e., the one based on the true objective function. We now compare nlmm-CMA using weighted recombination where weights are defined in Eq. (2.5) and intermediate recombination where weights are all equal to $1/\mu$: nlmm-CMA_I. Results are presented in Table 2.3. The algorithm nlmm-CMA outperforms nlmm-CMA_I in all cases suggesting that even if the exact ranking is not taken into account for assessing the quality of the meta-model in nlmm-CMA, this ranking is not random and still has an amount of information to guide CMA-ES.

2.3.3.4 Impact of initial parameters

In the tests presented so far, the initial parameters of the approximate ranking procedure are defined as follows: n_{init} is initialized at the beginning of the optimization process to λ , and n_{b} is set to $\max[1, (\frac{\lambda}{10})]$. Every generation g , the number of initial individuals evaluated n_{init} is adapted (increased or decreased) depending on the meta-model quality (Steps 5. and 6. in the procedure defined in Section 2.3.3.1). We denote by $n_{\text{init}}^{(g)}$ and $n_{\text{ic}}^{(g)}$ the values of n_{init} and n_{ic} respectively at generation g . The number of evaluations performed every generation g is $(n_{\text{init}}^{(g)} + n_{\text{ic}}^{(g)} \times n_{\text{b}})$. We quantify now the impact of the initial values of (n_{init} and n_{b}) on the total cost of the optimization process. The algorithm nlmm-CMA is compared to a similar version where initial parameters are chosen as small as possible, i.e., $n_{\text{init}}^{(0)}$ and n_{b} are equal to 1. Moreover, we consider two cases: (1) with update denoted nlmm-CMA₁, i.e., where initial parameters are adapted depending on the iteration cycle number (Steps 5. and 6. in the procedure defined in Section 2.3.3.1), and (2) without update denoted nlmm-CMA₂, i.e., parameters are equal to 1 during the entire optimization process (omitting steps 5. and 6. in the procedure defined in Section 2.3.3.1). We note that in case (1), the number of evaluations for each generation g is $(n_{\text{init}}^{(g)} + n_{\text{ic}}^{(g)})$. In case (2), every generation g , lmm-CMA evaluates $(1 + n_{\text{ic}}^{(g)})$ individuals, since $n_{\text{init}}^{(g)} = 1$. The results on different test functions are summarized in Table 2.3.

On the unimodal functions f_{Schw} , $f_{\text{Schw}^{1/4}}$, setting n_{init} and n_{b} as small as possible in every generation, is marginally more efficient than the default definition of initial parameters on small dimensions except for dimension $n = 8$ and $\lambda = 10$. On f_{Rosen} , nlmm-CMA₂ is

the most efficient compared to other approaches, except for dimension $n = 8$ and $\lambda = 10$ which can be justified by a higher number of unsuccessful runs compared to other approaches. On the multi-modal function f_{Ack} , modifying the initial parameter n_{init} does not have an important impact on the speedup of lmm-CMA (between 1.5 and 4). However on f_{Rast} , using a small initial n_{init} decreases considerably the probability of success of the optimization, from 0.95 to between 0.35 and 0.10 for dimension $n = 2$ and $\lambda = 50$, and from 0.60 to 0.10 for dimension $n = 5$ and $\lambda = 140$. These results confirm the initial parameter choice suggested in [87].

Table 2.3: SP1, standard deviations of the number of function evaluations for successful runs and speedup performance spu, to reach $f_{\text{stop}} = 10^{-10}$ of nlmm-CMA, nlmm-CMA_I (intermediate recombination and default initial parameters), nlmm-CMA₁ (default recombination, initial values of n_{init} and n_{b} set to 1) and nlmm-CMA₂ (default recombination type, $n_{\text{init}} = 1$ and $n_{\text{b}} = 1$ during the whole optimization process). The ratio of successful runs is denoted between brackets if it is < 1.0 .

Function	n	λ	ϵ	nlmm-CMA	spu	nlmm-CMA _I	spu	nlmm-CMA ₁	spu	nlmm-CMA ₂	spu
f_{Rosen}	2	6		252 ± 52	3.1	357 ± 67	2.2	250 ± 80	3.1	229 ± 53	3.4
	4	8		719 ± 54 [0.85]	3.0	833 ± 100	2.6	596 ± 55	3.7	575 ± 68	3.8
	8	10		2234 ± 202 [0.95]	2.4	2804 ± 256 [0.95]	1.9	2122 ± 133	2.5	2466 ± 207 [0.85]	2.1
f_{Schw}	2	6		87 ± 7	4.4	110 ± 10	3.5	75 ± 8	5.2	73 ± 7	5.3
	4	8		166 ± 6	5.4	220 ± 15	4.1	138 ± 6	6.5	136 ± 5	6.6
	8	10		333 ± 9	6.2	423 ± 15	4.9	374 ± 16	5.6	380 ± 21	5.5
$f_{\text{Schw}^{1/4}}$	16	12		855 ± 30	6.2	947 ± 24	5.6	794 ± 27	6.7	786 ± 37	6.8
	2	6		413 ± 25	3.3	550 ± 29	2.4	411 ± 20	3.3	398 ± 16	3.4
	4	8		971 ± 36	2.9	1320 ± 76	2.2	938 ± 32	3.1	909 ± 30	3.1
f_{NSphere}	8	10		2714 ± 41	2.2	2714 ± 257	2.2	2668 ± 40	2.2	2677 ± 36	2.2
	2	6	.35	109 ± 12	3.1	135 ± 19	2.5	92 ± 11	3.7	87 ± 9	3.9
	4	8	.25	236 ± 19	3.1	306 ± 40	2.4	216 ± 16	3.4	219 ± 16	3.4
f_{Ack}	8	10	.18	636 ± 33	2.4	788 ± 47	2.0	611 ± 35	2.5	619 ± 45	2.5
	16	12	.13	2156 ± 216	1.3	2690 ± 421	1.1	2161 ± 148	1.3	2195 ± 142	1.3
	2	5		227 ± 23	3.5	329 ± 29 [0.85]	2.4	226 ± 21 [0.95]	3.5	208 ± 19	3.8
f_{Rast}	5	7		704 ± 23 [0.90]	3.0	850 ± 43 [0.90]	2.5	654 ± 35 [0.95]	3.2	652 ± 32 [0.95]	3.2
	10	10		2066 ± 119 [0.95]	1.8	2159 ± 58	1.8	2394 ± 52 [0.80]	1.6	1925 ± 44	2.0
	2	50		524 ± 48 [0.95]	4.7	796 ± 68 [0.75]	3.1	569 ± 26 [0.35]	4.3	1365 ± 28 [0.10]	1.8
	5	140		4037 ± 209 [0.60]	2.6	5265 ± 313 [0.55]	2.0	13685 ± 257 [0.10]	0.8	7910 ± 82 [0.10]	1.3

2.4 Summary and discussions

In this chapter, we have introduced the stochastic optimizer CMA-ES, as well as an adaptive penalization with rejection technique in order to handle the optimization constraints.

We have explained that CMA-ES exhibits many invariances, a desirable property as it implies the generalization of results from one function to a class of functions and confer thus robustness and wider applicability of the method. In particular, CMA-ES is a *rank-based* search algorithm exploiting the objective function only through the relative ranking of solutions within the population. The rank-based property implies invariance of the algorithm on the class of functions class $f = \{g \circ f, g : \mathbb{R} \rightarrow \mathbb{R} \text{ strictly increasing}\}$ for any $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

In order to improve its performance when dealing with costly objective functions, the CMA-ES algorithm has been combined with local meta-models that are constructed using points from the archive of solutions—called the training set—evaluated on the (expensive) original objective function. The quality of the meta-models is appraised using an approximate ranking procedure that determines if the objective function predicted by the meta-model is good enough or more points should be evaluated on the original function. The resulting algorithm is called the local-meta-model CMA-ES (lmm-CMA) [87] (Section 2.3.1). In this chapter, the original acceptance criterion for the meta-models proposed for lmm-CMA has been shown to be too conservative for increasing population sizes (Section 2.3.2) and modified in order to maintain a reasonable speed-up when population sizes larger than the default one are used (Section 2.3.3). The proposed new variant is called the new-local-meta-model CMA-ES (nlmm-CMA).

In particular, we have investigated in this chapter the performances of the lmm-CMA algorithm coupling CMA-ES with local meta-models. On f_{Rosen} and $f_{\text{Schw}^{1/4}}$, we have shown that the speedup of lmm-CMA with respect to CMA-ES drops to one when the population size λ increases. This phenomenon has been explained by the too restrictive condition used to stop evaluating new points dedicated at refining the meta-model, namely requiring that the *exact* ranking of the $\mu = \lambda/2$ best solutions is preserved when evaluating a new solution on the exact objective function. To tackle this problem, we have proposed to relax the condition to: *the set of μ best solutions is preserved and the best individual is preserved*. The resulting new variant, nlmm-CMA outperforms lmm-CMA on the test functions investigated and the speedup with CMA-ES is between 1.5 and 7. Moreover, contrary to lmm-CMA it maintains a significant speedup, between 2.5 and 4, when increasing λ on f_{Rosen} , $f_{\text{Schw}^{1/4}}$ and f_{Rast} . The study of the impact of the recombination weights has shown that the default weights of CMA-ES are more appropriate than equal weights. The influence of two parameters, n_b and n_{init} , corresponding to the number of individuals evaluated respectively initially and in each iteration cycle has been investigated. We

have seen that setting those parameters to 1 during the whole optimization process can marginally improve the performances on uni-modal functions and some multi-modal test functions. However it increases the likelihood to be stuck in local minima for the Rastrigin function suggesting that the default parameter of Imm-CMA are still a good choice for nImm-CMA.

Chapter 3

Well placement optimization with CMA-ES and CMA-ES with meta-models

This chapter is based on the papers [26, 24]. In this chapter, we apply the CMA-ES algorithm to the well placement problem, with the adaptive penalization with rejection technique (introduced in Chapter 2) to handle constraints. Because genetic algorithms are quite often the method of choice in petroleum industry, we first show the improvement of applying CMA-ES over a GA on the synthetic benchmark reservoir case PUNQ-S3. In addition, because a reservoir simulation and thus the objective function is expensive, we apply the nlmm-CMA algorithm introduced in the previous chapter in order to save a number of evaluations by building a model of the problem. We validate the approach on the PUNQ-S3 case.

This chapter is structured as follows. Section 3.1 describes the problem formulation. In Section 3.2, CMA-ES is compared to a genetic algorithm on a synthetic reservoir case to show the contribution of the proposed optimization method. In Section 3.3, the reduction of the number of reservoir simulations is addressed by coupling CMA-ES with meta-models and the contribution of the whole methodology, i.e., CMA-ES with meta-models is demonstrated on a number of well location and trajectory optimization problems (with unilateral and multilateral wells).

3.1 The well placement optimization problem formulation

In this section, we describe the well placement optimization problem and explain the parameterization of the wells.

3.1.1 Objective function

The quality of a well placement decision is evaluated using an objective function that we aim at maximizing (good solutions have a high objective function value and we aim at finding the solution with the highest objective function value). The objective function associated with a well placement problem often evaluates the economic model of the decision and takes into account different costs such as prices of oil and gas, costs of drilling and costs of injection and production of water. Another alternative is to use the cumulative oil production or the barrel of oil equivalent (BOE). In this chapter, the objective function considered is the net present value NPV. Formally we want to find a vector of parameter \mathbf{p}_{\max} such that:

$$\text{NPV}(\mathbf{p}_{\max}) = \max_{\mathbf{p}} \{ \text{NPV}(\mathbf{p}) \} . \quad (3.1)$$

The NPV of a well configuration and trajectory represented by a vector of parameter \mathbf{p} is calculated using two terms, the expected revenue associated to \mathbf{p} denoted R and the drilling and completing cost of \mathbf{p} denoted C_d which is subtracted to the revenue term, i.e.,:

$$\text{NPV}(\mathbf{p}) = R(\mathbf{p}) - C_d(\mathbf{p}) . \quad (3.2)$$

The revenue term R is defined by summing the revenues from produced oil over all the wells, and subtracting the costs associated to produced water and to injected water. A discount rate –called also an annual percentage rate– is introduced to take into account the risk and uncertainty and the time value of money, that is oil produced earlier contributes more to the overall NPV. The detailed formula for the revenue term reads:

$$R = \sum_{n=1}^Y \left(\frac{1}{(1 + \text{APR})^n} \begin{bmatrix} Q_{n,o} \\ Q_{n,g} \\ Q_{n,w_a} \end{bmatrix}^T \begin{bmatrix} C_{n,o} \\ C_{n,g} \\ C_{n,w_a} \end{bmatrix} \right) , \quad (3.3)$$

where $Q_{n,p}$ is the field production of phase ph (either oil, gas or water denoted respectively o, g, w_a) at period n and $C_{n,p}$ is the profit or loss associated to this production. The annual percentage rate is denoted APR. The integer Y is the number of discount periods (years).

For the drilling and completing cost term C_d , we use the approximate formula used in [129] that proposes to estimate the drilling cost as the sum of two terms: the first term is proportional to the diameter of each lateral times the length of this lateral multiplied the logarithm of this lateral (taking into account that the cost is more than linear in the length), the second term adds up a fixed cost per junction, i.e.,:

$$C_d = \sum_{w=1}^{\mathcal{N}_w} \left(\sum_{k=0}^{\mathcal{N}_{\text{lat}}} [A \cdot d_w \cdot \ln(l_w) \cdot l_w]_{k,w} \right) + \sum_{m=1}^{\mathcal{N}_{\text{jun}}} [C_{\text{jun}}]_m , \quad (3.4)$$

3.1 The well placement optimization problem formulation

Table 3.1: Constants used to define the net present value (NPV).

Constant	Value
$C_{n,o}$	60 \$ / barrel
C_{n,w_a}	-4 \$ / barrel
$C_{n,g}$	0 \$ / barrel
APR	0.1
A	1000
d_w	0.1 m
C_{jun}	10^5 \$

where $k = 0$ represents the mainbore, $k > 0$ represents the laterals, l_w is the length of the lateral (in ft), d_w is the diameter of the mainbore (in ft), \mathcal{N}_w is the number of wells drilled, \mathcal{N}_{lat} is the number of laterals and A is a constant specific to the considered field containing conversion factors. C_{jun} is the cost of milling the junction and \mathcal{N}_{jun} is the number of junctions.

For this chapter, the constants used to define the NPV in Eqs. (3.3) and (3.4) are given in Table 3.1.

The computation of the NPV of a configuration \mathbf{p} requires to have a prediction of the quantity of oil, water and gas ($Q_{n,o}$, Q_{n,w_a} , $Q_{n,g}$) associated to \mathbf{p} in order to compute the revenue R given by Eq. (3.3). To compute those quantities we use a reservoir simulation which represents *the time consuming part* in the computation of the NPV objective function.

It is in general needed to impose different constraints on the well configuration to avoid finding both undrillable wells and wells that violates common engineering practices. The constraints handled in this thesis are as follows:

- maximum length of wells: $l_w < L_{max}$, for each well w to be placed;
- all wells must be inside the reservoir grid: $l_w = l_{inside}$, for each well w to be placed, where l_{inside} is the length of the well w inside the reservoir grid.

3.1.2 Well parameterization

In our approach, we want to be able to handle different possible configurations of multi-lateral wells. An illustrative scheme is given in Fig. 3.1. The terminology used to define each part of a multilateral well follows the terminology used in [77]. In general, a lateral can be defined by a line connecting two points. The mainbore is defined through the trajectory of its contiguous completed segments. Hence, we define a sequence of points where a deviation occurs $(P_{d,i})_{0 \leq i \leq \mathcal{N}_s}$ where \mathcal{N}_s is the number of segments. The starting point $P_{d,0} = P_0$ of the mainbore called the heel is represented by its Cartesian coordinates

3.1 The well placement optimization problem formulation

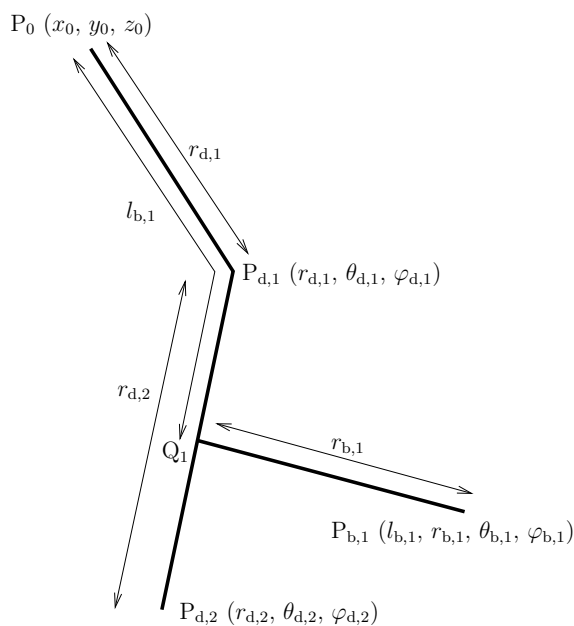


Figure 3.1: An example of a single multilateral well parameterization with two segments ($\mathcal{N}_s = 2$) and one branch ($\mathcal{N}_b = 1$).

(x_0, y_0, z_0) . Other intermediate points $(P_{d,i})_{1 \leq i \leq \mathcal{N}_s - 1}$ and the ending point P_{d,\mathcal{N}_s} called the toe are represented by their corresponding spherical coordinate system $(r_{d,i}, \theta_{d,i}, \varphi_{d,i})$ with respect to the basis $(P_{d,i-1}, \mathbf{u}_{d,i}^r, \mathbf{u}_{d,i}^\theta, \mathbf{u}_{d,i}^\varphi)$. We use spherical coordinates because they allow for straightforward control of the well lengths by imposing a box constraint whereas it would need to be handled by imposing a non linear constraint with Cartesian coordinates.

The wells are parameterized in a way to handle a number \mathcal{N}_b of branches and/or laterals as well.

The branch or lateral $j \in [1, \dots, \mathcal{N}_b]$ is defined by locating its ending point $P_{b,j}(l_{b,j}, r_{b,j}, \theta_{b,j}, \varphi_{b,j})$ where $(r_{b,j}, \theta_{b,j}, \varphi_{b,j})_{1 \leq j \leq \mathcal{N}_b}$ represents the spherical coordinates of $P_{b,j}$ with respect to the basis $(Q_j, \mathbf{u}_{b,j}^r, \mathbf{u}_{b,j}^\theta, \mathbf{u}_{b,j}^\varphi)$, Q_j is the starting point of the branch or the lateral j , and $l_{b,j}$ is the distance along the well between P_0 and Q_j .

The dimension D_w of the representation of a well denoted by w is as follows:

$$D_w = 3(1 + \mathcal{N}_s^w) + 4\mathcal{N}_b^w . \quad (3.5)$$

Hence, the dimension D of the problem of placing \mathcal{N}_w wells $(w_k)_{k=1, \dots, \mathcal{N}_w}$ is:

$$D = \sum_{k=1}^{\mathcal{N}_w} D_{w_k} . \quad (3.6)$$

An example of a single well parameterization is shown in Fig. 3.1. In this example,

\mathcal{N}_s is equal to two and \mathcal{N}_b is equal to one. The mainbore is then represented by three points P_0 and $(P_{d,i})_{1 \leq i \leq 2}$. The branch is represented by one point $P_{b,1}$. The corresponding dimension of the optimization problem is 13.

3.2 CMA-ES and a real-coded GA for the well placement problem

The choice of a stochastic optimization method was motivated by the ability of this type of algorithms to deal with non-smooth, non-convex and multi-modal functions. In addition, stochastic optimization does not require any gradients and can be easily parallelized. So far, the most popular stochastic approaches for tackling well placement have been genetic algorithms encoding the real parameters to be optimized as bit-strings. However, it is known in the stochastic algorithm community, that representing real vectors as bit strings leads to poor performance [122]. Recently, a comparison between binary and real representations on a well placement problem in a channelized synthetic reservoir model has been made, showing that the continuous variant outperforms the binary one [33].

This section compares a real-coded GA with CMA-ES on a well placement problem. To allow a fair comparison, both algorithms are used without parameter tuning. Indeed, tuning an algorithm requires some extra objective function evaluations that would need to be taken into account otherwise. Default parameters are used for the CMA-ES algorithm¹ and typical parameter value for the GA.

3.2.1 Well placement using CMA-ES

The initial population is normally drawn using a mean vector uniformly drawn in the reservoir. Parameters were defined according to default settings [71].

The population size λ is an important parameter of CMA-ES [71]. The default population size value equals $4 + \lfloor 3 \times \ln(D) \rfloor$, where D is the dimension of the problem. Independent restarts with increasing population size are suggested in [12]. In this thesis, the optimal tuning of the population size was not addressed. However, due to the difficulty of the problem at hand, we use a population size greater than the default value.

3.2.2 Well placement using GA

Genetic algorithms [78, 79] are stochastic search algorithms that borrow some concepts from nature. Similar to CMA-ES, GAs are based on an initial population of individuals. Each individual represents a possible solution to the problem at hand. Starting with

¹At the exception of the population size where the default setting is known to be good for non-rugged landscapes but needs to be increased otherwise [71].

3.2 CMA-ES and a real-coded GA for the well placement problem

Table 3.2: GA parameters: the probabilities to apply GA operators, i.e., crossover and mutation.

Constant	Value
crossprob	0.7
mutprob	0.1

an initial population of points called individuals or chromosomes, and at each iteration, candidate solutions evolve by selection, mutation and recombination until reaching the stopping criteria with a satisfactory solution. The correspondence between a solution and its representation needs to be defined. In general, simple forms like an array or a matrix of integer or bit elements are used. In this section, individuals are parameterized as defined for CMA-ES (see Section 3.1.2). Hence, well coordinates are defined using a real encoding. Elitism is used to make sure that the best chromosome would survive to the next generation. The used operators are defined as follows:

- The crossover starts with two parent chromosomes causing them to unite in points to create two new elements. The greater chromosome fitness' rank, the higher probability it will be selected. After selecting the two parents, crossover is applied with a probability denoted crossprob. To apply the crossover, we randomly draw an index i between 1 and D and a number c between 0 and 1. Let us denote the two parents by $(x_{1,j})_{1 \leq j \leq D}$ and $(x_{2,j})_{1 \leq j \leq D}$, then $x_{1,i} \leftarrow c \times x_{1,i} + (1 - c) \times x_{2,i}$ and $x_{2,i} \leftarrow c \times x_{2,i} + (1 - c) \times x_{1,i}$.
- The mutation, instead, starts with one individual and randomly changes some of its components. Mutation is applied to all chromosomes, except the one with the best fitness value, with a probability of mutation denoted mutprob. In this case, we randomly draw an index i . Let us denote the selected chromosome by $(x_j)_{1 \leq j \leq D}$, then $x_i \leftarrow \min_i + c \times (\max_i - \min_i)$, where \min_i and \max_i are the minimum and the maximum values that can be taken by the i^{th} coordinate of the chromosome and c is a number randomly drawn between 0 and 1.

The mutation and crossover probabilities are set to typical values (see Table 3.2)¹.

To handle the constraints, the genetic algorithm is combined with the Genocop III technique (Genetic Algorithm for Numerical Optimization of Constrained Problems) [47]. This procedure maintains two separate populations. The first population called the search population contains individuals which can be unfeasible. The second population, the reference population, consists of individuals satisfying all constraints (linear and non-linear), called reference individuals. Feasible search individuals and reference individuals

¹A good choice of the crossover probability is said to be in between 0.4 and 0.9 [128, 39], 0.6 and 0.8 [66], 0.6 and 0.95 [51, 55], 0.6 and 0.8 [120]. A good choice of the mutation probability is said to be in between 0.001 and 0.1 [39, 51, 55], 0.005 and 0.05 [120], 0.05 and 0.1 [128].

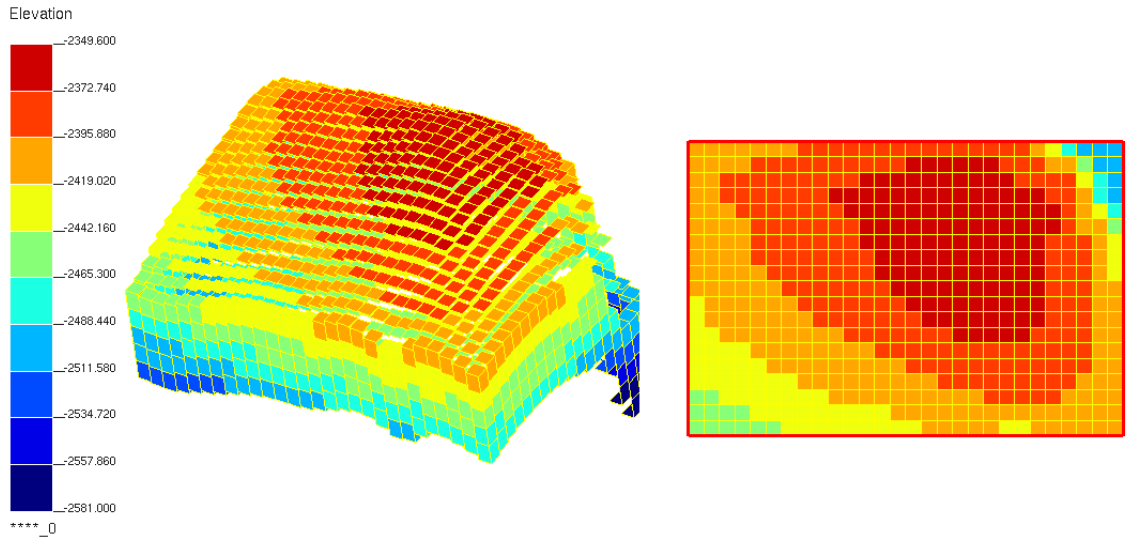


Figure 3.2: Elevation (in meters) and geometry of the PUNQ-S3 test case.

are evaluated directly using the objective function. However, unfeasible individuals are repaired before being evaluated. More details about Genocop III can be found in [97].

3.2.3 Well placement performance

All tests performed in the present chapter are conducted on the PUNQ-S3 test case [54]. PUNQ-S3 is a case taken from a reservoir engineering study on a real field, and qualified as a small-size industrial reservoir model. The model grid contains 19 cells in the x-direction, 28 cells in the y-direction and 5 cells in the z-direction. The cell sizes is 180m in the x and y directions and 18m in the z-direction. We suppose that the field does not contain any production or injection well initially. The elevation of the field and its geometry is shown in Fig. 3.2. We plan to drill two wells: one unilateral injector and one unilateral producer. The dimension of the problem is then equal to $12(= 6 \times 2)$.

To compare results obtained by both CMA-ES and the genetic algorithm, 14 runs were performed for each algorithm. A streamline simulator is used during the optimization. In this comparison, a bottomhole pressure imposed on the producer is fixed to 80 bar, and a bottomhole pressure imposed on the injector is fixed to 6.000 bar which is too high. This unrealistic value was used only for the sake of comparison between the two optimization methods.

The population size is set to 40 for both algorithms. The stopping criterion is also the same for both of the methods: a maximum number of iterations equal to 100. The size of the reference population for Genocop III is set to 60. Well lengths are constrained with a maximum well length $L_{\max} = 1000$ meters.

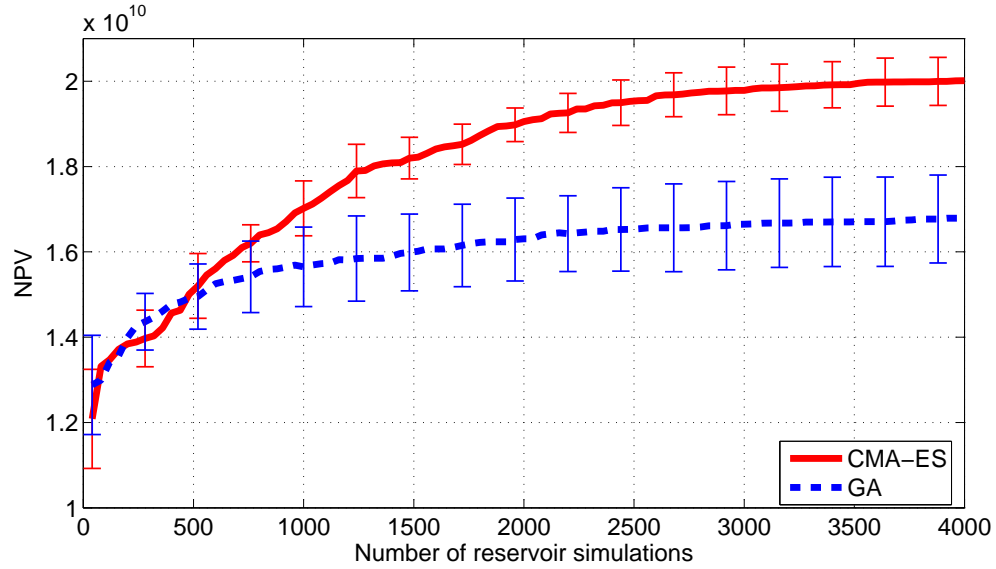


Figure 3.3: The mean value of NPV (in US dollar) and its corresponding standard deviation for well placement optimization using CMA-ES (*solid line*) and GA (*dashed line*). Fourteen runs are performed for each algorithm. Constraints are handled using an adaptive penalization with rejection technique for CMA-ES and using Genocop III for GA.

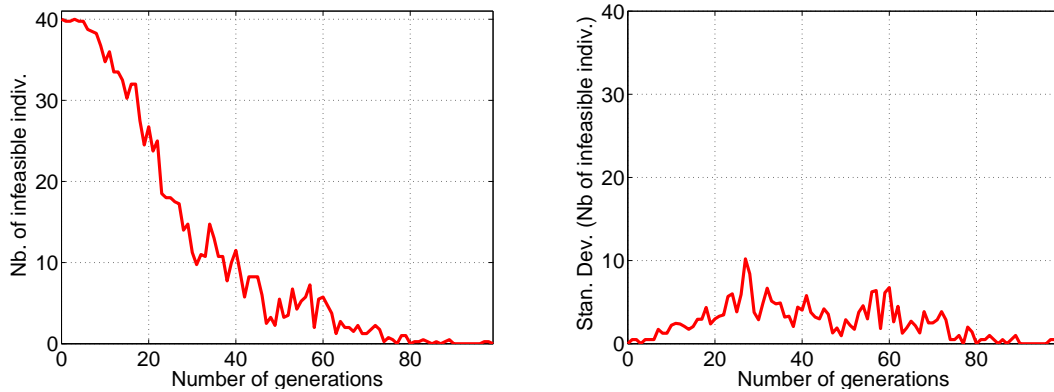


Figure 3.4: The mean number of unfeasible individuals per generation and its corresponding standard deviation using CMA-ES with an adaptive penalization with rejection technique. Here, we consider only unfeasible individuals far from the feasible domain, i.e., resampled individuals.

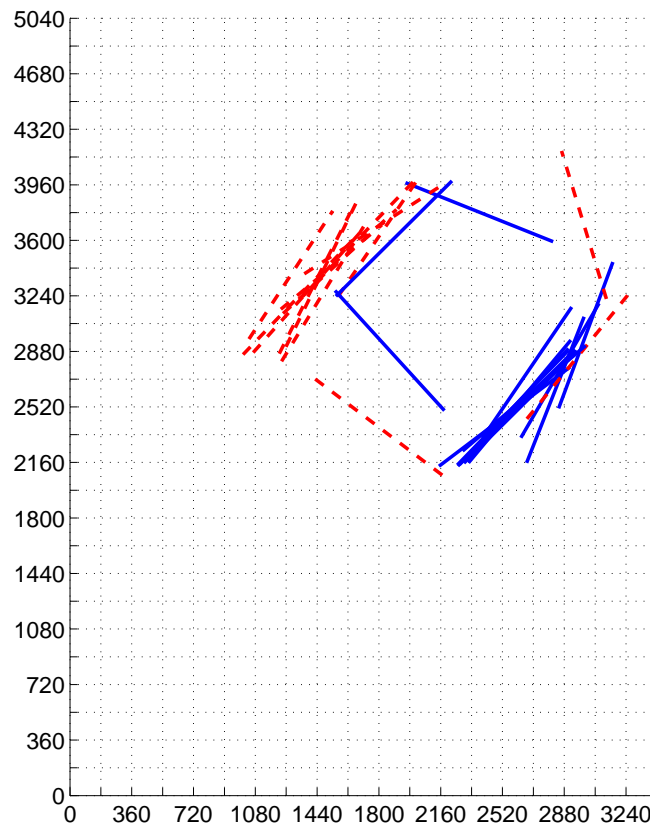


Figure 3.5: The positions of solution wells found by 14 runs of CMA-ES projected on the top face of the reservoir. Injectors are represented by (*dashed line*). Producers are represented by (*solid line*).

Fig. 3.3 shows the average performance and its standard deviation of the well placement optimization using both algorithms measured by the overall best objective function value. It is clear that CMA-ES outperforms the GA: the genetic algorithm adds only 40% to the best NPV obtained by a randomly sampled configuration, i.e., in the first generation of the optimization. However, CMA-ES adds 80%.

Fig. 3.4 shows that CMA-ES handles the used constraints successfully. The number of well configurations resampled, i.e., far from the feasible domain, approaches to 0 at the end of the optimization. Fig. 3.4 shows that after a number of iterations, the majority of the well configurations generated by CMA-ES are either feasible or close to the feasible domain.

Fig. 3.5 shows the positions of “optimum” wells obtained from 14 runs using CMA-ES. CMA-ES succeeds in defining in 11 runs of the 14 performed the same potential zone to place the producer and the injector. This region gives an NPV between $\$1.99 \times 10^{10}$ and

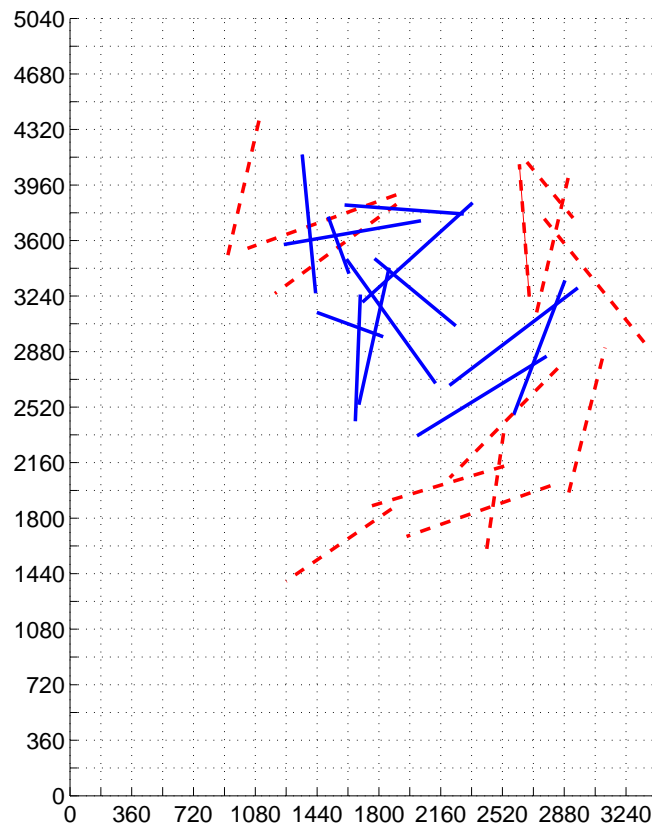


Figure 3.6: The positions of solution wells found by 14 runs of the GA projected on the top face of the reservoir. Injectors are represented by (*dashed line*). Producers are represented by (*solid line*).

3.3 Application of CMA-ES with meta-models on the PUNQ-S3 case

$\$2.05 \times 10^{10}$. In the other three runs, CMA-ES finds each time a different local optimum with NPV values equal to: $\$1.83 \times 10^{10}$, $\$1.95 \times 10^{10}$ and $\$2.05 \times 10^{10}$. Despite the large number of local optima, CMA-ES succeeds in providing satisfactory results on 93 % of the runs, if we consider that a run is satisfactory if it gives an NPV greater or equal to $\$1.95 \times 10^{10}$.

For the genetic algorithm, 14 runs were performed to trace different “optimum” well configurations in Fig. 3.6. Well configurations are not concentrated in some well-defined regions and have an NPV mean value equal to $\$1.68 \times 10^{10}$ with a standard deviation equal to 1.06×10^9 . The GA leads to well configurations dispersed over a large zone. The maximum value of NPV obtained by the GA is equal to $\$1.86 \times 10^{10}$ and it corresponds to a well configuration close to a well configuration obtained by CMA-ES with an NPV $\$2.05 \times 10^{10}$.

Results confirm that CMA-ES is able to find in the majority of the runs a solution in the same potential region. In 93% of the runs on the considered test case, CMA-ES finds a well configuration with a satisfactory NPV value. However, the GA has difficulties to define this potential region and seems to prematurely converge in different regions. Premature convergence in the GA is most certainly due to the lack of mechanisms that (1) would play the role of the step-size mechanism in CMA-ES which is able to increase the step-size in linear environments and (2) would play the role of the covariance matrix adaptation mechanism allowing to adapt the main search directions (elongate / shrink certain directions and learn the principal axis of the problem) to solve efficiently ill-conditioned problems. Without this latter mechanism on ill-conditioned problems, it is common to observe premature convergence.

3.3 Application of CMA-ES with meta-models on the PUNQ-S3 case

In this section we apply CMA-ES with meta-models on the well placement optimization problem. The proposed approach is able to handle different possible well configurations as defined in Section 3.1.2. The use of local meta-models instead of a global one is motivated by the fact that we want the algorithm to be able to handle multi-modal functions or unimodal functions where a global quadratic model would model poorly the function. In the following, we use the variant nlmm-CMA₂ defined in Section 2.3.3.4. For nlmm-CMA₂, $(1 + n_{ic})$ individuals are evaluated for a given generation where n_{ic} is the number of iteration cycles needed to satisfy the meta-model acceptance criterion. In this section, the performance of the approach is demonstrated on two cases.

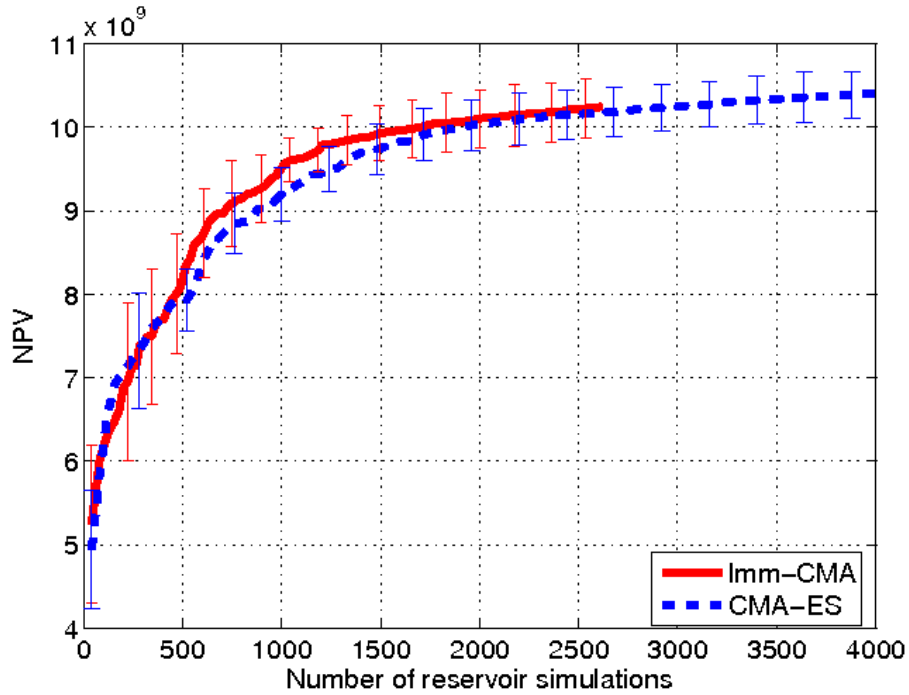


Figure 3.7: The mean value of NPV (in US dollar) and its corresponding standard deviation for well placement optimization using CMA-ES with meta-models (*solid line*) and CMA-ES (*dashed line*). Ten runs are performed for each algorithm. Constraints are handled using an adaptive penalization with rejection technique.

3.3.1 Placement of one unilateral producer and one unilateral injector

In this application, we consider a placement problem of one unilateral injector and one unilateral producer on the PUNQ-S3 case. Parameters of the problem are the same as for the example in Section 3.2.3, except for the following differences:

- a commercial reservoir simulator is used to evaluate field productions of each phase instead of the streamline simulator;
- the bottomhole pressure imposed on the producer is fixed to 150 bar;
- the bottomhole pressure imposed on the injector is fixed to 320 bar.

To define the parameters of the meta-model, we choose k , the number of individuals used to evaluate the meta-model, equal to 100. Meta-models are used when the training set contains at least 160 couples of points with their evaluations. For each method, i.e., CMA-ES and CMA-ES with local meta-models (Imm-CMA), 10 runs were performed. The evolution of the NPV mean value in term of the mean number of reservoir simulations is represented in Fig. 3.7.

Fig. 3.7 shows that, for the same number of reservoir simulations, combining CMA-ES with meta-models allows to reach higher NPV values compared to CMA-ES, given a

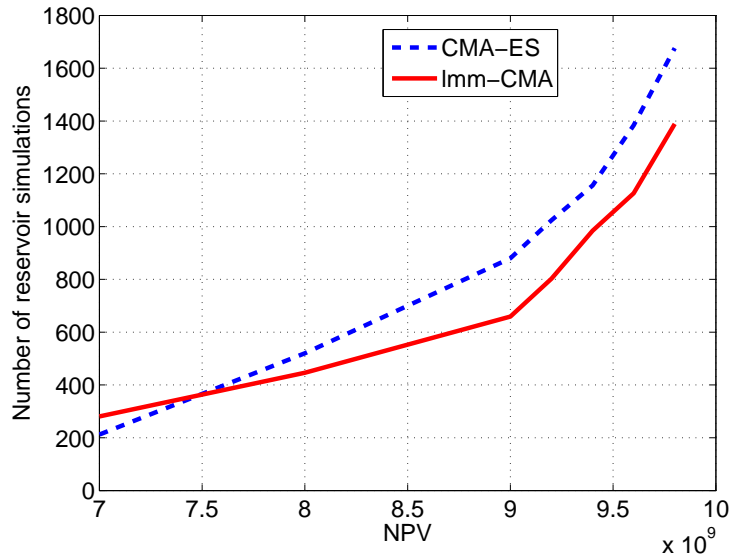


Figure 3.8: The mean number of reservoir simulations needed to reach a given NPV value using CMA-ES with meta-models (*solid line*) and CMA-ES (*dashed line*). Ten runs are performed for each algorithm.

restricted budget of reservoir simulations. A better representation is to show the mean number of reservoir simulations needed to reach a certain value of NPV for CMA-ES and for CMA-ES with meta-models (Fig. 3.8). To reach an NPV value of 9×10^9 , Imm-CMA requires only 659 reservoir simulations, while CMA-ES requires 880 reservoir simulations. If we consider that an NPV equal to 9×10^9 is satisfactory, using meta-models reduces the number of reservoir simulations by 25%. For an NPV value equal to 9.6×10^9 , the use of meta-models reduces the number of reservoir simulations by 19%. Figs. 3.7 and 3.8 highlight the contribution of meta-models in reducing the number of reservoir simulations. Results show also that, in addition to reducing the number of objective function evaluations, the method still succeeds in reaching high NPV values and results are similar to those obtained by CMA-ES. As for the example in Section 3.2.3, the well placement optimization still succeeds in identifying in the majority of the runs the same potential region to contain optimum wells. In the following, we present detailed results obtained only by one of the solution well configurations proposed by Imm-CMA. The selected solution well configuration is denoted *optimized config* in the sequel. *Optimized config* is then compared to two configurations designed after some trials in a way to represent the decision of a reservoir engineer (denoted *config.1* and *config.2*). The locations and trajectories of the considered well configurations are shown in Fig. 3.9.

The engineer’s proposed configurations were defined according to the SoPhiH map (Fig. 3.9) which represents the distribution of the hydrocarbon pore volume over the n_{layers}

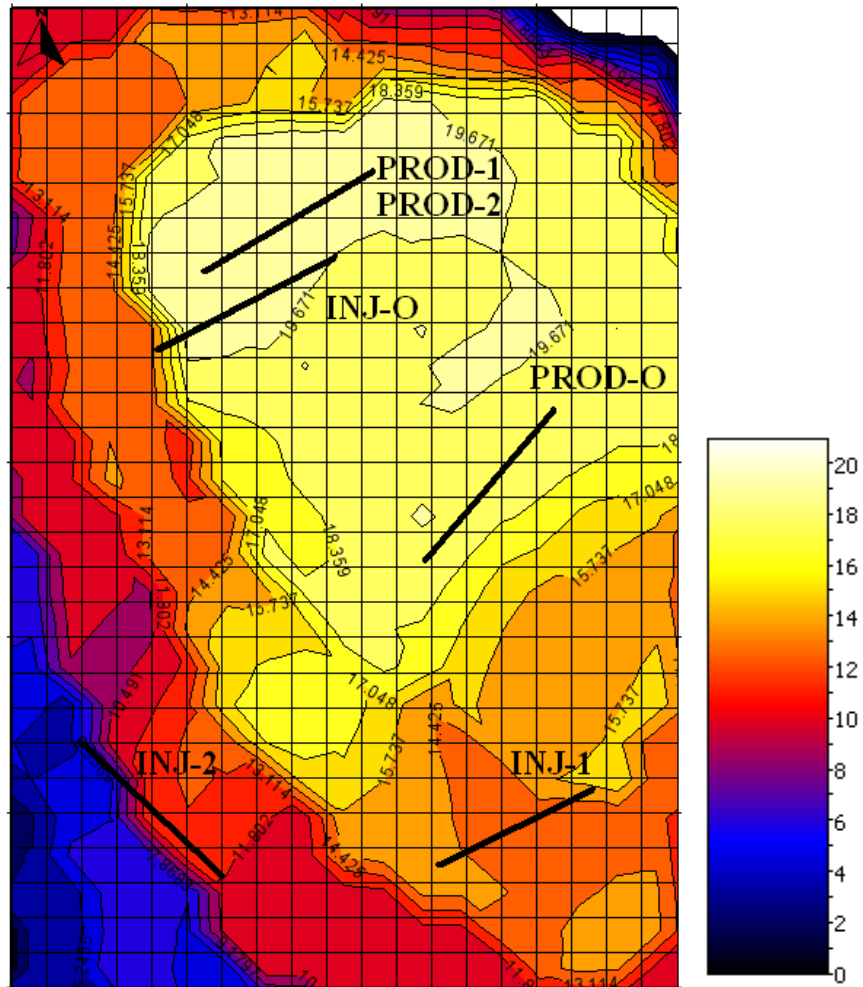


Figure 3.9: The SoPhiH map, with solution well configuration obtained using CMA-ES with meta-models (PROD-O, INJ-O) and two engineer's proposed well configurations (PROD-1, INJ-1 and PROD-2, INJ-2).

3.3 Application of CMA-ES with meta-models on the PUNQ-S3 case

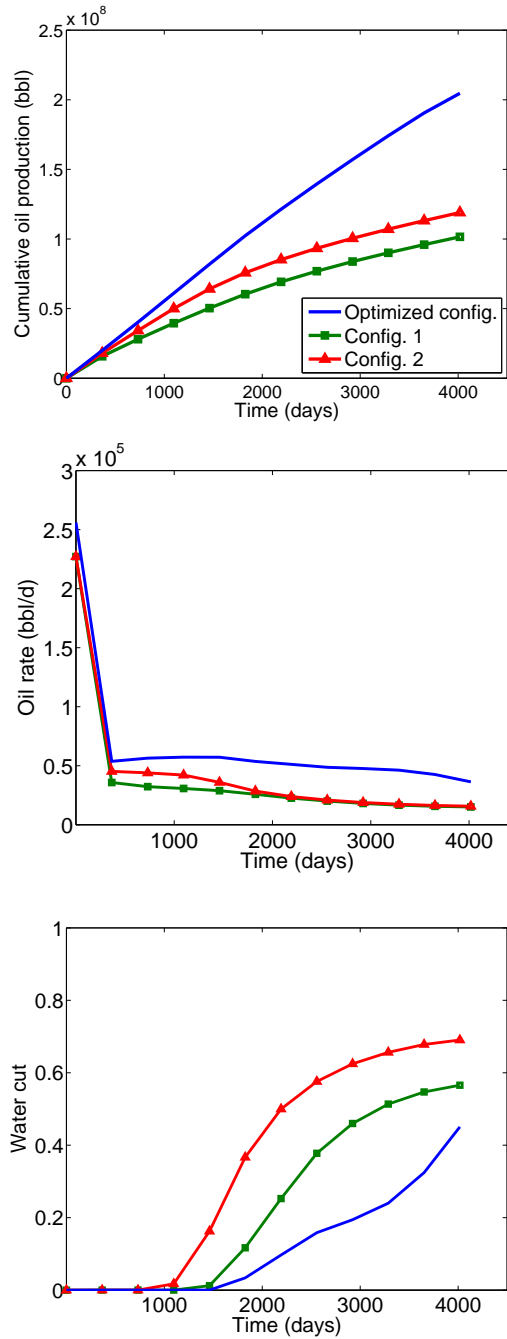


Figure 3.10: Production curves for an optimized solution using CMA-ES with meta-models (*optimized config.*) and two engineer's proposed configurations (*config.1* and *config.2*).

layers defined by $\sum_{k=1}^{n_{\text{layers}}} (\mathbf{H}_k \times \phi \times S_o)$, where \mathbf{H}_k is the gross thickness of the layer k , S_o is the oil saturation and ϕ is the porosity. PROD- c and INJ- c denote respectively the producer and the injector corresponding to the well configuration c . The well configuration is either *config.1*, *config.2* or *optimized config* denoted respectively 1, 2, O. Engineer’s proposed wells are horizontal wells where producers (PROD-1 = PROD-2) are placed in the top layer ($k = 1$) and injectors in the bottom layer ($k = 5$). However, producers and injectors in *optimized config* are inclined wells placed in the layer ($k = 3$). The engineer’s proposed producer is placed in the region with the highest SoPhiH value.

Fig. 3.10 shows the production curves of the considered well configurations. The cumulative oil production for *optimized config*, during the 11 simulated years equals 205 MMbbl. However, *config.1* offers only 119 MMbbl and *config.2* offers 102 MMbbl. Therefore, the optimization methodology adds 72% to the best considered engineer’s proposed well configuration. *Optimized config* offers also the smallest water cut (0.45 for *optimized config*, 0.57 for *config.1* and 0.69 for *config.2*).

3.3.2 Placement of one multi-segment producer

In this application, we consider a placement problem of one multi-segment well on the PUNQ-S3 case. We suppose that an injector is already placed in the reservoir. It corresponds to the well denoted INJ-O in Fig. 3.9. We plan to drill a multi-segment well with two completed segments. The dimension of the problem is then equal to 9(= 6 + 3). The different parameters of the problem are the same as in the example in Section 3.3.1, except for the population size which is equal to 30. Ten runs were performed with a maximum number of iterations equal to 100.

Fig. 3.11 shows the evolution of the average performance of the well placement, i.e., NPV mean values and the corresponding standard deviation. Optimizing the placement of one multi-segment producer offers an NPV equal to $\$1.10 \times 10^9 \pm 4.37 \times 10^7$. To reach an NPV mean value of $\$1.10 \times 10^9$, the optimization process requires only 504 reservoir simulations.

The positions of solution wells are shown in Figs. 3.12 and 3.13. In this application, the used methodology succeeds in reaching NPV values greater than $\$1.09 \times 10^9$ and in defining an “optimum” well configuration in the same potential region for all the performed runs. Therefore, performing only one run can be conclusive and can ensure converging to a solution well with a satisfactory NPV.

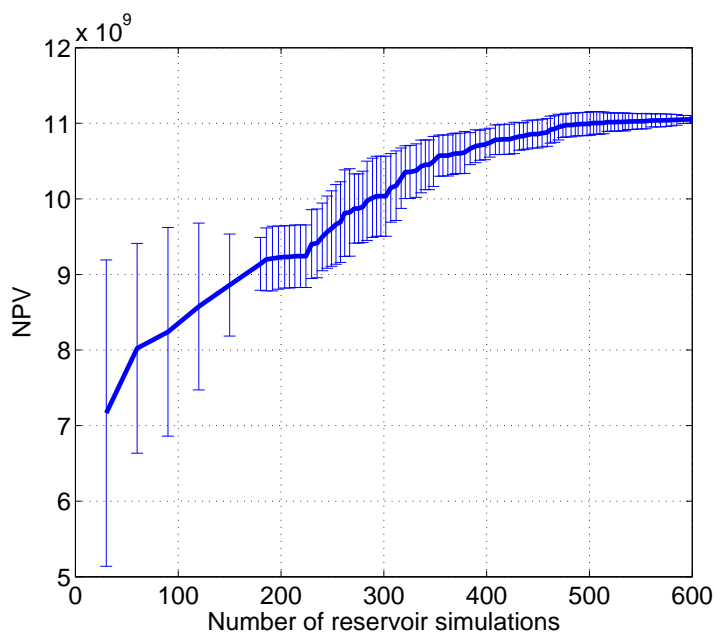


Figure 3.11: The mean value of NPV (in US dollar) and its corresponding standard deviation for well placement optimization using CMA-ES with meta-models of one multi-segment well. Ten runs are performed.

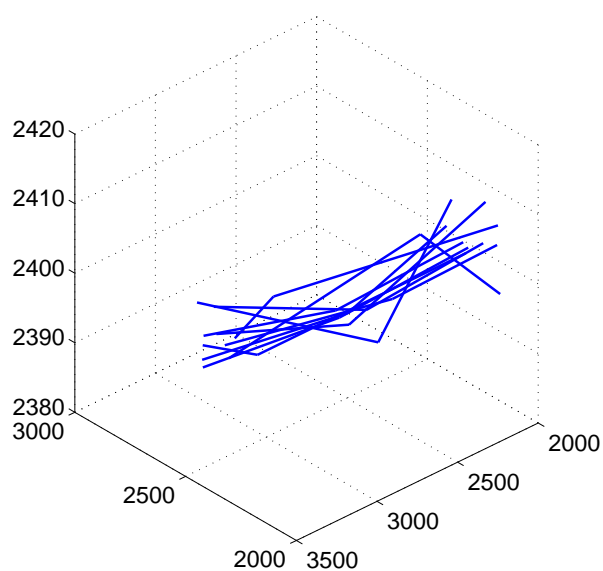


Figure 3.12: The positions of solution multi-segment producers found by 10 runs of CMA-ES with meta-models. A zoom on the region containing the solution wells is performed.

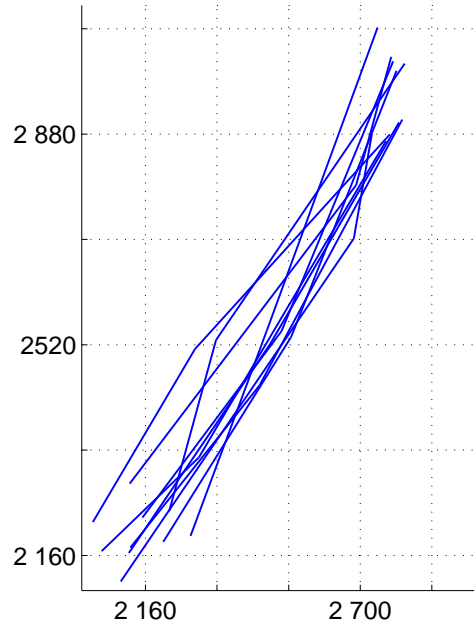


Figure 3.13: The positions of solution multi-segment producers found by 10 runs of CMA-ES with meta-models projected on the top face of the reservoir. A zoom on the region containing the solution wells is performed.

3.4 Summary and discussions

In this chapter, the stochastic optimization method CMA-ES was applied on a well placement problem. A technique based on adaptive penalization with rejection was developed to handle well placement constraints with CMA-ES. Results showed that this technique ensures that after a number of iterations, the majority of well configurations generated by CMA-ES are either feasible or close to the feasible domain. The optimization with CMA-ES was compared to a GA which is the most popular method used in well placement optimization in the literature. Both algorithms were used without parameter tuning allowing for a direct fair comparison of the results. Indeed parameter tuning requires extra function evaluations that should be taken into account when presenting comparison results. In addition, we think that parameter tuning should be done by the designer of the algorithm and not the user as it is unrealistic to waste expensive function evaluations for correcting the weakness of the design phase. The CMA-ES example shows that providing parameter-free algorithms with robust setting is possible to achieve. CMA-ES was shown to outperform the genetic algorithm on the PUNQ-S3 case by leading to a higher net present value (NPV). Moreover, CMA-ES was shown to be able to define potential regions containing optimal well configurations. On the other hand, the genetic algorithm converged to solutions located in different regions for every performed run. In addition

those solutions are associated to much smaller NPV values than the solutions found by CMA-ES. On the PUNQ-S3 case, the mean NPV value found by GA is $\$1.68 \times 10^{10}$. However, the mean NPV value found by CMA-ES is $\$2.01 \times 10^{10}$. The ability of CMA-ES to find much higher NPV values and to converge to the same region of the search space, has been explained by its advanced adaptation mechanism that allows the algorithm, on ill-conditioned non-separable problems, to adapt in an efficient way its sampling probability distribution.

To tackle the computational issue related to the number of reservoir simulations performed during the optimization, an application of nlmm-CMA algorithm is demonstrated on the PUNQ-S3 case. The use of meta-models was shown to offer similar results (solution well configurations and the corresponding NPV values) as CMA-ES without meta-models and moreover to reduce the number of simulations by 19-25% to reach a satisfactory NPV. The comparison of the obtained results with some engineer's proposed well configurations showed that the proposed optimization methodology is able to provide better well configurations in regions that might be difficult to determine by reservoir engineers.

The results presented in this chapter has demonstrated the potential huge benefit of applying the CMA-ES methodology over more established stochastic techniques for reservoir applications.

Chapter 4

Local-meta-model CMA-ES for partially separable functions

This chapter is based on the paper [23]. In this chapter, we propose a new variant of the covariance matrix adaptation evolution strategy with local meta-models for optimizing partially separable functions. We propose to exploit partial separability by building at each iteration a meta-model for each element function (or *sub-function*) using a full quadratic local model. The performance of the proposed algorithm is shown on a number of mathematical test functions.

This chapter is structured as follows. Section 4.1 defines a general notion of partial separability. In Section 4.2, we propose a new variant of CMA-ES with meta-models for partially separable functions. The performance of this variant is evaluated in Section 4.3 on a number of partially separable test functions. The choice of the number of points used to build the meta-model is also described and the computational cost is discussed.

In the following, we denote the objective function to be optimized by $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

4.1 Partial separability and problem modeling

A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is partially separable if it can be written as a sum of sub-functions, also called *element functions*, each depending on a fewer number of variables. Often the particular case where each sub-function depends on a subset of variables of the original function is defined as partial separability. For instance the Rosenbrock function in Table 4.1 writes:

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} h(x_i, x_{i+1}) , \quad (4.1)$$

where $\mathbf{x} = (x_i)_{1 \leq i \leq n}$ and $h(x_i, x_{i+1}) = \alpha(x_i^2 - x_{i+1})^2 + (x_i - 1)^2$ and is thus partially separable with each sub-function depending on the subset of variables $[(x_i, x_{i+1})]_{i=1, \dots, n-1}$. This particular case of partially separable function is considered for instance in [21, 38, 45].

A more general definition, given in [102], considers that each sub-function can depend on a number of variables that are a *linear combination* of a subset of variables.

In this thesis we consider a generalization of the previous definitions allowing non-linear combinations of the subset of variables. More precisely a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is said partially separable if there exists an integer $N > 1$, a set of integers $(n_i)_{1 \leq i \leq N}$ with $n_i < n$, for all $i = 1, \dots, N$, a set of explicit functions $(\Phi^i : \mathbb{R}^n \rightarrow \mathbb{R}^{n_i})_{1 \leq i \leq N}$ and a set of functions $(f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R})_{1 \leq i \leq N}$, such that f can be written as $f(\mathbf{x}) = \sum_{i=1}^N f_i(\Phi^i(\mathbf{x}))$. The sub-functions or element functions $(f_i)_{1 \leq i \leq N}$ depend on a number n_i of parameters called *element variables*. The functions Φ^i will be called *mapping functions*. Note that the setting of [102] is recovered by taking $\Phi^i = U^i$ where U^i is a linear mapping from \mathbb{R}^n to \mathbb{R}^{n_i} .

For a given partially separable function, there exists “theoretically” an infinite number of ways to define the element functions and mapping functions. However, one has usually a restricted knowledge about the structure of the problem that determines the modeling choice. We can argue that we only know in general that the problem can be decomposed as a sum of element functions depending on fewer variables, and that there is thus no reason to encode non-linearity in the variable dependencies. However, a motivating example for our general definition is the well placement optimization problem, in which we will show in Chapter 5 that a suitable way to model the objective function is to suppose that the profit corresponding to a given well depends only on its location and on the distances of this well to the others. Using the distances between the wells as an element variable implies using a nonlinear combination of the parameters of the problem (see Chapter 5).

In the well placement problem also, the objective function is computed using a numerical software (reservoir simulator) able to simulate for a given set of well placements the quantity of oil, water and gas that can be extracted from *each well*. Consequently one has access to the function value of *each element function*. In the following we will also assume not only that the function is partially separable but also that *one has access to the function value of each element function*. As argued above this assumption is reasonable as it models the case for the well placement problem. History matching is another problem in petroleum engineering in which this assumption is reasonable. In history matching problems, we want to adjust the reservoir model until it closely reproduces the past behavior of the reservoir (historical production and pressures). For this problem also, we can define the objective function as a sum of a number of sub-functions defined for each well and calculated when evaluating the objective function [44].

Exploiting partial separability or separability is a common approach to enhance performances of optimization algorithms, in particular when dealing with large scale optimization. For instance a trust region algorithm for minimizing partially separable functions

Table 4.1: Test functions. For the block-rotated ellipsoid, \mathbf{Q} is a 2×2 rotation matrix with each column being a uniformly distributed unit vector.

Name	Function
Rosenbrock	$f_{\text{Rosen}}^\alpha(\mathbf{x}) = \sum_{i=1}^{n-1} \left(\alpha \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right)$
Rosenbrock $^{\frac{1}{2}}$	$f_{\text{Rosen}^{\frac{1}{2}}}^\alpha(\mathbf{x}) = \sum_{i=1}^{n-1} \left(\alpha \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right)^{\frac{1}{2}}$
Block-rotated ellipsoid 2D	$f_{\text{BlockElli-2D}}^\alpha(x, y) = \sum_{i=1}^2 \left(\alpha^{\frac{i-1}{n-1}} \cdot (\mathbf{Q} \times (x, y))^2 \right)$
Block-rotated ellipsoid	$f_{\text{BlockElli}}^\alpha(\mathbf{x}) = \sum_{i=1}^{n-1} (f_{\text{BlockElli-2D}}^\alpha(x_i, x_{i+1}))$

was proposed in [38]. Separability was also exploited within CMA-ES. A method where the covariance matrix was constrained to be diagonal has been proposed in [109].

4.2 lmm-CMA for partially separable functions

This section introduces a new algorithm based on nlmm-CMA and exploiting the partial separability of the objective function. This algorithm will be called the partially separable local-meta-model CMA-ES (p-sep lmm-CMA).

In our proposed approach, the partial separability of the objective function is exploited when building the meta-models. The optimization process defined by CMA-ES is not altered. The idea behind exploiting the problem structure when building the meta-model, is to improve the quality of the approximate model. Hence, the better the quality of the model is, the easier the acceptance criteria can be satisfied, the less evaluations are performed.

Let us consider a partially separable function f . As in Section 4.1, we consider that f has N element functions $(f_i)_{1 \leq i \leq N}$. For each element function, we associate a mapping function Φ^i such that $f(\mathbf{x}) = \sum_{i=1}^N f_i \circ \Phi^i(\mathbf{x})$. We suppose that when evaluating a point \mathbf{x} on f , we have access to the evaluations $(f_i \circ \Phi^i(\mathbf{x}))_{1 \leq i \leq N}$ as well.

In Chapter 2, an approximate function \hat{f} for a given objective function f is defined using a locally weighted regression based on the training set containing both evaluated points and their values on f . In this chapter, we propose to build a meta-model for each element function f_i that we denote by \hat{f}_i . The meta-model \hat{f} of f is then defined by:

$$\hat{f} = \sum_{i=1}^N \hat{f}_i \circ \Phi^i . \quad (4.2)$$

The meta-model \hat{f}_i of each element function f_i is built in a way quite similar to the

meta-model \hat{f} of f defined for the (n)lmm-CMA in Section 2.3.1.1. The training set is built by storing for every evaluated point \mathbf{x} , $\Phi^i(\mathbf{x})$ and its corresponding values on f_i , i.e., $f_i(\Phi^i(\mathbf{x}))$. Let us consider an individual \mathbf{q} for which $\Phi^i(\mathbf{q}) \in \mathbb{R}^{n_i}$ has to be evaluated on the approximate model of f_i . Assuming that the training set contains a sufficient number m_i of elements, we select the $k_i \in \mathbb{N}$ nearest points $(\Phi^i(\mathbf{x}_j), j = 1, \dots, k_i)$ to $\Phi^i(\mathbf{q})$ using the Mahalanobis distance d_i with respect to a matrix \mathbf{C}_i , defined for a given point $\mathbf{z} \in \mathbb{R}^n$ as:

$$d_i(\Phi^i(\mathbf{z}), \Phi^i(\mathbf{q})) = \sqrt{(\Phi^i(\mathbf{z}) - \Phi^i(\mathbf{q}))^T \mathbf{C}_i^{-1} (\Phi^i(\mathbf{z}) - \Phi^i(\mathbf{q}))}, \quad (4.3)$$

where \mathbf{C}_i is an $n_i \times n_i$ matrix adapted to the local shape of the landscape of f_i (see below).

Similarly to Section 2.3.1.1, a full quadratic meta-model is used. Using a vector $\beta_i \in \mathbb{R}^{\frac{n_i(n_i+3)}{2}+1}$, \hat{f}_i is defined for a given point $\mathbf{z} \in \mathbb{R}^n$, for which we denote $\Phi^i(\mathbf{z}) = (\tilde{u}_1, \dots, \tilde{u}_{n_i})$ as:

$$\hat{f}_i(\Phi^i(\mathbf{z}), \beta_i) = \beta_i^T \tilde{\mathbf{z}}_i^T, \quad (4.4)$$

where $\tilde{\mathbf{z}}_i = (\tilde{u}_1^2, \dots, \tilde{u}_{n_i}^2, \tilde{u}_1\tilde{u}_2, \dots, \tilde{u}_{n_i-1}\tilde{u}_{n_i}, \tilde{u}_1, \dots, \tilde{u}_{n_i}, 1)$. The full quadratic meta-model is built by minimizing the following criterion with respect to β_i :

$$B(\mathbf{q}) = \sum_{j=1}^{k_i} \left[\left(\hat{f}_i(\Phi^i(\mathbf{x}_j), \beta_i) - f_i(\Phi^i(\mathbf{x}_j)) \right)^2 \times K \left(\frac{d_i(\Phi^i(\mathbf{x}_j), \Phi^i(\mathbf{q}))}{h} \right) \right]. \quad (4.5)$$

$K(\cdot)$ is the kernel weighting function defined as in Section 2.3.1.1 by $K(\zeta) = (1 - \zeta^2)^2$, and h is the bandwidth defined by the distance d_i of the k_i^{th} nearest neighbor data point to \mathbf{q} . For a given element function, k_i must be greater or equal to $k_{i,\min} = \frac{n_i(n_i+3)}{2} + 1$. k_i is chosen to be equal to $2 \times k_{i,\min}$. The choice of k_i will be discussed in Section 4.3.3. The sufficient size of the training set denoted above by m_i must be then greater or equal to k_i .

Hence, the approximate function of f which corresponds to $\hat{f}(\mathbf{x}) = \sum_{i=1}^N \hat{f}_i(\Phi^i(\mathbf{x}))$ is incorporated into CMA-ES using the approximate ranking procedure as detailed in Section 2.3.

It remains now to describe how the matrices $(\mathbf{C}_i)_{1 \leq i \leq N}$ are obtained. They are built in an iterative manner. At each iteration, after the approximate ranking procedure, each of the λ candidate solutions denoted $(\mathbf{X}_m)_{1 \leq m \leq \lambda}$ and sampled according to Eq. (2.2) has been either evaluated on f or has an associated approximate meta-models value given by Eq. (4.2). Thus for each i , the vectors $\Phi^i(\mathbf{X}_m) \in \mathbb{R}^{n_i}$ have either been evaluated on f_i or have an associated estimate of f_i provided by \hat{f}_i . We then consider the vectors $\Phi^i(\mathbf{X}_m) \in \mathbb{R}^{n_i}$ for $1 \leq m \leq \lambda$ and rank them according to \tilde{f}_i where \tilde{f}_i equals f_i if \mathbf{X}_m was evaluated on f and \hat{f}_i otherwise. The ordered μ best solutions according to \tilde{f}_i are used as input variables in Algorithm 1, to update the covariance matrix \mathbf{C}_i .

Algorithm 1: CMA-Update($\mathbf{x}_1, \dots, \mathbf{x}_\mu$)

1. given parameters $(\omega_i)_{1 \leq i \leq \mu}$, c_σ , c_c , c_{cov} , μ_{cov} , d_σ . Set $\mu_{\text{eff}} = 1 / \sum_{i=1}^{\mu} \omega_i^2$
 2. given $\mathbf{m} \in \mathbb{R}^n$, $\mathbf{p}_\sigma \in \mathbb{R}^n$, $\mathbf{p}_c \in \mathbb{R}^n$, $\sigma \in \mathbb{R}$ and $\mathbf{C} \in \mathbb{R}^{n \times n}$ from last iteration
 3. $\mathbf{m}^- \leftarrow \mathbf{m}$
 4. $\mathbf{m} \leftarrow \sum_{i=1}^{\mu} \omega_i \mathbf{x}_i$
 5. $\mathbf{p}_\sigma \leftarrow (1 - c_\sigma) \mathbf{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma)} \mu_{\text{eff}} \mathbf{C}^{-\frac{1}{2}} \frac{\mathbf{m} - \mathbf{m}^-}{\sigma}$
 6. $\mathbf{p}_c \leftarrow (1 - c_c) \mathbf{p}_c + \sqrt{c_c(2 - c_c)} \mu_{\text{eff}} \frac{\mathbf{m} - \mathbf{m}^-}{\sigma}$
 7. $\mathbf{C}_\mu = \sum_{i=1}^{\mu} \omega_i \frac{(\mathbf{x}_i - \mathbf{m}^-)(\mathbf{x}_i - \mathbf{m}^-)^\top}{\sigma^2}$
 8. $\sigma \leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma\|}{\mathbb{E}\|\mathcal{N}(0, \mathbf{I})\|} - 1\right)\right)$
 9. $\mathbf{C} \leftarrow (1 - c_{\text{cov}}) \mathbf{C} + \frac{c_{\text{cov}}}{\mu_{\text{cov}}} \mathbf{p}_c \mathbf{p}_c^\top + c_{\text{cov}} \left(1 - \frac{1}{\mu_{\text{cov}}}\right) \times \mathbf{C}_\mu$
-

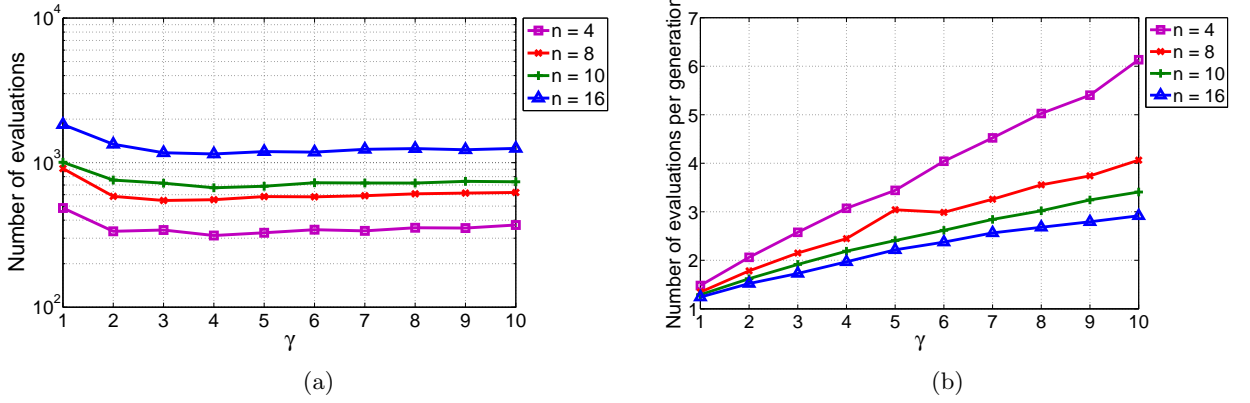


Figure 4.1: (a) Average number of evaluations of the p-sep lmm-CMA on f_{Rosen}^{100} to reach f_{stop} for varying population sizes $\lambda = \gamma \times \lambda_{\text{default}}$. (b) Average number of evaluations per generation of the p-sep lmm-CMA on f_{Rosen}^{100} for varying population sizes $\lambda = \gamma \times \lambda_{\text{default}}$.

In Algorithm 1, the parameters $(\omega_i)_{1 \leq i \leq \mu}$, c_σ , c_c , c_{cov} , μ_{cov} , d_σ are chosen with default values as defined in [71]. Initial values for \mathbf{p}_σ , \mathbf{p}_c and \mathbf{C} used in Algorithm 1 are also set to default as in [71]. Initial values for \mathbf{m} and σ are set to $\Phi^i(\mathbf{m}^{(0)})$ and $\sigma^{(0)}$ where $\mathbf{m}^{(0)}$ and $\sigma^{(0)}$ are the initial mean vector and step-size of (n)lmm-CMA. The idea behind this adaptation procedure is the same as the one of the adaptive encoding proposed in [68]. However in adaptive encoding, step-size update is not needed and different normalizations for the weights depending on the step-length are introduced. Though we believe that the adaptive encoding update is more robust numerically, it has not been tested for this thesis.

Table 4.2: Modeling of the partially separable functions tested.

Name	n_M	N	$f_i(\mathbf{u} = (u_j)_{1 \leq j \leq n_M})$	$\Phi_i(\mathbf{v} = (v_j)_{1 \leq j \leq n})$
Rosenbrock	2	$(n-1)$	$f_i(\mathbf{u}) = \alpha \cdot (u_1^2 - u_2)^2 + (u_1 - 1)^2$	$\Phi_i(\mathbf{v}) = (v_i, v_{i+1})$
	4	$\frac{n-1}{3}$	$f_i(\mathbf{u}) = \alpha \cdot (u_1^2 - u_2)^2 + (u_1 - 1)^2$ $+ \alpha \cdot (u_2^2 - u_3)^2 + (u_2 - 1)^2$ $+ \alpha \cdot (u_3^2 - u_4)^2 + (u_3 - 1)^2$	$\Phi_i(\mathbf{v}) = (v_{3i-2}, v_{3i-1}, v_{3i}, v_{3i+1})$
Rosenbrock $^{\frac{1}{2}}$	2	$(n-1)$	$f_i(\mathbf{u}) = \left(\alpha \cdot (u_1^2 - u_2)^2 + (u_1 - 1)^2 \right)^{\frac{1}{2}}$	$\Phi_i(\mathbf{v}) = (v_i, v_{i+1})$
Block-rotated ellipsoid	2	$(n-1)$	$f_i(\mathbf{u}) = f_{\text{BlockElli-2D}}^\alpha(u_1, u_2)$	$\Phi_i(\mathbf{v}) = (v_i, v_{i+1})$

4.3 Evaluation of p-sep lmm-CMA

In this section we describe the functions used to evaluate p-sep lmm-CMA. We show the performance of this method compared to CMA-ES. The optimal bandwidth used to build the meta-model is also investigated and the computational cost of the approach is discussed.

4.3.1 Test functions

The p-sep lmm-CMA is evaluated on the partially separable test functions f_{Rosen}^1 , f_{Rosen}^{100} , f_{Rosen}^{10000} , f_{Rosen}^{100} , $f_{\text{Rosen}}^{\frac{1}{2}}$ and $f_{\text{BlockElli}}$ defined in Table 4.1. For the block-rotated ellipsoid, \mathbf{Q} is a 2×2 rotation matrix sampled uniformly anew for every run performed. The performance of the method is measured using the success performance SP1 defined as the average number of evaluations for successful runs divided by the ratio of successful runs, needed to reach a stopping objective value $f_{\text{stop}} = 10^{-10}$, except for $f_{\text{Rosen}}^{\frac{1}{2}}$ for which $f_{\text{stop}} = 10^{-5}$. We perform 20 independent runs to measure SP1. The runs are randomly initialized in the intervals $[-5, 5]$ for f_{Rosen}^1 , f_{Rosen}^{100} , f_{Rosen}^{10000} and $f_{\text{Rosen}}^{\frac{1}{2}}$ and $[-10, 10]$ for $f_{\text{BlockElli}}$. Each test function is modeled by defining a number N of element functions, a number n_M of element variables for each element function, a set of element functions denoted by $f_i : \mathbb{R}^{n_M} \rightarrow \mathbb{R}$ and a set of mapping functions $\Phi_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n_M}$, such that $f = \sum_{i=1}^N f_i \circ \Phi_i$. The modeling of each test function is shown in Table 4.2. The block-rotated ellipsoid function is defined using quadratic element functions. For the other tested functions, the defined element functions are not quadratic.

4.3.2 Performance of p-sep lmm-CMA

Results on the test functions are presented in Table 4.3 showing the performance of p-sep lmm-CMA compared to CMA-ES and to some tests with nlmm-CMA. For each test, by defining the value of n_M , we refer to the corresponding modeling defined in Table 4.2. It is

4.3 Evaluation of p-sep lmm-CMA

Table 4.3: Success performance SP1, i.e., the average number of function evaluations for successful runs divided by the ratio of successful runs, standard deviations of the number of function evaluations for successful runs and speedup performance spu, to reach $f_{\text{stop}} = 10^{-10}$ of p-sep lmm-CMA, nlmm-CMA and CMA-ES (for $f_{\text{Rosen}\frac{1}{2}}^{100}$, $f_{\text{stop}} = 10^{-5}$). The ratio of successful runs is denoted between brackets if it is < 1.0 . The number of element variables of each element function is denoted by n_M .

Function	n	n_M	λ	p-sep lmm-CMA	spu	nlmm-CMA	spu	CMA-ES
f_{Rosen}^1	4	2	8	189 \pm 13	5.1	297 \pm 20	3.2	964 \pm 192
	8	2	10	308 \pm 20	6.5	932 \pm 52	2.2	2006 \pm 118
	10	2	10	353 \pm 20	6.8	1482 \pm 169	1.6	2418 \pm 204
	16	2	12	465 \pm 20	8.6			4023 \pm 310
	20	2	12	548 \pm 34	9.1			4978 \pm 374
	32	2	14	755 \pm 32	10.3			7777 \pm 347
	40	2	15	871 \pm 41	11.2			9799 \pm 602
f_{Rosen}^{100}	4	2	8	485 \pm 47 [0.80]	4.7	647 \pm 67 [0.95]	3.5	2269 \pm 254 [0.85]
	8	2	10	910 \pm 71 [0.80]	6.5	2602 \pm 264 [0.85]	2.3	5883 \pm 727 [0.90]
	10	2	10	1006 \pm 99 [0.95]	7.6	3727 \pm 300 [0.90]	2.1	7644 \pm 765 [0.95]
	16	2	12	1834 \pm 117 [0.90]	8.6			15781 \pm 1360 [0.85]
	16	4	12	7162 \pm 1112 [0.95]	2.2			15781 \pm 1360 [0.85]
	20	2	12	2533 \pm 361 [0.90]	10.4			26366 \pm 3249 [0.85]
	32	2	14	4628 \pm 144 [0.95]	13.2			60948 \pm 2668 [0.90]
	40	2	15	6527 \pm 226 [0.95]	15.2			99346 \pm 3502 [0.85]
f_{Rosen}^{10000}	4	2	8	1333 \pm 238 [0.95]	5.3	2637 \pm 715 [0.90]	2.7	7032 \pm 944 [0.90]
	8	2	10	2745 \pm 246	6.6	10287 \pm 468 [0.85]	1.8	18216 \pm 1683 [0.95]
	10	2	10	5552 \pm 429 [0.75]	4.5	16280 \pm 843 [0.85]	1.5	25037 \pm 3160 [0.95]
	16	2	12	10583 \pm 398 [0.80]	5.9			62903 \pm 4441 [0.90]
	20	2	12	14749 \pm 431 [0.90]	6.3			93545 \pm 6566 [0.95]
$f_{\text{Rosen}\frac{1}{2}}^{100}$	4	2	8	544 \pm 48 [0.70]	4.8	909 \pm 75 [0.75]	2.9	2620 \pm 342 [0.95]
	8	2	10	1008 \pm 67 [0.80]	7.0	2549 \pm 262 [0.95]	2.8	7006 \pm 762
	10	2	10	1299 \pm 178 [0.95]	10.4	4685 \pm 518 [0.90]	2.9	13517 \pm 1288 [0.75]
	16	2	12	3346 \pm 223 [0.90]	9.9			33154 \pm 3568 [0.90]
	20	2	12	6797 \pm 878 [0.85]	10.0			68136 \pm 5363 [0.80]
	32	2	14	20751 \pm 2116 [0.85]	14.6			302039 \pm 40915 [0.65]
$f_{\text{BlockElli}}^{10000}$	4	2	8	226 \pm 11	6.6			1500 \pm 89
	8	2	10	392 \pm 14	8.2			3220 \pm 196
	10	2	10	472 \pm 17	8.7			4093 \pm 173
	16	2	12	670 \pm 37	9.8			6566 \pm 284

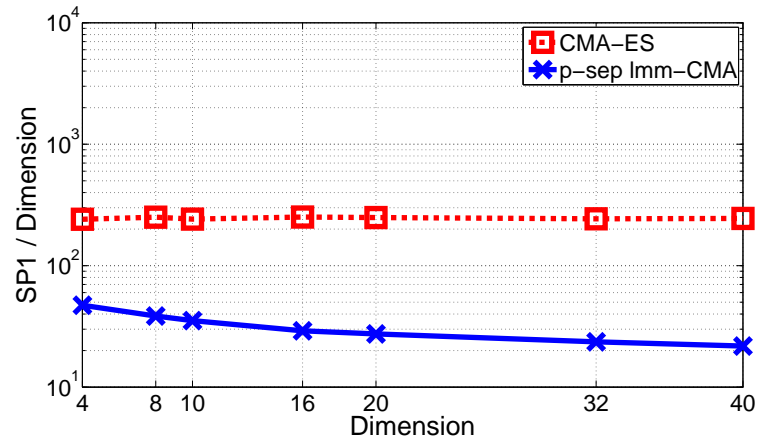
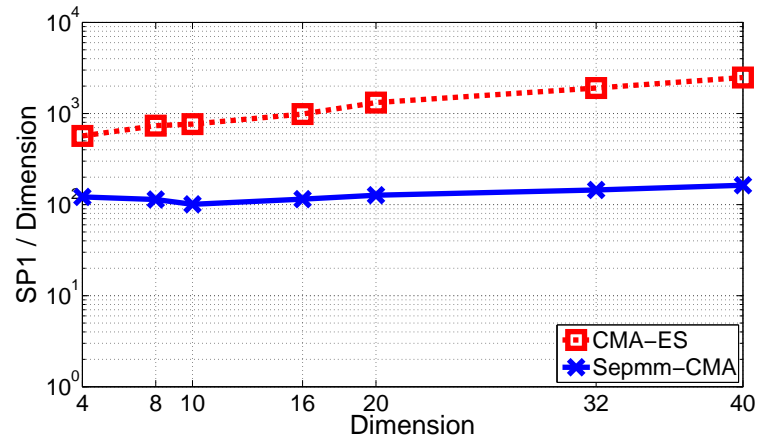
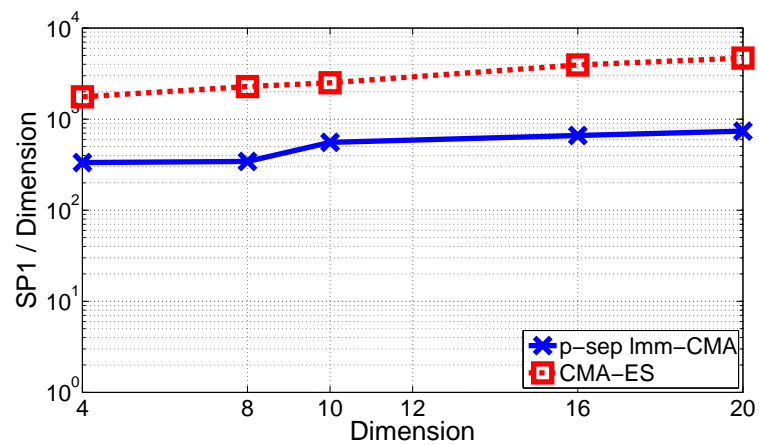
(a) f_{Rosen}^1 (b) f_{Rosen}^{100} (c) f_{Rosen}^{10000}

Figure 4.2: Success performance SP1 over the dimension of the problem on f_{Rosen}^α , with $\alpha = 1, 10^2$ and 10^4 for dimensions in between 4 and 40. The dimension of the sub-functions n_M equals 2.

clear that exploiting the partial separability within CMA-ES with meta-models improves the performance of CMA-ES with a speedup in-between 4.5 and 15.

For element functions with fixed n_M equal to 2, p-sep lmm-CMA offers an increasing speedup with increasing dimensions of the problem as shown in Fig. 4.2. The algorithm p-sep lmm-CMA performs better with increasing dimensions since it breaks the curse of dimensionality when building the meta-model: for a problem of dimension n , building the meta-model is equivalent to building N meta-models of dimension n_M .

Using greater number of parameters for each separated meta-model decreases the speedup obtained by the approach. On f_{Rosen}^{100} for a dimension 16, the speedup, decreases from 8.6 to 2.2 for corresponding values of n_M respectively equal to 2 and 4.

At each iteration at least n_b function evaluations are performed on the true function in order to check the accuracy of the meta-models. The parameter n_b is set to $\max[1, (\frac{\lambda}{10})]$. This setting is introduced in order to be able to add a significant amount of information at each iteration by enriching the training set. It is in particular important when dealing with large population sizes. For increasing population sizes λ , i.e., for increasing values of μ , we need an increasing number of points evaluated at each iteration cycle to be able to have a significant impact on the ranking of population.

Moreover, a better setting of n_b would also depend on the dimension of the problem as for increasing dimensions, i.e., for increasing numbers k (or k_i) of points to build the meta-model, we need an increasing number of points evaluated at each iteration cycle to be able to change significantly the meta-model and then the ranking of the population.

The minimum number of evaluations performed at each iteration n_b limits the speedup that can be achieved by our approach. We show that for some test functions, we are able to reach this maximum speedup of λ/n_b . For f_{Rosen}^{100} with $n = 40$ and for $f_{\text{Rosen}\frac{1}{2}}^{100}$ with $n = 20$, we reach a speedup equal to λ since n_b is equal to 1 in these tests.

Since we reach the maximal speedup allowed by the approach on the Rosenbrock function, we asked ourselves whether we can further reduce the number of overall function evaluations needed to reach a target by increasing the population size λ . The default population size denoted λ_{default} value equals $4 + \lfloor 3 \times \ln(n) \rfloor$. Fig. 4.1(a) shows the influence of the population size on the performance of p-sep lmm-CMA. We perform 20 independent runs on f_{Rosen}^{100} for dimensions $n = 4, 8, 10$ and 16, and $n_M = 2$ with $f_{\text{stop}} = 10^{-10}$. The tested population sizes are written as $\lambda = \gamma \times \lambda_{\text{default}}$ where γ is in-between 1 and 10. Tests were performed with similar parameters: n_{init} initialized to λ_{default} and n_b equal to $\max[1, (\frac{\lambda_{\text{default}}}{10})]$. A training set containing k_i elements randomly sampled is loaded at the beginning of every run in order to use the meta-models from the first generation, for all the tests. Results show that $\lambda = 4 \times \lambda_{\text{default}}$ gives the minimum number of evaluations to reach f_{stop} and improves the performance by a factor between 1.5 and 2 over the default

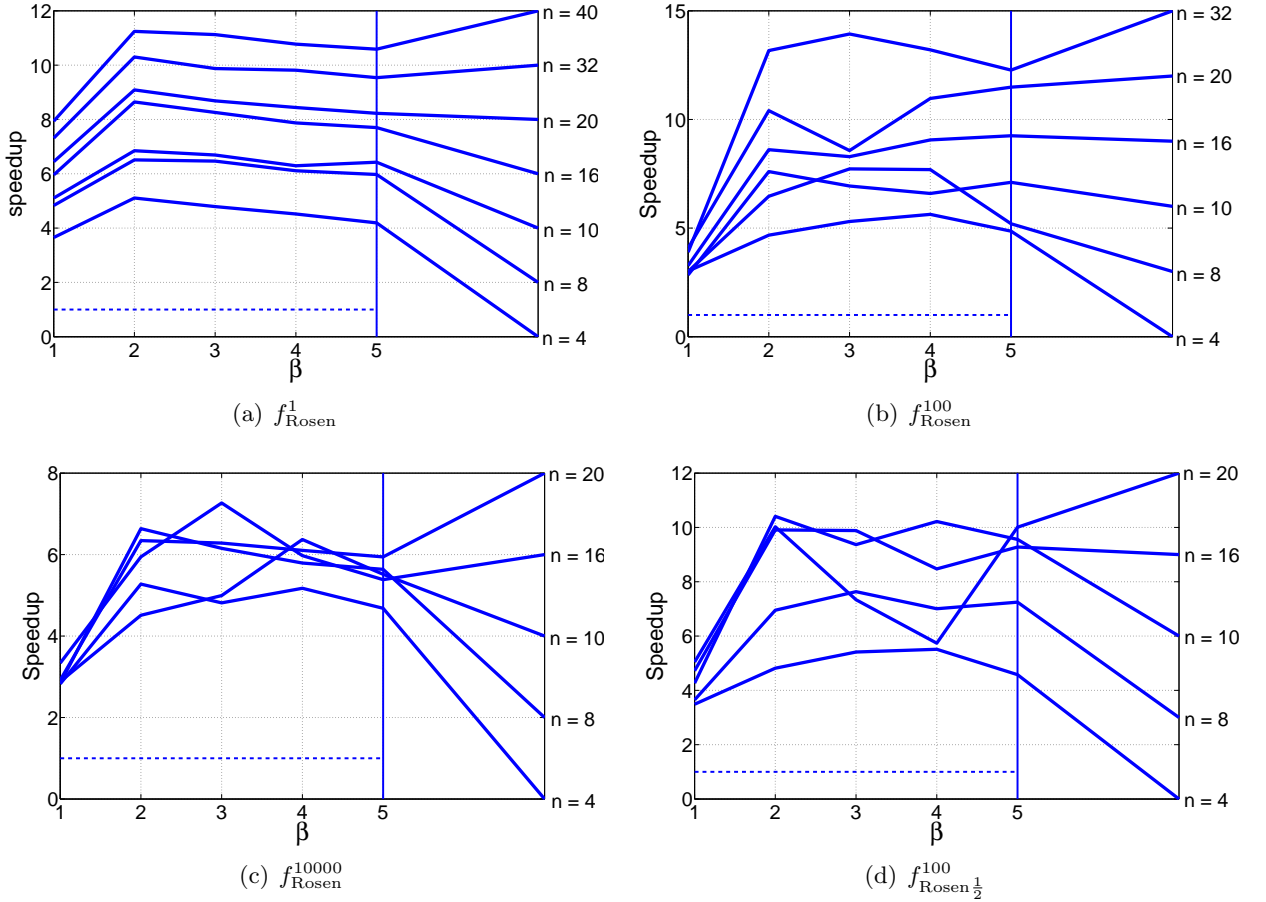


Figure 4.3: Average speedup with respect to CMA-ES to reach f_{stop} with a varying number of points used to build the meta-model $k_i = \beta \times k_{i,\text{min}}$ where $k_{i,\text{min}} = \frac{n_i(n_i+3)}{2} + 1$. Each point corresponds to 20 runs performed.

population size. For $\gamma > 4$, the performance of p-sep Imm-CMA stagnates. We observe in Fig. 4.1(b) that the number of evaluations per generation increases linearly for increasing population sizes.

4.3.3 Optimal bandwidth for building partially separated meta-models

Let us consider an element function f_i with a number of element variables n_i . The optimal bandwidth depends on the number of points k_i used to build the meta-model. As shown in Section 4.2, k_i must be greater or equal to $k_{i,\text{min}} = \frac{n_i(n_i+3)}{2} + 1$. In this section, we investigate the influence of the choice of k_i on the performance of p-sep Imm-CMA. We perform 20 independent runs on f_{Rosen}^α for $\alpha = 1, 10^2, 10^4$ and $f_{\text{Rosen}\frac{1}{2}}^{100}$ for different dimensions in-between 4 and 40. Results are shown in Fig. 4.3, where k_i is written as $k_i = \beta \times k_{i,\text{min}}$ for $\beta = 1, 2, 3, 4$ and 5. We find that for 14 tests over the 23 tests performed on the test functions with different dimensions, a good estimate of the optimal β is equal to 2. Moreover, for the other tests, choosing a value of β equal to 2 is a

reasonable choice since it offers a speedup close to best one found, except for f_{Rosen}^{100} with dimensions 10 and 16.

4.3.4 Computational cost

The internal cost of the optimization procedure is dominated by the evaluation of the objective function and the construction of the meta-model.

For p-sep lmm-CMA, building a meta-model consists in finding in the training set the k_i sorted nearest points to the point to be evaluated and then solving Eq. (4.5). Let us consider a training set with a size m . To find and sort the best k_i points, we begin by sorting the first k_i points of the training set using a heapsort algorithm which has a complexity of $k_i \log k_i$. Then, we compare the other $(m - k_i)$ points with the selected k_i points until finding its position which adds at worst a complexity of $(m - k_i) \times k_i$. Thus, finding and sorting the best k_i points needs $O(k_i \log k_i + (m - k_i)k_i) = O(m \times k_i)$. According to Section 4.3.3, the optimal bandwidth k_i is equal to $n_i(n_i + 3) + 2$. Thus, finding and sorting the points to evaluate the meta-model needs $O(m \times n_i^2)$. Moreover, solving Eq. (4.5) is dominated by a $k_i \times k_i$ matrix inversion and thus has a complexity of n_i^6 .

Let us denote by N_e the number of evaluations on the true objective function and by c_e the complexity of one single objective function evaluation. Let us denote also by N_m the number of built meta-models. The complexity of p-sep lmm-CMA is then equal to: $N_e c_e + N_m n_i^2(m + n_i^4)$.

4.4 Summary and discussions

In this chapter we have investigated the exploitation of partial separability of the objective function to enhance the performances of CMA-ES coupled with local meta-models. We have defined p-sep lmm-CMA, a new variant of CMA-ES with meta-models for partially separable functions. In this variant, we build separate meta-models for each element function, instead of building one meta-model for the whole objective function. We have shown that the speedup of p-sep lmm-CMA with respect to CMA-ES is in-between 4.5 and 15 for the tested functions. For f_{Rosen}^{100} with a dimension 40 and for $f_{\text{Rosen}\frac{1}{2}}^{100}$ with a dimension 20, we reach a speedup equal to λ which corresponds to the theoretical maximum speedup allowed by the approach. In general, the maximum speedup that can be achieved equals λ/n_b as at least n_b evaluations on the true function are performed at each iteration. We have shown on the standard Rosenbrock function that increasing the population size allows to decrease significantly (by a factor between 1.5 and 2) the number of evaluations

to reach a given target. The optimal population size on the Rosenbrock function is shown to be equal to $4 \times \lambda_{\text{default}}$.

Chapter 5

Partially separated meta-models with CMA-ES for well placement optimization

This chapter is based on the paper [25]. In the well placement optimization problem, the objective function (e.g., the NPV) can usually be split into local components referring to each of the wells that moreover depends in general on a smaller number of principal parameters, and thus can be modeled as a partially separable function. In this chapter, we propose to apply p-sep lmm-CMA (defined in Chapter 4) on the well placement problem, i.e., to exploit the partial separability of the objective function when using CMA-ES coupled with meta-models, by building partially separated meta-models. Thus, different meta-models are built for each well or set of wells, which results in a more accurate modeling. The approach is shown on the PUNQ-S3 case.

This chapter is structured as follows. Section 5.1 defines p-sep lmm-CMA for the well placement problem. In Section 5.2, we demonstrate the contribution of the proposed approach in reducing the number of reservoir simulations on the synthetic benchmark reservoir case PUNQ-S3 [54].

5.1 p-sep lmm-CMA for well placement optimization

In this chapter, we propose to build a meta-model for each well or set of wells to be placed, instead of one meta-model for all the wells.

In order to apply p-sep lmm-CMA (defined in Chapter 4), we need to define the different element functions and their corresponding dependencies. As mentioned in Chapter 4, for a given partially separable function, there exists “theoretically” an infinite number of ways to define the element functions and mapping functions. However in this chapter, we

propose to investigate building one meta-model for each well (already drilled and to be drilled) approximating its NPV.

Let us consider a reservoir case with a number \mathcal{N}_w of wells to be drilled. We suppose that we have also \mathcal{N}_{wd} wells already drilled. We denote by $(NPV_i)_{1 \leq i \leq \mathcal{N}_w}$ the NPVs corresponding to the wells to be drilled and by $(NPV_i)_{(\mathcal{N}_w+1) \leq i \leq (\mathcal{N}_w+\mathcal{N}_{wd})}$ the NPVs corresponding to the wells already drilled.

Therefore, the objective function corresponding to the NPV of the field is equal to the sum of the different element functions corresponding to the NPV of each well, i.e., $(NPV_i)_{1 \leq i \leq (\mathcal{N}_w+\mathcal{N}_{wd})}$.

Let us denote by $\{m_1, \dots, m_{\mathcal{N}_w}\}$ the number of parameters defining the position of the wells to be placed, and by $(\mathbf{W}_j \in \mathbb{R}^{m_j})_{1 \leq j \leq \mathcal{N}_w}$ these parameters. Thus, the NPV, as well as the NPVs corresponding to each well depends on $(\mathbf{W}_j)_{1 \leq j \leq \mathcal{N}_w}$:

$$NPV = \sum_{i=1}^{\mathcal{N}_w+\mathcal{N}_{wd}} NPV_i , \quad (5.1)$$

$$NPV((\mathbf{W}_j)_{1 \leq j \leq \mathcal{N}_w}) = \sum_{i=1}^{\mathcal{N}_w+\mathcal{N}_{wd}} NPV_i((\mathbf{W}_j)_{1 \leq j \leq \mathcal{N}_w}) . \quad (5.2)$$

As reflected in the previous equation, in general, the NPV_i of a given well i depends on all the wells¹, however, in order to use the p-sep lmm-CMA, we will assume that the NPV_i of a well essentially depends on a fewer number of parameters.

In this chapter, we will assume that the NPV_i of a given well essentially depends on the considered well and that the impact of other wells is represented only by distances between the considered well and the others. For each well denoted by i , we define the following parameters:

- dp_i : the minimum distance between the well i and the other producers;
- di_i : the minimum distance between the well i and the other injectors.

The minimum distance between two wells is defined by the minimum Euclidean distance between the two trajectories of the considered wells. In order to calculate the meta-model, we now suppose that the NPVs of the wells to be drilled, i.e., $(NPV_i)_{1 \leq i \leq \mathcal{N}_w}$ can be approximated using only the parameters defining the location and trajectory of the considered well and its corresponding dp_i and di_i . The NPV of the well already drilled, i.e., $(NPV_i)_{(\mathcal{N}_w+1) \leq i \leq (\mathcal{N}_w+\mathcal{N}_{wd})}$ can be approximated using only two parameters: dp_i and di_i .

¹Except when dealing with non-communicating reservoir regions, and if each of the wells has to be placed in one of these regions

Therefore, the built meta-model $\hat{\text{NPV}}$ can be written as follows:

$$\hat{\text{NPV}}((\mathbf{W}_j)_{1 \leq j \leq N_w}) = \sum_{i=1}^{N_w} \hat{\text{NPV}}_i(\underbrace{\mathbf{W}_i, \text{dp}_i, \text{di}_i}_{\in \mathbb{R}^{m_i} \times \mathbb{R} \times \mathbb{R}}) + \sum_{i=N_w+1}^{N_w+N_{\text{wd}}} \hat{\text{NPV}}_i(\text{dp}_i, \text{di}_i) \quad , \quad (5.3)$$

where $\hat{\text{NPV}}_i$ denotes the meta-model approximating NPV_i .

After that, to incorporate the built meta-model $\hat{\text{NPV}}$ into CMA-ES, we use the approximate ranking procedure as described in the variant nimm-CMA_2 defined in Section 2.3.3.4 with only one difference related to the acceptance criterion of the meta-model: in this case, we use a less conservative criterion in which the meta-model is accepted if it succeeds in keeping only the best well configuration unchanged.

In the next section, we will see how the approach can be applied for a well placement problem and the number of function evaluations that can be saved in the optimization process.

5.2 Application of p-sep lmm-CMA on the PUNQ-S3 case

This section shows an application of p-sep lmm-CMA on the benchmark reservoir case PUNQ-S3 [54]. This application is compared to the CMA-ES optimizer and to the variant of CMA-ES with meta-models (nimm-CMA)¹. As shown in previous examples, the model contains $19 \times 28 \times 5$ grid blocks. The elevation of the field is shown in Fig. 3.2. An injection well denoted I1 is already drilled. Fig. 5.1 represents the SoPhiH map which represents the distribution of the hydrocarbon pore volume over the n_{layers} layers, and defined by $\sum_{k=1}^{n_{\text{layers}}} (\text{H}_k \times \phi \times S_o)$, where H_k is the gross thickness of the layer k , S_o is the oil saturation and ϕ is the porosity. The location of I1 is also shown in Fig. 5.1, where I1 is an inclined well drilled in the layer 3.

We propose to drill 3 unilateral producers (denoted P1, P2 and P3) to maximize the NPV. The dimension of the problem is then equal to: $6 \times 3 = 18$. A producer limit bottomhole pressure is fixed to 150 bar, and an injector limit bottomhole pressure is fixed to 320 bar. A maximum length of 1000 m is imposed on the 3 producers to be drilled.

The population size λ is set, for all the methods used, to 60. The different optimizers are run with a stopping criterion corresponding to a maximum number of reservoir simulations equal to 1000. Other parameters of the optimization method were set to default settings.

As shown in Section 5.1, the built meta-model for the element functions $(\text{NPV}_i)_{i=1, \dots, 3}$ will only depend on *eight parameters* (compared to eighteen if we would use all the original variables), and the meta-model for the element function NPV_4 will only depend on *a single*

¹In this chapter, we use the variant nimm-CMA_2 (defined in Section 2.3.3.4), as used in Chapter 3.

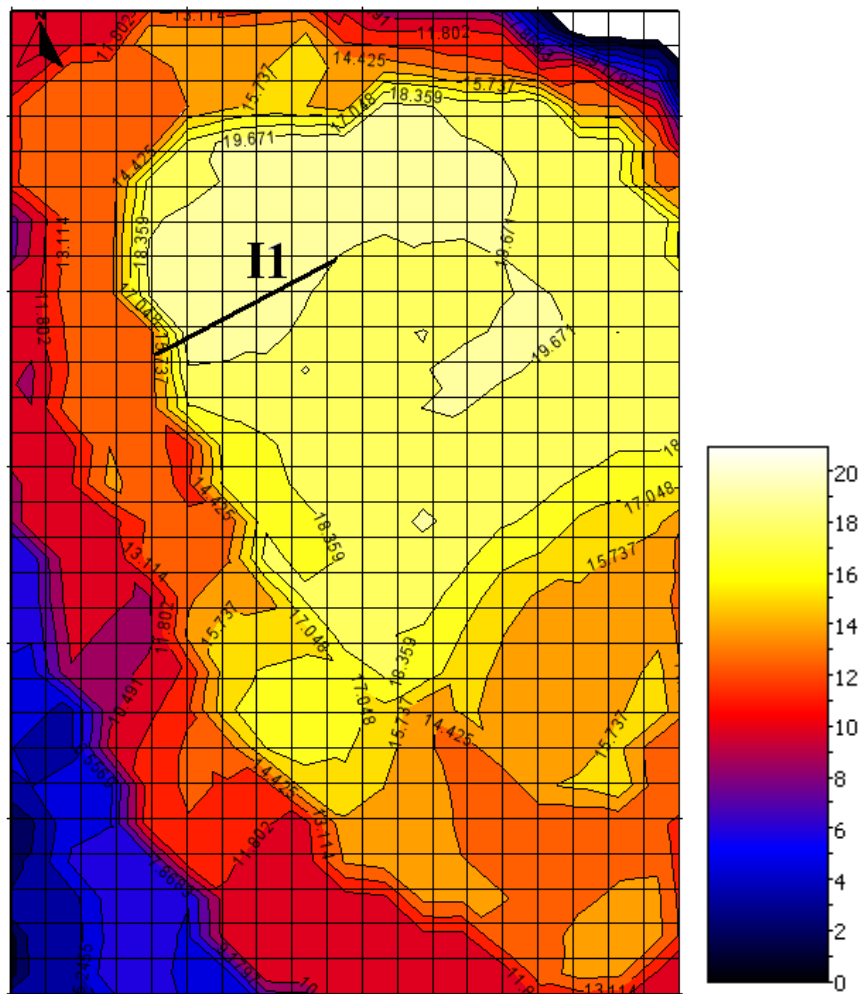


Figure 5.1: The SoPhiH map with the location of the injector already drilled I1.

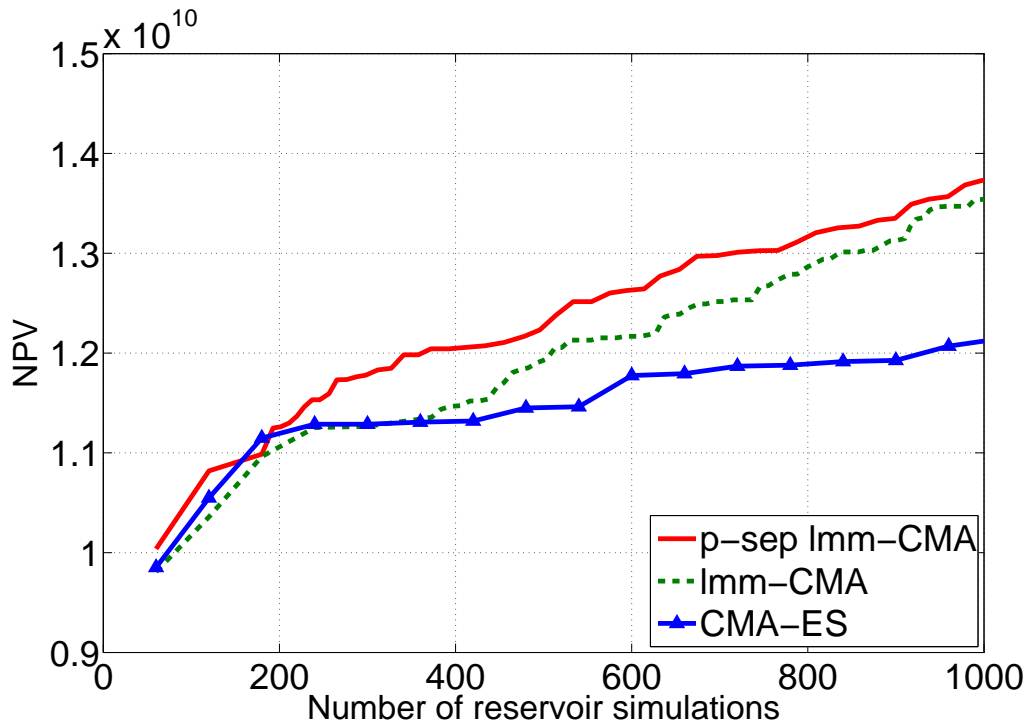


Figure 5.2: The mean value of NPV (in US dollars) for well placement optimization using CMA-ES with partially separated meta-models denoted p-sep lmm-CMA (*solid line*), CMA-ES with meta-models denoted lmm-CMA (*dash line*) and CMA-ES (\triangle). Ten runs are performed for each method.

*parameter*¹:

$$\hat{\text{NPV}}(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3) = \hat{\text{NPV}}_1(\mathbf{P}_1, dp_1, di_1) + \hat{\text{NPV}}_2(\mathbf{P}_2, dp_2, di_2) + \hat{\text{NPV}}_3(\mathbf{P}_3, dp_3, di_3) + \hat{\text{NPV}}_4(dp_4), \quad (5.4)$$

where $\mathbf{P}_i \in \mathbb{R}^6$ denotes the vector of parameters defining the position of the well P_i .

The number of points used to build the partially separated meta-models, is chosen to be equal to 90 (according to Section 4.3.3), and the meta-model is used when the training set (storing the performed evaluations) contains at least 150 elements, i.e., before performing 150 simulations, all the points are evaluated with the true objective function, and the partially separated meta-model is not used.

Fig. 5.2 shows the average performance of the proposed method, i.e., CMA-ES with partially separated meta-models (p-sep lmm-CMA). Results are reported together with those obtained using CMA-ES and CMA-ES with meta-models (nlmm-CMA). The performance of each method is evaluated on ten independent runs, where for each run, we report the best obtained NPV value after each generation. These values correspond to true values of the objective function, i.e., obtained with a reservoir simulation².

¹Here, we have only one single parameter dp_i , since the only injector we have is the considered injector.

²The CMA-ES with meta-models method ensures by construction that at least each generation the best

During the first iterations of the optimization, the performance of the 3 used algorithms is equivalent. For p-sep lmm-CMA, the meta-model is used if the training set contains at least 150 performed reservoir simulation results. Therefore, at the beginning of the optimization, the meta-model is not used which justifies the equivalent results for the three optimizers.

For nlmm-CMA, building the meta-model requires more reservoir simulations compared to partially separated meta-models. Non-partially separated meta-models depend on 18 parameters. In the performed runs, the meta-model is built using 300 performed reservoir simulations ($k = 300$) and used when the training set contains at least 350 objective function evaluations. Hence, before reaching 350 simulations, nlmm-CMA and CMA-ES are equivalent.

Except at the beginning of the optimization in which all the optimizers are equivalent, it is clear that CMA-ES with partially separated meta-models outperforms the other methods, when considering a restricted budget of 1000 reservoir simulations. The context of restricted budget of simulations is imposed to consider real applications in which the number of simulations is generally limited to several hundreds or at most a few thousands, due to the CPU time required by a simulation.

For a given number of reservoir simulations equal to 600, p-sep lmm-CMA is able to find a well configuration with an NPV equal to $\$1.26 \times 10^{10}$. However, CMA-ES reaches only an NPV equal to $\$1.17 \times 10^{10}$ and nlmm-CMA offers only a maximum NPV equal to $\$1.21 \times 10^{10}$. As a conclusion, using a restricted budget of reservoir simulations, exploiting the partial separability allows reaching greater NPV values compared to CMA-ES and nlmm-CMA.

To reach a value of NPV equal to $\$1.20 \times 10^{10}$, CMA-ES with partially separated meta-models requires 370 reservoir simulations. However, to reach the same value of NPV, using standard meta-models requires 510 reservoir simulations, and when using CMA-ES without meta-models, we need 930 reservoir simulations. Therefore, using partially separated meta-models saves 60% of the number of reservoir simulations compared to CMA-ES (without meta-models). The contribution of exploiting the partial separability is shown when comparing p-sep lmm-CMA with nlmm-CMA. Exploiting the partial separability of the objective function saves 28% of the number of reservoir simulations compared to CMA-ES with standard meta-models approach.

Fig. 5.3 shows one of the obtained solution well configurations, with an NPV value equal to $\$1.38 \times 10^{10}$. Although, each of the performed runs proposes in general a different solution, the majority of the solution well configurations are located in the same regions.

point is evaluated with the true objective function, i.e., each iteration, the best obtained well configuration is evaluated using a reservoir simulation.

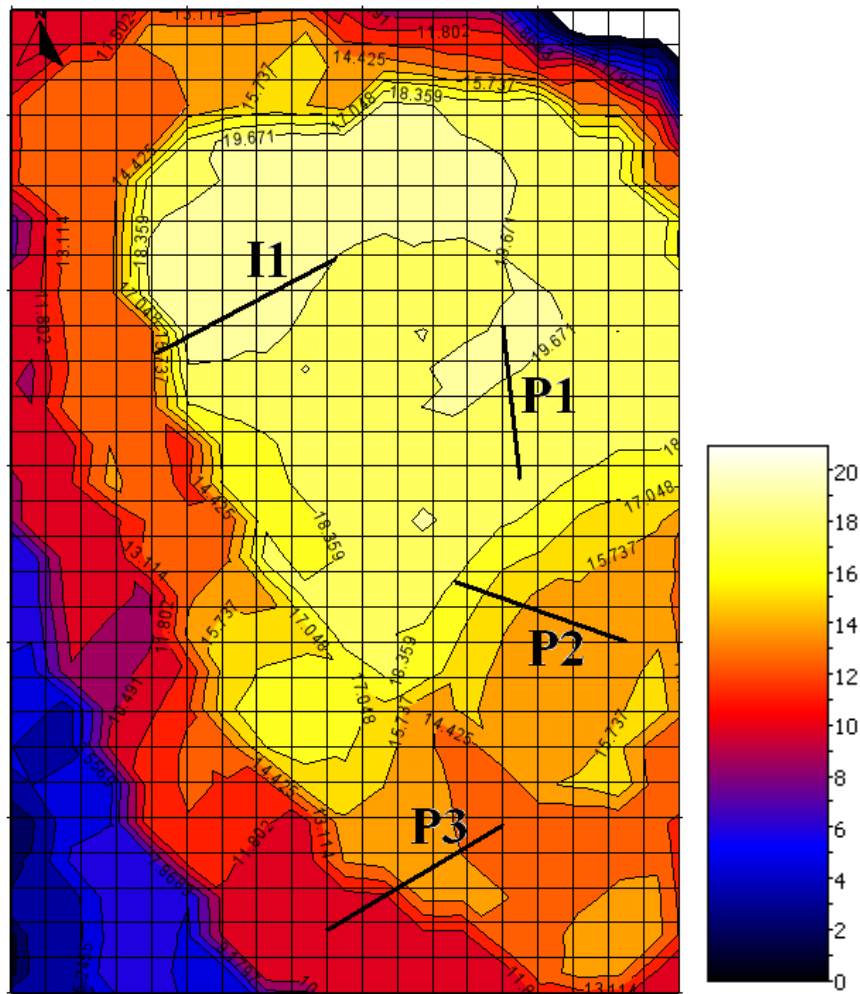


Figure 5.3: The SoPhiH map with the location of the injector already drilled I1, and solution producers (P1, P2 and P3).

Fig. 5.4 shows a typical optimization process performed using CMA-ES with separated meta-models, i.e., with p-sep ImmCMA. Fig. 5.4 shows the evolution of the NPV (the best at each generation and the overall best) as well as the evolution of the parameters encoding the three wells.

5.3 Summary and discussions

In this chapter we have shown on the synthetic benchmark reservoir case PUNQ-S3 that using p-sep Imm-CMA algorithm leads to an important reduction of the number of reservoir simulations (around 60%) compared to the optimizer CMA-ES. The important savings in the number of reservoir simulations are justified by the reduced number of parameters required to build the meta-model of the element functions.

The proposed approach exploiting the partial separability of the objective function can also be combined with any other stochastic optimizer, in order to reduce the computational cost of the optimization.

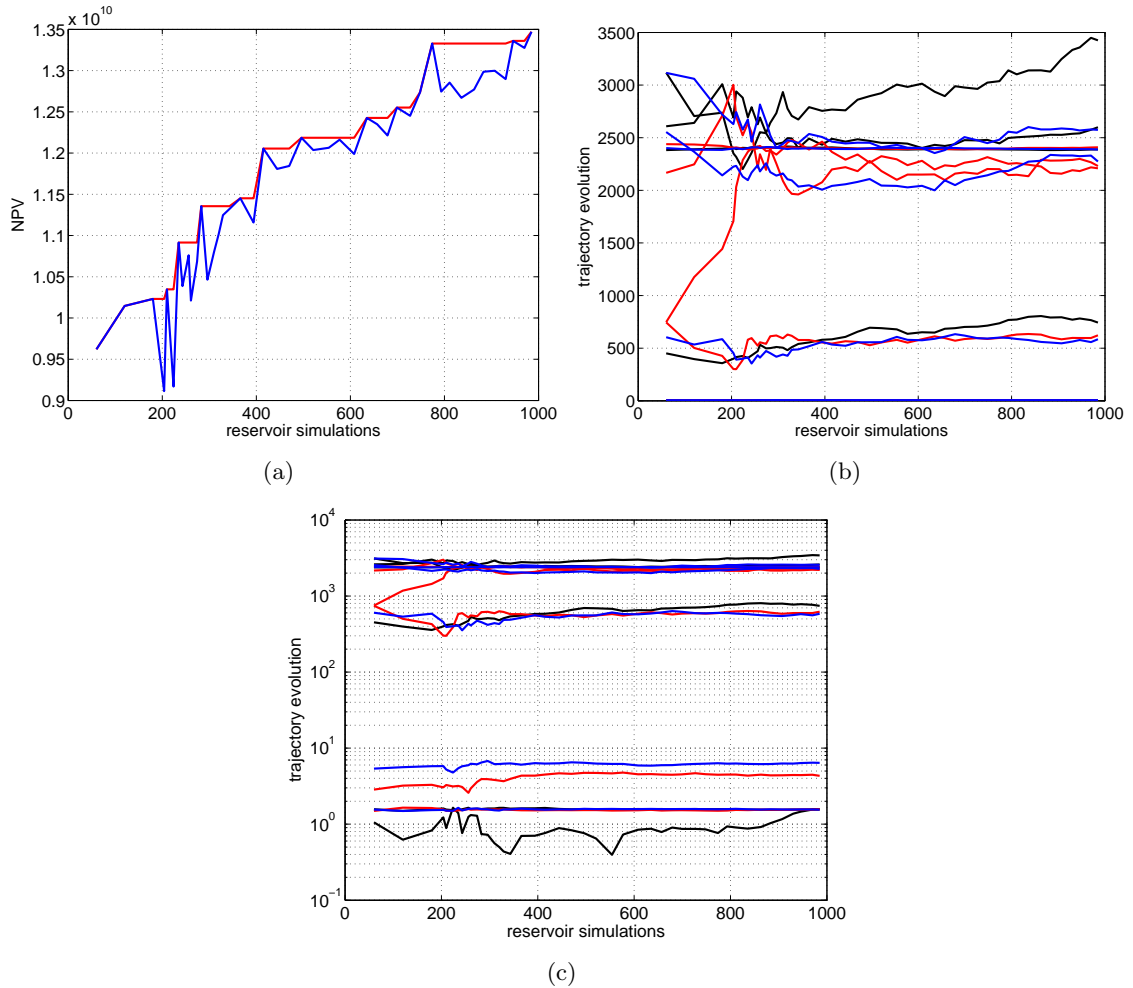


Figure 5.4: The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with partially separated meta-model, i.e., p-sep lmmCMA. The three figures depict one of the ten performed runs of p-sep lmm-CMA. In (a), the evolution of the best overall NPV value (in red) and the best NPV obtained each generation (in blue) is depicted. In (b), the evolution of the well trajectory parameters, where each well is plotted using a different color representing three group of parameters is depicted. The group of angles encoding each well is shown in the lower part of the figure (values below 10). The group of well lengths is shown in the intermediate part of the figure (the three curves with values around 500). The group of Cartesian coordinates of the wells is shown in the upper part of the figure. In (c) the evolution of the well trajectory parameters on the log-scale is depicted.

Chapter 6

Well placement optimization under geological uncertainty

In the well placement problem, as well as in many other field development optimization problems, geological uncertainty is considered as a key source of risk affecting the viability of field development projects. The problem arises when we have multiple possible geological realizations of the reservoir. The multiple realizations are generated using geostatistical techniques and in general deemed equiprobable. Let us consider an objective function to optimize denoted by f and a number \mathcal{N}_r of geological realizations denoted by $(R_i)_{i=1, \dots, \mathcal{N}_r}$. The key issue here is that for each scenario, i.e., for each well configuration when optimizing well placement, we have \mathcal{N}_r possible values of the objective function, one for each realization where each will be denoted for a given well configuration \mathbf{x} by $f(\mathbf{x}, R_i)$ corresponding to a given realization R_i .

This chapter addresses the problem of how to define the objective function to optimize when dealing with uncertainty for well placement and whether we should perform evaluations on all the possible realizations in order to define the objective function.

This chapter is structured as follows. Section 6.1 provides a detailed literature review for well placement optimization under geological uncertainty. Section 6.2 defines a new approach to handle geological uncertainty for well placement using the neighborhood. In Section 6.3, we demonstrate the contribution of the proposed approach in capturing the geological uncertainty and in reducing the number of reservoir simulations on the synthetic benchmark reservoir case PUNQ-S3 [54].

6.1 Optimization under uncertainty: a literature review

The problem of optimization under geological uncertainty shares many similarities with the problem of optimizing noisy functions.

A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be noisy if the only measurable value of f on $\mathbf{x} \in \mathbb{R}^n$ is a random variable that can be written as $\mathcal{F}(f(\mathbf{x}), z)$ where f is a time-invariant function and z is a noise often assumed to be normally distributed with a zero mean and variance σ^2 , and denoted by $\mathcal{N}(0, \sigma^2)$. The noise can be also defined differently (e.g., Cauchy distributed), and can be either additive or multiplicative. A common approach to optimize noisy functions is to estimate the fitness function by the expected value defined as follows:

$$f(\mathbf{x}) = \int_{-\infty}^{+\infty} [\mathcal{F}(f(\mathbf{x}), z)] p(z) dz \quad , \quad (6.1)$$

where $p(z)$ is the probability density function of the noise. Thus, a common way to approximate the expected fitness function is by averaging over a number of random samples:

$$f(\mathbf{x}) \simeq \frac{1}{\mathcal{N}_s} \sum_{i=1}^{\mathcal{N}_s} [\mathcal{F}(f(\mathbf{x}), z_i)] \quad , \quad (6.2)$$

where \mathcal{N}_s denotes the number of samples called also the sample size.

In the context of field development optimization under geological uncertainty, we are dealing with a finite number of realizations, and the measurable fitness values correspond to the values $f(\mathbf{x}, R_i)_{i=1, \dots, \mathcal{N}_r}$. Therefore, the objective function corresponds in general to:

$$f(\mathbf{x}) = \frac{1}{\mathcal{N}_r} \sum_{i=1}^{\mathcal{N}_r} [f(\mathbf{x}, R_i)] \quad . \quad (6.3)$$

However due to the expensive computational effort required to evaluate the objective function over one realization R_i , the expected fitness function is often approximated in a way to use a fewer number of samples instead of using all the realizations. Thus, one common way to approximate the expected objective function here is again by averaging over a number of samples $\mathcal{N}_s \leq \mathcal{N}_r$.

In the following, we briefly review the existing approaches often used in optimization under uncertainty. On the one hand we review the approaches defined by the optimization community mainly to cope with noise but that can be extended to the different field development optimization under geological uncertainty. On the other hand we review the approaches already applied in the petroleum community to cope with geological uncertainty.

6.1.1 Optimization community

This section summarizes the different ways to handle uncertainty within the evolutionary optimization community. A detailed overview of the existing approaches addressing uncertainties in evolutionary optimization is presented in [84]. Let us suppose in this sec-

tion then that the function f to optimize is a noisy function. The approaches to handle uncertainty can be mainly divided into two categories.

6.1.1.1 Explicit Averaging

Using mean of several samples for each individual The simplest and the most common way to address the uncertainty issue is to define the objective function for each point by averaging over a number of samples (Eq. (6.2)). Increasing the sample size \mathcal{N}_s is equivalent to reducing the variance of estimating the objective function.

In general, the objective function is defined using an averaged sum of a *constant sample size*. In this case, for each single evaluation of the expected objective function, one needs to evaluate the objective function on \mathcal{N}_s samples.

In the context of costly objective functions, depending on the number of samples, there is a compromise between the computational cost of the optimization and the accuracy of the estimation of the objective function. Increasing (respectively, decreasing) the number of samples tends to improve (respectively, worsen) the accuracy of the estimated objective function, but on the other hand it tends also to increase (respectively, reduce) the computational cost of the optimization. The idea of using an *adapted sample size* during the optimization was first proposed in [3, 4]. In [4], it is shown that adapting the number of samples performs better than using constant sample sizes, and it is suggested to increase the sample size with the generation number and to use a higher number of samples for individuals with higher estimated variance. An other way to adapt the sample size is based on an individual's probability to be among a number of the best individuals [121]. Recently, an other approach relying on the rank based selection operators was proposed in [73]. In [76], an adaptive uncertainty handling procedure is proposed, based on selection races [93].

Using the neighborhood for each individual An alternative approach to defining the objective function as an averaged sum of a number of samples (constant or adapted) is to define the objective function using the neighborhood points already evaluated [106, 29, 28, 27, 112, 113]. The general idea has first been suggested in [27] in which it is suggested to estimate the fitness as a weighted average of the neighborhood with a linearly decreasing weight function up to some fixed maximum distance. In [106, 28, 29], a locally weighted regression is used for estimation. This technique is shown to be a good solution to improve the accuracy of the estimated objective function without increasing the computational cost.

6.1.1.2 Implicit Averaging

When increasing the population size, the probability to obtain similar points is higher. Thus, a way to cope with noise is to simply increase the population size [52]. In this case, with a large population size, the influence of noise on a given point can be reduced due to the evaluations on other similar points. Conflicting conclusions [52, 7, 8, 60] were shown in the literature when comparing explicit averaging and implicit averaging.

6.1.2 Petroleum community

Several studies in the literature have addressed the problem of optimization under geological uncertainty not only on the well placement problem but also on other field development optimization problems. Optimization under geological uncertainty in the petroleum community considers always a finite number of realizations \mathcal{N}_r and models the objective function following Eq. (6.3). In the following we briefly review the approaches to handle uncertainty in optimization within the petroleum community.

To the best of our knowledge, all the studies that consider a number \mathcal{N}_r multiple possible realizations of the reservoir, use the approach “Using mean of several samples for each individual”. Moreover, all the studies in the literature, except the approach proposed in [126] that will be detailed later, perform \mathcal{N}_r reservoir simulations for every single objective function evaluation. Although sharing this common similarity, the proposed approaches introduce different formulations of the objective function.

In [116, 115, 103, 35], the objective function is formulated as the expected value of the net present value over all the realizations, as shown in Eq. (6.3). In [35], the authors tackle the problem of closed-loop production optimization using the optimizer EnOpt [37, 36] which is applied to the geological model ensemble updated by either EnKF [49] or EnRML [62].

In [129, 2, 5], multiple geostatistical realizations of the reservoir are considered in the formulation of the objective function:

$$f(\mathbf{x}) = \frac{1}{\mathcal{N}_r} \sum_{i=1}^{\mathcal{N}_r} [f(\mathbf{x}, R_i)] + r\sigma \ , \quad (6.4)$$

where $r \in \mathbb{R}$ is the risk factor and σ is the standard deviation of f on \mathbf{x} over the realizations, defined as follows:

$$\sigma = \sqrt{\frac{1}{\mathcal{N}_r} \sum_{i=1}^{\mathcal{N}_r} (f(\mathbf{x}, R_i) - \langle f(\mathbf{x}) \rangle)^2} \ , \quad (6.5)$$

where:

$$\langle f(\mathbf{x}) \rangle = \frac{1}{\mathcal{N}_r} \sum_{i=1}^{\mathcal{N}_r} [f(\mathbf{x}, R_i)] \ . \quad (6.6)$$

The term $r\sigma$ in Eq. (6.4) is used to take into account the decision maker's attitude toward risk. A positive r indicates a risk-prone attitude, a negative r indicates a risk-averse attitude and an $r = 0$ indicates a risk-neutral attitude. This formulation is close to the formulations defined in [64, 104] using utility functions.

In [10], a more general formulation of the objective function is defined as follows. A genetic algorithm is used, in which at each iteration only a predefined percentage of the individuals, chosen according to a set of scenario attributes, is simulated. For the simulated individuals, the authors in [10] propose to perform again \mathcal{N}_r reservoir simulations for each well configuration \mathbf{x} in order to evaluate the values of $f(\mathbf{x}, R_i)$ on all realizations and then to derive the cumulative distribution function $\text{cdf}\{f\}$ on \mathbf{x} . From this distribution, the values of $f^{10}(\mathbf{x})$, $f^{50}(\mathbf{x})$ and $f^{90}(\mathbf{x})$ are determined. The value f^{10} on \mathbf{x} denotes the value of f on \mathbf{x} corresponding to a probability of 0.1, i.e., there is a probability 0.1 that the value of f on \mathbf{x} will be less than f^{10} on \mathbf{x} . The value f^{10} on \mathbf{x} can be written as $\text{cdf}\{f\}^{-1}(0.1)$. The values $f^{50}(\mathbf{x})$ and $f^{90}(\mathbf{x})$ are defined in a way similar to $f^{10}(\mathbf{x})$. The objective function is then formulated as follows:

$$f(\mathbf{x}) = r_{10}f^{10}(\mathbf{x}) + r_{50}f^{50}(\mathbf{x}) + r_{90}f^{90}(\mathbf{x}) , \quad (6.7)$$

where the parameters r_{10} , r_{50} and r_{90} are defined according to the decision maker's attitude toward risk. A risk-neutral attitude corresponds to the case where $(r_{10}, r_{50}, r_{90}) = (0, 1, 0)$ which may be similar to the definition in Eq. (6.3). However, a risk-averse investor tends to increase the value of r_{10} , and a risk-prone investor tends to increase the value of r_{90} .

Another way to formulate the objective function under geological uncertainty is to optimize the worst case scenario using a min-max problem formulation [30]. This approach is used in [5] to optimize smart well controls.

The only approach selecting only a number of samples instead of all the realizations is defined in [126]. The approach is based on the so-called retrospective optimization [34, 127] and divides the problem as a number of subproblems, where the initial solution of the current subproblem is simply the returned solution from the previous subproblem. Each point to be evaluated is approximated by the average over a number of realizations, where the number of selected realizations is increased from subproblem to subproblem. The approach implies then defining a sequence of samples. The example shown in [126] considers a well placement problem on 104 permeability and porosity realizations and therefore defines subproblems with a sequence $\{20, 15, 10, 5\}$ of iterations and a sequence $\{1, 5, 16, 21, 104\}$ of sample sizes. Although the authors suggest further testing of the overall framework to determine the appropriate sequence of sample sizes, an answer can

6.2 Well placement under uncertainty with CMA-ES using the neighborhood

be the work on adapting automatically the sample sizes already proposed in [121, 73] but still demanding in the number of objective function evaluations.

6.2 Well placement under uncertainty with CMA-ES using the neighborhood

This section proposes a new approach to handle geological uncertainty for well placement. The proposed approach focuses on reducing the uncertainty by using the objective function evaluations of already evaluated individuals in the neighborhood. In this section, we propose then to apply an approach based on using the neighborhood for each individual. We define a CMA-ES optimizing an estimated fitness defined on a given point using a weighted average of a small number of evaluations on the considered point and a number of evaluations already performed on the neighborhood (up to some fixed maximum distance) with a decreasing weight function depending on the Mahalanobis distance with respect to the covariance matrix \mathbf{C} defined by CMA-ES. Although considering a Mahalanobis distance with respect to $\sigma^2\mathbf{C}$ is suspected to be a better choice (since we are using a fixed maximum distance to select the neighbors), it has not been tested in this thesis.

Let us consider a well placement optimization problem with a number of wells (producers and/or injectors) to be placed. Let us denote by n the dimension of the problem, i.e., the number of parameters needed to encode the wells to be placed. The wells to be placed can be parameterized as defined in Section 3.1.2. Without loss of generality, we will consider in the sequel the NPV as the objective function that we aim to optimize, unless otherwise explicitly stated. Thus, we want to find a vector of parameter $\mathbf{p}_{\max,R} \in \mathbb{R}^n$ such that:

$$\text{NPV}^R(\mathbf{p}_{\max,R}) = \max_{\mathbf{p}} \{ \text{NPV}^R(\mathbf{p}) \} , \quad (6.8)$$

where NPV^R is the averaged sum of the NPVs of a given well configuration represented by a vector of parameters \mathbf{p} over all the realizations:

$$\text{NPV}^R(\mathbf{p}) = \frac{1}{\mathcal{N}_r} \sum_{i=1}^{\mathcal{N}_r} \text{NPV}(\mathbf{p}, R_i) . \quad (6.9)$$

In the proposed approach, we define a so-called estimated objective function that will be optimized instead of the true objective function NPV^R defined in Eq. (6.9). The estimated function will be denoted in the sequel by NPV^E . Thus in the proposed approach, contrary to what is shown in Eq. (6.8), we will try to find the vector of parameter $\mathbf{p}_{\max,E} \in \mathbb{R}^n$ such that:

$$\text{NPV}^E(\mathbf{p}_{\max,E}) = \max_{\mathbf{p}} \{ \text{NPV}^E(\mathbf{p}) \} . \quad (6.10)$$

6.2 Well placement under uncertainty with CMA-ES using the neighborhood

The simplest case in which solving Eq. (6.8) is equivalent to solving Eq. (6.10), is when NPV^E is a monotonic transformation of NPV^R . However in this thesis, we do not aim to define an estimated objective function NPV^E such that we can prove that Eq. (6.10) is equivalent to Eq. (6.8). Our aim is that by solving Eq. (6.10), we can propose good points with high NPV^R values (see below for the definition of NPV^E).

To optimize NPV^E , we propose to use the CMA-ES optimizer. During the optimization process, we build a database –called also training set– in which after every performed reservoir simulation for a given point \mathbf{x} on a realization R , we store the point \mathbf{x} together with its corresponding evaluation $\text{NPV}(\mathbf{x}, R)$.

It remains now to define the estimated objective function NPV^E for a given point (well configuration) denoted by a vector of parameters \mathbf{p} :

1. At the beginning of the optimization and until reaching a given number \mathcal{N}_{sim} of performed reservoir simulations, we define a number of reservoir simulations \mathcal{N}_s^1 ($\leq \mathcal{N}_r$) to be performed on \mathbf{p} , and a set of \mathcal{N}_s^1 randomly drawn integers $\{j_1, \dots, j_{\mathcal{N}_s^1}\} \subset \{1, \dots, \mathcal{N}_r\}$. We perform then \mathcal{N}_s^1 reservoir simulations on \mathbf{p} on the realizations $(R_i)_{i=j_1, \dots, j_{\mathcal{N}_s^1}}$, and we add each of the obtained simulation results $(\mathbf{p}, \text{NPV}(\mathbf{p}, R_i))$ to the training set.

The estimated objective function on the point \mathbf{p} reads as follows:

$$\text{NPV}^E(\mathbf{p}) = \frac{1}{\mathcal{N}_s^1} \sum_{i=1}^{\mathcal{N}_s^1} \text{NPV}(\mathbf{p}, R_{j_i}) . \quad (6.11)$$

In this case, the evaluation of NPV^E requires a number \mathcal{N}_s^1 of reservoir simulations.

2. If more than \mathcal{N}_{sim} reservoir simulations are performed, we perform the following steps.

We begin by defining a number of reservoir simulations \mathcal{N}_s^2 ($\leq \mathcal{N}_r$) to be performed on \mathbf{p} , and a set of randomly drawn integers $\{j_1, \dots, j_{\mathcal{N}_s^2}\} \subset \{1, \dots, \mathcal{N}_r\}$. We perform then \mathcal{N}_s^2 reservoir simulations on \mathbf{p} on the realizations $(R_i)_{i=j_1, \dots, j_{\mathcal{N}_s^2}}$, and we add each of the obtained simulation results $(\mathbf{p}, \text{NPV}(\mathbf{p}, R_i))$ to the training set.

We also define a maximum number of neighbor points $\mathcal{N}_{n, \text{max}} \in \mathbb{N}$ that can be used in the definition of NPV^E . We select then at most the $\mathcal{N}_{n, \text{max}}$ nearest points to \mathbf{p} from the training set. Here, we select only the points with a distance less or equal to a given fixed distance of selection denoted by d_{max} . We denote by \mathcal{N}_n the number of selected points and by $(\mathbf{x}_i)_{1 \leq i \leq \mathcal{N}_n}$ the selected points¹. The distance used for this

¹For each selected point \mathbf{x}_i for the training set, we have a corresponding evaluation on a given realization. For the sake of notation simplicity we will denote the corresponding stored evaluation by $\text{NPV}(\mathbf{x}_i, R_i)$ although it is not necessarily evaluated on realization R_i .

6.3 Application of CMA-ES using the neighborhood approach on the PUNQ-S3 case

purpose is the Mahalanobis distance with respect to the current covariance matrix \mathbf{C} of CMA-ES defined for two given points $\mathbf{z}_1 \in \mathbb{R}^n$ and $\mathbf{z}_2 \in \mathbb{R}^n$ by $d_{\mathbf{C}}(\mathbf{z}_1, \mathbf{z}_2) = \sqrt{(\mathbf{z}_1 - \mathbf{z}_2)^T \mathbf{C}^{-1} (\mathbf{z}_1 - \mathbf{z}_2)}$.

The estimated objective function on \mathbf{p} reads as follows:

$$\text{NPV}^E(\mathbf{p}) = \frac{1}{S} \left[\sum_{i=1}^{\mathcal{N}_s^2} (p_i \text{NPV}(\mathbf{p}, R_{j_i})) + \sum_{i=1}^{\mathcal{N}_n} (\tilde{p}_i \text{NPV}(\mathbf{x}_i, R_i)) \right], \quad (6.12)$$

where $p_i = 1$, $\tilde{p}_i = \left(1 - \left(\frac{d_{\mathbf{C}}(\mathbf{x}_i, \mathbf{p})}{d_{\max}}\right)^2\right)^2$ and $S = \sum_{i=1}^{\mathcal{N}_s^2} p_i + \sum_{i=1}^{\mathcal{N}_n} \tilde{p}_i$. In this case, the evaluation of NPV^E requires only a number \mathcal{N}_s^2 of reservoir simulations.

The parameters \mathcal{N}_{sim} , \mathcal{N}_s^1 , \mathcal{N}_s^2 and $\mathcal{N}_{n, \max}$ are not meant to be in the users' choice. Typical values are $\mathcal{N}_{n, \max} = 2 \times \mathcal{N}_r$, $\mathcal{N}_{\text{sim}} = 2 \times \mathcal{N}_r$, $\mathcal{N}_s^1 = 1$ and $\mathcal{N}_s^2 = 1$. A users' choice is the maximum distance of selection for the neighborhood d_{\max} , and which is a problem-dependent constant. An investigation of the impact of the choice of d_{\max} will be briefly shown in the next section through some examples.

An estimated standard deviation can also be included in the formulation of the estimated objective function NPV^E . In this case, the estimated objective function, which will not be tested in this chapter, can be formulated as follows:

$$\text{NPV}^E(\mathbf{p}) = m^E + r\sigma^E(\mathbf{p}), \quad (6.13)$$

where:

$$m^E = \frac{1}{S} \left[\sum_{i=1}^{\mathcal{N}_s^2} (p_i \text{NPV}(\mathbf{p}, R_{j_i})) + \sum_{i=1}^{\mathcal{N}_n} (\tilde{p}_i \text{NPV}(\mathbf{x}_i, R_i)) \right], \quad (6.14)$$

and

$$\sigma^E(\mathbf{p}) = \sqrt{\frac{1}{S} \left[\sum_{i=1}^{\mathcal{N}_s^2} (p_i (\text{NPV}(\mathbf{p}, R_{j_i}) - m^E)^2) + \sum_{i=1}^{\mathcal{N}_n} (\tilde{p}_i (\text{NPV}(\mathbf{x}_i, R_i) - m^E)^2) \right]}. \quad (6.15)$$

6.3 Application of CMA-ES using the neighborhood approach on the PUNQ-S3 case

In this section, we apply the CMA-ES using the neighborhood approach –that we will call in the sequel the “using the neighborhood” approach– on the well placement problem on the benchmark reservoir case PUNQ-S3 [54]. As shown in previous examples in Chapters 3 and 5, the model contains $19 \times 28 \times 5$ grid blocks, and the elevation and the geometry of the field is shown in Fig. 3.2. We consider 20 geological realizations that will

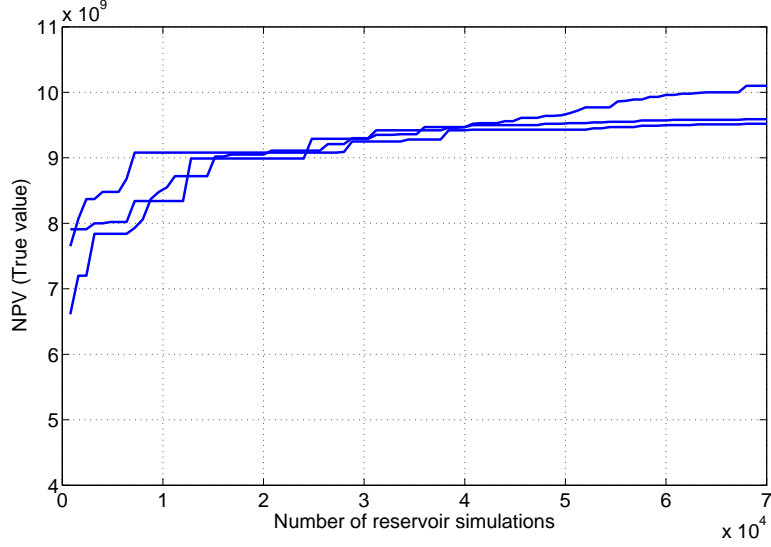


Figure 6.1: The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the mean of samples” approach. The best mean value of the NPV over the 20 possible realizations, i.e., NPV^R is shown. Three runs are performed.

be again denoted by $(R_i)_{i=1, \dots, 20}$. Each realization defines one possible porosity map and one possible permeability map. In these examples, the number of realization N_r is then equal to 20.

We plan to drill two wells: one unilateral injector and one unilateral producer. The dimension of the problem is then equal to $12(= 6 \times 2)$. In all the following applications, we use CMA-ES as an optimization algorithm with a population size equal to 40.

As a reference approach, we perform three independent runs in which we optimize the objective function NPV^R as defined in Eq. (6.9). In this reference approach, we perform for each well configuration to be evaluated 20 reservoir simulations. The reference approach will be called in the sequel the “using the mean of samples” approach. Fig. 6.1 shows the evolution of the best mean value of NPV^R , i.e., the NPV over the 20 possible realizations, for the three performed runs. The “using the mean of samples” approach is shown to be able to reach a mean value of NPV^R equal to $\$9 \times 10^9$ using 15200 reservoir simulations. It is able also to reach a mean value of NPV^R equal to $\$9.3 \times 10^9$ using 31200 reservoir simulations and a mean value of NPV^R equal to $\$9.5 \times 10^9$ using 44400 reservoir simulations.

To evaluate the “using the neighborhood” approach, we use typical values of the parameters N_{sim} , N_s^1 , N_s^2 and $N_{n,max}$ as defined in Section 6.2, i.e., $N_{n,max} = 2 \times N_r$, $N_{sim} = 2 \times N_r$, $N_s^1 = 1$ and $N_s^2 = 1$. We begin by choosing the maximum distance of selection for the neighborhood d_{max} equal to 4000.

Fig. 6.2 shows the evolution of the optimization process for three independent runs

6.3 Application of CMA-ES using the neighborhood approach on the PUNQ-S3 case

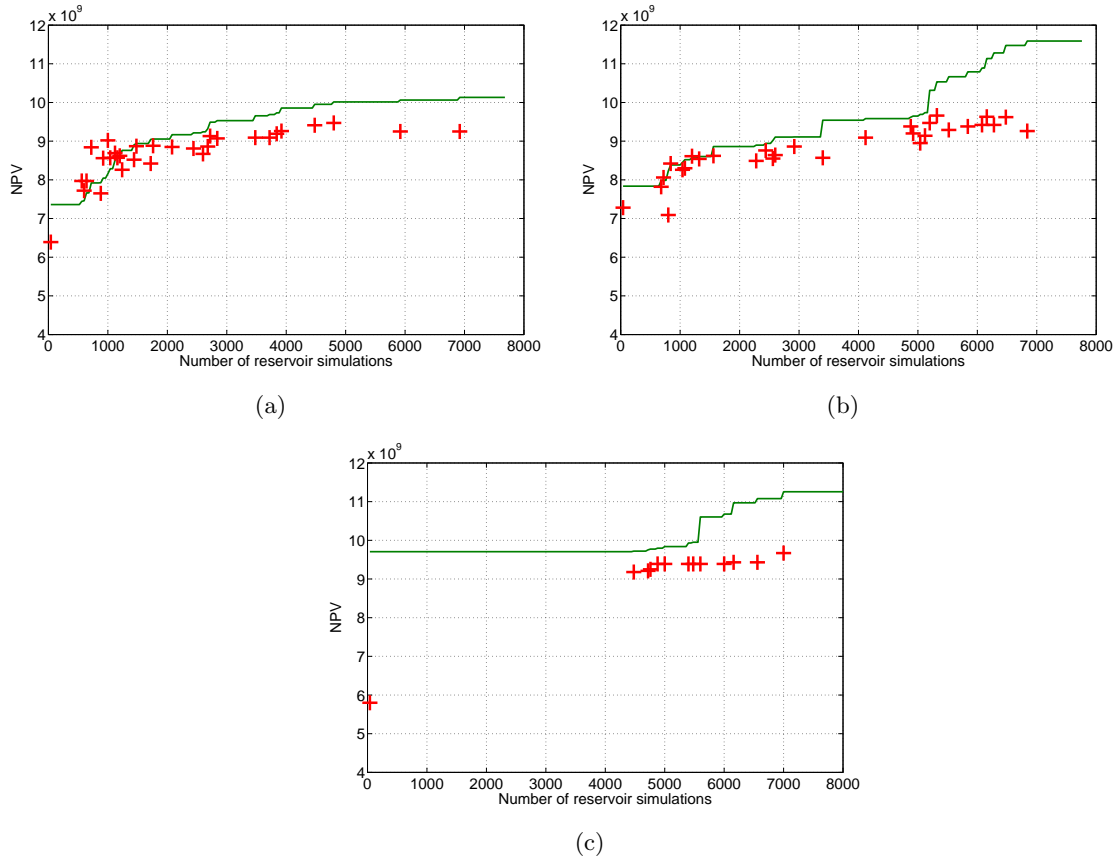


Figure 6.2: The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the neighborhood” approach, for three independent runs in (a), (b) and (c). The evolutions of the best estimated objective function, i.e., NPV^E are drawn with green lines. The evaluations on the true objective function over the 20 possible realizations, i.e., NPV^R are depicted with red crosses. The maximum distance of selection for the neighborhood d_{max} is equal to 4000.

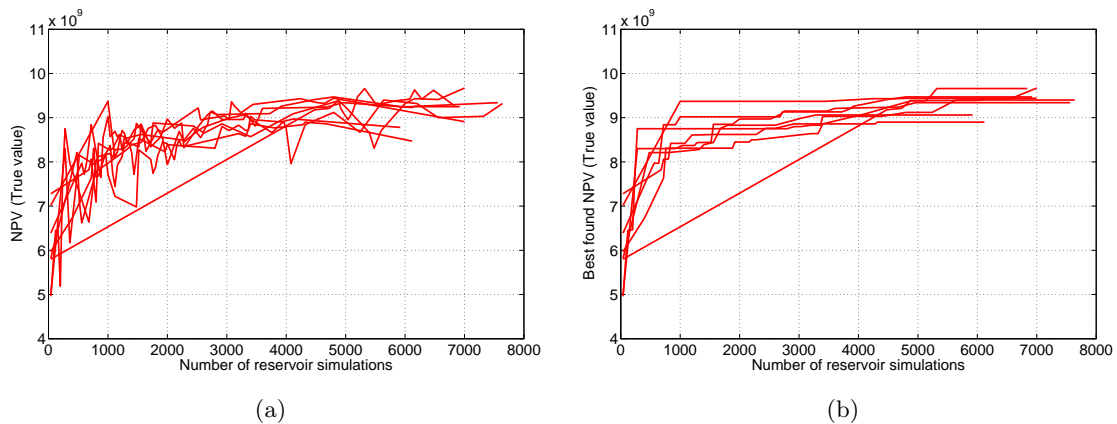


Figure 6.3: The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the neighborhood” approach for eight independent runs. (a) shows the evolution of the evaluations on NPV^R . (b) shows the evolution of the best found evaluation on NPV^R . The maximum distance of selection for the neighborhood d_{max} is equal to 4000.

6.3 Application of CMA-ES using the neighborhood approach on the PUNQ-S3 case

of CMA-ES with the “using the neighborhood” approach. The evolutions of the best estimated objective function, i.e., NPV^E are drawn with green lines. During the optimization process, each new overall best point found on NPV^E , is evaluated on NPV^R . The evaluations performed on NPV^R are depicted with red crosses. Fig. 6.2 shows that when optimizing NPV^E , we are able to propose good points according to NPV^R (points with an NPV^R greater than $\$9 \times 10^9$). Moreover, NPV^R tends to increase with an increasing number of performed reservoir simulations.

Fig. 6.2(c) shows a particular run in which the best NPV^E value found at the first generation is equal to $\$9.7 \times 10^9$. This value is calculated according to Eq. (6.11), and thus calculated using only one single reservoir simulation (with one single random realization). Indeed, with a single reservoir simulation to evaluate one point, the estimated objective function can not in general propose a good point according to NPV^R . Consequently, the best point found at the first generation according to NPV^E has a “bad” NPV^R value equal to $\$5.8 \times 10^9$. Thus, the optimization process does not propose for 112 iterations a new overall best point to be evaluated on NPV^R . The performance of this run can be avoided either by evaluating more often points using NPV^{R1} or simply by using more reservoir simulations for each point to be evaluated at the beginning of the optimization, i.e., choosing $\mathcal{N}_s^1 \geq 2$.

We show in Fig. 6.3 the performance of eight independent runs of CMA-ES with the “using the neighborhood” approach. Fig. 6.3(a) shows the evolution of the evaluations performed on NPV^R . The evaluated points correspond to the best overall points found during the optimization process of NPV^E . Fig. 6.3(b) shows the evolution of the best evaluation performed on NPV^R . Seven runs out of the eight performed runs (88%) are able to reach an NPV^R value greater than to $\$9 \times 10^9$, using a mean number of reservoir simulations equal to 2851. Consequently the reduction of the number of reservoir simulations to reach an NPV^R greater than to $\$9 \times 10^9$ when using the “using the neighborhood” approach compared to the “using the mean of samples” approach is equal to 81%. Six runs out of eight performed runs (75%) are able to reach a value of NPV^R greater than to $\$9.3 \times 10^9$, using a mean number of reservoir simulations equal to 4307, which offers a reduction of the number of reservoir simulations when comparing to the “using the mean of samples” approach equal to 86%. However, only two runs out of the eight performed runs (25%) are able to reach a value of NPV^R greater than to $\$9.5 \times 10^9$. The mean number of reservoir simulations required to reach this value is 6160. Consequently the reduction of the number of reservoir simulations to reach an NPV^R greater than to $\$9.5 \times 10^9$ when comparing to the “using the mean of samples” approach is again equal to 86%.

¹For example, one can evaluate the best found point according to NPV^E at each iteration on NPV^R

6.3 Application of CMA-ES using the neighborhood approach on the PUNQ-S3 case

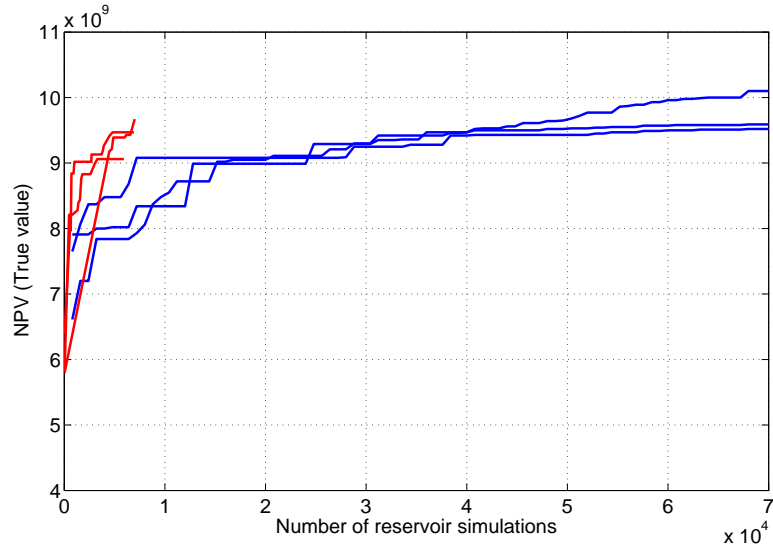


Figure 6.4: The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the mean of samples” approach and the “using the neighborhood” approach. The evolution of the best found evaluation on NPV^R for the “using the neighborhood” approach is drawn with red lines. The evolution of the best found evaluation on NPV^R for the “using the mean of samples” approach is drawn with blue lines. Three independent runs are performed for each approach. For the “using the neighborhood” approach, the maximum distance of selection for the neighborhood d_{\max} is equal to 4000.

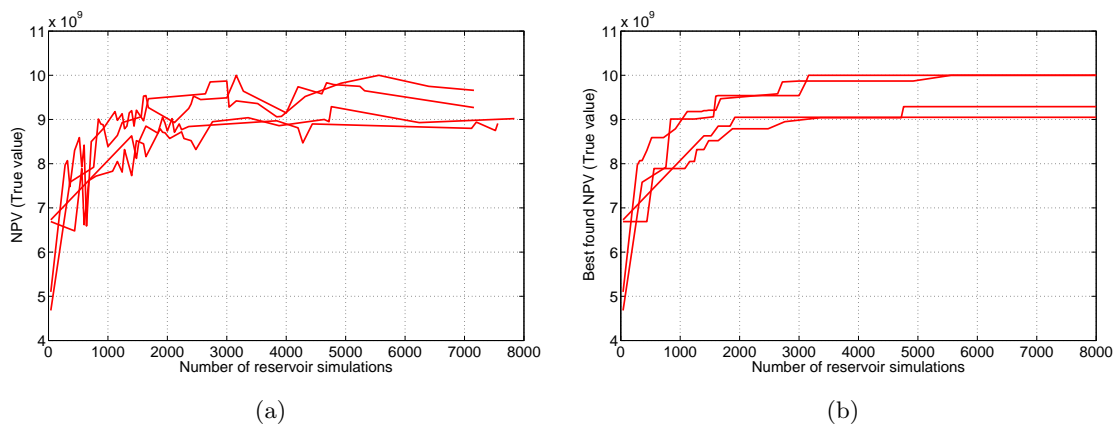


Figure 6.5: The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the neighborhood” approach for four independent runs. (a) shows the evolution of the evaluations on NPV^R . (b) shows the evolution of the best found evaluation on NPV^R . The maximum distance of selection for the neighborhood d_{\max} is equal to 3000.

6.3 Application of CMA-ES using the neighborhood approach on the PUNQ-S3 case

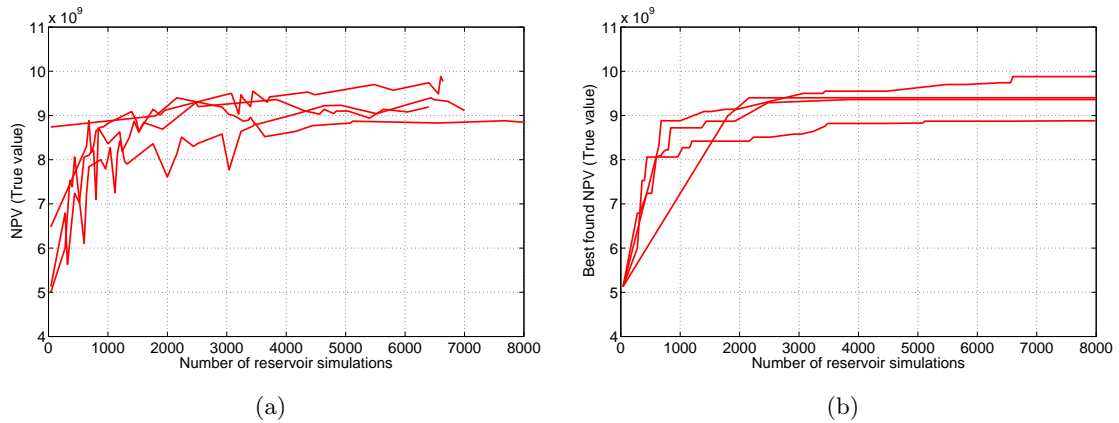


Figure 6.6: The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using the neighborhood” approach for four independent runs. (a) shows the evolution of the evaluations on NPV^R . (b) shows the evolution of the best found evaluation on NPV^R . The maximum distance of selection for the neighborhood d_{\max} is equal to 6000.

Three runs of CMA-ES with the “using the neighborhood” approach are shown together with the three performed runs of CMA-ES with the “using the mean of samples” approach in Fig. 6.4. Results show that although the “using the neighborhood” approach does not guarantee finding the best values of NPV^R found by the “using the mean of samples” approach when comparing with the “using the mean of samples” approach, the number of reservoir simulations is reduced significantly by more than 81%.

The impact of the choice of the maximum distance of selection for the neighborhood d_{\max} is shown in Figs. 6.5 and 6.6. Comparing the results in Figs. 6.5, 6.3 and 6.6 (with $d_{\max} = 3000, 4000$ and 6000) shows that the approach is not very sensitive to the choice of d_{\max} .

In the sequel, we compare the “using the neighborhood” approach with another approach in which the estimated objective function to be optimized is equal to an evaluation on a randomly chosen realization. This approach is called the “using one realization” approach. In this approach, we also evaluate on NPV^R only the overall new best points found on the estimated objective function. Figs. 6.7 and 6.8 show the evolution of the optimization process for three independent runs of CMA-ES with the “using one realization” approach. In Fig. 6.7, the evolutions of the best estimated objective function are again drawn with green lines. If we compare the “using the neighborhood” and the “using one realization” approaches through Figs. 6.2 and 6.7, it is clear that contrary to the “using the neighborhood” approach which is shown to be able to capture the geological uncertainty, the “using one realization” approach is shown to be not able to propose good points with high NPV^R . The three performed runs with the “using one realization” approach are not able to reach an NPV^R value greater than $\$9 \times 10^9$.

6.3 Application of CMA-ES using the neighborhood approach on the PUNQ-S3 case

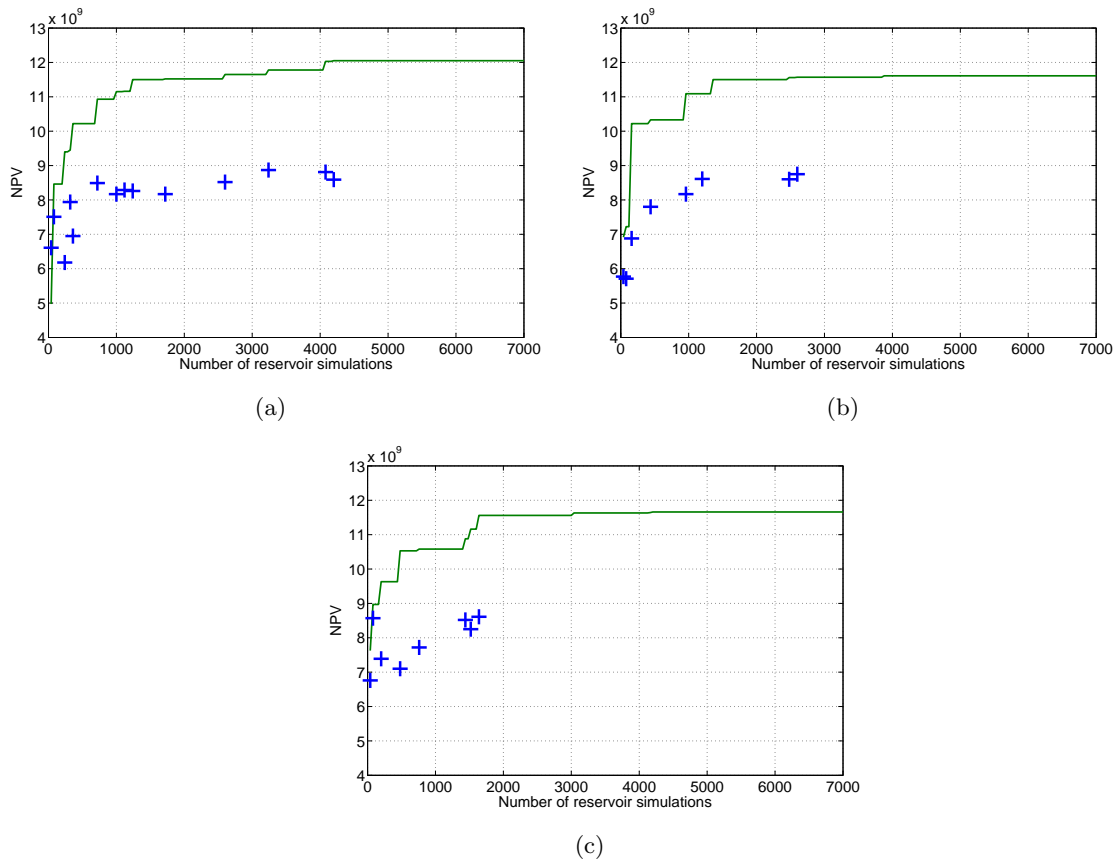


Figure 6.7: The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using one realization” approach, for three independent runs in (a), (b) and (c). The evolutions of the best estimated objective function (equal to an evaluation on a randomly chosen realization) are drawn with green lines. The evaluations on the true objective function over the 20 possible realizations, i.e., NPV^R are depicted with blue crosses.

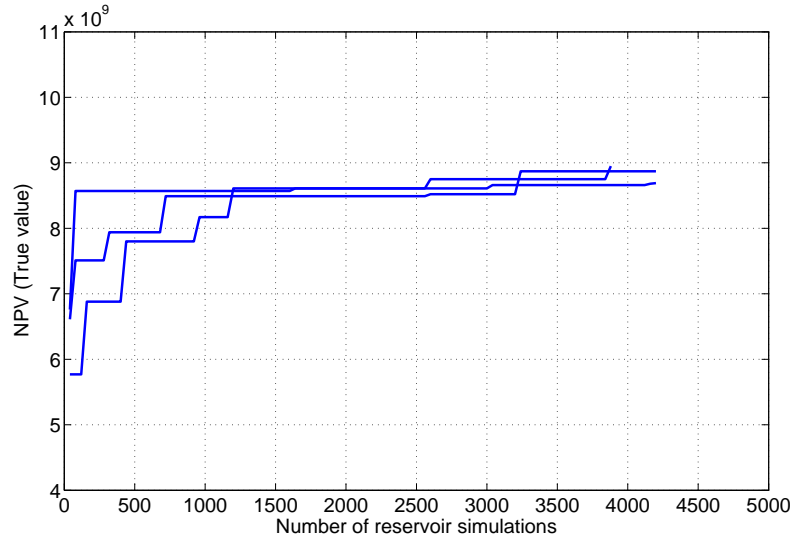


Figure 6.8: The evolution of the well placement optimization process on the PUNQ-S3 case using CMA-ES with the “using one realization” approach. The best mean value of the NPV over the 20 possible realizations, i.e., NPV^R is shown. Three runs are performed.

6.4 Summary and discussions

In this chapter, we have defined a new approach to handle geological uncertainty for well placement using the objective function evaluations of already evaluated individuals in the neighborhood. The proposed approach is compared to a reference approach using the mean of samples of each individual. We have shown on the synthetic benchmark reservoir case PUNQ-S3 that although the proposed approach does not guarantee finding always the best values found by the reference approach, the number of reservoir simulations is reduced significantly by more than 81%.

Chapter 7

Conclusions and perspectives

7.1 Conclusions

In this thesis, we have contributed to the research area of optimizing well placement (locations and trajectories of the wells to be drilled) by addressing the following challenges (presented in Section 1.3):

- (I) The non-smoothness, the multi-modality, the non-convexity and the high dimensionality of the objective function;
- (II) The expensive cost of the objective function;
- (III) The geological uncertainty handling problem.

The problem (I) was addressed in Chapter 3 by applying the stochastic optimizer CMA-ES. We have shown that CMA-ES outperforms the genetic algorithm on the PUNQ-S3 case by leading to a higher net present value (NPV). Moreover, CMA-ES was shown to be able to define potential regions containing optimal well configurations. The ability of CMA-ES to find much higher NPV values and to converge to the same region of the search space, has been explained by its advanced adaptation mechanism that allows the algorithm, on ill-conditioned non-separable problems, to adapt in an efficient way its sampling probability distribution.

The problem (II) was addressed by defining two new algorithms aiming at reducing the number of objective function evaluations, based on meta-models whose underlying idea is to replace some (true) function evaluations during the optimization process by the function values given by the meta-model. Meta-models can be considered as a computationally cheaper replacement of the objective function. This consideration is justified by the context of costly objective function for the well placement problem. The new-local-meta-model CMA-ES, denoted nlm-CMA (Chapter 2) was proposed in order to mitigate some defects of the already existing local-meta-model CMA-ES (lmm-CMA) when dealing with large population sizes. The partially separable local-meta-model CMA-ES, denoted

p-sep lmm-CMA (Chapter 4) was proposed leading to an important speedup compared to the standard CMA-ES when dealing with partially separable functions. Exploiting the partial separability of the objective function is motivated by the well placement problem, in which the reservoir simulations can output the production of each single well (assumed to be depending on a fewer parameters) though the objective is to maximize the production of all the wells. The proposed algorithms (nlmm-CMA and p-sep lmm-CMA) were then applied on the well placement problem in Chapters 3 and 5. Results had demonstrated the potential huge benefit of applying the proposed algorithms in reducing the number of objective function evaluations on the well placement problem, and which can be extended to other reservoir applications. The use of nlmm-CMA was shown on the benchmark reservoir case PUNQ-S3 to be able to offer similar results (solution well configurations and the corresponding NPV values) as CMA-ES without meta-models and moreover to reduce the number of simulations by around 20% to reach a satisfactory NPV. The use of p-sep lmm-CMA was also shown on PUNQ-S3 to lead to an important reduction of the number of reservoir simulations of around 60% compared to CMA-ES.

Finally, the problem (III) was addressed in Chapter 6 by defining a new approach to optimize under geological uncertainty with a reduced number of reservoir simulations. The approach uses the objective function evaluations of already simulated well configurations in the neighborhood of each well configuration. The proposed approach is shown to be able to capture the geological uncertainty using a reduced number of reservoir simulations. On the benchmark reservoir case PUNQ-S3, the proposed approach is able to reduce significantly the number of reservoir simulations by more than 80% compared to the reference approach using all the possible realizations for each well configuration. The research in the area of reducing the number of reservoir simulations when optimizing under geological uncertainty has been relatively scarce, and it is expected that an efficient algorithm such as the proposed one, should provide a renewed interest in this area.

7.2 Perspectives

Several extensions to the present research can be mentioned. For the two algorithms nlmm-CMA and p-sep lmm-CMA defined respectively in Chapters 2 and 4, we have only focused on local quadratic meta-models. However other types of meta-models could be used like kriging and radial basis functions as we have no a priori that quadratic meta-models are the best models to use for practical purposes.

In Chapter 4 when defining p-sep lmm-CMA, we have shown on f_{Rosen}^{100} that using a population size $\lambda = 4 \times \lambda_{\text{default}}$ gives the minimum number of evaluations and improves the performance by a factor between 1.5 and 2 over the default population size. Future

investigations are needed to define the optimal population size depending on the dimension of the problem and the dimension of the sub-problems, over a wide range of test functions.

In Chapter 5 when applying p-sep lmm-CMA on the well placement problem, we have assumed that each element function depends on the parameters defining the considered well and the minimum distances between the considered well and the other wells. However, further studies are useful in order to define the best parameters for each element function and thus improve the accuracy of the partially separated meta-models.

Moreover, the work in Chapters 4 and 5 was motivated by the fact that we need to exploit more information on the well placement problem and that we need to incorporate these information into the original algorithm in order to obtain consequent improvements. Chapters 4 and 5 deal with the partial separability of the objective function. Another approach could be to exploit some a priori information such as well allocation factors and connectivity using the work developed in [41].

The problem of optimization under geological uncertainty remains an open research problem. The approach proposed in Chapter 6 can be considered as an initial work to define a robust optimization algorithm handling the uncertainty with a restricted budget of function evaluations. This work can be extended and enhanced by numerous means, mainly by using an adaptive strategy to define the parameters of the algorithm. Although we suspect that the used values of the parameters defining the number of performed reservoir simulations \mathcal{N}_s^1 and \mathcal{N}_s^2 (equal to one) can be a good choice, we believe that a procedure to adapt the number of function evaluations can improve the performance of the approach and generalize it to a wider range of problems. Indeed, the adaptation of \mathcal{N}_s^1 and \mathcal{N}_s^2 aims at controlling the quality of the estimated function and thus at deciding whether the estimated function is sufficiently accurate or if one needs more evaluations on other realizations to improve the quality of the estimated function. In Chapter 6, the proposed approach is compared to a reference approach using the mean of samples of each individual. However, more comparisons with recent competing approaches is recommended, in particular with the approach in [73].

Finally, applying the developed methods to other Geosciences optimization problems (e.g., history matching, production optimization) is suggested since we believe that many other Geosciences problems could be successfully handled with the developed methods.

Nomenclature

Included in the following list are common symbols employed throughout the thesis. Other, more specific symbols are defined as used.

\mathbf{C}	covariance matrix of a multivariate normal distribution
C_d	cost of drilling
di_i	minimum distance between the well i and the other injectors
dp_i	minimum distance between the well i and the other producers
H_k	gross thickness of a layer with an index k
k	number of points used to build the meta-model
k_i	number of points used to build the sub-meta-model (approximating an element function) with an index i
L_{\max}	maximum well length
\mathbf{m}	mean of a multivariate normal distribution
n	dimension of the problem
n_b	number of individuals evaluated on the meta-model each iteration cycle
n_{ic}	number of iteration cycles needed to satisfy the meta-model acceptance criterion
n_{init}	number of initial individuals evaluated on the meta-model
n_{layers}	number of layers in the reservoir
n_M	number of element variables for an element function
\mathcal{N}_{jun}	number of junctions
\mathcal{N}_{lat}	number of laterals
\mathcal{N}_r	number of geological realizations
\mathcal{N}_s	number of samples (sample size)
\mathcal{N}_w	number of wells
\mathcal{N}_{wd}	number of wells already drilled
R	revenue
R_i	geological realization with an index i
S_o	oil saturation

Symbols

λ	number of individuals created each generation (population size)
μ	number of individuals of the current generation used to create the individuals of the next generation
ϕ	porosity
σ	step-size; standard deviation

Abbreviations

APR Annual Percentage Rate
BOE Barrel of Oil Equivalent
CPU Central Processing Unit
NPV Net Present Value

Superscripts

E estimated
 R true
 (g) at generation g

Résumé en Français

(extended abstract in French)

L'état de l'art de la gestion des réservoirs a été récemment fortement influencé par le développement technologique. De nos jours, les technologies de forage ont connu de grands progrès, en particulier le forage directionnel. Par conséquent, les ingénieurs de réservoir bénéficient de l'utilisation des différentes architectures de configurations des puits, à savoir des architectures verticales, horizontales, ainsi que des architectures plus complexes, afin d'améliorer la productivité du réservoir.

Les Environnements, les zones et les conditions dans lesquelles les champs de pétrole et de gaz sont actuellement découverts, sont beaucoup plus complexes et difficiles à exploiter. D'une part, les champs existants sont de plus en plus déplétés, et par conséquent plus marginaux. A moins d'avoir un moyen d'optimiser leurs productivités et de prendre des mesures correctives, il serait difficile de justifier de continuer à investir dans la production de ces champs existants pour des raisons économiques [14]. D'autre part, les nouvelles découvertes ont aussi besoin d'un schéma de production optimal pour être économiquement viables.

L'un des problèmes les plus importants qui doivent être abordés afin de maximiser la valeur liquidative d'un projet donné est de décider d'une façon optimale où forer les puits. Une décision de placement des puits affecte considérablement la récupération des hydrocarbures, et donc la valeur liquidative du projet. En général, une telle décision est difficile à prendre puisqu'un placement optimal dépend d'un grand nombre de paramètres tels que les hétérogénéités du réservoir, les failles et les fluides en place. En outre, la complexité des configurations des puits, e.g. les puits non-conventionnels, implique des

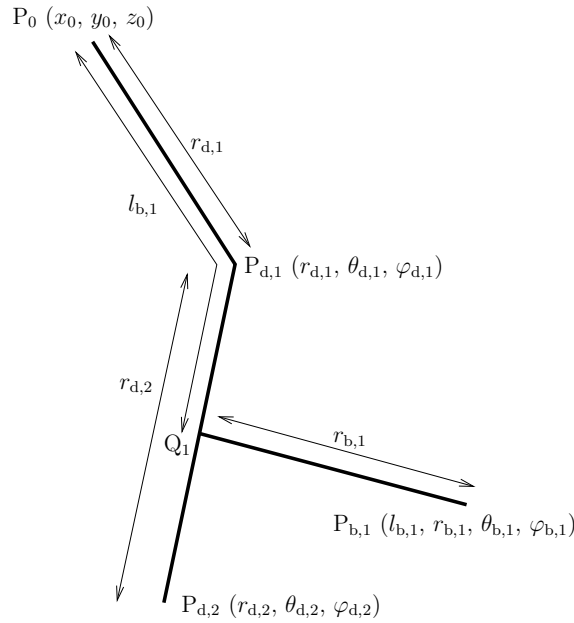


Figure .1: Un exemple de paramétrage d'un puits multilatéral ayant deux segments et une branche.

difficultés supplémentaires, telles que la concentration des investissements et la difficulté d'intervention sur les puits ¹.

La décision de placement des puits est formulée comme étant un problème d'optimisation:

- la fonction objectif à optimiser, qui est estimée en utilisant un simulateur de réservoir, évalue les aspects économiques du projet;
- les paramètres du problème encodent les positions des différents puits (qui comportent leurs emplacements et trajectoires).

Nous définissons l'emplacement d'un puits donné par la position de son point de départ, et nous définissons la trajectoire d'un puits donné par les positions de sa *bore* principale et les différents latérales (le cas échéant).

Si le nombre des puits à placer ainsi que leurs types (injecteur ou producteur) sont fixés, les paramètres encodant les positions des puits sont des nombres réels, et la fonction objectif f est une fonction d'un sous-ensemble de \mathbb{R}^n , où n , qui correspond au nombre de paramètres, est égal à la somme des nombres de paramètres nécessaires pour encoder chaque position de puits à placer. Un exemple de paramétrage d'un puits multilatéral est représenté dans la figure. .1.

¹Le forage d'un puits coûte en général entre 1 million de dollars et 30 millions de dollars.

Formellement, nous cherchons à trouver un vecteur de paramètres $\mathbf{p}_{\max} \in \mathbb{R}^n$ tel que:

$$f(\mathbf{p}_{\max}) = \max_{\mathbf{p}} \{f(\mathbf{p})\} \quad , \quad (.1)$$

où \mathbf{p} désigne le vecteur de paramètres à optimiser encodant les positions et les trajectoires des configurations des puits.

L'objectif principal de cette thèse est de proposer une procédure permettant de résoudre le problème d'optimisation de placement des puits, en particulier l'emplacement des puits et leurs trajectoires (défini dans l'Eq. (.1)). La procédure proposée doit à la fois offrir la valeur liquidative maximale tout en utilisant un nombre techniquement abordable de simulations de réservoir.

Cela implique les défis suivants, à savoir:

- (I) L'irrégularité, la multimodalité, la non-convexité et la dimensionnalité élevée de la fonction objectif;
- (II) Le coût élevé de la fonction objectif;
- (III) Le problème de traitement des incertitudes géologiques.

Considérant l'état de l'art de l'optimisation, le choix de l'algorithme CMA-ES [74] semble a priori adapté pour attaquer le problème (I). En effet, CMA-ES est reconnu comme l'un des plus puissants optimiseurs sans-dérivés pour l'optimisation continue [70]. CMA-ES est à la fois un algorithme rapide et robuste de recherche locale, présentant une convergence linéaire sur des classes larges de fonctions, et un algorithme de recherche global, si on relance l'algorithme tout en augmentant la taille de population. CMA-ES, contrairement à la plupart des autres algorithmes évolutionnaires, est un algorithme quasi sans-paramètres ¹.

Dans l'industrie pétrolière, CMA-ES n'a été appliqué, au meilleur de notre connaissance, que dans deux études antérieure à ce travail: une caractérisation des conductivités de fracture pour l'inversion des tests de puits [32], une optimisation de placement des puits mais en utilisant des attributs simples (par exemple, les indices de productivité) [43]. Une application plus récente sur l'optimisation de placement des puits a été publiée dans [116, 115].

Pour s'attaquer au problème (II), nous proposons d'étudier le couplage de l'optimiseur CMA-ES avec des *surrogates* (ou méta-modèles). Dans ce contexte, nous cherchons à définir une variante efficace de CMA-ES couplé avec des méta-modèles, capable de réduire significativement le nombre de simulations de réservoir. Par ailleurs, nous visons à exploiter

¹Seule la taille de la population est suggéré d'être ajustée par l'utilisateur, afin de tenir compte de la robustesse du paysage fonction objective.

les connaissances sur le problème d’optimisation, en particulier ladite séparabilité partielle de la fonction objectif afin de réduire davantage le nombre de simulations de réservoir.

Enfin, pour s’attaquer au problème (III), nous visons à définir une approche (pour CMA-ES) capable de capturer l’incertitude géologique avec un coût nettement réduit de simulations de réservoir. Dans ce contexte, nous cherchons à définir une approche qui utilise un nombre réduit de simulations de réservoir (typiquement un) pour chaque configuration des puits, au lieu d’effectuer des simulations de réservoir sur toutes les réalisations géologiques possibles.

Résumé des contributions

Dans ce qui suit, on présente un résumé des contributions de la présente thèse.

Nous avons abordé le problème (I) lié à l’irrégularité, la multimodalité, la non-convexité et la dimensionnalité élevée de la fonction objectif dans le problème de placement des puits, et nous avons présenté:

Une première application réussie de CMA-ES sur le problème de placement des puits. (Résultats publiés dans [26, 24])

Nous proposons une nouvelle méthodologie pour l’optimisation des positions et trajectoires des puits. Cette méthodologie basée sur des populations est un algorithme de recherche stochastique appelée “Covariance Matrix Adaptation - Evolution Strategy” (CMA-ES) [74]. Nous proposons d’utiliser une nouvelle technique de pénalisation adaptative avec rejet pour traiter les contraintes.

Vu que les algorithmes génétiques (AGs) sont très souvent la méthode choisie dans l’industrie pétrolière, nous montrons la contribution de l’application de CMA-ES par rapport à un AG sur le cas réservoir de benchmark synthétique PUNQ-S3 [54]. Pour permettre une comparaison équitable, les deux algorithmes sont utilisés sans réglage de paramètres du problème, les réglages standard sont utilisés pour AG et les paramètres par défaut pour CMA-ES. Nous montrons que notre nouvelle approche est plus performante que l’algorithme génétique (voir figure. .2): généralement, elle conduit à la fois à une plus grande valeur actuelle nette (NPV) et à une réduction significative du nombre de simulations de réservoir nécessaire pour atteindre une bonne configuration de puits.

Après cette application de CMA-ES sur le problème de placement des puits, nous avons abordé le problème (II) liée au coût élevé de la fonction objectif, et nous avons proposé deux nouveaux algorithmes:

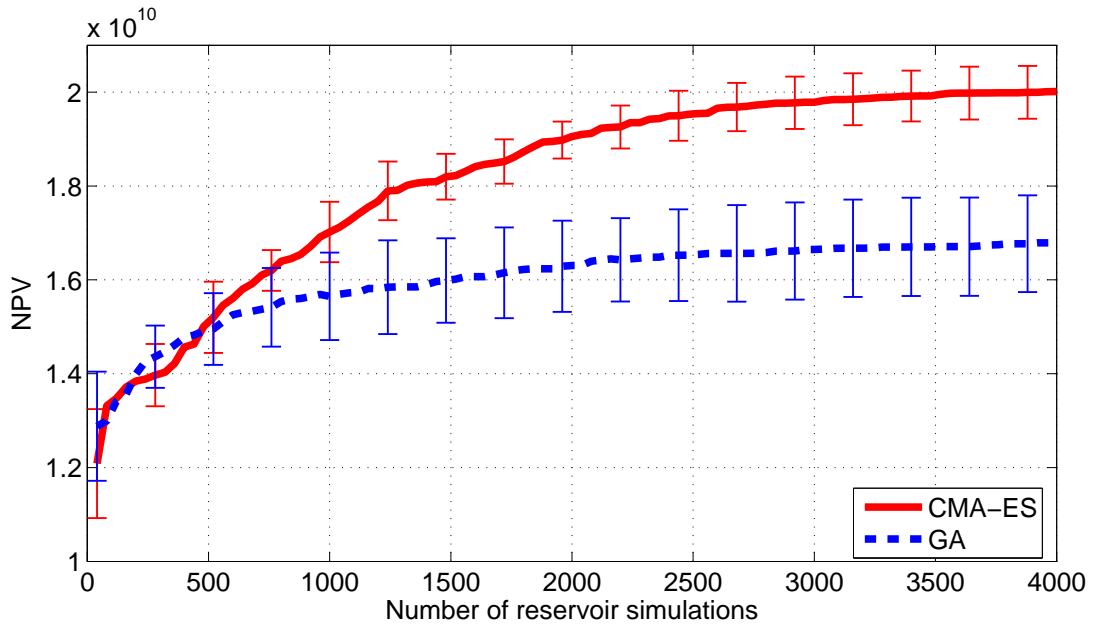


Figure .2: Valeur moyenne du NPV (en dollars) et l'écart type correspondant pour un problème de placement des puits utilisant CMA-ES (*courbe continue*) et AG (*courbe discontinue*). Quatorze tests sont effectués pour chaque algorithme.

Une nouvelle variante de CMA-ES avec des méta-modèles locaux. (Résultats publiés dans [22])

Le local-méta-modèle CMA-ES (Imm-CMA) [87], couplant des méta-modèles locaux quadratiques avec CMA-ES, est étudié. La dépendance de l'algorithme par rapport à la taille de population est analysée et les limites de l'approche pour des tailles de population plus grandes que celle par défaut, sont démontrées. Une nouvelle variante pour décider quand le méta-modèle est accepté, est proposée - appelé le nouveau-local-méta-modèle CMA-ES (nlmm-CMA).

Une nouvelle variante de CMA-ES avec des méta-modèles locaux pour les fonctions partiellement séparables. (Résultats publiés dans [23])

Nous proposons une nouvelle variante de CMA-ES avec des méta-modèles locaux pour l'optimisation des fonctions partiellement séparables - appelée la partiellement séparable local-méta-modèle CMA-ES (p-sep Imm-CMA). Nous proposons d'exploiter la séparabilité partielle en construisant à chaque itération un méta-modèle pour chaque fonction élément (ou *sous-fonction*) en utilisant un modèle quadratique complet local. Nos résultats démontrent que l'exploitation de la séparabilité partielle conduit à une accélération importante par rapport à l'algorithme CMA-ES standard. Nous montrons sur les fonctions testées que l'accélération augmente avec les dimensions, pour une dimension fixe de la fonction élément. Sur la fonction Rosenbrock standard, l'accélération maximale de λ est

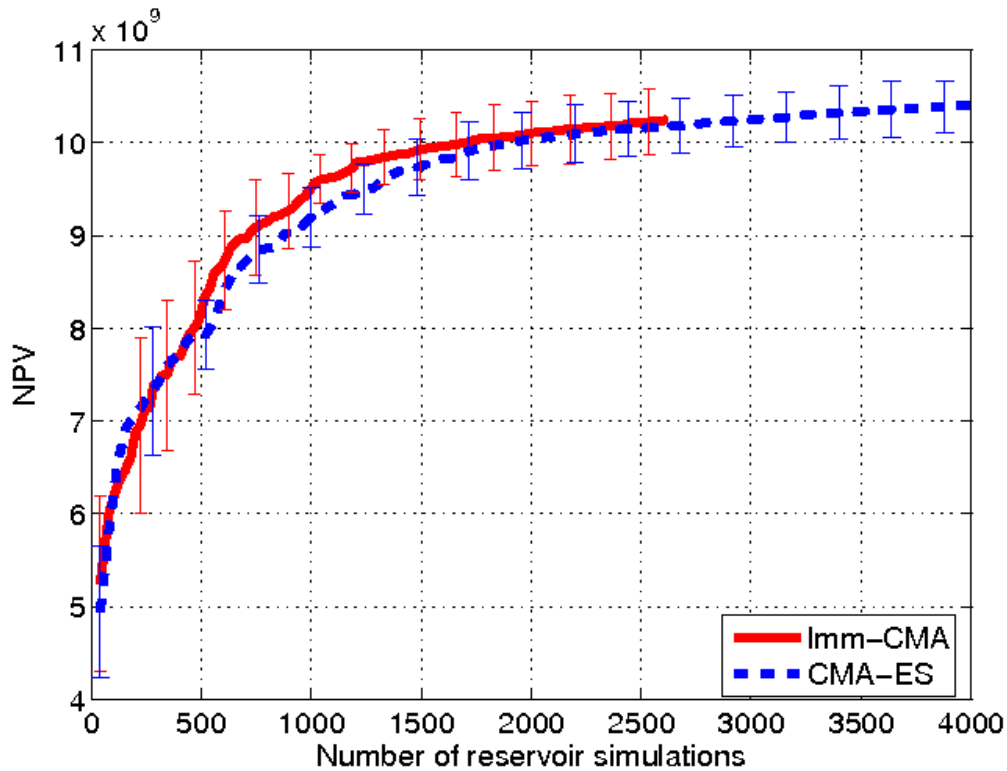


Figure .3: Valeur moyenne du NPV (en dollars) et l'écart type correspondant pour un problème de placement des puits utilisant CMA-ES avec des méta-modèles (*courbe continue*) et CMA-ES (*courbe discontinue*). Dix tests sont effectués pour chaque algorithme.

atteinte avec la dimension 40 en utilisant des fonctions éléments de dimension 2, où λ est la taille de population. Nous montrons également que des accélérations plus importantes peuvent être atteintes en augmentant la taille de population.

Maintenant, nous avons appliqué les deux nouveaux algorithmes proposés sur le problème de placement des puits:

Une réduction significative du nombre de simulations de réservoir pour le problème de placement des puits. (Résultats publiés dans [26, 24, 25])

Nous proposons d'appliquer CMA-ES avec des méta-modèles locaux (nlmm-CMA) sur le problème de placement des puits, où pour chaque configuration de la population, un modèle approché convexe quadratique est construit en utilisant les vraies évaluations de la fonction objectif collectées pendant le processus d'optimisation. Le couplage de CMA-ES avec les méta-modèles conduit à une amélioration significative, autour de 20 % pour le cas réservoir de benchmark synthétique PUNQ-S3 (voir figure. .3).

Par ailleurs, nous proposons également d'appliquer p-sep Imm-CMA sur le problème de placement des puits, en construisant des méta-modèles partiellement séparés pour chaque

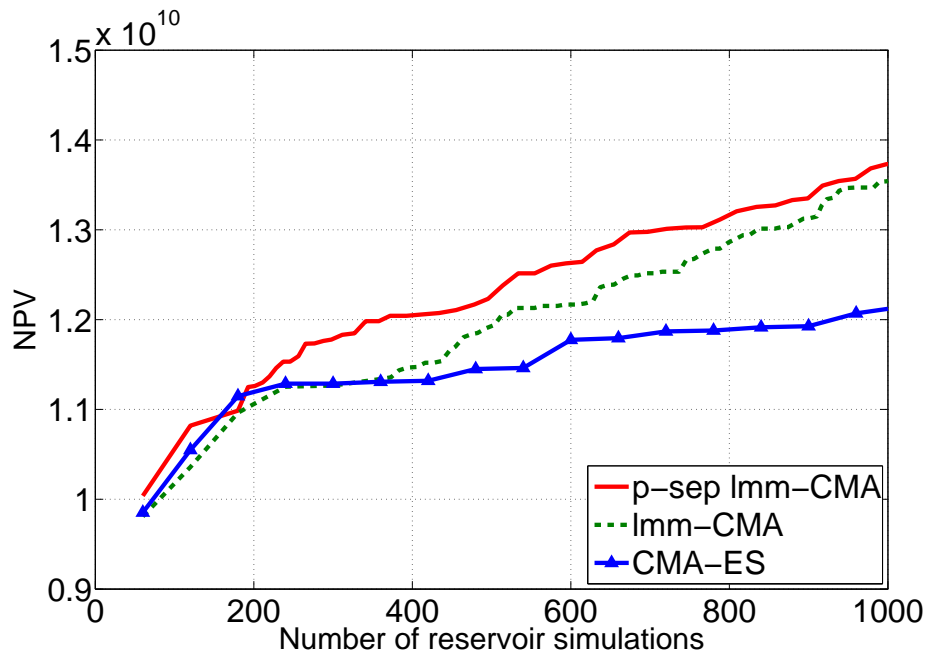


Figure .4: Valeur moyenne du NPV (en dollars) pour un problème de placement des puits utilisant CMA-ES avec des méta-modèles partiellement séparés, i.e., p-sep Imm-CMA (*courbe continue*), CMA-ES avec des méta-modèles, i.e., Imm-CMA (*courbe discontinue*) et CMA-ES (\triangle). Dix tests sont effectués pour chaque algorithme.

puits ou ensemble de puits, qui se traduit par une modélisation plus précise. Nos résultats montrent qu’en exploitant la séparabilité partielle de la fonction objectif, nous obtenons une diminution significative du nombre de simulations de réservoir nécessaire pour trouver la configuration “optimale” des puits, en considérant un budget restreint de simulations de réservoir (voir figure. .4).

Une nouvelle approche pour traiter l’incertitude géologique pour le problème de placement des puits.

Nous proposons une nouvelle approche pour traiter l’incertitude géologique pour le problème de placement des puits avec un nombre réduit de simulations de réservoir. Nous proposons d’utiliser une seule réalisation avec le voisinage de chaque configuration des puits afin d’estimer sa fonction objectif, au lieu d’utiliser multiples réalisations. L’approche est appliquée sur le cas réservoir de benchmark synthétique PUNQ-S3 et démontré être capable de capturer l’incertitude géologique en utilisant un nombre réduit de simulations de réservoir (voir figure. .5).

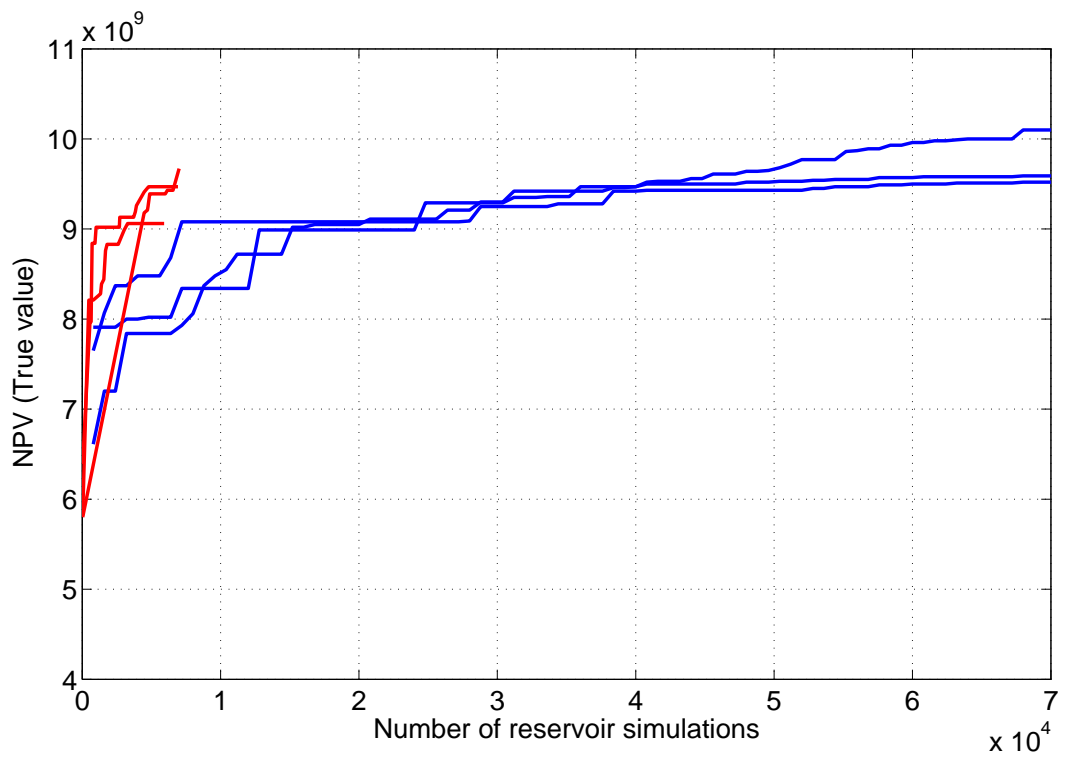


Figure .5: L'évolution du meilleur NPV (en dollars) obtenu pour un problème de placement des puits utilisant CMA-ES avec une approche utilisant la moyenne des échantillons (*courbe bleue*), et avec l'approche proposée utilisant le voisinage (*courbe rouge*). Trois tests sont démontrés pour chaque algorithme.

Perspectives

Plusieurs extensions de ces travaux présentés dans cette thèse peuvent être mentionnées. Pour les deux algorithmes nlmm-CMA et p-sep-lmm CMA définies respectivement dans les chapitres 2 et 4, nous nous sommes uniquement focalisé sur des méta-modèles quadratiques locaux. Cependant, d'autres types de méta-modèles peuvent être utilisés comme le krigeage et les fonctions à base radiale, puisque que nous n'avons aucun a priori que les méta-modèles quadratiques sont les meilleurs modèles à utiliser dans la pratique.

Dans le chapitre 4, lors de la définition p-sep-lmm CMA, nous avons montré sur la fonction f_{Rosen}^{100} qu'une taille de population $\lambda = 4 \times \lambda_{(\text{par défaut})}$ offre le nombre minimum d'évaluations, et améliore ainsi les performances de l'algorithme avec un facteur entre 1.5 et 2 par rapport à l'utilisation de la taille de population par défaut. D'autres investigations sont nécessaires pour définir la taille de population optimale en fonction de la dimension du problème et de la dimension des sous-problèmes, sur une large gamme de fonctions test.

Dans le chapitre 5, lors de l'application de p-sep-lmm CMA sur le problème de placement des puits, nous avons supposé que chaque élément fonction dépend des paramètres du puits considéré, ainsi que des distances minimales entre le puits considéré et les autres puits. Cependant, des études complémentaires sont utiles afin de définir les meilleurs paramètres pour chaque élément fonction, et d'améliorer ainsi la précision des méta-modèles partiellement séparés.

En outre, le travail dans les chapitres 4 et 5 a été motivé par le fait que nous avons besoin d'exploiter le maximum possible d'informations sur le problème de placement des puits, et que nous avons besoin d'intégrer ces informations dans l'algorithme original afin d'obtenir des améliorations significatives. Les chapitres 4 et 5 traitent la séparabilité partielle de la fonction objectif. Une autre approche pourrait consister à exploiter certaines informations a priori comme les facteurs d'allocation des puits et la connectivité, en utilisant les travaux développés dans [41].

Le problème d'optimisation sous incertitude géologique demeure un problème de recherche ouvert. L'approche proposée dans le chapitre 6 peut être considéré comme un travail initial visant à définir un algorithme d'optimisation robuste pouvant gérer les incertitudes avec un budget restreint d'évaluations de la fonction objectif. Ce travail peut être étendu par de nombreux moyens, notamment en utilisant une stratégie adaptative pour définir les paramètres de l'algorithme. Bien que nous soupçonnons que les valeurs des paramètres utilisées pour définir le nombre de simulations de réservoir réalisées \mathcal{N}_s^1 et \mathcal{N}_s^2 (égal à un) peuvent représenter un bon choix, nous considérons qu'une procédure adaptative du nombre d'évaluations de la fonction objectif peut améliorer la performance de l'approche, et ainsi la généraliser à un plus large éventail de problèmes. En effet,

l'adaptation de \mathcal{N}_s^1 et \mathcal{N}_s^2 , vise à contrôler la qualité de la fonction objectif estimée, et donc à décider si la fonction estimée est suffisamment précise ou si des évaluations sur d'autres réalisations sont nécessaires pour améliorer la qualité de la fonction estimée.

Enfin, l'application des méthodes développées sur d'autres problèmes d'optimisation en Géosciences (e.g., calage d'historiques, optimisation de la production) est suggérée car nous pensons que de nombreux autres problèmes en Géosciences pourraient être traités avec succès avec les méthodes mises au point dans cette thèse.

References

- [1] S. I. Aanonsen, A. L. Eide, L. Holden, and J. O. Aasen. Optimizing reservoir performance under uncertainty with application to well location. In *SPE Annual Technical Conference and Exhibition*, number SPE 30710, October 1995.
- [2] I. Aitokhuehi, L. J. Durlofsky, V. Artus, B. Yeten, and K. Aziz. Optimization of advanced well type and performance. In *9th European Conference on the Mathematics of Oil Recovery (ECMOR IX)*. EAGE, August 2004.
- [3] A. Aizawa and B. W. Wah. Dynamic control of genetic algorithms in a noisy environment. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 48–55, 1993.
- [4] A. Aizawa and B. W. Wah. Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, 2:97–122, 1994.
- [5] A. H. Alhuthali, A. Datta-Gupta, B. Yuen, and J. P. Fontanilla. Optimizing smart well controls under geologic uncertainty. *Journal of Petroleum Science and Engineering*, 73(1-2):107 – 121, 2010.
- [6] D. V. Arnold. Optimal weighted recombination. In A. H. Wright, M. D. Vose, K. A. De Jong, and L. M. Schmitt, editors, *Foundations of Genetic Algorithms*, volume 3469 of *Lecture Notes in Computer Science*, pages 215–237. Springer Berlin / Heidelberg, 2005.
- [7] D. V. Arnold and H.-G. Beyer. Efficiency and mutation strength adaptation of the $(\mu/\mu_i, \lambda)$ -ES in a noisy environment. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature PPSN VI*, volume 1917 of *Lecture Notes in Computer Science*, pages 39–48. Springer Berlin / Heidelberg, 2000.
- [8] D. V. Arnold and H.-G. Beyer. Local performance of the $(\mu/\mu_I, \lambda)$ -ES in a noisy environment. *Foundations of Genetic Algorithms 6*, pages 127–141, 2001.

-
- [9] L. Arnold, A. Auger, N. Hansen, and Y. Ollivier. Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles. *ArXiv e-prints*, June 2011.
- [10] V. Artus, L. J. Durlofsky, J. Onwunalu, and K. Aziz. Optimization of nonconventional wells under uncertainty using statistical proxies. *Computational Geosciences*, 10:389–404, 2006.
- [11] A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *IEEE Congress on Evolutionary Computation*, pages 1777–1784, 2005.
- [12] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1769–1776. IEEE Press, 2005.
- [13] A. Auger and N. Hansen. Approximate evolution strategy using stochastic ranking. In *IEEE Congress on Evolutionary Computation*, pages 745–752, 2006.
- [14] T. Babadagli. Development of mature oil fields - a review. *Journal of Petroleum Science and Engineering*, 57(3-4):221 – 246, 2007.
- [15] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1:1–23, 1993.
- [16] W. Bangerth, H. Klie, V. Matossian, M. Parashar, and M. F. Wheeler. An automatic reservoir framework for the stochastic optimization of well placement. *Cluster Computing*, 8(4):255–269, 2005.
- [17] W. Bangerth, H. Klie, M. F. Wheeler, P. L. Stoffa, and M. K. Sen. On optimization algorithms for the reservoir oil well placement problem. *Computational Geosciences*, 10(3):303–319, 2006.
- [18] P. Bayer and M. Finkel. Evolutionary algorithms for the optimization of advective control of contaminated aquifer zones. *Water Resources Research*, 40(6), 2004.
- [19] B. L. Beckner and X. Song. Field development planning using simulated annealing - optimal economic well scheduling and placement. In *SPE annual technical conference and exhibition*, number SPE 30650, October 1995.
- [20] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural and Multidisciplinary Optimization*, 17(1):1–13, 1999.

-
- [21] A. Bouaricha and J. J. Morè. Impact of partial separability on large-scale optimization. *Computational Optimization and Applications*, 7(1):27–40, 1997.
- [22] Z. Bouzarkouna, A. Auger, and D. Y. Ding. Investigating the local-meta-model CMA-ES for large population sizes. In C. Di Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A. Esparcia-Alcazar, C.-K. Goh, J. Merelo, F. Neri, M. Preuß, J. Togelius, and G. Yannakakis, editors, *Applications of Evolutionary Computation*, volume 6024 of *Lecture Notes in Computer Science*, pages 402–411. Springer Berlin / Heidelberg, 2010.
- [23] Z. Bouzarkouna, A. Auger, and D. Y. Ding. Local-meta-model CMA-ES for partially separable functions. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation, GECCO '11*, pages 869–876, New York, NY, USA, 2011. ACM.
- [24] Z. Bouzarkouna, D. Y. Ding, and A. Auger. Using evolution strategy with meta-models for well placement optimization. In *12th European Conference on the Mathematics of Oil Recovery (ECMOR XII)*. EAGE, 2010.
- [25] Z. Bouzarkouna, D. Y. Ding, and A. Auger. Partially separated meta-models with evolution strategies for well placement optimization. In *SPE EUROPEC/EAGE Annual Conference and Exhibition*, number SPE 143292, 2011.
- [26] Z. Bouzarkouna, D. Y. Ding, and A. Auger. Well placement optimization with the covariance matrix adaptation evolution strategy and meta-models. *Computational Geosciences*, 16(1):75–92, 2011.
- [27] J. Branke. Creating robust solutions by means of evolutionary algorithms. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, PPSN V*, pages 119–128, London, UK, 1998. Springer-Verlag.
- [28] J. Branke and C. Schmidt. Faster convergence by means of fitness estimation. *Soft Comput.*, 9(1):13–20, January 2005.
- [29] J. Branke, C. Schmidt, and H. Schmeck. Efficient fitness estimation in noisy environments. In *Proceedings of Genetic and Evolutionary Computation*, pages 243–250, 2001.
- [30] R. Brayton, S. Director, G. Hachtel, and L. Vidigal. A new algorithm for statistical circuit design based on quasi-newton methods and function splitting. *IEEE Transactions on Circuits and Systems*, 26(9):784–794, 1979.

-
- [31] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970.
- [32] J. Bruyelle and A. Lange. Automated characterization of fracture conductivities from well tests. In *SPE EUROPEC/EAGE annual conference and exhibition*, number SPE 121172, June 2009.
- [33] A. Y. Bukhamsin, M. M. Farshi, and K. Aziz. Optimization of multilateral well design and location in a real field using a continuous genetic algorithm. In *SPE/DGS Saudi Arabia Section Technical Symposium and Exhibition*, number SPE 136944, April 2010.
- [34] H. Chen and B. Schmeiser. Stochastic root finding via retrospective approximation. *IIE Transactions*, 33(3):259–275, 2001.
- [35] Y. Chen. *Ensemble-based closed-loop production optimization*. PhD thesis, University of Oklahoma, 2008.
- [36] Y. Chen and D. S. Oliver. Ensemble-based closed-loop optimization applied to brugge field. In *SPE reservoir simulation symposium*, number SPE 118926, February 2009.
- [37] Y. Chen, D. S. Oliver, and D. Zhang. Efficient ensemble-based closed-loop production optimization. *SPE Journal*, 14(4):634–645, 2009.
- [38] B. Colson and P. L. Toint. Optimizing partially separable functions without derivatives. *Optimization Methods and Software*, 20(4-5):493–508, 2005.
- [39] A. Cordoba, R. Vilar, A. Lavrov, A. B. Utkin, and A. Fernandes. Multi-objective optimisation of lidar parameters for forest-fire detection on the basis of a genetic algorithm. *Optics & Laser Technology*, 36(5):393 – 400, 2004.
- [40] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.
- [41] P. S. da Cruz, R. N. Horne, and C. V. Deutsch. The quality map: A tool for reservoir uncertainty quantification and decision making. *SPE Reservoir Evaluation & Engineering*, 7(1):6–14, 2004.
- [42] L. Damp and L. F. Gonzalez. Optimisation of the nose of a hypersonic vehicle using dsmc simulation and evolutionary optimisation. In *5th AIAA ASSC Space Conference*, September 2005.

-
- [43] D. Y. Ding. Optimization of well placement using evolutionary algorithms. In *SPE EUROPEC/EAGE annual conference and exhibition*, number SPE 113525, June 2008.
- [44] D. Y. Ding and F. Mckee. Using partial separability of the objective function for gradient-based optimizations in history matching. In *SPE reservoir simulation symposium*, number SPE 140811, February 2011.
- [45] N. Durand and J.-M. Alliot. Genetic crossover operator for partially separable functions. In *Proceedings of the third annual Genetic Programming Conference*, 1998.
- [46] T. A. El-Mihoub, A. A. Hopgood, L. Nolle, and A. Battersby. Hybrid genetic algorithms: A review. *Engineering Letters*, 13(2):124–137, 2006.
- [47] A. A. Emerick, E. Silva, B. Messer, L. F. Almeida, D. Szwarcman, M. A. C. Pacheco, and M. M. B. R. Vellasco. Well placement optimization using a genetic algorithm with nonlinear constraints. In *SPE reservoir simulation symposium*, number SPE 118808, February 2009.
- [48] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.
- [49] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *J. Geophys. Res.*, 99:10143–10162, 1994.
- [50] C. L. Farmer, J. M. Fowkes, and N. I. M. Gould. Optimal well placement. In *12th European Conference on the Mathematics of Oil Recovery (ECMOR XII)*. EAGE, 2010.
- [51] K. P. Ferentinos, K. G. Arvanitis, and N. Sigrimis. Heuristic optimization methods for motion planning of autonomous agricultural vehicles. *J. of Global Optimization*, 23(2):155–170, June 2002.
- [52] J. M. Fitzpatrick and J. J. Grefenstette. Genetic algorithms in noisy environments. *Machine Learning*, 3(2):101–120, 1988.
- [53] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.
- [54] F. J. T. Floris, M. D. Bush, M. Cuypers, F. Roggero, and A.-R. Syversveen. Methods for quantifying the uncertainty of production forecasts: A comparative study. *Petroleum Geoscience*, 7:87–96, 2001.

-
- [55] D. B. Fogel. An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks*, 5:3–14, 1994.
- [56] L. J. Fogel. Autonomous automata. *Industrial research*, 4:14–19, 1962.
- [57] F. Forouzanfar, G. Li, and A. C. Reynolds. A two-stage well placement optimization method based on adjoint gradient. In *SPE annual technical conference and exhibition*, number SPE 135304, September 2010.
- [58] A. I. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):50 – 79, 2009.
- [59] S. Gelly, S. Ruetten, and O. Teytaud. Comparison-based algorithms are robust and randomized algorithms are anytime. *Evolutionary Computation*, 15(4):411–434, December 2007.
- [60] H. Georg Beyer. Towards a theory of ‘evolution strategies’. some asymptotical results from the $(1, +\lambda)$ -theory. *Evolutionary Computation*, 1:165–188, 1993.
- [61] D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computing*, 24:23–26, 1970.
- [62] Y. Gu and D. S. Oliver. An iterative ensemble Kalman filter for multiphase fluid flow data assimilation. *SPE Journal*, 12:438–446, 2007.
- [63] B. Guyaguler and R. N. Horne. Optimization of well placement. *Journal of Energy Resources Technology*, 122(2):64–70, 2000.
- [64] B. Guyaguler and R. N. Horne. Uncertainty assessment of well placement optimization. In *SPE annual technical conference and exhibition*, number SPE 87663, September-October 2001.
- [65] B. Guyaguler, R. N. Horne, L. Rogers, and J. J. Rosenzweig. Optimization of well placement in a gulf of mexico waterflooding project. In *SPE annual technical conference and exhibition*, number SPE 63221, October 2000.
- [66] P. Hajela and E. Lee. Genetic algorithms in truss topological optimization. *International Journal of Solids and Structures*, 32(22):3341 – 3357, 1995.
- [67] M. Handels, M. J. Zandvliet, D. R. Brouwer, and J. D. Jansen. Adjoint-based well-placement optimization under production constraints. In *SPE reservoir simulation symposium*, number SPE 105797, February 2007.

-
- [68] N. Hansen. Adaptive encoding: How to render search coordinate system invariant. In G. Rudolph et al., editors, *Parallel Problem Solving from Nature - PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 205–214. Springer Berlin / Heidelberg, 2008.
- [69] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Research Report 6828, INRIA, 2009.
- [70] N. Hansen, A. Auger, and P. P. R. Ros, S. Finck. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In *GECCO '10: Proceedings of the 12th annual conference comp on Genetic and evolutionary computation*, pages 1689–1696, New York, NY, USA, 2010. ACM.
- [71] N. Hansen and S. Kern. Evaluating the CMA evolution strategy on multimodal test functions. In X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. Rowe, P. Tino, A. Kabán, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 282–291. Springer Berlin / Heidelberg, 2004.
- [72] N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation. *Evolutionary Computation*, 11(1):1–18, 2003.
- [73] N. Hansen, A. S. P. Niederberger, L. Guzzella, and P. Koumoutsakos. A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *IEEE Transactions on Evolutionary Computation*, 13(1):180–197, 2009.
- [74] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [75] N. Hansen, R. Ros, N. Mauny, M. Schoenauer, and A. Auger. Impacts of Invariance in Search: When CMA-ES and PSO Face Ill-Conditioned and Non-Separable Problems. *Applied Soft Computing*, 11:5755–5769, 2011.
- [76] V. Heidrich-Meisner and C. Igel. Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 401–408, New York, NY, USA, 2009. ACM.
- [77] A. D. Hill, D. Zhu, and M. J. Economides. *Multilateral Wells*. Society of Petroleum Engineers, 2008.

-
- [78] J. H. Holland. Outline for a logical theory of adaptive systems. *J. ACM*, 9(3):297–314, 1962.
- [79] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, 1975.
- [80] R. Hooke and T. A. Jeeves. "direct search" solution of numerical and statistical problems. *Journal of the association for computing machinery*, 8:212–229, 1961.
- [81] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 1988.
- [82] M. Jebalia, A. Auger, and N. Hansen. Log linear convergence and divergence of the scale-invariant (1+1)-ES in noisy environments. *Algorithmica*, 59(3):425–460, 2011. 10.1007/s00453-010-9403-3.
- [83] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005.
- [84] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments - a survey. *IEEE Trans. Evolutionary Computation*, 9(3):303–317, 2005.
- [85] C. V. D. K. P. Norrena. Automatic determination of well placement subject to geostatistical and economic constraints. In *SPE International Thermal Operations and Heavy Oil Symposium and International Horizontal Well Technology Conference*, number SPE 78996, November 2002.
- [86] J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE international conference on neural networks*, pages 1942–1948, 1995.
- [87] S. Kern, N. Hansen, and P. Koumoutsakos. Local meta-models for optimization using evolution strategies. In T. Runarsson, H.-G. Beyer, E. Burke, J. Merelo-Guervós, L. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature - PPSN IX*, volume 4193 of *Lecture Notes in Computer Science*, pages 939–948. Springer Berlin / Heidelberg, 2006.
- [88] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [89] D. Kourounis, D. Voskov, and K. Aziz. Adjoint methods for multicomponent flow simulation. In *12th European Conference on the Mathematics of Oil Recovery (EC-MOR XII)*. EAGE, 2010.

-
- [90] D. G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society*, 52:119–139, 1951.
- [91] H. Langouët. *optimisation sans dérivées sous contraintes*. PhD thesis, Université Nice - Sophia Antipolis, June 2011.
- [92] C. Li and P. H. Heinemann. A comparative study of three evolutionary algorithms for surface acoustic wave sensor wavelength selection. *Sensors and Actuators B: Chemical*, 125(1):311 – 320, 2007.
- [93] O. Maron and A. W. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11:193–225, 1997. 10.1023/A:1006556606079.
- [94] N. S. Mera. Passive gamma tomography reconstruction of layered structures in nuclear waste vaults. *Inverse Problems*, 23(1):385 – 403, 2007.
- [95] Z. Michalewicz. Heuristic methods for evolutionary computation techniques. *Journal of Heuristics*, 1(2):177–206, 1996.
- [96] Z. Michalewicz, D. Dasgupta, R. G. L. Riche, and M. Schoenauer. Evolutionary algorithms for constrained engineering problems. *Computers & Industrial Engineering Journal*, 30(4):851–870, 1996.
- [97] Z. Michalewicz and G. Nazhiyath. Genocop III: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In D. B. Fogel, editor, *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, pages 647–651, Piscataway, NJ, USA, 1995. IEEE Press.
- [98] G. Montes, P. Bartolome, and A. L. Udias. The use of genetic algorithms in well placement optimization. In *SPE Latin American and Caribbean Petroleum Engineering Conference*, number SPE 69439, March 2001.
- [99] A. Morales, H. Nasrabadi, and D. Zhu. A modified genetic algorithm for horizontal well placement optimization in gas condensate reservoirs. In *SPE annual technical conference and exhibition*, number SPE 135182, September 2010.
- [100] L. Nakajima and D. J. Schiozer. Horizontal well placement optimization using quality map definition. In *Canadian International Petroleum Conference*, number 2003-053. Petroleum Society of Canada, June 2003.
- [101] J. A. Nelder and R. Mead. A simplex-method for function minimization. *Computer journal*, 7:308–313, 1965.

-
- [102] J. Nocedal. Large scale unconstrained optimization. In *The state of the art in numerical analysis*, pages 311–338. Oxford University Press, 1996.
- [103] J. Onwunalu and L. J. Durlofsky. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, 14(1):183–198, 2010.
- [104] U. Ozdogan and R. N. Horne. Optimization of well placement under time-dependent uncertainty. *SPE Reservoir Evaluation & Engineering*, 9(2):135–145, 2006.
- [105] Y. Pan and R. N. Horne. Improved methods for multivariate optimization of field development scheduling and well placement design. In *SPE annual technical conference and exhibition*, number SPE 49055, September 1998.
- [106] I. Pänke, J. Branke, and Y. Jin. Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. *IEEE Transactions on Evolutionary Computation*, 10(4):405–420, 2006.
- [107] M. J. D. Powell. The NEWUOA software for unconstrained optimization without derivatives. In *Large Scale Nonlinear Optimization*, pages 255–297, 2006.
- [108] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. PhD thesis, Technical University of Berlin, 1971.
- [109] R. Ros and N. Hansen. A simple modification in CMA-ES achieving linear time and space complexity. In G. Rudolph et al., editors, *Parallel Problem Solving from Nature - PPSN X, 10th International Conference Dortmund, Germany, September 13-17, 2008, Proceedings*, volume 5199 of *Lecture Notes in Computer Science*, pages 296–305. Springer, 2008.
- [110] F. Rosenblatt. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.
- [111] T. P. Runarsson. Constrained evolutionary optimization by approximate ranking and surrogate models. In X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. Rowe, P. Tino, A. Kabán, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 401–410. Springer Berlin / Heidelberg, 2004.
- [112] Y. Sano and H. Kita. Optimization of noisy fitness functions by means of genetic algorithms using history of search. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from*

-
- Nature PPSN VI*, volume 1917 of *Lecture Notes in Computer Science*, pages 571–580. Springer Berlin / Heidelberg, 2000.
- [113] Y. Sano and H. Kita. Optimization of noisy fitness functions by means of genetic algorithms using history of search with test of estimation. volume 1917, pages 360–365, 2002.
- [114] P. Sarma and W. H. Chen. Efficient well placement optimization with gradient-based algorithms and adjoint models. In *SPE intelligent energy conference and exhibition*, number SPE 112257, February 2008.
- [115] R. Schulze-Riegert, M. Bagheri, M. Krosche, N. Kück, and D. Ma. Multiple-objective optimization applied to well path design under geological uncertainty. In *SPE reservoir simulation symposium*, number SPE 141712, February 2011.
- [116] R. Schulze-Riegert, M. Dong, K. L. Heskestad, M. Krosche, H. Mustafa, K. Stekolschikov, and M. Bagheri. Well path design optimization under geological uncertainty: Application to a complex north sea field. In *SPE Russian Oil and Gas Conference and Exhibition*, number SPE 136288, October 2010.
- [117] H.-P. Schwefel. *Evolutionstrategie und numerische Optimierung*. PhD thesis, Technical University of Berlin, 1975.
- [118] D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computing*, 24:647–656, 1970.
- [119] J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37:332–341, 1992.
- [120] M. Srinivas and L. M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24:656–667, 1994.
- [121] P. Staggé. Averaging efficiently in the presence of noise. In A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN V*, volume 1498 of *Lecture Notes in Computer Science*, pages 188–197. Springer Berlin / Heidelberg, 1998.
- [122] P. D. Surry and N. J. Radcliffe. Real representations. In *Foundations of Genetic Algorithms 4*. Morgan Kaufmann, 1996.

-
- [123] R. Thangaraj, M. Pant, A. Abraham, and P. Bouvry. Particle swarm optimization: Hybridization perspectives and experimental illustrations. *Applied Mathematics and Computation*, 217(12):5208–5226, 2011.
- [124] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.
- [125] S. Vlemmix, G. J. P. Joosten, D. R. Brouwer, and J. D. Jansen. Adjoint-based well trajectory optimization in a thin oil rim. In *SPE EUROPEC/EAGE annual conference and exhibition*, number SPE 121891, June 2009.
- [126] H. Wang, D. E. Ciaurri, and L. J. Durlofsky. Optimal well placement under uncertainty using a retrospective optimization framework. In *SPE Reservoir Simulation Symposium*, number SPE 141950, February 2011.
- [127] H. Wang and B. W. Schmeiser. Discrete stochastic optimization using linear interpolation. In *Proceedings of the 40th Conference on Winter Simulation, WSC '08*, pages 502–508. Winter Simulation Conference, 2008.
- [128] L. C. Wrobel and P. Miltiadou. Genetic algorithms for inverse cathodic protection problems. *Engineering Analysis with Boundary Elements*, 28(3):267 – 277, 2004.
- [129] B. Yeten, L. J. Durlofsky, and K. Aziz. Optimization of nonconventional well type, location and trajectory. *SPE Journal*, 8:200–210, 2003.
- [130] M. J. Zandvliet, M. Handels, G. M. van Essen, D. R. Brouwer, and J. D. Jansen. Adjoint-based well-placement optimization under production constraints. *SPE Journal*, 13(4):392–399, 2008.
- [131] H. Zhao, C. Chen, S. Do, G. Li, and A. C. Reynolds. Maximization of a dynamic quadratic interpolation model for production optimization. In *SPE reservoir simulation symposium*, number SPE 141317, February 2011.
- [132] D. I. Zubarev. Pros and cons of applying proxy-models as a substitute for full reservoir simulations. In *SPE Annual Technical Conference and Exhibition*, number SPE 124815, October 2009.

Abstract: The amount of hydrocarbon recovered can be considerably increased by finding optimal placement of non-conventional wells. For that purpose, the use of optimization algorithms, where the objective function is evaluated using a reservoir simulator, is needed. Furthermore, for complex reservoir geologies with high heterogeneities, the optimization problem requires algorithms able to cope with the non-regularity of the objective function. The goal of this thesis was to develop an efficient methodology for determining optimal well locations and trajectories, that offers the maximum asset value using a technically feasible number of reservoir simulations.

In this thesis, we show a successful application of the Covariance Matrix Adaptation - Evolution Strategy (CMA-ES) which is recognized as one of the most powerful derivative-free optimizers for continuous optimization. Furthermore, in order to reduce the number of reservoir simulations (objective function evaluations), we design two new algorithms. First, we propose a new variant of CMA-ES with meta-models, called the new-local-meta-model CMA-ES (nlmm-CMA), improving over the already existing variant of the local-meta-model CMA-ES (lmm-CMA) on most benchmark functions, in particular for population sizes larger than the default one. Then, we propose to exploit the partial separability of the objective function in the optimization process to define a new algorithm called the partially separable local-meta-model CMA-ES (p-sep lmm-CMA), leading to an important speedup compared to the standard CMA-ES.

In this thesis, we apply also the developed algorithms (nlmm-CMA and p-sep lmm-CMA) on the well placement problem to show, through several examples, a significant reduction of the number of reservoir simulations needed to find optimal well configurations. The proposed approaches are shown to be promising when considering a restricted budget of reservoir simulations, which is the imposed context in practice.

Finally, we propose a new approach to handle geological uncertainty for the well placement optimization problem. The proposed approach uses only one realization together with the neighborhood of each well configuration in order to estimate its objective function instead of using multiple realizations. The approach is illustrated on a synthetic benchmark reservoir case, and is shown to be able to capture the geological uncertainty using a reduced number of reservoir simulations.

Résumé: La quantité d'hydrocarbures récupérés peut être considérablement augmentée si un placement optimal des puits non conventionnels forer, peut être trouvé. Pour cela, l'utilisation d'algorithmes d'optimisation, où la fonction objectif est évaluée en utilisant un simulateur de réservoir, est nécessaire. Par ailleurs, pour des réservoirs avec une géologie complexe avec des hétérogénéités élevées, le problème d'optimisation nécessite des algorithmes capables de faire face à la non-régularité de la fonction objectif. L'objectif de cette thèse est de développer une méthodologie efficace pour déterminer l'emplacement optimal des puits et leurs trajectoires, qui offre la valeur liquidative maximale en utilisant un nombre techniquement abordable de simulations de réservoir.

Dans cette thèse, nous montrons une application réussie de l'algorithme "Covariance Matrix Adaptation - Evolution Strategy" (CMA-ES) qui est reconnu comme l'un des plus puissants optimiseurs sans-dérivés pour l'optimisation continue. Par ailleurs, afin de réduire le nombre de simulations de réservoir (évaluations de la fonction objectif), nous concevons deux nouveaux algorithmes. Premièrement, nous proposons une nouvelle variante de la méthode CMA-ES avec des méta-modèles, appelé le nouveau-local-méta-modèle CMA-ES (nlmm-CMA), améliorant la variante déjà existante de la méthode local-méta-modèle CMA-ES (lmm-CMA) sur la plupart des fonctions de benchmark, en particulier pour des tailles de population plus grande que celle par défaut. Ensuite, nous proposons d'exploiter la séparabilité partielle de la fonction objectif durant le processus d'optimisation afin de définir un nouvel algorithme appelé la partiellement séparable local-méta-modèle CMA-ES (p-sep lmm-CMA), conduisant à une réduction importante en nombre d'évaluations par rapport à la méthode CMA-ES standard.

Dans cette thèse, nous appliquons également les algorithmes développés (nlmm-CMA et p-sep lmm-CMA) sur le problème de placement des puits pour montrer, à travers plusieurs exemples, une réduction significative du nombre de simulations de réservoir nécessaire pour trouver la configuration optimale des puits. Les approches proposées sont révélées prometteuses en considérant un budget restreint de simulations de réservoir, qui est le contexte imposé dans la pratique.

Enfin, nous proposons une nouvelle approche pour gérer l'incertitude géologique pour le problème d'optimisation de placement des puits. L'approche proposée utilise seulement une réalisation, ainsi que le voisinage de chaque configuration, afin d'estimer sa fonction objectif au lieu d'utiliser multiples réalisations. L'approche est illustrée sur un cas de réservoir de benchmark, et se révèle être en mesure de capturer l'incertitude géologique en utilisant un nombre réduit de simulations de réservoir.