Mixed nonlinear optimization for integer and real variables: application to well location problem in reservoir engineering

Claire Lizon

Mixed nonlinear optimization for integer and real variables :
application to well location problem in reservoir engineering

Optimisation non linéaire mixte en variables entières et réelles:
application au problème de placement des puits en ingénierie de réservoir

par

Claire LIZON

Thèse pour obtenir le titre de
Docteur en Sciences de l'Ecole Polytechnique

Soutenue le 16 Décembre 2015

Ecole Polytechnique
Laboratoire Informatique

Bjarne FOSS, NTNU	Rapporteur
Sébastien LE DIGABEL, Ecole Polytechnique Montréal	Rapporteur
Ala BEN-ABBES, EDF	Examinateur
Sonia CAFIERI, ENAC	Examinateur
Marie-Christine COSTA, ENSTA	Présidente du jury
Leo LIBERTI, Ecole Polytechnique	Directeur de thèse
Claudia D'AMBROSIO, Ecole Polytechnique	Co-Directrice de thèse
Delphine SINOQUET, IFPEN	Encadrante IFPEN

Remerciements

La condition du thésard peut parfois être un peu solitaire, heureusement de nombreuses personnes ont été là pour contribuer au bon déroulement de mes travaux, et il est donc important de les remercier comme il se doit.

Ainsi je souhaiterais en premier lieu remercier Leo Liberti, mon directeur de thèse, pour avoir permis que cette thèse soit possible et pour ses précieux conseils. J'adresse un grand merci à Claudia D'Ambrosio et Delphine Sinoquet pour leur encadrement, leur gentillesse, leur patience et leur conseils.

Je remercie aussi Sébastien Le Digabel pour avoir accepté d'être un rapporteur de la thèse et pour les corrections qu'il y a fait porter. Also a few words in English to express my gratitude to Bjarne Foss, who accepted to be one referee and gave precious advices for the improvement of my thesis manuscript. Je remercie également Ala Ben-Abbes, Sonia Cafieri et Marie-Christine Costa pour leur participation au jury. Je les remercie pour le temps qu'ils ont consacré à la lecture de cette thèse.

Je souhaite aussi remercier Zakia Benjelloun-Touimi, chef du département de mathématiques appliquée, pour sa gentillesse en particulier durant ma dernière année de thèse. Je remercie également Frédéric Delbos, dont j'ai partagé le bureau durant les premiers mois de ma thèse, pour sa bonne humeur, son aide et ses sages conseils. Je remercie également Miguel Munoz Zuniga pour avoir relu une partie de ma thèse. Je souhaiterais remercier Éric Flauraud, pour sa patience et ses explications concernant le modèle de simulation d'écoulement en milieux poreux. Je remercie aussi Guillaume Enchery pour son aide dans l'utilisation du simulateur. Je tiens également à remercier Didier Ding, pour ses conseils pour l'incorporation des puits au modèle 1D. Je tiens aussi à remercier Quang Huy Tran pour son aide en LaTeX, ainsi qu'Isabelle Faille pour la relecture du rapport et ses corrections lors de ma mi-thèse. Je voudrais aussi remercier Michaëlle Le Ravalec pour ses conseils et son aide dans la mise en place de cas tests. Il m'est impossible de ne pas remercier Steven Thomas pour son aide pour l'incorporation des branches dans l'utilisation du simulateur.

Je remercie toute les personnes du département de mathématiques appliquées et informatiques et plus particulièrement les thésards à mon arrivée en stage, Eugenio, Antoine, Carole, Mathieu, Simon, Franck et Ratiba, mes compagnons de thèses et parfois de bureaux, Tassadit, Soleiman, Benoît, Pierre, Thibaut, Huong et Aboubakar. Merci également aux thésards et stagiaires de réservoir, Caroline, Benjamin, Arthur, Nicolas et Sandra, pour nos repas du vendredi.

Je remercie aussi tous les thésards et postdocs du LIX, Pierre-Louis, Raouia, Ky,

Gustavo, Youssef, Nicolas, Luca et Olivier. Merci particulièrement à Sonia pour son aide, ses conseils et sa bonne humeur.

Je remercie bien sur les personnes qui se sont dévouées pour relire et corriger l'anglais de mon manuscrit : ma marraine Cécile, mon père, et mon amie Caroline. Merci d'avoir aussi rapides et efficaces.

Merci à tous mes amis de Jussieu qui ont pu me soutenir de plus ou moins près, Alexis, Émilie, Joëlle, PP et Ève (Force et Courage étaient plus que nécessaires!). Je remercie aussi Claire dont l'amitié m'est depuis longtemps précieuse et dont les visites à Paris tombaient toujours à point. Merci aussi à Joséphine, mon amie d'enfance, je n'oublie pas les moments passés qui m'ont permis de me changer les idées quand ça n'allait pas. Merci à Caroline pour son aide, sa bonne humeur et les vacances passées ensemble. Enfin un grand merci Jérémy pour son soutien durant ces trois années.

Merci à mon ami pour m'avoir accompagnée dans les périodes difficiles et pour les bons moments partagés.

Un grand merci à ma famille, en particulier à mon père et ma mère, pour leur soutien et leur encouragements. Un remerciement spécial pour mon grand-père qui a toujours cru en moi.

Résumé

L'objectif de la thèse est de développer une méthode adaptée au problème d'optimisation de la production d'hydrocarbures. Il s'agit d'une optimisation conjointe de la production d'hydrocarbures et du coût du forage (à l'aide par exemple d'une fonction objective de type Net Present Value) pour le développement d'un champ pétrolier ou gazier. On recherche ainsi une configuration optimale du schéma de production modélisé par des variables à valeurs entières, le nombre de puits injecteurs et producteurs, le nombre de branches, complétées par des variables continues comme la position des puits dans le réservoir, la longueur des branches, etc. Les fonctions à optimiser et les contraintes sont calculées à partir des réponses d'un simulateur d'écoulement des fluides dans le réservoir, coûteux en temps de calcul : les réponses à optimiser sont les quantités d'huile, d'eau et de gaz produits, les quantités d'eau et/ou de gaz injecté (pour faciliter la production). La fonction objectif et les contraintes sont considérées comme des réponses d'un simulateur boîte noire et peuvent nécessiter plusieurs heures ou plusieurs jours en temps de calcul.

Au cours de cette thèse, nous nous sommes intéressés à différentes méthodes d'optimisation, et une étude bibliographique a permis d'identifier les méthodes adaptées à notre application. Le problème étant un problème de type boîte noire, nous avons étudié en particulier des méthodes d'optimisation sans dérivées, telles que les méthodes de recherche directe dont nous avons évalué les performances pour l'optimisation du placement, du nombre et du type de puits verticaux sur plusieurs cas de modèle de réservoir, de dimension et réalisme différents. Nous avons également étudié des méthodes basées sur un modèle de substitution par krigeage, populaire en optimisation continue, et avons utilisé une adaptation aux variables mixtes. Nous avons aussi travaillé sur le développement d'une nouvelle méthode de région de confiance étendue aux variables mixtes et l'avons évaluée sur deux cas simplifiés de placement de puits.

Après ce travail transverse sur différentes méthodes d'optimisation adaptées au cas boîte noire MINLP, nous proposons une méthodologie prenant en compte les caractéristiques de la fonction objectif, issue des résultats d'un simulateur boîte noire, tout en utilisant les méthodes MINLP classiques. Pour cela nous proposons d'optimiser le placement et le nombre de puits, ainsi que le nombre de branches, avec une méthode de résolution en deux étapes, en résolvant successivement deux sous-problèmes. La première étape consiste à résoudre un problème boîte noire MINLP avec une méthode de recherche directe (étudiée dans la première partie de la thèse), en optimisant le nombre et le type de puits verticaux.

Dans la seconde étape, nous définissons un problème MINLP à partir des données issues de la simulation de la solution à puits verticaux obtenue à la première étape. Dans ce nouveau problème MINLP, nous optimisons la trajectoire des puits en ajoutant de nouvelles branches aux puits producteurs verticaux existants. Les inconnues sont la position de leurs extrémités, et la production d'hydrocarbure est estimée sans utiliser le simulateur d'écoulement. Plusieurs modèles pour ce problème ont été écrits et évalués sur des cas tests de différentes dimensions en utilisant les solveurs BONMIN et SCIP.

Abstract

The aim of this thesis is to develop an adapted method to the hydrocarbon production optimization problem. It is a joint optimization of the hydrocarbon production and the drilling costs (using for instance an objective function of type Net Present Value) for the development of petroleum or gas field. Hence we search an optimal configuration of the production scheme which is modeled by integer variables, the number of injector and producer wells, the number of branches, completed by continuous variables as the well location in the reservoir, the length of the branches, etc. The functions to optimize and the constraints are computed from the outputs of a reservoir fluid flow simulator, costly in computational time: the outputs to optimize are the quantities of produced oil, water and gas, and the quantities of injected (to facilitate the production). The objective function and the constraints are considered as the outputs of a Black-Box simulator and can necessitate several hours or several days in computational time.

During this thesis, we considered different optimization methods, and a bibliographic study allowed us to identify methods suitable for our applications. The problem is a Black-Box optimization problem, hence we studied in particular derivative free optimization methods, such as direct search methods. We evaluated the performances of these methods for the optimization of the location, the number and the type of vertical wells on several reservoir model cases, of different dimension and realism. We also studied methods based on a substitution models with kriging, popular in continuous optimization, and used an adaptation to mixed variables. We also worked on the development of a new Trust Region method extended to mixed variables and evaluated it on two simplified cases of well placement.

After this transversal work on different optimization methods adapted to the Black-Box MINLP case, we propose a methodology taking in account the characteristics of the objective functions, computed from the outputs of a Black-Box fluid flow simulator, while using classical MINLP methods. To do so, we propose a two-step resolution method, by successively solving two sub-problems. The first step consists in solving a Black-Box MINLP problem with a direct-search method (studied in the first part of the thesis), by optimizing the number the type of vertical wells.

In the second step, we define a MINLP problem from the outputs of the simulation of the vertical well solution that was obtain at the first step. In this new MINLP problem, we optimize the well trajectory by adding new branches to the existing vertical producer wells. The unknowns are their extremities locations, and the hydrocarbon production is estimate without using the fluid flow simulator. We wrote several models for this problem, and evaluated them on test cases of different dimensions by using the solvers BONMIN and SCIP.

Contents

Introduction

Mixed-integer nonlinear optimization problems (MINLP) are defined as optimization problems whose variables have the constraint to take integer (or discrete) values, and for which the objective function and the constraints are described by functions that can be linear or non linear. These optimization problems appear in various applications at *IFP énergies nouvelles*, a public-sector research, innovation and training center active in the fields of energy, transport and environment. Among these applications are the optimization of petroleum transportation network, topological optimization in mechanics, setting optimization of experimental scheme (e.g., the choice of injection mode in motor calibration), optimization of production well placement in reservoir engineering, or optimization of injection scheme for geological storage of CO_2.

MINLP optimization problems are considered as particularly complex problems (Burer and Letchford [19], D'Ambrosio and Lodi [28]): this complexity lies in the strongly combinatorial aspect induced by integer variables and nonlinearities of the objective function and the constraints. Although MINLP optimization has benefited from the advances of the last thirty years in nonlinear optimization (NLP) and in mixed linear optimization (MILP), the size of the problems of this type globally solvable in an acceptable computational time by the current methods remains limited, and resolution time can drastically increase with the number of integer variables, and with the number and type of nonlinear functions. In addition, the problem becomes even more complex when one of the objective or constraint functions are Black-Box. In that case, the optimization problem is a Black-Box MINLP problem and its solving is even more complicated (Belotti et al. [14]).

In this thesis, we propose to develop a method adapted to the optimization problem of hydrocarbon production (Bouzarkouna et al. [17], Echeverría Ciaurri et al. [30], Ermolaev and Kuvichko [32]). It deals with a joint optimization of hydrocarbon production and drilling costs (using for instance an objective function of type Net Present Value) for the development of petroleum or gaseous field. Hence we seek an optimal configuration of the production scheme. The integer variables are the number of injector and producer wells, the number of branches, completed by continuous variables such as the wells and the branches location in the reservoir, the length of the branches, etc. The functions to optimize and the constraints are generally computed from the outputs of a reservoir fluid flow simulator, costly in computational time: the outputs to optimize are the quantities of produced oil, water and gas, and the quantities of injected water and gas (to facilitate the production). Localization constraints and number of wells constraints are added to the problem.

The application and its main difficulties are presented in the first chapter: the diversity of the variables and their number (number depending on the values taken by some integer variables), but also the Black-Box characteristic of the functions to optimize (nonlinearity, non-convexity), the computational cost of these functions and the size of these problems. We introduce also in this chapter several synthetic cases, with different levels of complexity and realism: A first mono-dimensional simplified test case, for which the study and the implementation were made to rapidly evaluate optimization methods, and a second more realistic and relatively simplified 2D case. The third introduced case is the PUNQ-S3 case (Floris et al. [35]), a 3D case widely used for the evaluation of optimization in reservoir characterization. Finally we present a second 3D case, derived from the SPE10 reservoir case (Project [64]). These last cases will allow us to evaluate the methods developed during this thesis on cases relatively realistic. Geologic notions, used in the methodology presented in the third chapter, and in fluid flow simulation are also presented in the first chapter.

In a second chapter, we focus on the optimization of the number and the location of vertical wells. We first present a direct-search method and the NOMAD solver (Le Digabel [48]), for which we evaluate the capacities through different test cases, first by using a 1D case, then the PUNQ-S3 case. We then present an optimization method using a substitution model based on a kriging method adapted to mixed variables, coupled to the Efficient Global Optimization (EGO, Jones et al. [44]), that we evaluate on a mono-dimensional case. Finally we introduce a new Trust-Region method adapted to mixed variables, and we evaluate it on a 1D case, then on a 3D case derived from the SPE10 case.

In the third chapter, we present the methodology that we elaborated and developed during the thesis for the optimization of the location and the number of wells and the number of branches. The methodology consists in two steps, each of which solving a subproblem. In the first step a Black-Box MINLP optimization problem is solved to find the placement of vertical wells. The solution of the first step is then analyzed to define the input of the second MINLP subproblem, which optimizes the design of the branches for each well. This second subproblem is treated through three different models. The methodology is tested and evaluated on a 2D case.

Introduction en français

Les problèmes d'optimisation non linéaire mixte en variables entières et réelles (MINLP) sont des problèmes d'optimisation pour lesquels certaines variables sont contraintes à prendre des valeurs entières (ou discrètes) et la fonction objective et les contraintes sont décrites par des fonctions qui peuvent être linéaires ou non linéaires. Ces problèmes d'optimisation apparaissent dans diverses applications à *IFP énergies nouvelles*, un organisme public de recherche, d'innovation industrielle et de formation intervenant dans les domaines de l'énergie, du transport et de l'environnement. Parmi ces applications nous pouvons citer l'optimisation de réseaux de conduites pétrolières, l'optimisation topologique en mécanique, l'optimisation de réglages de dispositifs expérimentaux (par exemple, le choix des modes d'injection en calibration des moteurs), l'optimisation du placement des puits pour la production en ingénierie de réservoir et l'optimisation du dispositif d'injection en stockage géologique du CO_2.

Les problèmes d'optimisation MINLP sont considérés comme des problèmes particulièrement difficiles (Burer and Letchford [19], D'Ambrosio and Lodi [28]) : cette complexité provient de l'aspect fortement combinatoire induit par les variables entières et des non-linéarités de la fonction objective et des contraintes. Bien que l'optimisation MINLP ait bénéficié des grandes avancées des 30 dernières années en optimisation non linéaire (NLP) et en optimisation mixte linéaire (MILP), la taille des problèmes de ce type solubles de manière globale en un temps de calcul acceptable par les méthodes actuelles reste limitée, le temps de résolution pouvant augmenter drastiquement avec le nombre de variables entières, ainsi qu'avec le nombre et la complexité des fonctions. Une difficulté peut s'ajouter au problème quand l'une des fonctions objectif ou de contraintes est dite "boîte-noire", le problème d'optimisation est alors dis "boîte-noire" MINLP, et sa résolution est plus compliquée encore (Belotti et al. [14]).

Dans cette thèse, nous proposons de développer une méthode adaptée au problème d'optimisation de la production d'hydrocarbures (Bouzarkouna et al. [17], Echeverría Ciaurri et al. [30], Ermolaev and Kuvichko [32]). Il s'agit d'une optimisation conjointe de la production d'hydrocarbures et du coût du forage (à l'aide d'une fonction objective de type NPV, Net Present Value) pour le développement d'un champ pétrolier ou gazier. On recherche ainsi une configuration optimale du schéma de production. Les variables à valeurs entières sont donc le nombre de puits injecteurs et producteurs, le nombre de branches, complétés par des variables continues comme la position des puits et des branches dans le réservoir, la longueur des branches, etc. Les fonctions à optimiser et les contraintes sont généralement des réponses d'un simulateur d'écoulement dans le réser-

voir, coûteux en temps de calcul : quantités d'huile, d'eau et de gaz produits, quantités d'eau et/ou de gaz injecté (pour faciliter la production), ces quantités étant cumulées sur la durée de production. Des contraintes de localisation dans le réservoir, et des contraintes sur le nombre de puits s'ajoutent au problème.

L'application et ses principales difficultés sont présentées dans le premier chapitre : la diversité des variables et leur nombre (nombre dépendant des valeurs prises par certaines variables entières) mais aussi le caractère "boîte-noire" des fonctions à optimiser (non-linéarité, non-convexité), le coût de calcul de ces fonctions et la taille de ces problèmes. Nous introduisons également dans ce chapitre plusieurs cas test synthétiques, ayant différent niveau de complexité et de réalisme. Un premier cas test simplifié mono-dimensionnel, dont l'étude et l'implémentation ont pour but d'évaluer rapidement des méthodes d'optimisation, et un second cas 2D plus réaliste et relativement simplifié. Le troisième cas introduit est le cas PUNQ-S3, un cas 3D largement utilisé pour l'évaluation des méthodes d'optimisation en caractérisation de réservoir, enfin nous présentons un second cas 3D, dérivé du cas SPE10 (Project [64]). Ces derniers cas nous permettrons d'évaluer les méthodes développées dans le cadre de cette thèse sur des cas relativement réalistes. Des notions en géologie, utilisées dans les méthodes présentées dans le troisième chapitre, et simulation d'écoulement sont également présentées dans ce premier chapitre.

Dans un second chapitre nous nous intéressons à l'optimisation du nombre et du placement de puits verticaux. Nous présentons d'abord une méthode de recherche directe (logiciel NOMAD, Le Digabel [48]), dont nous évaluons les capacités à travers différents cas tests, tout d'abord en utilisant un cas 1D, puis le cas PUNQ-S3. Enfin nous testons le solveur sur le cas 3D dérivé du cas SPE10 en ajoutant à la simulation une contrainte sur la production. Nous présentons ensuite une méthode d'optimisation utilisant un modèle de substitution basé sur le modèle de krigeage et adapté aux variables mixtes, couplé à l'algorithme Efficient Global Optimization (EGO, Jones et al. [44]), que nous testons sur un cas mono-dimensionnel. Enfin nous introduisons une nouvelle méthode de région de confiance adaptée aux variables mixtes, et l'évaluons sur un cas 1D, puis sur un cas 3D dérivé du cas SPE10.

Dans le troisième chapitre, nous présentons la méthodologie que nous avons mis en place au cours de la thèse pour l'optimisation du placement, du nombre de puits et du nombre de branches. La méthodologie se déroule en deux étapes, en résolvant successivement deux sous-problèmes. Dans un premier temps un problème d'optimisation Black-Box MINLP pour puits verticaux est résolu, puis la solution obtenue est analysée pour définir les données du second problème MINLP, optimisant le nombre et la position de branches. Ce deuxième sous-problème est traité au travers de trois différents modèles, dont nous détaillons les variables, paramètres et contraintes. La méthodologie est testée et évaluée sur un cas test 2D.

Chapter 1

Context

1.1 Introduction

As oil reserves are declining, a strong effort is underway in the petroleum industry to propose efficient extraction techniques with a view to increase oil field production. In this context, developing efficient simulation-based optimization methods for well placement and monitoring is a key challenge. Thanks to intelligent fields or Smart fields (AbdulKarim et al. [2], van den Berg et al. [75]), a large amount of information can be used to improve the knowledge of the reservoir, in particular to improve numerical models for generating oil production forecasts. These models, calibrated with history matching (Le Ravalec et al. [50], Oliver et al. [62], Schulze-Riegert and Ghedan [69]), have better forecasting capabilities, and can be used to make decisions related to the development of oil fields.

One of these decisions is the location and geometry (i.e., trajectory) of wells. In industry the decision to drill a well or not, and its location, is taken based on the so-called professional judgment of reservoir engineers. It requires the knowledge of both geological and engineering influencing parameters. Considering that the effects of those parameters are non linear, and evolve with time, it is unlikely that the reservoir engineer's experience will suffice to determine the best well configuration, even though the provided solutions yield satisfying production results. Although finding the provably best well configuration is also an excessively hard task for today's technology, optimization methods can lead to an improved configuration. Optimizing the number of wells, their trajectory, and their type (injector or producer) is a difficult problem and can be very costly in terms of reservoir fluid flow simulations. Optimizing well location has been observed to yield non-smooth cost functions (Bouzarkouna et al. [17], Isebor et al. [42]). Hence, in optimal well placement one should perform some kind of global search in order to avoid being trapped in local minima.

Different aspects of the well placement problem have received varying degrees of attention in the literature. The number of wells is one of these aspects, it can be fixed and given by the users (Bouzarkouna et al. [18]) or can be a variable of the optimization

problem as in Echeverría Ciaurri et al. [30]. Another important aspect in the optimization problem are the different ways of describing a well. Aliyev and Durlofsky [9] simply consider vertical wells, since this geometry requires fewer variables than deviated lateral wells (Humphries and Haynes [41]). Onwunalu and Durlofsky [63] study nonconventional wells, while the number of producer wells is fixed, the number of their lateral branches is optimized. Bouzarkouna et al. [18] optimize the location of a fixed number of wells composed of a fixed number of lateral segments, where wells are not necessarily rectilinear and can have branches. In reservoir engineering, a fundamental aspect is to account for uncertainties to reduce investment risks. Busby and Sergienko [20] propose a robust optimization of the well placement while accounting for geological uncertainties of the reservoir model (see also Busby et al. [21], Shirangi and Durlofsky [71], Wang et al. [78]). Chang et al. [22] propose a multi-objective optimization under geological uncertainties by maximizing the mean and minimizing the variance of the production values. The last aspect, mentioned here, in the optimization of oil production, is the well control, i.e., deciding the production rate of the injector or producer well, during the production time frame. A joint optimization of the number of wells and their control is proposed in Aliyev and Durlofsky [9], Forouzanfar and Reynolds [36], Humphries and Haynes [41].

Derivative Free Optimization (DFO) methods such as evolutionary algorithms and Direct Search methods are the most common methods to solve well placement problem. In Onwunalu and Durlofsky [63], an evolutionary algorithm, Particle Swarm Optimization (PSO, introduced by Kennedy and Eberhart [45] and inspired from the movement of birds or insects during migrations), are used for the optimization of the type and location of the wells. Forouzanfar and Reynolds [36] use a gradient based optimization. Direct search methods are also used to effectively solve this type of problem, as in Echeverría Ciaurri et al. [30], but these can be costly in computational time. Thus, recently several papers study methods coupling evolutionary algorithms and Mesh Adaptive Direct Search (MADS) methods. Aliyev and Durlofsky [9], Humphries and Haynes [41], Isebor et al. [42] combine the PSO stochastic method with MADS. Bouzarkouna et al. [18] implement "the covariance matrix adaptation evolution strategy" (CMA-ES) method in order to optimize the trajectory of a fixed number of wells, and couple the method with local metamodels, i.e., the objective function is replaced by different metamodels built for each well or set of wells. In Busby et al. [21] response surface models are built with kriging, and optimized under uncertainties with robust Efficient Global Optimization. Adjoint-based methods are used in Zandvliet et al. [79] to optimize the position of wells under production constraints. Generally, reducing computational time is a challenging aspect: Ermolaev and Kuvichko [32] work on the structure of the matrix, whereas Bellout et al. [12], Shirangi and Durlofsky [71] focus on parallelization.

The aim of this thesis is to optimize both the well placement and the number of wells and their trajectories. As including integer variables can lead to formulations that are computationally expensive to solve, we do not associate other aspects of reservoir engineering to the optimization problem, and thus choose a deterministic approach. The second chapter of this thesis is focused on Derivative Free Optimization methods, methods that are well-suited for solving Black-Box MINLP optimization problem and we propose a new trust region method for the optimization of the number, type and location of unilateral wells. In the third chapter, we propose a new approach for the optimization

of the number and location of branches for vertical producer wells, using both DFO methods and exact methods in MINLP optimization. The well placement optimization problem, its characteristics, and several reservoir models used for the evaluation of the optimization methods are presented in the remaining of this chapter.

1.2 Problem formulation

1.2.1 The problem statement

The formulation of the well placement decisions as an optimization problem requires an objective function. For a given well configuration, a reservoir fluid flow simulator evaluates the predictions of produced oil and gas volumes, and the predictions of produced and injected water and gas volumes. The objective function is computed from the predictions of the fluid flow simulator which is a Black-Box simulator: we do not have access to an analytic formula of the objective function, so we have a Black-Box optimization problem. The significant difficulty of Black-box problems is that we have no knowledge of the continuity, differentiability or convexity of the objective function. The variables of the problem describe the wells configuration: continuous variables are used for the location and the length of the wells and their branches (if any), and binary variables are used to characterize the type (injector/producer), status (drilled/not drilled), and the number of branches and their status.

The Mixed Integer Non Linear Programming (MINLP) optimization problem is the following:

$$\begin{cases} \max\limits_{x,y} f(x,y), \\ x \in X \subseteq \mathbb{R}^p \\ y \in \{0,1\}^q \\ g(x,y) \leq 0 \end{cases} \tag{1.1}$$

1.2.2 Objective function

The objective function depends on the petroleum optimization scheme. The two main possibilities are to maximize the quantity of produced oil, or to evaluate the revenue with the Net Present Value function (NPV). The NPV function is generally more accurate because it takes into account more information such as oil revenue, water management and drilling costs.

1.2.2.1 The Net Present Value (NPV) function

The Net Present Value (NPV) for a well configuration (x, y) is defined as follows

$$\begin{aligned} f_{NPV}(x,y) &= \sum_{n=1}^{N_I+N_P} \frac{1}{(1+\mathrm{R}^D)^{TD_n}} y_n (C^F l_n \log(l_n) + C^D) \\ &+ \sum_{t=1}^{T} \sum_{n=1}^{N_P} y_n (C_t^{PO} q_{tn}^{PO} - C_t^{PW} q_{tn}^{PW} - C_t^{PG} q_{tn}^{PG}) \\ &- \sum_{t=1}^{T} \sum_{n=N_P+1}^{N_P+N_I} y_n (C_t^{IW} q_{tn}^{IW} + C_t^{IG} q_{tn}^{IG}), \end{aligned} \tag{1.2}$$

where

- N_I and N_P are respectively the number of injector and producer wells,
- C^D is the drilling cost and $C^F l_n \log(l_n)$ accounts for the extra cost related to the length of well n (namely l_n),
- y_n is the status of well n, 1 if the well is drilled, 0 if it is not drilled,
- q_{tn}^{PO}, q_{tn}^{PW}, q_{tn}^{PG}, q_{tn}^{IW} and q_{tn}^{IG} are cumulative quantities for, respectively, produced oil, produced water, produced gas (for producer well n) and injected water, injected gas (for injector well n). They are computed by a reservoir fluid flow simulator corresponding to well configuration described by the variables (x, y) at a production period t.
- R^D is a discount factor (it implies that a stronger weight in the NPV function is assigned to wells that are drilled early in the production time frame) and TD_n is the period where well n is drilled,
- C_t^{PO}, C_t^{PW}, C_t^{PG}, C_t^{IW} and C_t^{IG} are positive weights associated with each term for production period t, and T is the number of production period.

1.2.2.2 Variant of NPV function: Simplified NPV function

The NVP function can be simplified by considering only the production of oil and water and the injection of water are considered, can be used. The function accounts for a fixed drilling cost, and production and injection cost are fixed for the whole production time frame. This function is used in the second chapter for the evaluation of optimization methods on a simplified test case. The function is a special case of the function (1.2) with the parameters values given in Tab. 1.1.

$$\begin{aligned} f_{NPV}(x,y) &= \sum_{n=1}^{N_I+N_P} y_n C^D \\ &+ \sum_{n=1}^{N_P} y_n (C^{PO} q_n^{PO} - C^{PW} q_n^{PW}) - \sum_{n=N_P+1}^{N_P+N_I} y_n (C^{IW} q_n^{IW}). \end{aligned} \tag{1.3}$$

Table 1.1: NPV function, parameter values

Parameters	**Values**
C_t^{PG}	0 \$/m^3
C_t^{IG}	0 \$/m^3
C_t^F	0 \$/m
R^D	0

1.2.2.3 Variant of NPV function: NPVmax

We introduce here a variant of the usual NPV function, which considers an unknown number of production periods. For each evaluation, instead of taking the NPV value at the end of the production time frame T, we maximize the NPV value in function of the number of production periods. By using the objective function (1.4), we obtain a well solution configuration and a time period production where the NPV function has reached its maximum. This function is used in the second chapter for the evaluation of an optimization method.

$$\begin{aligned} f_{NPV}(x,y) &= \sum_{n=1}^{N_I+N_P} y_n(C^F l_n \log(l_n) + C^D) \\ &+ \max_T \left(\sum_{t=1}^{T} \frac{1}{(1+\mathrm{R})^t} \left(\sum_{n=1}^{N_P} y_n(C^{PO} q_{tn}^{PO} - C^{PW} q_{tn}^{PW}) - \sum_{n=N_P+1}^{N_P+N_I} y_n(C^{IW} q_{tn}^{IW}) \right) \right). \end{aligned} \tag{1.4}$$

where R is a discount factor (it implies that early oil production is assigned a stronger weight in the NPV function than the late production).

1.2.3 Constraints

The optimization problem (1.1) can have constraints represented by:

$$g(x,y) \leq 0. \tag{1.5}$$

Several constraints can be introduced, and (1.5) is satisfied if they are all satisfied. We can have constraints on continuous variables to guaranty the behavior of the fluid flow simulator for the well configuration (x,y):

$$(x_n^1, \ldots, x_n^p) \in [x_L, x_U] \subseteq \mathbb{R}^p, \tag{1.6}$$

$$dist(x_n, x_m) > P, \forall n, m \in \{1, \ldots, N_P + N_I\}, n < m, \tag{1.7}$$

where

- $[x_L, x_U] \subseteq \mathbb{R}^p$ are the bounds of the reservoir,

- $dist(x_n, x_m)$ is a function giving the distance between wells n and m.

Constraints on integer variables are also introduced:

$$\underline{N_P} \leq \sum_{n=1}^{N_P} y_n \leq \overline{N_P}, \tag{1.8}$$

$$\underline{N_I} \leq \sum_{n=1+N_P}^{N_P+N_I} y_n \leq \overline{N_I}, \tag{1.9}$$

$$\underline{N} \leq \sum_{n=1}^{N_P+N_I} y_n \leq \overline{N}, \tag{1.10}$$

where

- $\underline{N_P}$, $\underline{N_I}$, $\underline{N} \in \mathbb{N}$ are respectively the minimum number of producer wells, injector wells, and total number of wells,
- $\overline{N_P}$, $\overline{N_I}$, $\overline{N} \in \mathbb{N}$ are respectively the maximum number of producer wells, injector wells, and total number of wells.

Breaking-Symmetry-Constraints can be used to avoid equivalent simulations, by ordering the first continuous coordinate of wells of a same type.

$$x_n^1 \leq x_n^1 + 1, \forall n \in 1, N_P - 1, \tag{1.11}$$

$$x_n^1 \leq x_n^1 + 1, \forall n \in 1 + N_P, N_p + N_I - 1. \tag{1.12}$$

A useful constraint is the water cut constraint: we apply a reactive control on each producer to avoid producing too much water which impacts negatively on the NPV. We set a reactive control by shutting off producers when the water cut is higher than a given threshold. The water cut is the ratio between the water rate produced and the sum of water and oil rates produced. In this manuscript, we consider the threshold $\frac{C_t^{PW}}{C^{POt}+C_t^{PW}} \leq 0.91$. With this reactive control, a reservoir will produce until the last well is shut in.

1.3 Reservoir model

A reservoir is composed of one (or more) subsurface rock formations containing liquid and/or gaseous hydrocarbons, of sedimentary origin. The reservoir rock is porous (contains empty volume) and permeable, and the structure is bounded by impermeable barriers which trap the hydrocarbons. Rocks are characterized by different properties, as their porosity, permeability. The capacity of a fluid (composed of water, hydrocarbons or gas) to move through a porous media, depends on the fluid, on the rock media, on other present fluids, and on the gravity. Given a reservoir model, i.e., the set of data that characterize a reservoir, a fluid flow simulator can evaluate the production of a field by solving an equations system.

1.3.1 Notions in reservoir modeling

In this section we give several definitions useful in reservoir modeling (Cossé [27]).

1.3.1.1 Definitions

Definition 1.1 *An oil field consists of a reservoir in a shape that will trap hydrocarbons and that is covered by an impermeable or sealing rock, it is an accumulation, pool or group of pools of oil in the subsurface.*

Definition 1.2 *A porous medium (or a porous material) is a material containing pores (voids). It is characterized by its porosity, ϕ equal to the ratio of the pores volume on the total volume. Porosity generally depends of the pressure P, in that case, the rock is compressible and its compressibility c_r is equal to:*

$$\frac{1}{\phi}\frac{\mathrm{d}\phi}{\mathrm{d}P}$$

Definition 1.3 *Isotropy is the property of directional uniformity in material such that physical properties do not vary in different directions. The contrary notion is the anisotropy.*

Definition 1.4 *Fluids in the pores can be composed of several phases: the "water" phase, the "liquid hydrocarbon" phase and the "gaseous hydrocarbon" phase. These phases have their own rate flow. Saturation S_α of α phase is the ratio of the pore volume occupied by α phase.*

$$\sum_\alpha S_\alpha = 1.$$

1.3.1.2 The equations system

In this section we present the fluid flow equations system. Details on the solving with finite volume scheme can be found in Eymard et al. [33] and the implementation in Aarnes et al. [1].
We note "w" the "water" phase, "o" the "liquid hydrocarbon" phase, and "g" the "gaseous hydrocarbon" phase, Ω the reservoir domain, and $[0,T]$ the production time.
Unknowns of the equations system are S_α, P_α, V_α, $\alpha \in \{g,w,o\}$, respectively saturation, pressure and velocity for α phase.

$$\frac{\partial \phi S_w}{\partial t}(x,t) + \mathrm{div}(V_w)(x,t) = h_w(x,t),\ x \in\ \Omega,\ t \in [0,T], \tag{1.13}$$

$$\frac{\partial \phi S_o}{\partial t}(x,t) + \mathrm{div}(V_o)(x,t) = h_o(x,t),\ x \in\ \Omega,\ t \in [0,T], \tag{1.14}$$

where h_α are source-terms due to the wells.
Initials conditions and boundary conditions (given quantities at $t = 0$ and on the bounds of the domain) have to be added to these equations.

1.3.1.2.a "Capillary pressure" law

The difference of pressure between phases is a known function, noted P_c, which depends on water saturation .

$$P_w(x,t) - P_o(x,t) = P_c(x, S_w(x,t)),\ x \in\ \Omega,\ t \in [0,T] \tag{1.15}$$

1.3.1.2.b Darcy's law

Darcy's law allows to link the velocity V_α to the pressure gradient of each phase α.

$$V_\alpha = -f_\alpha(S_w)K(\nabla P_\alpha - \rho_\alpha g),\ \text{in } \Omega \times [0,T], \tag{1.16}$$

where
ρ_α is the volumic mass of the component of phase α,
g is the gravity vector,
and f_α is a function that depends on the water saturation, the nature of component α, and takes into account pressure, viscosity and relative permeability.

1.3.1.3 Bi-phasic domain

Velocity V_α is replaced by its expression which is given by Darcy's law (1.16), in (1.13) and (1.14).

$$\frac{\partial \phi S_w}{\partial t} - \text{div}(f_w(s)K(\nabla P - \rho_w g)) = 0,\ \text{in } \Omega \times [0,T], \tag{1.17}$$

$$\frac{\partial \phi S_o}{\partial t} - \text{div}(f_o(s)K(\nabla P - \rho_o g)) = 0,\ \text{in } \Omega \times [0,T], \tag{1.18}$$

$$V_w = -f_w(S_w)K(\nabla P_w - \rho_w g),\ \text{in } \Omega \times [0,T], \tag{1.19}$$

$$V_o = -f_o(S_w)K(\nabla P_o - \rho_o g),\ \text{in } \Omega \times [0,T]. \tag{1.20}$$

with $f_w = \frac{k_{rw}}{\mu_w},\ f_o = \frac{k_{ro}}{\mu_o}$,
the relative permeability of fluid α, $k_{r\alpha}$,
the viscosity of fluid α, μ_α.

Depending on the model complexity, k_r and μ can be function of the point x.
Unknowns of the equations system are saturations S_α, pressures P_α and velocity V_α.
Initials and boundary conditions complete the equations system and allow its solving.

1.4 Reservoir model test cases

In this section four test cases are presented. They offer different levels of realism, complexity and size. Three optimization methods are tested to compare and illustrate their effects and advantages. The first case is a simplified 1 dimension toy problem implemented to evaluate the difficulties of the optimization problem. The second case is a relatively simple reservoir model based on a two-dimensional discretization. The third and fourth test cases are 3D reservoir models, the PUNQ-S3 case, a synthetic problem, well known in petroleum industry, and a reservoir model derived from the SPE10 case.

1.4.1 Mono-dimensional toy problem

In order to evaluate different optimization methods for our application, without being penalized by computational time, we developed a fluid flow simulation code in porous environment using a simplified equations system. The equations system is solved with MATLAB, using IMplicit Pressure Explicit Saturation (IMPES) discretization scheme. The program simulates the flow of water and oil with injector and producer wells on a 1 dimensional reservoir of 100 grid blocks with a dimension of 10 m. It is possible to choose the location and the number of wells in the reservoir. The study of the equations system, with simplifying hypotheses, was necessary for the setting of the case. In Lizon and Sinoquet [53] we build up two different cases. The permeability profiles along the 1D reservoir are shown in Fig. 1.1, for the case of heterogeneous permeability at the top of the figure, and for homogeneous permeability at the bottom of the figure. The choice of the finite difference stepsize for derivative estimation was also studied. Tab. 1.2 summarizes information of the case.

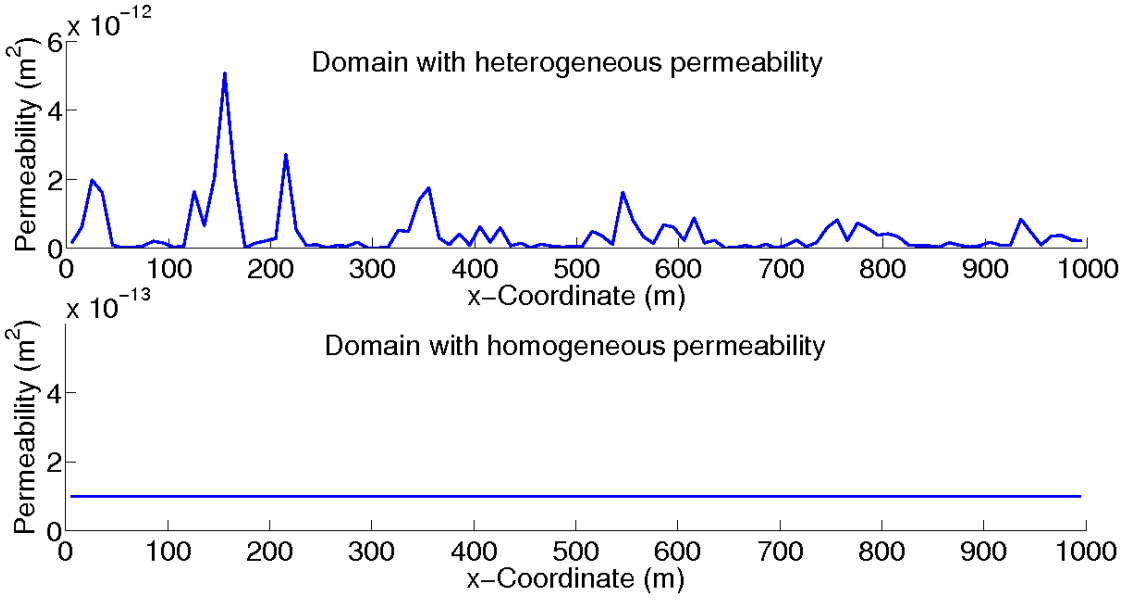

Figure 1.1: Permeability maps of the two studied cases.

Table 1.2: 1D case information model

Grid domain	**Grid Block Dimension**	**Simulator**
100×1	$Dx = 10$ m	MATLAB implementation

1.4.2 A two-dimensional case

The present case is a relatively simple reservoir model based on a two-dimensional discretization. The reservoir corresponds to an anticline trap with an aquifer at the bottom to ensure pressure support. The reservoir model has $830 \times 1 \times 50$ grid blocks, and the dimensions of each grid block are $DX = 2.5$ m, $DY = 10$ m, and $DZ = 1$ m. The resulting porosity and permeability models are shown in Figure 1.2 and Figure 1.3. The fluid flow simulator used is PumaFlow^{TM} [65]. Tab. 1.3 summarizes information of the case.

The reservoir presents two facies with volume fractions 40% and 60%. The mean and variance of the corresponding porosities are 0.4 and 0.005 for the first facies and 0.2 and 0.001 for the second one. A facies realization is first generated using a Fast Fourier Transform Moving Average (FFTMA) algorithm (Le Ravalec et al. [49]) and the Truncated Gaussian method (Chiles [23], Matheron et al. [57]) with a Gaussian variogram characterized by a horizontal range of 1250 m and a vertical range of 5 m. Thereafter, the two resulting facies are populated with porosity values applying again the FFTMA algorithm. The porosity realizations for each facies are characterized by an exponential variogram. The porosity model is converted into a permeability model using deterministic relationships between the logarithm of permeability K and porosity ϕ. For the first facies, the relation is $\log(K) = 5\phi + 0.4$, and for the second facies, the relation is $\log(K) = 7\phi + 0.3$.

Table 1.3: 2D case information model

Grid domain	**Grid Block Dimension**			**Simulator**
$830 \times 1 \times 50$	$Dx = 2.5$ m	$Dy = 10$ m	$Dz = 1$ m	PumaFlow^{TM}

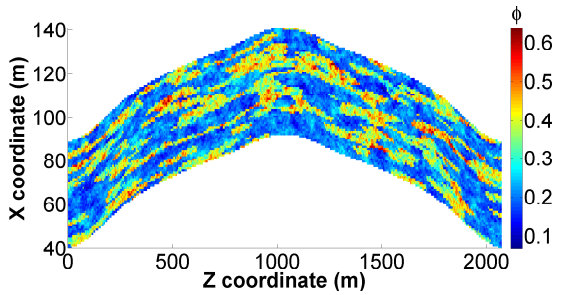

Figure 1.2: Porosity map

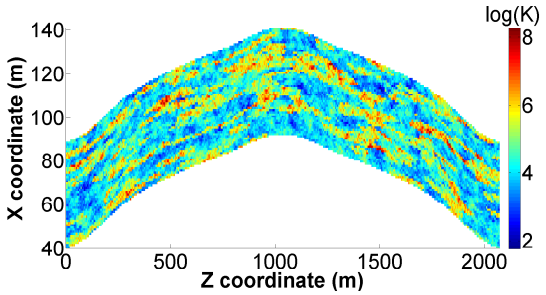

Figure 1.3: Permeability map (log).

1.4.3 A three-dimensional case, the PUNQ-S3 case

The PUNQ-S3 case (Floris et al. [35]), where PUNQ stands for Production forecasting with UNcertainty Quantification, is a 3D synthetic reservoir model derived from a real field of the North Sea Brent reservoir operated by Elf Exploration and Production. The problem was initially set up as a test case for the comparative study of inverse method for history matching. This case is a small synthetic case well known in industry. The geological model consists of a grid of $19 \times 28 \times 5$ cells (axes (x, y, z)), including 1761 active cells. The cell size is of $180m$ in x and y directions, and $18m$ in z direction. The model is composed of 5 geological layers with given permeability and porosity properties shown in Fig. 1.5. In Fig. 1.4 the PUNQ-S3 reservoir anticline geometry is represented. The fluid flow simulator used is PumaFlow^{TM} (PumaFlow^{TM} [65]). Tab. 1.4 summarizes information of the case.

Table 1.4: PUNQ case information model

Grid domain	**Grid Block Dimension**			**Simulator**
$19 \times 28 \times 5$	$Dx = 180$ m	$Dy = 180$ m	$Dz = 18$ m	PumaFlow^{TM}

PUNQ-S3 case can also be used to evaluate optimization methods for well placement problem, as in Bouzarkouna et al. [17], where the evolutionary algorithm CMA-ES is proposed for the determination of an optimal well configuration (with a fixed number of wells).

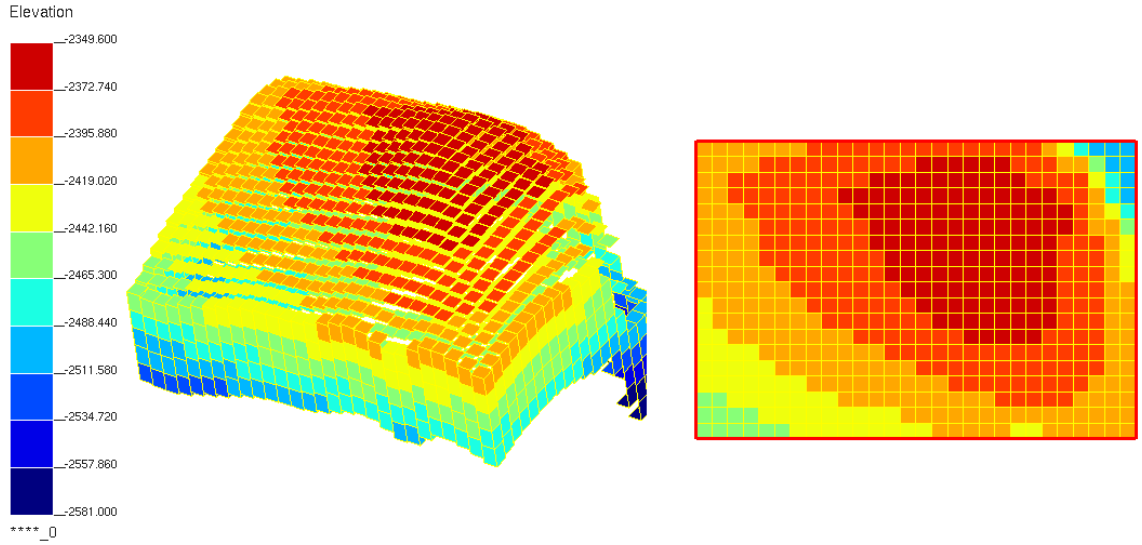

Figure 1.4: PUNQ-S3 case geometry Bouzarkouna et al. [17].

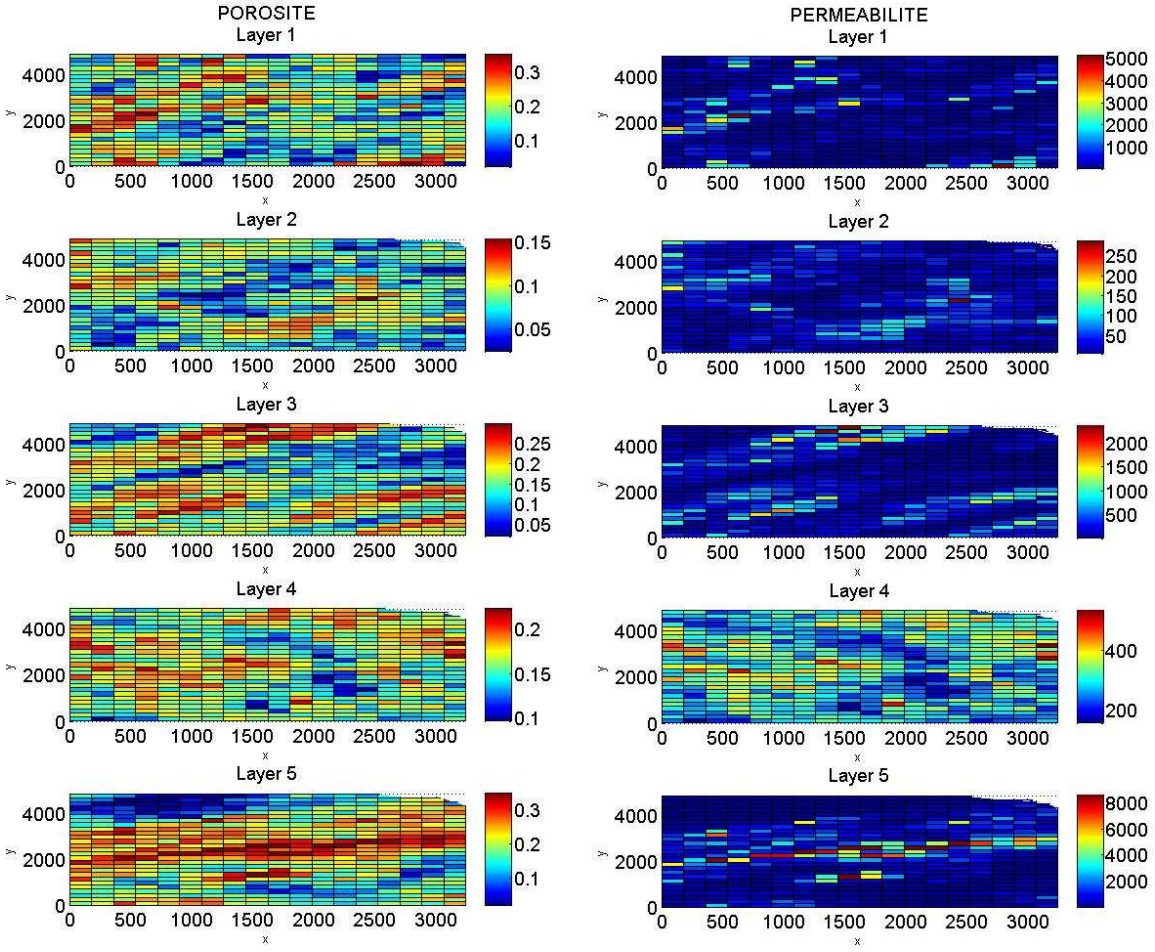

Figure 1.5: 5 layers of PUNQ-S3 case: porosity and permeability maps.

1.4.4 A three-dimensional case, the SPE10 case

This case is derived from the SPE10 reservoir case Project [64]. We consider a subpart of the whole reservoir, the Upper Ness sequence (see FIG. 1.6) composed of 50 layers of 2 ft with 60×220 cells of dimension 20 ft $\times$ 10 ft. In order to reduce the computational time of the simulation, the grid is upscaled to a grid of dimension $30 \times 110 \times 25$, with each cell of dimension 40 $ft \times 20$ $ft \times 4$ ft. The conditions at the injector well are an injecting rate of 5000 bbl/day (reservoir conditions), and a maximum injection bottom hole pressure of 10000 psi. The bottom hole pressure for producers is of 4000 psi.
Two fluid flow simulators are used, PumaFlow^{TM} [65] and a MATLAB implementation (Lie [51], Toolbox [73]). TAB. 1.5 summarizes information of the case.

Table 1.5: SPE10 case information model.

Grid domain	**Grid Block Dimension**			**Simulators**
$30 \times 110 \times 25$	$Dx = 40$ ft	$Dy = 20$ ft	$Dz = 4$ ft	PumaFlow^{TM} MRST Toolbox [73]

In FIG. 1.6 we represent the permeability in $log(m^2)$ (left: horizontal, right: vertical) of the SPE 10 reservoir model. Only upper Ness sequence is represented and considered.

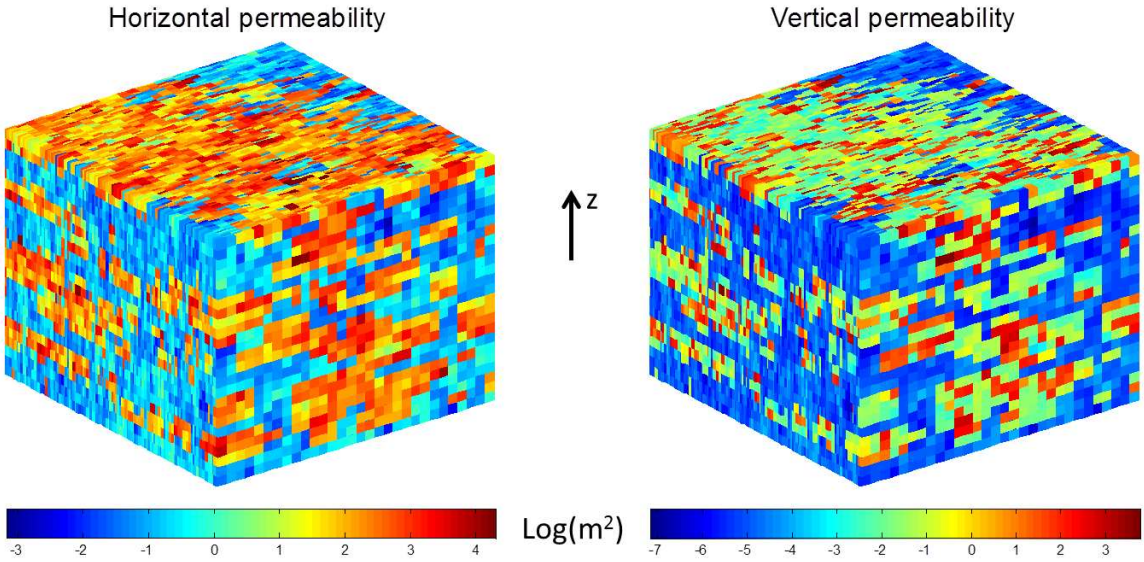

Figure 1.6: Permeability in $log(m^2)$ (left: horizontal, right: vertical) of the SPE 10 reservoir model for the upper Ness sequence.

Chapter 2

Unilateral well placement

The well placement problem requires the use of a fluid flow simulator seen as a black box, implying the availability of objective function values but no derivative information. The methods of interest in this case are Derivative Free Optimization methods (DFO) Conn et al. [26]. They use exclusively objective function values evaluated during the optimization, without substituting the objective function gradient by approximations (e.g., with finite difference). The objective is to solve an optimization problem, without knowledge on the derivatives of the objective function and constraints.

Quite a number of DFO methods have been developed over the years. Three groups are significant for our purpose: the Direct Search methods, the Sampling methods and the model based methods.

Direct Search methods, as the Pattern-Search method (Hooke and Jeeves [40]), evaluate and compare the objective function in points chosen depending on geometric criteria. The method is extended with Generalized Pattern Search (GPS) which allows a global convergence (Torczon [74]), and is improved with Mesh Adaptive Direct Search (MADS) by using dense search directions to ensure and accelerate the global convergence (Abramson et al. [6], Audet and Dennis Jr. [11]) .

Sampling methods or metaheuristics consist in exploring the domain of the feasible points. Sampling methods are usually inspired from natural systems in physics, (Simulated annealing), in biology, as in Genetic algorithms with the notion of natural selection introduced by Darwin at the 18th century. Some methods are also inspired from ethology, as Particle Swarm Optimization algorithms (Kennedy and Eberhart [45]) and ant colony optimization algorithms (Colorni et al. [24], Dorigo [29]), that were first implemented for combinatorial optimization and then adapted for continuous optimization Bilchev and Parmee [15]. Metaheuristics are generally iterative stochastic algorithms, they research a feasible solution while learning the characteristics of the problem to find an estimation of the better solution.

We can also find optimization methods using substitution models, known as surrogates, that replace the objective function by an estimation, cheaper in computational time to evaluate, such as Kriging methods, and other response surface as Radial Basis functions, Cosso, Sparce Grid, and Trust region methods. Trust region methods are one

the most important algorithm in solving NLP problems. They have several advantages including that they are based on quadratic models, useful for the information about the curvature function.

Other mixed integer DFO methods for engineering can be mentioned, such as García-Palomares et al. [39], J. [43], Laguna et al. [46], Liuzzi et al. [52], Müller et al. [59, 60], Newby and Ali [61], Vicente [76].

As DFO methods are well adapted for the optimization of Black-Box functions such as the NPV function, we selected and studied three of these methods. In a first section, we present a general algorithm for direct search methods, and present and test the NOMAD solver in which these methods are implemented. In a second section, we present a surrogate model based on a kriging method for mixed variables. A Derivative Free Trust Region method for mixed variables is proposed in the third section. We use and evaluate the methods by solving the well placement problem, for vertical or rectilinear wells, on different reservoir model cases, the simplified 1D case (see Section 1.4.1), and two 3D cases, the PUNQ-S3 (see Section 1.4.3), and the SPE10 test case (see Section 1.4.4).

2.1 Pattern-Search

Pattern-Search, introduced by Hooke and Jeeves [40], is a directed direct-search method. It uses a sampling, in a finite number of points, of the objective function. At each iteration, a predefined number of points is evaluated, these values allowing to determine the next points to be evaluated. A set of direction with adapted properties is used to guide the optimization. This method uses meshes of different sizes on which the functions values are evaluated and kept in memory. First proofs of convergence for NLP problems can be found in Torczon [74]. The method was adapted for the MINLP case by Audet and Dennis Jr [10], and Abramson [3] added a notion of filter for the handling of constraints with GPS methods. Constrained MINLP problems are studied in Abramson et al. [5] with GPS methods, and in Abramson et al. [6] with MADS. These methods are implemented in the NOMAD solver uses these methods.

2.1.1 Algorithm

Each iteration of the Pattern-Search algorithm has two steps, the search step and the poll step. The search step is used to find a point such as its objective function gives a better value than the current point noted x_k. If such a point is found, the search step is successful, otherwise a the poll step is used to find a better point by evaluating points in a mesh centered on the current point. The iteration is successful if one of the two steps gives a better point. At the end of the iteration, α_k, the current size of the mesh, is updated according to the failure or success of the iteration. $\mathcal{D}$ is the set of poll directions. In GPS methods $\mathcal{D}$ must be a positive basis to ensure the convergence: each element of the variable domain $\mathbb{R}^n$ can be expressed as a unique nonnegative linear combination of vectors in $\mathcal{D}$. In MADS methods the set $\mathcal{D}$ must be a positive spanning set, i.e., several nonnegative linear combinations of its elements must span the variables domain $\mathbb{R}^n$.

Directed Direct-Search Algorithm

Initialization

Choose x_0, $\alpha_0 > 0$, $0 < \beta_1 \leq \beta_2 < 1$ and $\gamma \geq 1$. Let $\mathcal{D}$ be a set of positive basis, and let α_{tol} be the smallest mesh size. Let $k = 0$.

While the mesh size $\alpha_k \geq \alpha_{tol}$.

(1.) **Search Step**

Try to compute a point that satisfies $f(x) < f(x_k)$ by evaluating the objective function in a finite number of points. If a better point is found, $x_{k+1} = x$, the iteration and the search step are successful, Go to the Mesh Update Step.

(2.) **Poll step**

Choose a positive basis $\mathcal{D}_k$ from the set $\mathcal{D}$. Order the set of poll points $\mathcal{P} = \{x_k + \alpha_k d \mid d \in \mathcal{D}_k\}$. Start the evaluations of f at poll points given the chosen order. If an evaluated poll point $x_k + \alpha_k d_k$ satisfies $f(x_k + \alpha_k d_k) < f(x_k)$, the polling is stopped, $x_{k+1} = x_k + \alpha_k d_k$, the iteration and the poll step is successful. Otherwise, the iteration (and the poll step) failed, and $x_{k+1} = x_k$.

(3.) **Mesh Update**

If the iteration is successful, the mesh size remains unchanged or increases: $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$. Otherwise, the mesh size is reduced: $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$. $k = k + 1$.

Pattern-Search methods are expensive in function evaluations. Mesh Adaptive Direct Search (MADS), by Audet and Dennis Jr. [11], uses more diversified sets of directions for the poll step. The well placement problem is a MINLP problem with constraints, so in the next two sections, we explain how to handle mixed integer variables and constraints in MADS methods.

2.1.1.1 Handling mixed integer variables

We aim at solving the following MILP problem:

$$\begin{cases} \min f(x, y) \\ x \in X \subseteq \mathbb{R}^n \\ y \in Y \subset \mathbb{N}^p, \end{cases} \tag{2.1}$$

where $x \in X \subseteq \mathbb{R}^n$ is a continuous variable, and $y \in Y \subset \mathbb{N}^p$ an integer variable. In Audet and Dennis Jr [10], the Mesh Adaptive Direct Search methods are generalized to mixed integer variables. A mixed integer neighborhood is proposed to define a local optimality of a point (x, y).

Point $(x, y) \in X \times Y$ is a local minimizer of f with respect to a set of neighborhood $\mathcal{N}(u^c, u^i) \subset X \times Y$ if there exists $\varepsilon > 0$ such as $f(x, y) \leq f(v^c, v^i)$ for all (v^c, v^i) in the set

$$X \cap \bigcup_{(t^c,t^i)\in\mathcal{N}(x,y)} \Big(B(t^c, \varepsilon) \times t^i\Big)$$

where $B(t^c, \varepsilon)$ is a ball of radius ε around t^c.
During exploration step and poll step, the integer part of the neighborhood set to evaluate optimality depends on the size of the mesh at iteration k.

2.1.1.2 Handling constraints

$$\begin{cases} \min f(x) \\ C(x) \leq 0 \\ x \in X \subseteq \mathbb{R}^n \end{cases} \tag{2.2}$$

Filter algorithms treat the optimization problem as bi-objective: minimize the objective function f and a nonnegative aggregate constraint violation function h.
Function h is defined as:

$$h(x) \begin{cases} = 0 & \Leftrightarrow C(x) \leq 0, \\ > 0 & \Leftrightarrow C(x) > 0. \end{cases}$$

$h_X(x) = h(x) + \psi_X(x)$ with ψ_X the indicator function of X:

$$\psi_X(x) = \begin{cases} 0 & \text{if } x \in X, \\ \psi_X(x) = +\infty & \text{else.} \end{cases}$$

Notion of dominance needs to be introduced: $x \prec x'$ if and only if $f(x) \leq f(x')$ and $h(x) \leq h(x')$ with either $f(x) < f(x')$ or $h(x) < h(x')$.

$x \preceq x'$ implies $x \prec x'$ or $f(x) = f(x')$ and $h(x) = h(x')$.

A filter $\mathcal{F}$ is a finite set of infeasible points of X and is defined such a way that no pair (x, x') in the filter are in the relation $x \prec x'$. Then a point x' is said to filtered if either $x' \succeq x$ for some $x \in \mathcal{F}$, or if $h_X(x') \geq h_{max}$, with h_{max} a positive upper bound on feasibility, or if x' is feasible and $f(x') \geq f^F$, with f^F the least objective function value found that far at a feasible point. Otherwise the point x' is unfiltered.

At iteration k, the set of filtered points $\overline{\mathcal{F}}_k$ is written as follow:

$$\overline{\mathcal{F}}_k = \bigcup_{x\in\mathcal{F}_k} \{x' | x' \preceq x\} \cup \{y | h_X(x') \leq h_{\max}\} \cup \{x' | h_X(x') = 0, f(x') \geq f^F\}$$

The exploration and poll steps presented in Section 2.1.1 are successful if unfiltered points are found. At the end of the iteration, unfiltered points are added to the filter $\mathcal{F}_{k+1}$ and the dominated points are removed.

In MADS methods, Audet and Dennis Jr. [11] combine aspects of filter algorithms to handle the constraints, and a "barrier" approach to maintain feasibility with respect to X.

2.1.2 The NOMAD solver

NOMAD is a solver, developed and implemented by Le Digabel [48]). The implemented algorithms are MADS and Generalized Pattern-Search (GPS) for Black-Box under constraints non linear optimization.

2.1.2.1 Available methods

In NOMAD, it is possible to choose the optimization algorithm between MADS and GPS; it is also possible to determine the set of poll directions.

The constraints can be handled in different ways. Some constraints have to be satisfied to allow the evaluation of the objective function. Other constraints can be violated and the objective function can still be evaluated, so the search space can be explored. NOMAD allows the definition of constraints that can be relaxed or not.

The user can also implement a search step with a strategy adapted to his problem.

Another advantage is the handling of "hidden constraints". When a point is rejected by the simulation, NOMAD avoids the neighborhood of this point. Substitution models, provided by the user, can be used by NOMAD in the poll step to order the most promising points before evaluating the "true" function. Discrete variables can be handled in NOMAD. Other available algorithms are adapted for the bi-objective optimization, for parallel executions. A Variable Neighborhood Search (VNS) algorithm to escape local minima is also available. Also, an opportunistic strategy can be chosen, it consists in terminating the evaluations of a list of points at a given step of the algorithm as soon as an improved value is found.

2.1.2.2 Use of NOMAD solver

NOMAD can be used in two different modes: batch and library. The batch mode is intended for a basic and simple use of the MADS method, while the library mode allows more flexibility. For example, in batch mode, users must define their separate Black-Box program that will be called with system calls by Nomad. In library mode users may define their Black-Box function as a C++ code that will be directly called by Nomad without system calls and temporary files. NOMAD is more adapted to Black-Box problems. The use of NOMAD requires to know well the problem in order to choose the suitable method amongst the available ones.

2.1.3 Applications of NOMAD

The NOMAD solver is used on challenging Black-Box location problems, such as groundwater supply and remediation in Fowler et al. [38], and the measure of snow water equivalent (an important factor for hydroelectric power generation) in Alarie et al. [8]. NOMAD is also used for optimization problems in thermal insulation (Abramson [4]). In this section, we use and evaluate the NOMAD solver capabilities on the well placement problem with three reservoir test cases.

2.1.3.1 Application on a 1D Case

The NOMAD solver uses Pattern-Search methods and can be used for Black-Box MINLP optimization. We use the MATLAB Opti toolbox.
The NOMAD solver allows us to solve MINLP optimization problems by using the Mixed Variable Programming algorithm. Wells are represented by two variables. In this section, we solve an MINLP optimization problem on a 1 dimensional test case presented in 1.4.1, for each permeability pattern (heterogeneous and homogeneous) given in Fig. 1.1.
The Black-Box MINLP optimization problem is written as:

$$\begin{cases} \min -NPV(x,y) \\ x \in [x_L, x_U] \subseteq \mathbb{R}^{n+m} \\ y \in \{0,1\}^{n+m} \\ g(x,y) \leq 0 \end{cases} \tag{2.3}$$

with the design variables $x \in \mathbb{R}^{n+m}$, and $y \in \{0,1\}^{n+m}$ representing respectively the locations of the wells and their status (active or not). $n = 2$ is the fixed number of producer wells and $m = 2$ the fixed number of injector wells. x_L, x_U are the bound vectors for $[1, 1000]^{n+m}$. The resulting optimization problem deals with 4 continuous variables and 4 binary variables.
We introduce symmetry breaking constraints to avoid obtaining two different solutions, which are symmetrical therefore equivalent: wells of a same type are ordered according to their coordinates. The number of active wells is the effective number of wells in the well configuration. We introduce lower and upper bounds for the number of active injector and producer wells. The considered NPV function is defined in Section 1.3 and its parameters are defined in Tab. 2.1.

Table 2.1: NPV function, parameter values.

Parameters	**Values**
C^{PO}	100 \$/m
C^{PW}	5 \$/m
C^{IW}	4 \$/m
C^{D}	1/6 \$

We launch the NOMAD solver from different starting points. The choice of the initial point is important: by giving a "bad" configuration from the start, the solution found is better in comparison to the initial NPV value, but solutions obtained by fixing the number of wells for each possible combination of binary (see Appendix A) are better. At this point we can remark that some configurations were evaluated several times uselessly because coordinates of inactive wells were modified.
We want to see if solutions found on NLP problems (the number of wells are fixed), represented in Appendix A, could be improved with NOMAD. We use as initial point the

two wells solution (1 injector, 1 producer) obtained by the fmincon MATLAB function from optimization toolbox and add to it two inactive wells (1 injector, 1 producer). The results are deceiving, the optimal solution found is usually the initial configuration. The reason is the inefficiency of the search step: sets of evaluated variables with 3 or 4 active wells did not provide better NPV function value, 4 wells configurations were rejected. Then we activate the wells that were added to the initial configuration (wells in blue in FIG. 2.1, FIG. 2.2, FIG. 2.3 and FIG. 2.4), and we obtain similar solutions in comparison to fmincon. The method demands 300 to 400 functions evaluations, this is much better than the fmincon function which required sometimes more than 100 evaluations for a single iteration. The starting point and solution of different runs of NOMAD are represented in FIG. 2.1 and FIG. 2.2 for the heterogeneous permeability pattern, and in FIG. 2.3 and FIG. 2.4 for the homogeneous pattern. The location of the wells in blue on the top configuration are providing from the optimization with fmincon. To this configuration, one active well is added in FIG. 2.1 and FIG2.3, and two active wells in FIG. 2.2 and FIG. 2.4. Under the starting configuration, we can observe the solution found by the NOMAD solver. In FIG. 2.1 and FIG. 2.2, we can observe that NOMAD improves the NPV function without obtaining a different number of active wells. In the homogeneous case, the 3 active wells starting configuration is improved by inactivating a producer well (see FIG. 2.3), and the 4 active wells starting configuration is improved by inactivating a injector well (see FIG. 2.4).

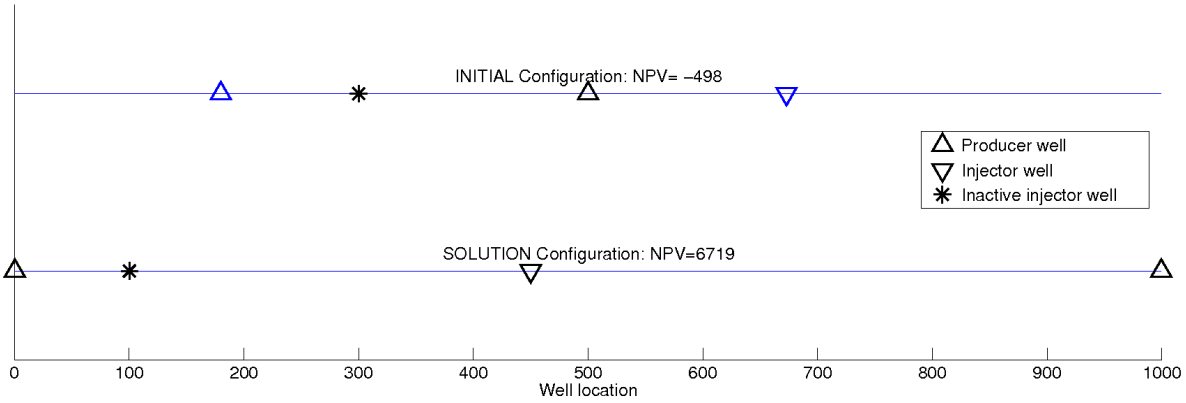

Figure 2.1: Optimization, initial configuration: 1 injector active well and 2 producer active wells, heterogeneous permeability.

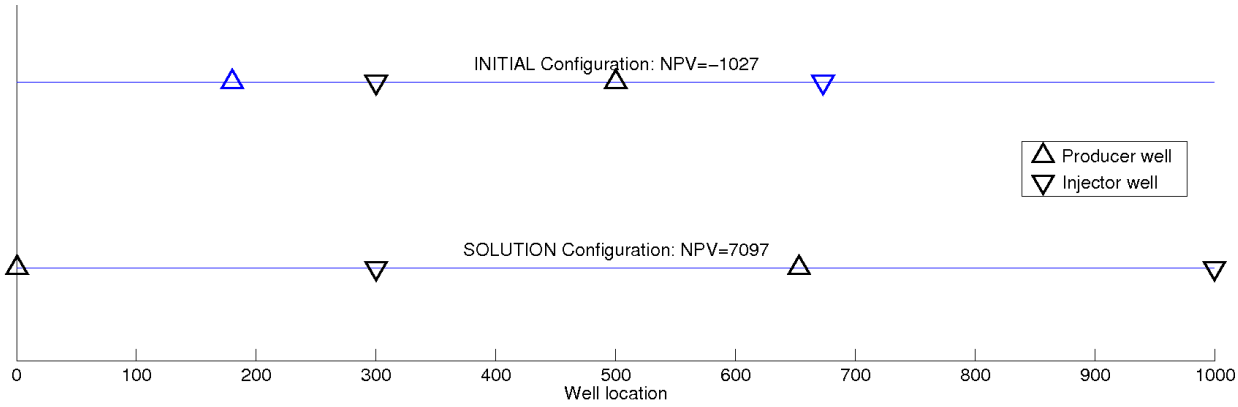

Figure 2.2: Optimization, initial configuration: 4 active wells, heterogeneous permeability.

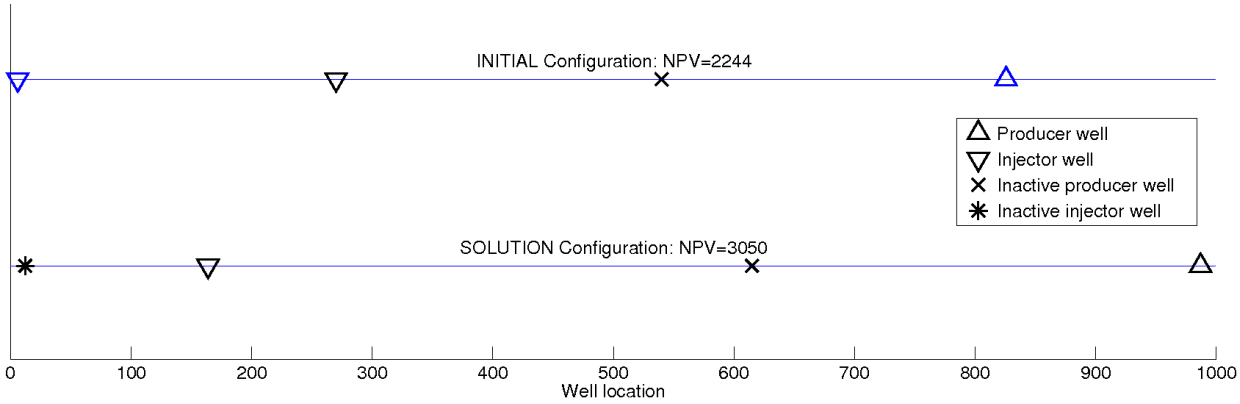

Figure 2.3: Optimization, initial configuration: 1 injector active well and 2 producer active wells, homogeneous permeability.

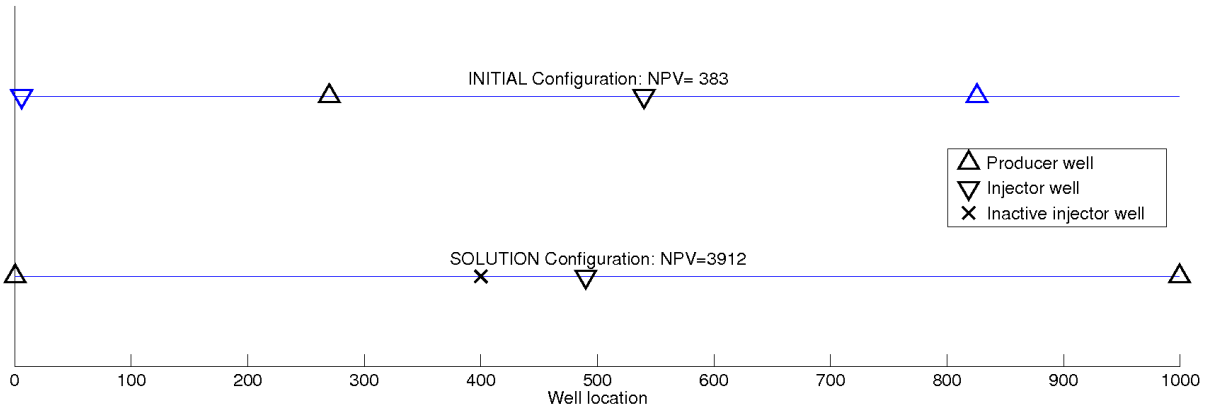

Figure 2.4: Optimization, initial configuration: 4 active wells, homogeneous permeability.

The results given by NOMAD are promising, but an adaptation of the software, to our application, is necessary. It could be interesting to implement a search step adapted to our application, e.g., modify coordinates of a well only if this well is active. This may avoid to evaluate several different configurations but equivalent because active wells are the same, which is uselessly costly in terms of evaluations and in simulation time.
Otherwise, the search step modifies first only continuous variables, and then modifies discrete variables. This is another aspect that would be interesting to add in our own search step: modify simultaneously continuous and binary variables. Another thing to explore is the use of surrogate models to substitute the cost function. The surrogate model being less costly to evaluate, the computational time decreases.

2.1.3.2 Application on a 3D Case: the PUNQ-S3 case

Now that we tested NOMAD on a 1D toy problem, we test it on a 3D test case, more complex. First we define the NPV function to optimize, then the constraints.

2.1.3.2.a The NPV function

The implemented NPV function is the function (1.2) presented in Section 1.2.2. Values of the objective function parameters for the test case PUNQ S3 are detailed in Tab. 2.2.

Table 2.2: NPV function parameters.

Parameters	**Values**
C_t^*	$C^*/(1+0.1)^t$
C^D	-1 \$
C^F	-200 \$/m
C^{PO}	600 \$/m^3
C^{PW}	-42 \$/m^3
C^{IW}	-24 \$/m^3
R^D	0

2.1.3.2.b Constraint definition

Bound constraints on optimization variables are introduced with the aim to limit the research domain and to define an adapted variable normalization. Binary variables, indicating if a well is closed or not, remain unchanged whereas continuous variables, associated to the coordinates of the well trajectory extremities, are normalized by:

$$x_{norm} = 2x/(u-l),$$

with l and u the lower and upper bounds of the constraints variable x, respectively. Well length is also bounded to an interval, typically:

$$180\text{m} \leq l \leq 1000\text{m}.$$

To avoid solutions that are non physical, or even that cannot be simulated, the following constraints are introduced:

I. Well trajectory must be in the reservoir. In the studied case, the reservoir is defined by its width and a grid of points giving the height of the top reservoir.

II. Well trajectories must not cross themselves, which can be translated by the following tests on trajectories taken 2 by 2:

- are trajectories $[A_1; A_2]$ and $[B_1; B_2]$ coplanar ?
  We compute the cross product $A_1A_2 \otimes B_1B_2$.
  If $\|A_1A_2 \otimes B_1B_2\| > 0$ and $< A_1B_1, A_1A_2 \otimes B_1B_2 > \geq 0$, then the 2 trajectories are coplanar,
- in the case of coplanar trajectories, the sign of $s(s-1)$ is tested with $s =< A_1B_1 \otimes B_1B_2, A_1A_2 \otimes B_1B_2 > / \|A_1A_2 \otimes B_1B_2\|$.
  If $s(s-1) < 0$ then the 2 trajectories do not cross.

To avoid the abortion of the simulation, we check that these constraints are satisfied and in case of violation the associated wells are closed.

2.1.3.2.c Optimization on 4 vertical wells (run1)

A first optimization is done with 4 vertical wells: the 12 optimization parameters are lateral coordinates of each well and their status (open or closed). It results in 8 continuous variables and 4 binary variables.

The solution is represented in FIG. 2.5: the 4 wells are open, the producer wells get closer of the top of the anticline, and the injector wells are located in periphery. The NPV function increases from a value of 2.10^9\$ to $1.2\,10^{10}$\$ (see FIG. 2.6). The evolution of iterates is represented in FIG. 2.7 and we can remark two important results, the status active or inactive of the wells varied during the optimization, and the region explored by the algorithm for continuous variables. Producer wells P1 and P2, represented by squares, explore mainly the top of the reservoir, while injector wells I1 and I2, represented by circles, explore the bottom of the reservoir. More than 600 simulations were necessary to obtain this solution (the stopping criteria of minimal threshold was reached), with 4.5 hours of execution time.

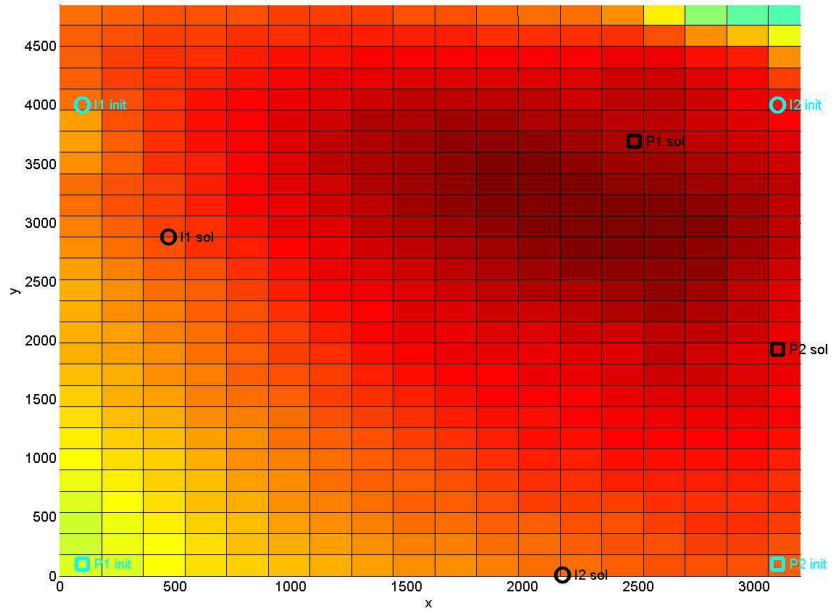

Figure 2.5: Initial and final location of optimized wells (run 1).

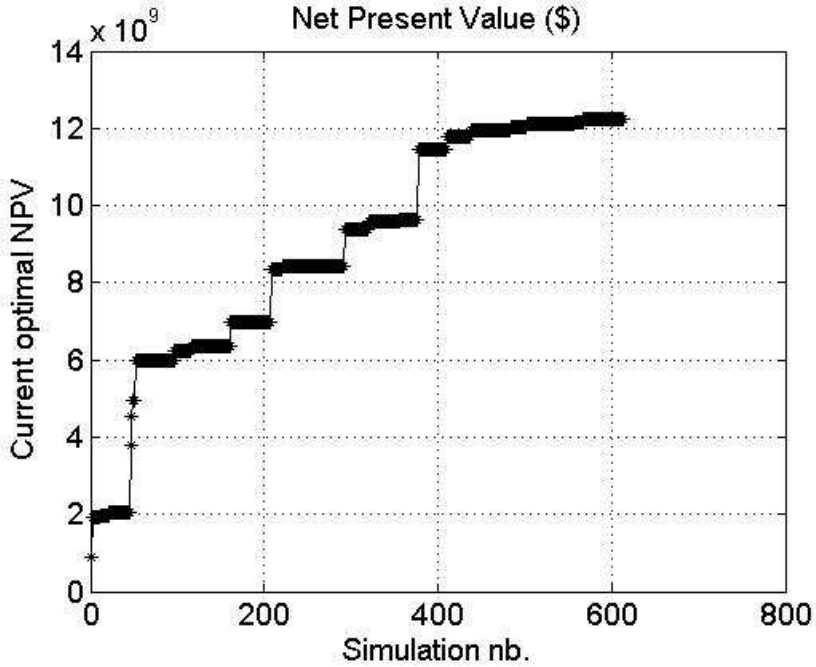

Figure 2.6: Evolution of the best objective NPV function along iterations of optimization (run1).

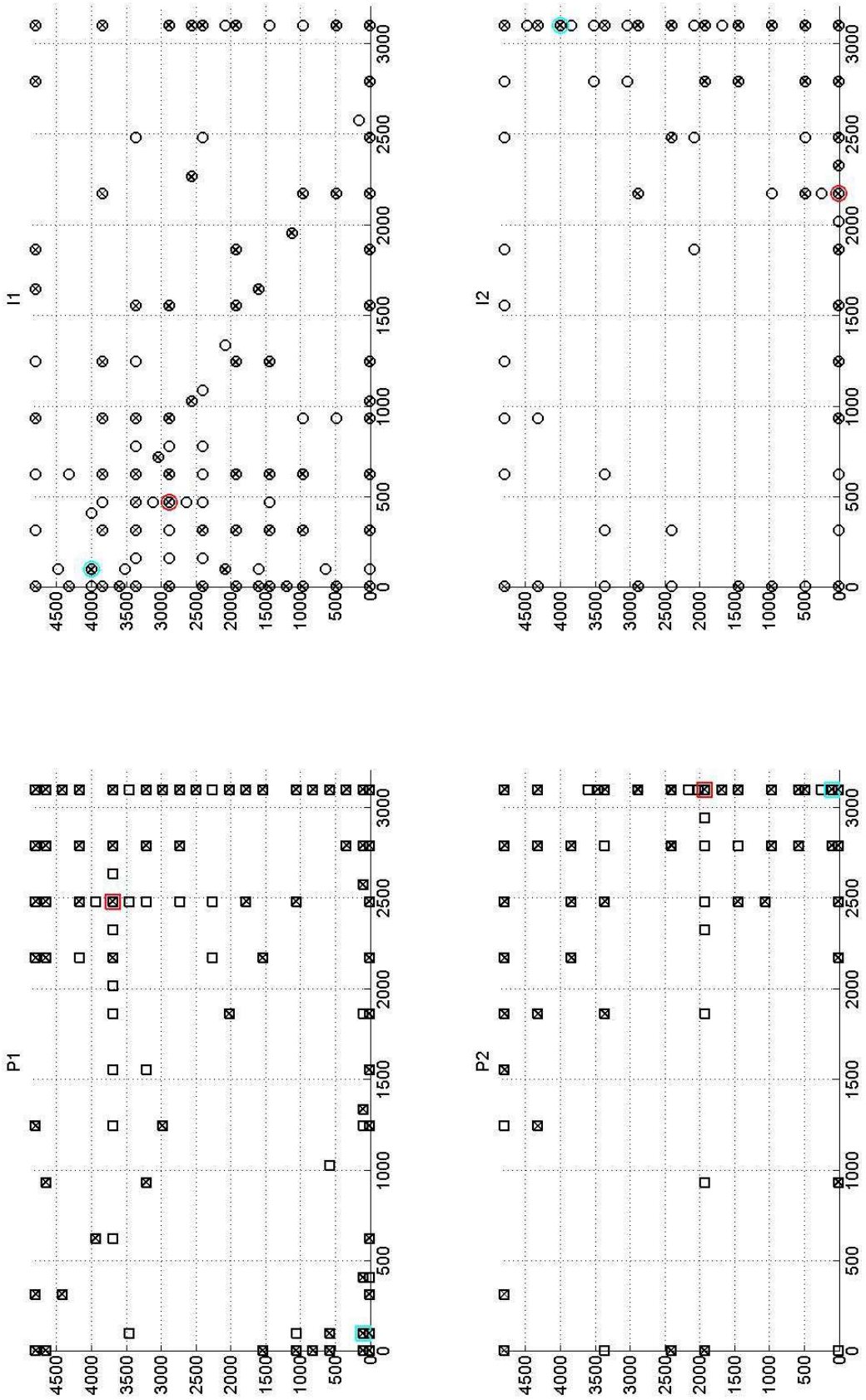

Figure 2.7: Movements of wells during optimization iterations (run1). A crossed symbol indicates that the well is closed. In blue: the initial solution, in red: final solution.

2.1.3.2.d Optimization of 4 non vertical wells (run2)

A second optimization on 4 wells (28 variables) with non vertical lateral trajectories is launched, starting from the obtained solution by the optimization of 4 vertical wells (Fig. 2.8). The configuration obtained as the solution (Fig. 2.9) corresponds to 4 open wells with 2 practically horizontal wells, 1 injector well and 1 producer well whereas the 2 other trajectories are more vertical. Fig. 2.12 illustrates the different trajectories taken by the wells during the optimization, again we can note that producer wells, represented by squares, explore mainly the top of the reservoir, while injector wells, represented by circles, explore the bottom of the reservoir. These trajectories are represented in Fig. 2.11 for the 200 first iterations of the optimization, thus we can observe more clearly the different locations taken by the wells and see that the status active or inactive of the wells varied during the optimization. The gain in terms of NPV function between the solution of 2 optimizations (vertical well and non vertical lateral well) is of $3.8\,10^9$\$. More than 1100 simulations were necessary to reach the convergence of the algorithm, with 8 hours of execution time.

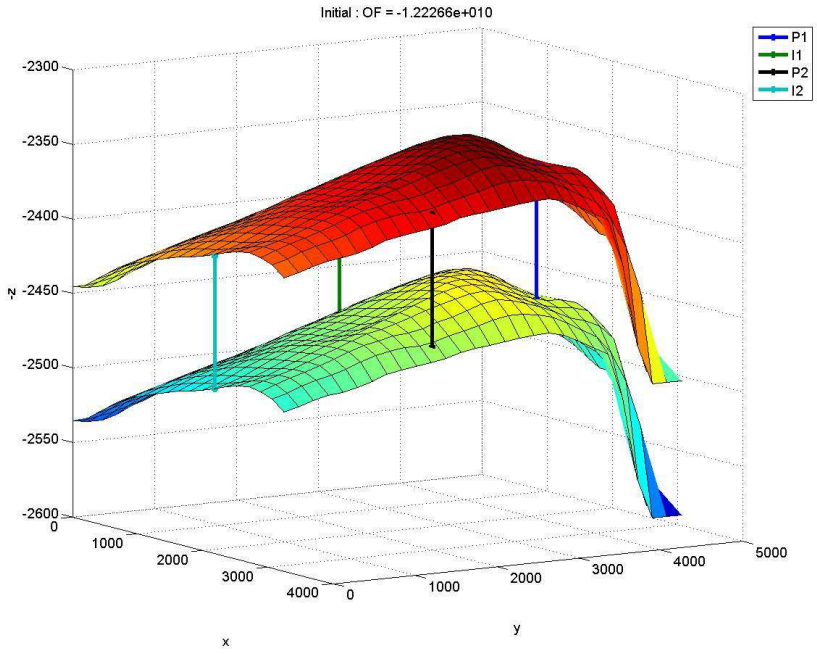

Figure 2.8: Initial location of well for the optimization (run2): solution of optimization (run1).

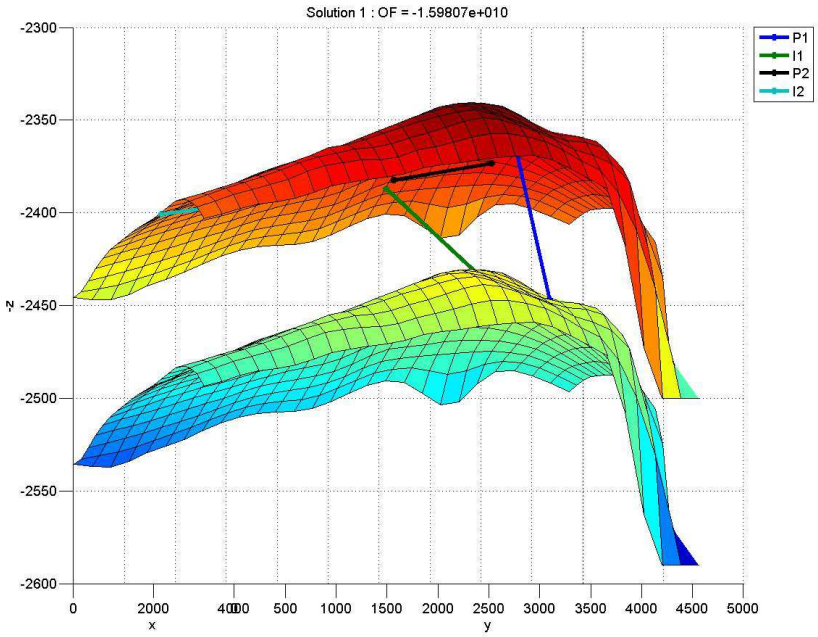

Figure 2.9: Solution of optimization (run2).

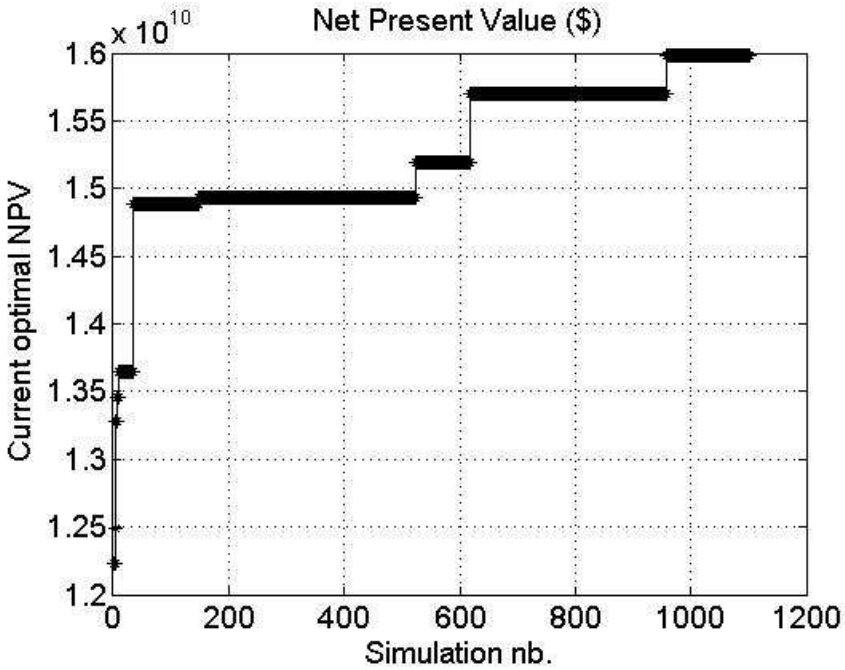

Figure 2.10: Evolution of the best objective NPV function along iterations of optimization (run2).

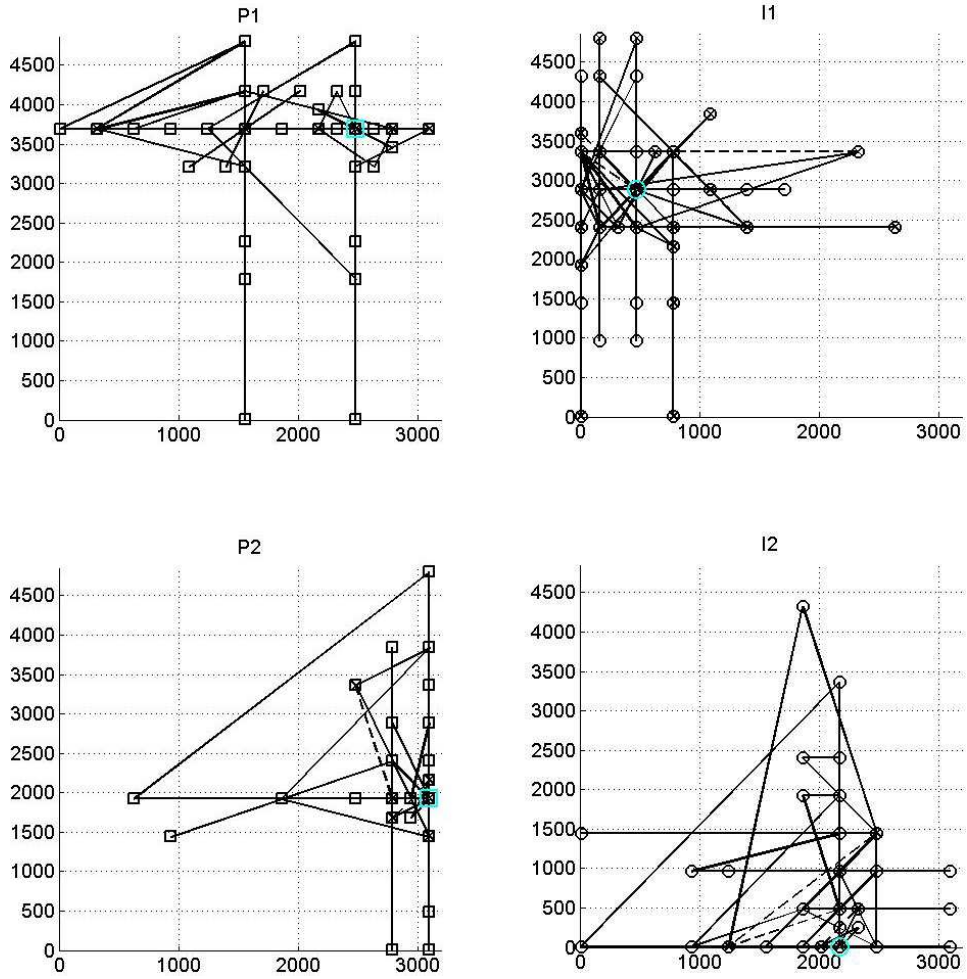

Figure 2.11: Trajectories of wells during the 200 first optimization iteration (run2). Dashed line indicates that the associated well is closed. In blue: the initial solution.

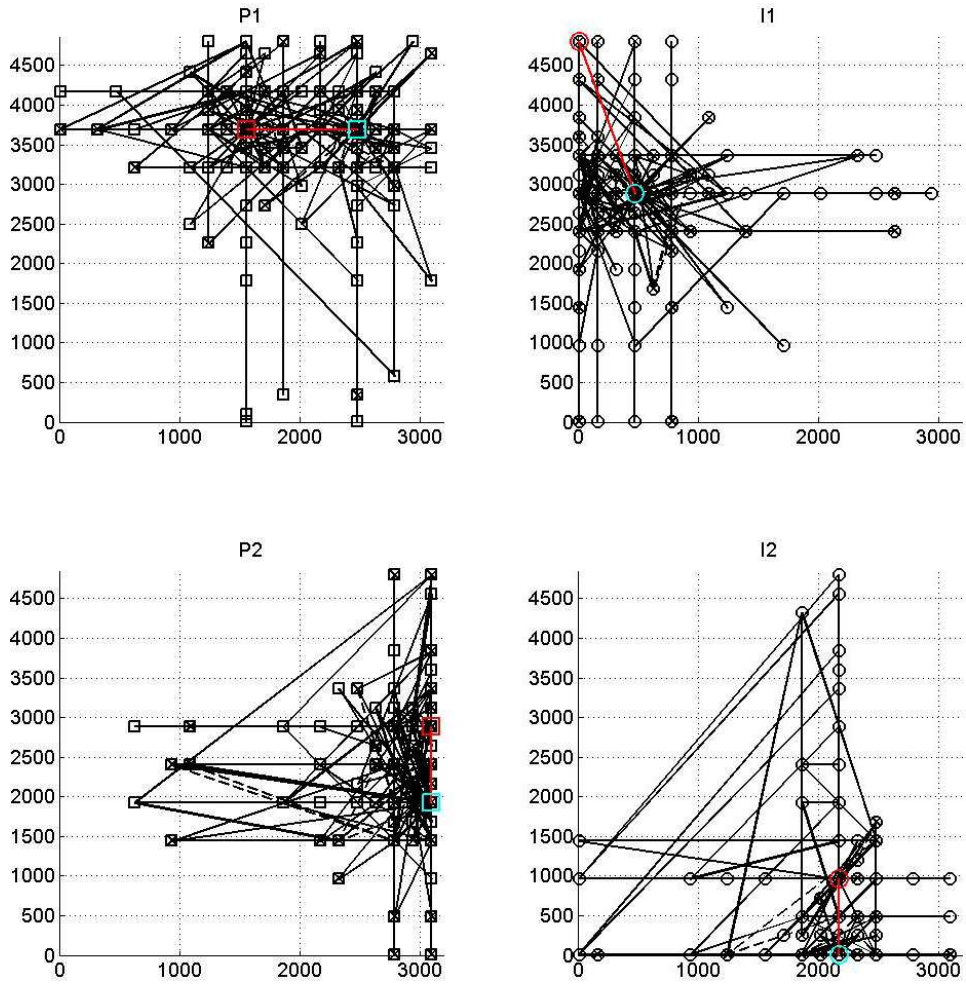

Figure 2.12: Trajectories of wells during optimization iteration (run2). Dashed line indicates that the associated well is closed. In blue: the initial solution, in red: the final solution.

2.1.3.3 Test on a 3D case

In this section we use the NOMAD solver on a 3D case, presented in Section 1.4.4. We optimize the location of vertical wells, and the number of vertical producer wells. The maximum number of producer well considered is 4, and we optimize the location of one vertical injector well. We use the NPVmax function, presented in Section 1.4, and give the parameters values in Tab. 2.3. The Optimization is made with a maximum bottom hole pressure of 4000 psi at producer wells, and an injection rate of 5000 bbl/day. The maximum time production is 10 years, and the production is subject to a water cut constraint (see Section 1.2.3).

The optimization problem is written as:

$$\begin{cases} \min -NPV(x,y) \\ x \in [x_L, x_U] \subseteq \mathbb{R}^p \\ y \in \{0,1\}^q \\ g(x,y) \leq 0 \end{cases} \tag{2.4}$$

where $x = \{x^1_{P1}, x^2_{P1}, x^1_{P2}, x^2_{P2}, x^1_{P3}, x^2_{P3}, x^1_{P4}, x^2_{P4}, x^1_I, x^2_I\}$, (x^1_{Pi}, x^2_{Pi}) are the two coordinates necessary to define vertical producer well $i = 1 \dots 4$, and (x^1_I, x^2_I) are the coordinates of the injector well,

$y = \{y_1, y_2, y_3, y_4\}$ are binary values associated with the status of the producer well (0: inactive well, 1: active well), x_L and x_U are the bound vectors for x^1 and x^2.

The resulting optimization problem deals with 10 continuous variables and 4 binary variables.

480 evaluations of the objective function were necessary to solve the optimization problem with NOMAD, with 7 days of execution time.

Table 2.3: NPV function, parameters values.

Parameters	**Values**
C^{PO}	50 \$/bbl
C^{PW}	5 \$/bbl
C^{IW}	5 \$/bbl
C^F	45 \$/ft
C^D	1e7 \$
R	0.1

Tab. 2.4 summarizes the different results obtained for each well configuration: the reference configuration, presented in Section B.3, the starting point of the optimization, and the solution of the optimization. The reference configuration and the starting point have both 4 active producer wells, whereas the solution has only one producer. Despite this single producer well, the cumulative volume of produced oil is greater than the volume of the reference configuration. The cumulative volume of produced water is a bit more of half the starting point water produced volume. As the injecting rate is fixed, the total volume of injected water depends of the production time frame, 7 years for the starting point, and the solution, 6 years for the reference configuration.

Table 2.4: NPV function values.

	Starting point	Solution	Reference configuration
Number of active producers	4	1	4
NPV value	50.73*e*6 $	160.49*e*6 $	115.63*e*6 $
Total cumulative produced oil	4.62*e*6 *bbl*	5.96*e*6 *bbl*	5.32*e*6 *bbl*
Total cumulative produced water	8.49*e*6 *bbl*	4.91*e*6 *bbl*	5.70*e*6 *bbl*
Total cumulative injected water	12.87*e*6 *bbl*	12.87*e*6 *bbl*	11.03*e*6 *bbl*
Period of maximum production	7 years	7 years	6 years

In FIG. 2.13 we can see the evolution of the value of the best NPV value at each simulation. In FIG. 2.14, the NPV function for each year of production is represented for the starting point, the optimization solution and the reference configuration.

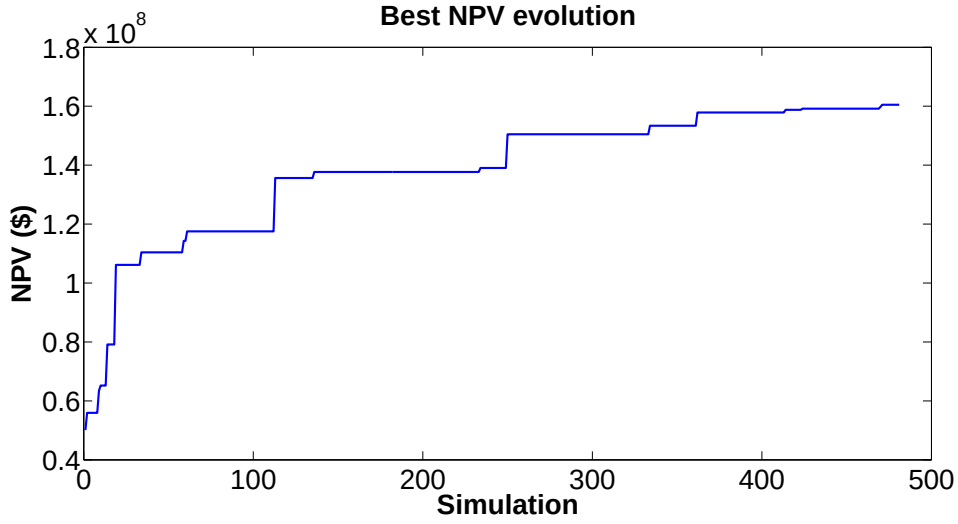

Figure 2.13: Best NPV during optimization.

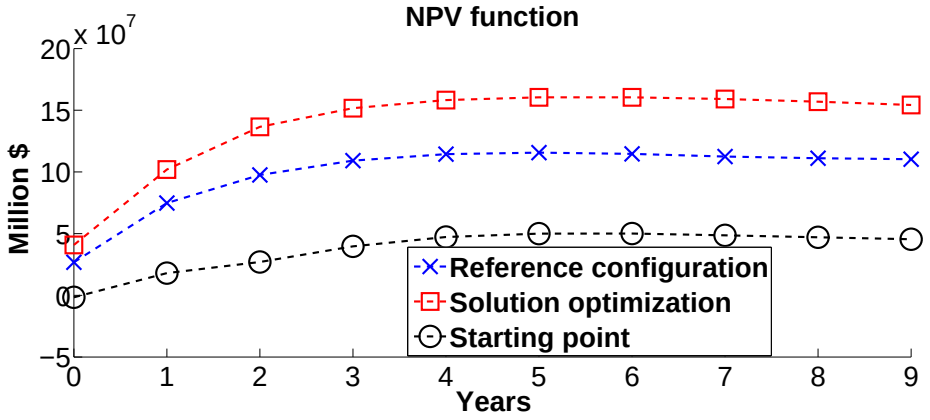

Figure 2.14: NPV function.

2.1.3.3.a Starting point

In FIG. 2.15 we represented the cumulative water production for each producer wells of the starting configuration. Well $P3$ (in red) started to produce water in the first year of production, and its production stopped after 5 years of production. This is due to the water cut constraint that was not satisfied any more (see FIG. B.4 in Appendix B), thus the well was shut in. In FIG. 2.16, we can also see that the oil production was stopped for Well $P3$ after 5 years. We can observe in FIG. 2.16 that the increase of oil production of wells $P1$, $P2$ and $P4$ was higher after five years, which is not a good well configuration because a stronger weight in the NPV function is associated to early production than to late production.

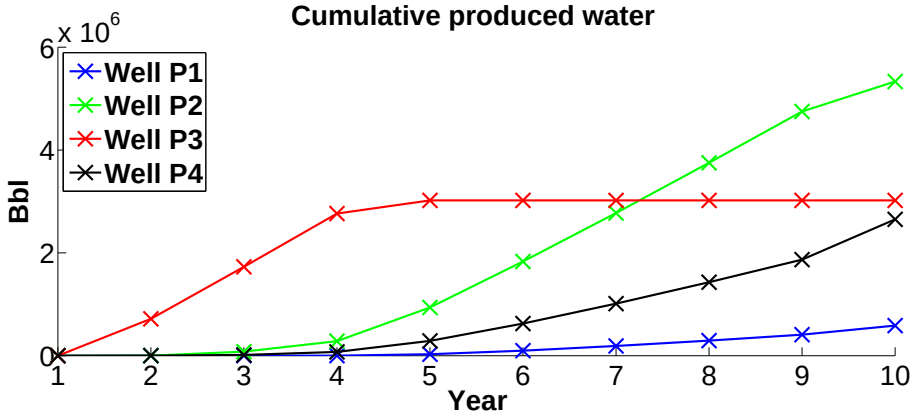

Figure 2.15: Cumulative water production of the starting point of the optimization (1 injector well and 4 producer wells).

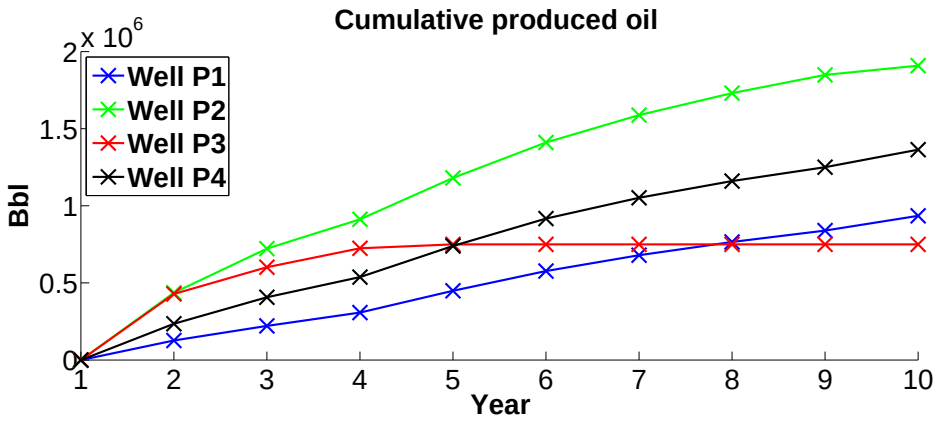

Figure 2.16: Cumulative oil production of the starting point of the optimization (1 injector well and 4 producer wells).

2.1.3.3.b Optimization solution

The cumulative water production of the optimization solution is represented in FIG. 2.17. Water production started for in the third years of production, and we can observe an higher increase after 6 years, which is the time production of the NPVmax function. FIG. 2.18 shows that half of the oil production was accumulated in the 4 first years.

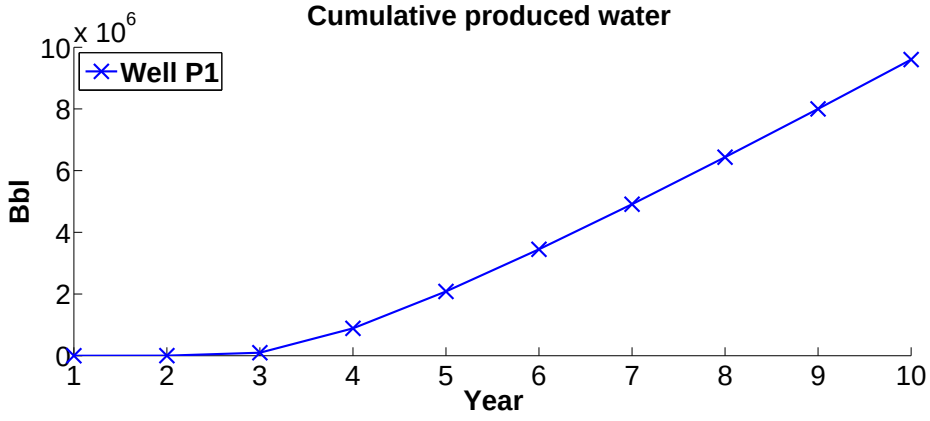

Figure 2.17: Cumulative water production of the solution configuration the optimization (1 injector well and 1 producer well).

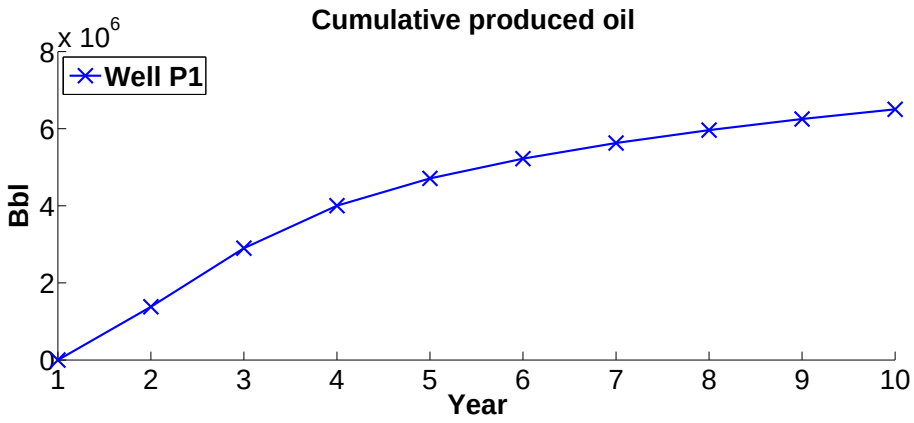

Figure 2.18: Cumulative oil production of the solution configuration the optimization (1 injector well and 1 producer well).

2.1.3.3.c Reference configuration

We consider the reference configuration case: 4 producer wells at each corner of the domain, and 1 injector well at the center. Cumulative water production in FIG. 2.19 and cumulative oil production in FIG. 2.20. In FIG. 2.19 we represented the cumulative water production for each producer wells of the reference configuration. The water production of Well $P4$ (in black) stopped after 9 years (the water cut constraint that was not satisfied any more (see FIG. B.12 in Appendix B), thus the well was shut in. In FIG. 2.20, we can also see that the oil production was stopped for Well $P4$ (in black) after 9 years. FIG. 2.20 shows that half of the oil production was simulated in the 3 first years.

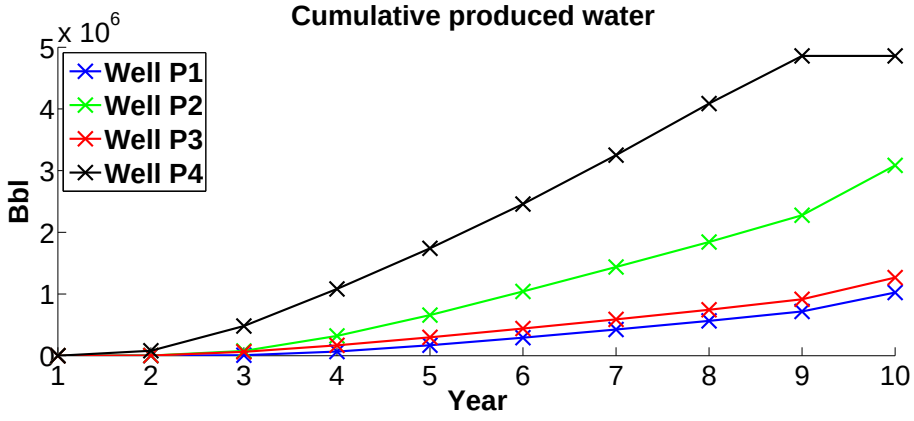

Figure 2.19: Cumulative water production of the reference configuration the optimization (1 injector well and 4 producer wells).

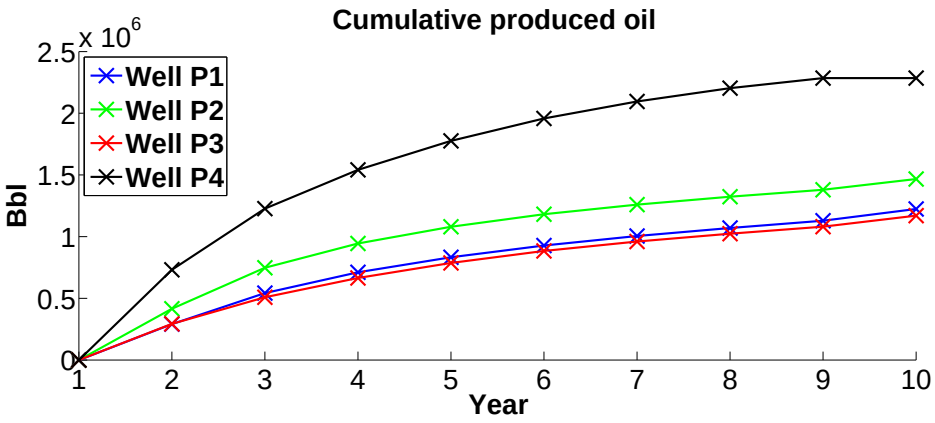

Figure 2.20: Cumulative oil production of the reference configuration the optimization (1 injector well and 4 producer wells).

2.1.4 Conclusion

We used the NOMAD solver, in which are implemented MADS methods, to optimize the number and the location of vertical or rectilinear wells on several reservoir model cases. The NOMAD solver was first evaluated on the 1D case (see Section 1.4.1) for two permeability patterns. For each of these permeability case, we used two optimization starting points and we obtained different NPV function solution values and configurations with different number of active wells. Hence depending on the initialization, the optimization with NOMAD can be trapped in local minima.

The NOMAD solver was also tested on the PUNQ-S3 case (see Section 1.4.3), and optimized the number and location of 4 vertical wells. A good solution was obtained in 600 simulations. Starting from this solution, a second problem optimizing the number and location of rectilinear wells allows to obtain a better configuration in 1100 simulations. Hence the first simple problem allow to obtain a good initial guess for the second more advanced problem.

Finally, we optimized the location and number of vertical producer wells and the location of 1 injector well on the SPE10 test case (see Section 1.4.4). The solution, with a well configuration of 1 producer well and 1 injector, is obtained in 7 days and 480 simulations. This solution is better than the reference point of 4 wells (with 4 producer wells at each corner of the reservoir and one single centered injector wells), in terms of NPV function and of oil production. We can observe here the necessity to include the number of wells in the optimization, since it allow to obtain configuration that reduce the drilling costs and increase the oil production.

NOMAD allows to obtain satisfying results for the optimization of the placement and number of wells, even though the method requires numerous evaluations of the objective function which is costly in CPU time, thus it could be interesting to couple the algorithm with substitution models.

2.2 Kriging and Efficient Global Optimization (EGO)

In this section we treat the optimization problem by using surrogate models. To minimize the number of expensive evaluations, the objective function is replaced by an approximation model, computed from a limited sample of simulated points. First we present a kriging method adapted for mixed variables, then we present two optimization algorithms, a basic optimization approach, and the well-known EGO (Efficient Global Optimization) algorithm proposed by Jones et al. [44].

2.2.1 General surrogate model for mixed quantitative and qualitative variables

We propose to use the popular kriging method approach to approximate the objective function by the mean function of an underlying stochastic process. Zhou et al. [80] proposed an adapted kriging approach for mixed variables.

Let $w = (x, z)$ an input vector, where $x = (x_1, \ldots, x_I)$ is a vector in $\mathbb{R}^I$ and represents the quantitative factors, and $z = (z_1, \ldots, z_J)$ is a vector of J qualitative factors. Each component z_j, with $j \in \{1, \ldots, J\}$, has b_j levels that are categorical and not supposed ordinal. We note $\mathcal{C}$ the set of all level combinations for qualitative factor $\mathcal{C} = \{c_1, \ldots, c_m\}$, where $m = \prod_{j=1}^{J} b_j$. Then we note $w = (x, c)$ with $c \in \mathcal{C}$, and consider the following model:

$$y(w) = f^t(w)\beta + \varepsilon(w) \tag{2.5}$$

where

- $f(w) = (f_1(w), \ldots, f_p(w))$ is a set of p regression functions,
- $\beta = (\beta_1, \ldots, \beta_p)$ is a vector of unknown coefficients,
- $\varepsilon(w)$ is a stationary Gaussian process with mean 0, variance σ^2, and a parametrized covariance function.

$\varepsilon(w)$ is the residual of the regression model $f^t(w)\beta$ in Expression (2.5). A valid correlation structure for this model is:

$$cor(\varepsilon(w), \varepsilon(w')) = cor(\varepsilon_c(x), \varepsilon_{c'}(x')) = \tau_{c,c'} R(x, x'). \tag{2.6}$$

where $\varepsilon_c(x) = \varepsilon(x, c)$, $\tau_{c,c'} = \tau_{c',c}$ is the cross-correlation between categories c and c', and $R(x, x')$ is the a correlation function for quantitative values. A common choice for R is the Gaussian correlation function, leading to Expression (2.7):

$$cor(\varepsilon(w), \varepsilon(w')) = \tau_{c,c'} \prod_{i=1}^{I} \exp\left(-\phi_i (x_i - x_i')^2\right), \tag{2.7}$$

where the unknown roughness parameters ϕ_i are grouped in $\Phi = \{\phi_i\}$.

In order to have a valid correlation function, the $m \times m$ matrix $T = \{\tau_{r,s}\}$ has to be positive symmetric definite with unit diagonal elements (Qian et al. [67]), to be nonsingular. Nevertheless, the optimal tuning of $\{\tau_{r,s}\}$ is a difficult problem which can be efficiently simplified by using a hypersphere decomposition (introduced for financial applications by Rebonato and Jaeckel [68]). It consists in 2 steps, first a Cholesky decomposition is applied to T, $T = LL^t$, where L is a lower triangular matrix with strictly positive diagonal elements. In the second step, each row vector $(l_{r,1}, \ldots, l_{r,r})$ of matrix L is modeled as coordinates of a surface point in an r-dimensional unit hypersphere:

$$\begin{cases} l_{r,s} &= \prod_{j=1}^{s-1} \sin(\theta_{r,j}) \cos(\theta_{r,s}), \text{ for } s = 1, \ldots, r-1, \\ l_{r,r} &= \prod_{j=1}^{r-1} \sin(\theta_{r,j}). \end{cases} \tag{2.8}$$

where $\theta_{r,s} \in]0, \pi[$. This restriction leads to strictly positive elements of matrix $\Theta = \{\theta_{r,s}\}$ in (2.8). Thanks to trigonometric properties, it is easy to verify that the product LL^t will be symmetrical with diagonal elements equal to 1.
It requires less parameters to compute one matrix for each qualitative variable than to use one for all the combinations between all variables as in (2.7). The expression of the correction function becomes:

$$cor(\varepsilon(w), \varepsilon(w')) = \prod_{j=1}^{J} \tau^j_{z_j, z'_j} \prod_{i=1}^{I} \exp\left(-\phi_i (x_i - x'_i)^2\right) \tag{2.9}$$

The number of parameters Θ required in Expression (2.7) is $(m-1)(m-2)/2$, whereas the number of parameters needed in Equation (2.9) is $\sum_{j=1}^{J} (b_j - 1)(b_j - 2)/2$.

The unknown parameters of Model (2.5) are Φ, Θ, β, and σ^2. Given a sample $W = \{w^1, \ldots, w^n\}$ of n inputs values, maximum likelihood is used to estimate these parameters, the resulting estimators are $\widehat{\Phi}$,$\widehat{\Theta}$, $\widehat{\beta}$, and $\widehat{\sigma^2}$. Then a non-biased estimator is given by expression (2.10):

$$\widehat{y}(w) = f(w)\widehat{\beta}^t + \widehat{r}(w)\widehat{R}^{-1}(y - F\widehat{\beta}). \tag{2.10}$$

where

- $\widehat{r}(w) = (cor(\varepsilon(w), \varepsilon(w^1)), \ldots, cor(\varepsilon(w), \varepsilon(w^n)))$ is a vector composed of the correlations between w and each sample value,
- $\widehat{R}$ is the estimated correlation matrix of R using equation (2.7),
- F is a $n \times p$ matrix of the regression values $f(w^i) = (f_1(w^i), \ldots, f_p(w^i))$ for each w^i in W, $i = 1 \ldots, n$,
- $\widehat{y}(w)$ is the prediction of the model for the value w.

To implement and test the model, we use the MATLAB toolbox DACE (Lophaven et al. [56]) for the computation of the quantitative correlation part. For the qualitative part, a matrix T is computed for each combination of the qualitative values.

2.2.2 Model quality evaluation

The model quality is estimated with two values: the Q2 expressed in (2.11) and the RMSE (Root Mean Square Error) expressed in (2.12). Both take into account errors between simulated points and observed points, and Q2 gives more weight to the points closed from the mean of the observed points.

$$Q2 = 1 - \frac{\sum\limits_{w \in \overline{W}} (\hat{y}(w) - y(w))^2}{\sum\limits_{w \in \overline{W}} ((y(w) - mean(y(w)))^2)}, \tag{2.11}$$

where $\hat{y}$ is the estimation of the model, y the values, and $\overline{W}$ is the validation sample on which the model is evaluated.

$$RMSE = \sqrt{\frac{\sum\limits_{w \in \overline{W}} (\hat{y}(w) - y(w))^2}{|\overline{W}|}}. \tag{2.12}$$

2.2.3 Kriging model evaluation on a mixed integer NPV function

In this section we evaluate the efficiency of the kriging method on NPV function associated with the 1D reservoir model presented in Section 1.4.1. Models are built with Gaussian correlations, and constant regression function. We use different size of sample to build the kriging models to evaluate its accuracy, depending on the sizes of the experimental plan. After building the different samples for the construction of the models, and for its validation, we rejected points that did not satisfy constraints of distance between wells. So, the actual size of the samples is not the initial size.

The different configurations for the number and type of well are the categories of the function. We can have at most 3 wells, and at least 1 well of each type. There are only three different possible categories, one composed of two wells (1 injector, 1 producer), and two others composed of three wells, (1 injector, 2 producer and 2 injector, 1 producer).

We use sliced Latin Hypercube design based on Qian [66]. This design is a Latin Hypercube Sample (LHS) for the continuous factors and is sliced into groups of smaller Latin Hypercube designs associated with different categorical levels. In this case, we generate a sliced Latin hypercube design with m slices, where each slice of n runs corresponds to one category. We do not use the same size of LHS for each category. We have three categories, which do not require the same amount of information to be modeled. We note n_2 the size of sample of 2 wells, and n_3 the size of sample of 3 wells.

We build models from sliced LHS samples composed of 3 concatenated LHS: the first slice is a LHS of size 50 for the configuration of two wells, and the two other slices are LHS of size 100 for the two configurations of 3 wells. We use also a sliced LHS for the validation of the model. We evaluate the kriging model on the validation sliced LHS, the results for each slices are represented in Appendix C.2, and summarized in Tab. 2.5 and Tab. 2.6.

2.2.3.1 Validation of the kriging model build from the sliced LHS, sample of size $n_2 = 50$, and $n_3 = 100$

Here we test the kriging model built from the sliced LHS of 250 points on the validation sliced LHS. In FIG. 2.21 we can see that several points have residuals more than 1000, which means the model prediction was not accurate on these points. Crossplot is displayed in FIG. 2.22, and we can see that they are some irregular points, but most of the predictions are reliable.

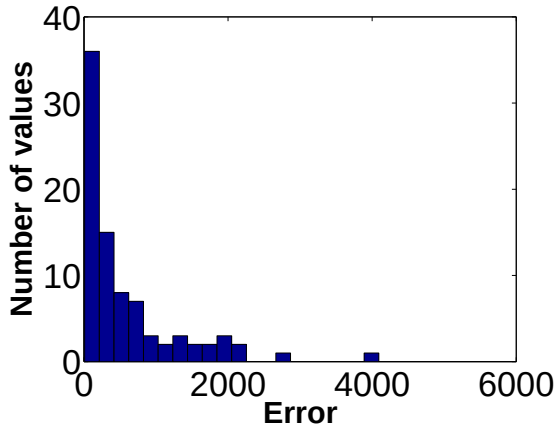

Figure 2.21: Residuals of the validation sliced LHS, estimation with kriging model built with a sliced LHS $n_2 = 50$, $n_3 = 100$.

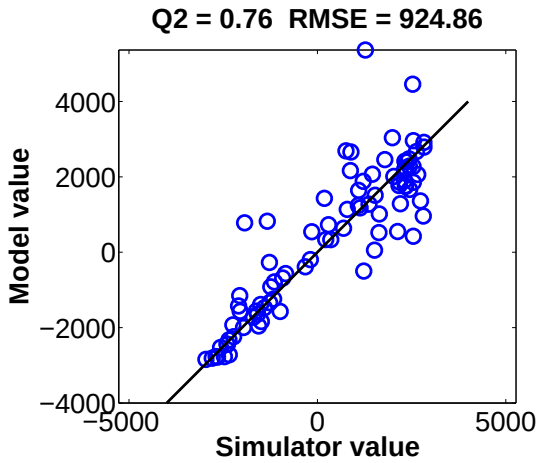

Figure 2.22: Crossplot, tested on the validation sliced LHS, model built with a sliced LHS $n_2 = 50$, $n_3 = 100$.

Table 2.5: Q2 and RMSE results of quantitative and qualitative model, model built from sliced LHS $n_2 = 50$, $n_3 = 100$.

Validation sample	Sliced LHS	Slice 1 1 producer 1 injector	Slice 2 1 producer 2 injector	Slice 3 2 producer 1 injector
Q2 model built from corresponding slice	–	0.994	0.79	0.46
Q2 model built from sliced LHS	0.76	0.99	0.89	0.27
RMSE model built from corresponding slice	–	156	829	1225
RMSE model built from sliced LHS	924	181	727	1425

2.2.3.1.a Test on the 3 sliced LHS, sample of size $n_2 = 100$ and $n_3 = 200$

Here we test the kriging model built from the sliced LHS of 500 points on the validation sliced LHS. In Fig. 2.23 we can see that most of the validations points have residuals under of 1000. Crossplot is displayed in Fig. 2.24, and we can see that they are some irregular points as in the model built from 250 points (see Fig. 2.22).

Table 2.6: Q2 and RMSE results of quantitative and qualitative model, model built from sliced LHS $n_2 = 100$, $n_3 = 200$.

Validation sample	Sliced LHS	Slice 1 1 producer 1 injector	Slice 2 1 producer 2 injector	Slice 3 2 producer 1 injector
Q2 model built from corresponding slice	–	0.991	0.97	0.88
Q2 model built from sliced LHS	0.89	0.951	0.82	0.87
RMSE model built from corresponding slice	–	386	305	580
RMSE model built from sliced LHS	620	457	767	601

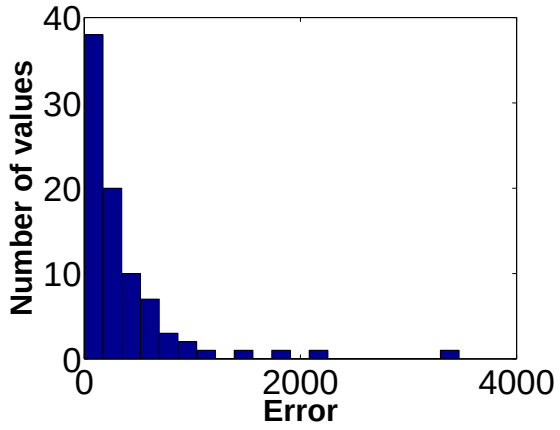

Figure 2.23: Residuals of the validation sliced LHS, estimation with kriging model built with a sliced LHS $n_2 = 100$, $n_3 = 200$.

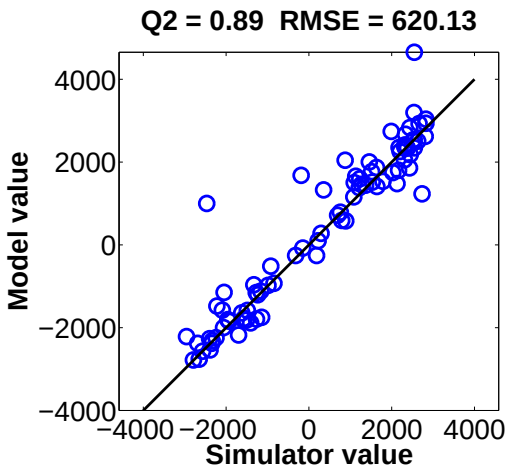

Figure 2.24: Crossplot, tested on the validation sliced LHS, model built with a sliced LHS $n_2 = 100$, $n_3 = 200$.

A large number of points is required to build a reliable model, but the models we obtained can give us an acceptable approximation, that can be used to guide the optimization.

2.2.4 Optimization with kriging

In this section, we present two optimization algorithms based on kriging models, a basic optimization approach, and the known EGO algorithm of Jones et al. [44]. Then we test and compare the two algorithms by maximizing the NPV function on the 1D reservoir model.

2.2.4.1 Optimization with kriging: a basic algorithm

A first idea to optimize with surrogate kriging models, is simply to successively optimize the model using NOMAD (see Section 2.1.2), and update the model with the solution of the optimization.

Optimization Algorithm

---

0. **Initialization**

   Let W_0 be the initial sample, $\mathcal{M}_0$ be the kriging model obtained from the sample W_0. Let ϵ, and let *maxiter* be the maximum number of iterations. Note (x^*, c^*) the best current solution, and f^* its associated objective function value.

1. **Iteration k: Maximize the model**

   Solve optimization problem, and note (x^k, c^k) its solution, and $\widehat{f}^k$ its objective function value:

$$\begin{cases} \min\limits_{x,c} \mathcal{M}_k(x, c) \\ x \in X \subseteq \mathbb{R}^n \\ c \in \mathcal{C}. \end{cases}$$

   Compute $f(x^k, c^k)$.

   If $f(x^k, c^k) < f^*$, then let $(x^k, c^k) \rightarrow (x^*, c^*)$, and $f(x^k, c^k) \rightarrow f^*$.

   If $|f(x^k, c^k) - \widehat{f}_k| > \epsilon$ or $k < maxiter$ Go to the Update Model Step. Otherwise algorithm stopping criteria has been reached.

2. **Update Model** Let $W_{k+1} = W_k \cup (x^k, c^k)$.

The basic optimization algorithm presented in Section 2.2.4.1 assumes the accuracy of a model built up with a limited number of function evaluations, and might get trapped in a local minimum.

2.2.4.2 Optimization with kriging: EGO algorithm

Efficient Global Optimization (EGO) algorithm, proposed by Jones et al. [44], allows optimization with surrogate models while compromising between the prediction of the model and the associated estimated error. At each iteration, we aim at maximizing the expected improvement. This optimization problem helps for the decision of improving the approximation (by evaluating a point where prediction error is high) or exploit the

approximation of the model (by evaluating a point where the approximation model is low). This optimization problem is solved with NOMAD.
Improvement in a new point is defined as:

$$I(x,c) = \begin{cases} (f^* - f(x,c)) & \text{if } f(x,c) < f^*, \\ 0 & \text{else.} \end{cases} \tag{2.13}$$

where f^* is the current best objective function value.

Expected improvement is then:

$$\mathbb{E}(I) = \begin{cases} (f^* - \hat{f})\Phi\frac{f^*-\hat{f}}{\hat{\sigma}} + \hat{\sigma}\Phi'\frac{f^*-\hat{f}}{\hat{\sigma}} & \text{if } \hat{\sigma} > 0, \\ 0 & \text{if } \hat{\sigma} = 0. \end{cases} \tag{2.14}$$

2.2.4.3 Numerical results

In this section we optimize the well configuration in a 1D reservoir model (see 1.4.1) for a maximum number of 3 wells, and at least one well of each type (producer, injector) using the NPV function defined in Section 1.2.2.2. We use the two algorithms defined in Section 2.2.4.1 and Section 2.2.4.2. We use two kriging models, built from different size of sliced LHS, composed of 250 and 500 points.

The MINLP formulation of the well placement is the following (we minimize -NPV which is equivalent to maximize NPV)

$$\begin{cases} \min -NPV(x,y) \\ x \in [x_L, x_U] \subseteq \mathbb{R}^{n+m} \\ c \in \{0,1\}^{n+m} \\ g(x,c) \leq 0 \end{cases} \tag{2.15}$$

with the quantitative variable $x \in \mathbb{R}^{n+m}$, and the binary variable, considered as a qualitative variable, $c \in \{0,1\}^{n+m}$ representing respectively the locations of the wells ($n = 2$ is the fixed number of producer wells and $m = 2$ the fixed number of injector wells) and their status (active or not), x_L, x_U are the bound vectors for $[1, 1000]^{n+m}$. The resulting optimization problem deals with 4 quantitative variables and 4 qualitative variables. Additional constraints are introduced in order to force the number of injector and producer wells to be larger than 1, and the maximal number of wells to be 3.

2.2.4.3.a Optimization test: basic algorithm

In Fig. 2.25 and Fig. 2.26 we represent the different locations of the wells during the optimization process for respectively kriging model built from 250 and 500 points. Red points represent the kriging points and blue squares the well location. We can observe that the optimization process did not explore the entire domain, but only some subsets of the variable space.

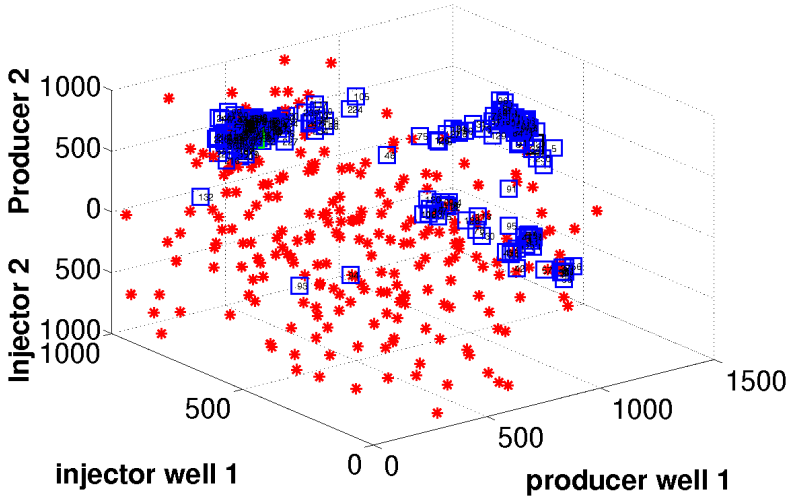

Figure 2.25: Well locations, represented by blue squares, during the basic algorithm optimization process associated with kriging model built from 250 points. The red stars represent the location of kriging points.

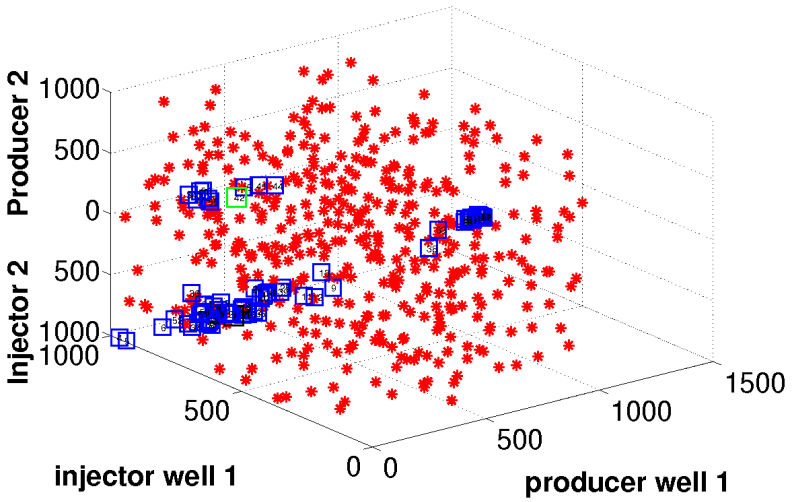

Figure 2.26: Well locations, represented by blue squares, during the basic algorithm optimization process associated with kriging model built from 500 points. The red stars represent the location of kriging points.

In FIG. 2.27 and FIG. 2.28 we represent the NPV function value of the wells configuration in red, and the corresponding approximated value in blue circles, during the basic algorithm optimization process associated respectively with kriging model built from 250 and 500 points. In green is represented the best NPV value and its corresponding approximated value.

The best NPV value is equal to 6475 with the model built with 250 points, and is equal to 6325 for with other model. Those values are found after, respectively, 160 and

42 algorithm iterations, hence a total of reservoir simulator evaluations of respectively 410 and 542. The first conclusion of these first runs is that using a model built from a larger number of kriging points might not help to obtain better NPV function, and it only increases the number of preliminary simulations to build the kriging model.

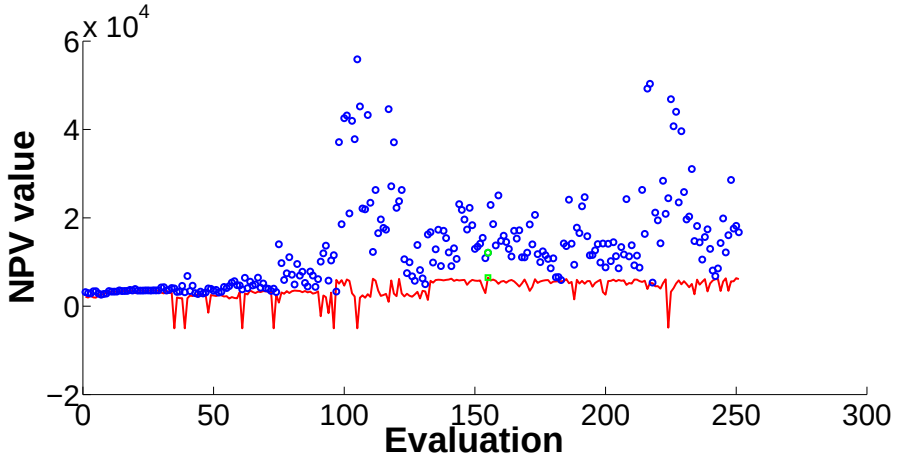

Figure 2.27: Estimated values (in blue circles) and associated NPV (in red) values during the basic algorithm optimization using model built from 250 points. The best NPV value (in green) of 6475 is obtained after 150 iterations.

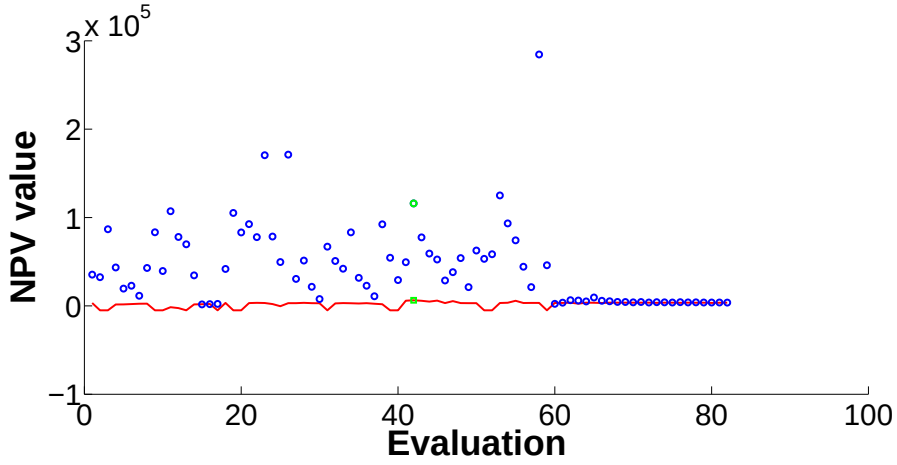

Figure 2.28: Estimated values (in blue circles) and associated NPV (in red) values during the optimization using model built from 500 points. The best NPV value (in green) of 6325 is obtained after 42 iterations.

2.2.4.3.b Optimization test: EGO algorithm

In FIG. 2.29 and FIG. 2.30 we represent the different locations of the wells during the optimization, and the corresponding approximated values of the model.

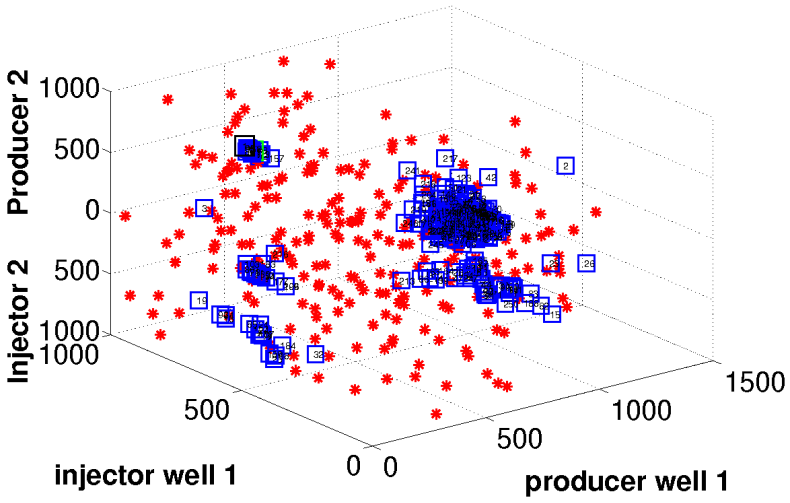

Figure 2.29: Well locations, represented by blue squares, during the EGO optimization process associated with kriging model built from 250 points. The red stars represent the location of kriging points.

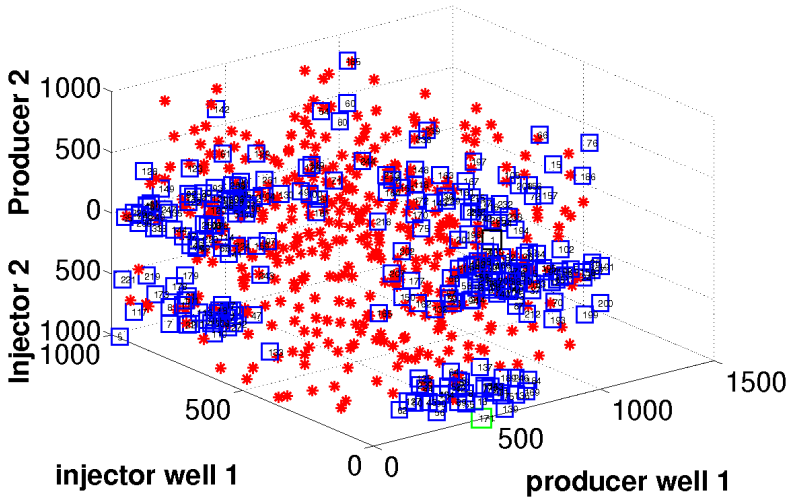

Figure 2.30: Well locations, represented by blue squares, during the basic algorithm optimization process associated with kriging model built from 500 points. The red stars represent the location of kriging points.

In FIG. 2.31 and FIG. 2.32 we represented the NPV function value of the wells configuration during the EGO optimization process associated respectively with kriging model

built from 250 and 500 points. Best NPV values, 6775 for the optimization using kriging model built with 250 points, and 6675 using kriging model built with 500 points, were found after respectively 25 and 170 algorithm iterations, hence a total of reservoir simulator evaluation of respectively 275 and 670.

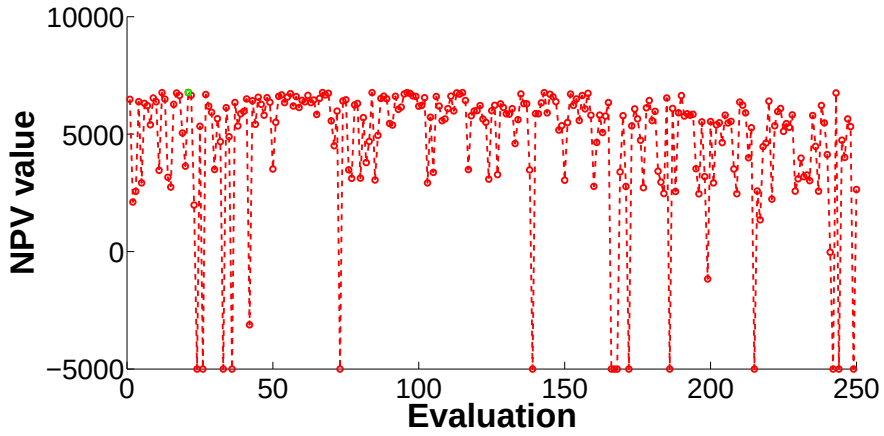

Figure 2.31: NPV value during the optimization with EGO algorithm using model built from 250 points. The best NPV value (in green) of 6775 is obtained after 25 iterations.

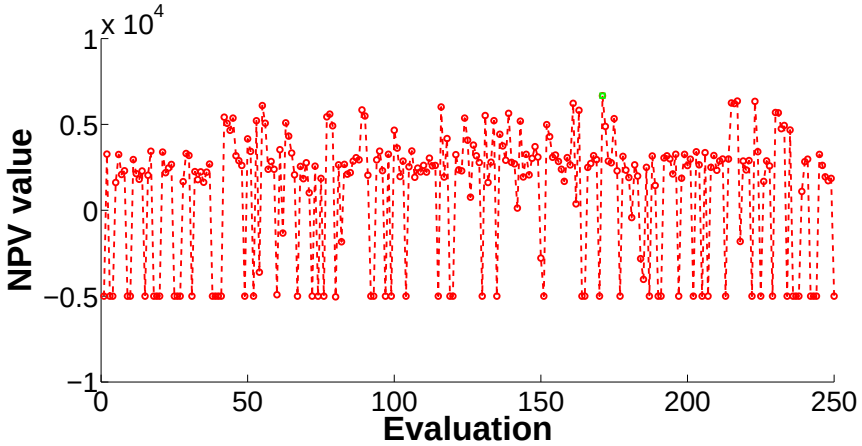

Figure 2.32: NPV value during the optimization with EGO algorithm using model built from 500 points. The best NPV value (in green) of 6625 is obtained after 170 iterations.

2.2.4.3.c Conclusion on numerical results

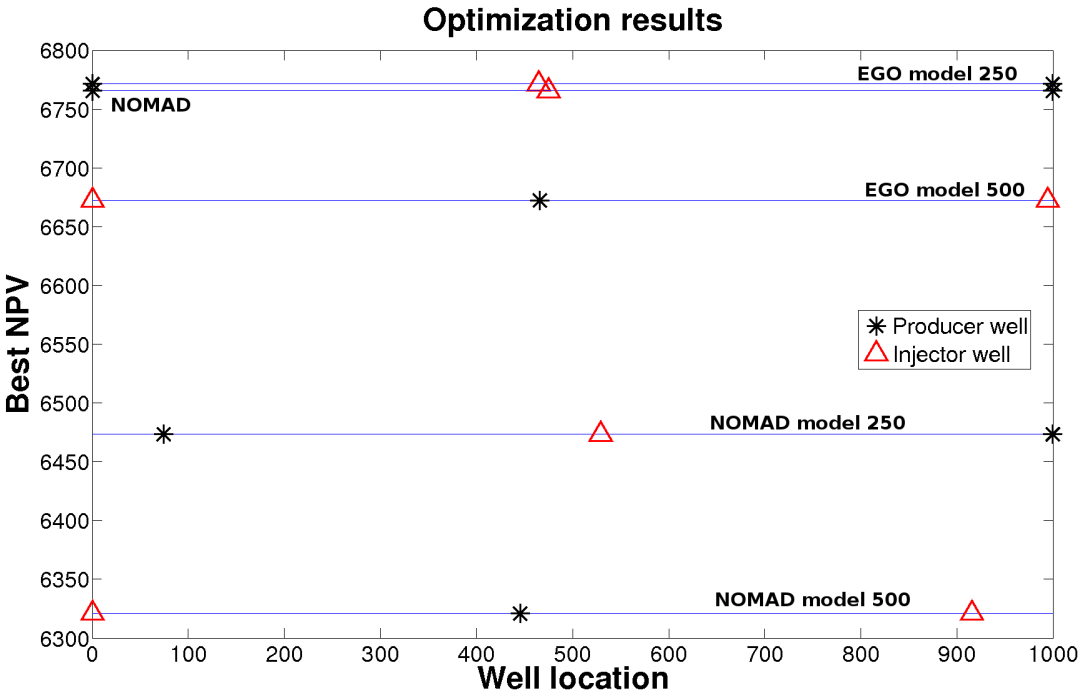

Figure 2.33: NPV solutions obtained by each method of optimization

In Fig. 2.33 we can compare the results of optimization for the two algorithms with a 3 wells configuration solution obtained with NOMAD alone (a constraint to have three wells was also included). The two lower results are obtained by using the basic optimization algorithm presented in Section 2.2.4.1 associated with kriging models built from two different initial sliced LHS. We see that the variable space is less explored during optimization using the model built from 250 points than during optimization with the model built from 500 points. Also the model built with 250 point requires 1 day of NPV function evaluation, in comparison the 500 points model requires 1-2 day of NPV function evaluation.The different well configurations between the two results suggest that a local minimum was reached with optimization using the model built from 500. We can also see in Fig. 2.33 that similar results are obtained by solving the optimization problem with NOMAD, and by using EGO algorithm associated with the kriging model built from 250 points: NPV values and wells configurations are similar. The number of simulations necessary to reach the best NPV value with EGO algorithm was of 275 (250 kriging points + 25 iterations, equivalent to 1 day of execution time), while more than 300 simulations were required with NOMAD alone. However the optimization with EGO did not end before the maximum authorized number of iterations was reached. The use of EGO algorithm and a 500 points sample is less effective, the best NPV obtained is lower than the best value obtained with NOMAD alone. Configurations of active wells were not the same, with a total of 670 simulations (2 days of execution time) to reach the best NPV value. We remark that the results of the EGO algorithm with the 250 points kriging sample also gives a better NPV function value, in fewer iterations, than the 3

wells solution obtained with NOMAD on the same problem in Section 2.1.3.1 represented in FIG. 2.1, with similar configurations (2 producer wells, and 1 injector well between them). From these results, we can say that using a large sample to build a kriging model for optimization purposes is not more effective, and is obviously more costly. These optimization run show that the EGO algorithm is well adapted for the well placement problem. We also observe that this method is more efficient than NOMAD for this 1D case.

2.2.5 Conclusion

In this section we presented a kriging model adapted for quantitative and qualitative variables. We tested this method to approximate the NPV function on a 1D reservoir. First results show that we do not need a larger number of data points to build a model that can guide the optimization. Then we presented two algorithms for solving optimization problem using kriging, a basic approach and the EGO algorithm that maximizes the expected improvement at each iteration. We tested and compared the two algorithms by maximizing the NPV function associated to the 1D reservoir model (see Section 1.4.1. We obtained similar results using NOMAD or the EGO algorithm. The results with EGO are slightly better in terms of objective function value and in terms of total number of reservoir simulations. However the difference is not significant enough and the results may depend on the starting point or the initial kriging sample. The method was tested on a 1D reservoir case, with a limited number of wells described by only one quantitative variable and one qualitative variable. For larger dimension cases, the size of the problem, and the number of simulations to reach convergence, may increase exponentially. Hence the method should be limited to vertical wells, to reduce the number of quantitative variables to one per well. However, this is encouraging to continue the exploitation of the quantitative and qualitative kriging method, and test the optimization with EGO and smaller initial kriging samples. We could also consider to compute in parallel the evaluation of these samples. To limit the number of simulations due to the size of the problem and applied efficiently the method on larger dimension reservoir case, the EGO algorithm should be apply as a search step in NOMAD and and thus allow to benefit from the advantage of the two methods.

2.3 A trust region method for Black-Box MINLP

In this section we propose to extend the popular Derivative Free Trust Region (DFTR) method to mixed variables optimization problems, the new approach is a joint work with A. R. Conn (IBM Watson). We test the method on the 1D reservoir model, and the 3D SPE10 reservoir model case.

2.3.1 Black-box MINLP formulation

The problem formulation we address is the following

$$\begin{cases} \min\limits_{x,y} \ f(x,y) \\ x \in [x_L, x_U] \subseteq \mathbb{R}^p \\ y \in \{0,1\}^q \end{cases} \tag{2.16}$$

where $x \in \mathbb{R}^p$, $y \in \{0,1\}^q$ are decision variables,

$f : \mathbb{R}^n \to \mathbb{R}$ a Black-box function of x, y (with $n = p + q$) which is supposed to be at least twice differentiable in every argument,

$x_L,\ x_U \in \mathbb{R}^p$ are bounds for the continuous variables.

Formulation (2.16) arises in simulation optimization settings, where an expensive simulator takes as input a certain number of continuous, x, and integer, y, values and then produces a function value, $f(x, y)$, as output.

Like all such methods, see for example Conn et al. [25, 26], Langouët et al. [47] our Derivative Free Trust Region method aims at iteratively improve an existing current feasible point (x^k, y^k) at iteration k of the Black-Box problem by optimizing an approximate model $\mathcal{M}_k$ of (2.16) valid in a set $\mathcal{R}_k \in \mathbb{R}^n$, which is called the trust region. We assume that $\mathcal{M}_k$ is itself a Mathematical Program (MP), and we denote by $f_{\mathcal{M}}$ its objective function. The new candidate point $(\tilde{x}, \tilde{y})$ is the solution of (MP) restricted to $\mathcal{R}_k$, so it is not necessarily locally optimal with respect to (2.16). To quantify the quality of this solution, we use the quality criterion defined in the next section.

2.3.2 Quality of the new candidate point

Define

$$\rho = \frac{f(\tilde{x},\tilde{y}) - f(x^k,y^k)}{f_{\mathcal{M}}(\tilde{x},\tilde{y}) - f_{\mathcal{M}}(x^k,y^k)} \tag{2.17}$$

as a quality measure for solution $(\tilde{x}, \tilde{y})$.

Computing ρ requires, (i) an (expensive) evaluation of $f(x, y)$, i.e., a call to the simulator, and, (ii) $\mathcal{M}_k$ will be defined in such a way that the denominator in (2.17) is always non positive.

Solution $(\tilde{x}, \tilde{y})$ is considered as "good", "ok" or "bad" according to ρ being in certain intervals:

- if ρ is close or larger than 1, i.e., $\rho \in [1 - \epsilon_{good}, +\infty[$: the model is either locally very close to the original problem, or that the decrease in objective function value of the original problem (2.16) is higher than the corresponding decrease in $f_{\mathcal{M}}$. So $(\tilde{x}, \tilde{y})$ is "good"

- if $\rho \in [\epsilon_{ok}, \epsilon_{good}[$ then $(\tilde{x}, \tilde{y})$ is just "ok". The improvement in the model engenders a moderate improvement in the original problem.

- if $\rho \in [-\infty, \epsilon_{ok}[$ then $(\tilde{x}, \tilde{y})$ is "bad" since an improvement in the model yields a tiny improvement or a worsening in the original problem.

2.3.3 The trust region

The trust region $\mathcal{R}_k$ is centered at current iterate (x^k, y^k) and is defined in different way for x and y: a ball centered in x^k defined with a l_∞-norm: a box $[\underline{x}, \overline{x}] = x^k - \Delta_k, x^k + \Delta_k$, and a "*local branching*" constraint that defines the neighborhood of y^k, see Fischetti and Lodi [34], by limiting the number of flips in binary variable values to K

$$\sum_{j:y_j^k=0} y_j + \sum_{j:y_j^k=1} (1 - y_j) \leq K. \tag{2.18}$$

The left hand side represents the Hamming distance with respect to y^k, the most natural measure in a discrete space.

The size of the trust region is modified accordingly to the concordance between the improvement predicted by model $\mathcal{M}_k$ and the actual improvement measured by ρ:

- if the new candidate point is good, we accept it and the trust region R_k is enlarged and centered at this new point,
- if the new candidate point is just acceptable ("ok"), we accept it, and center the trust region at the new point but keep its size,
- if the new candidate point is "bad", we keep the trust region centered at the previous point (x^k, y^k) and decrease its size.

In all cases, the model is updated with the new point (x^k, y^k) and the associated simulated objective function $f(x^k, y^k)$.

2.3.4 The model

The aim of the model $\mathcal{M}$ is to provide an approximation of the original problem (2.16), which should be accurate within the trust region $\mathcal{R}$. We assume that the objective function $f_\mathcal{M}$ of $\mathcal{M}$ approximates f. Since f is a Black-Box function and we have no access to derivatives, we can only use previous points $\{x_l,\ y_l,\ f_l\ |l < k\}$ (where $f_l = f(x_l, y_l)\ \forall l < k$) to construct an approximation $f_\mathcal{M}$.

The model of the objective function f is a linear or quadratic function interpolating the known values of f (the model can also be obtained by regression) $f_\mathcal{M}(x, y) = \alpha + \beta^T x + \gamma^T y + \frac{1}{2}(x, y)^T \Gamma (x, y)$,

with $\alpha \in \mathbb{R}, \beta \in \mathbb{R}^p, \gamma \in R^q, \Gamma$ a symmetric matrix of $\mathbb{R}^{n\times n}$.

In order to construct this model, we solve the auxiliary problem which computes the (n+1)(n+2)/2 coefficients in $(\alpha, \beta, \gamma, \Gamma)$ by imposing the interpolation of simulated values at previous iterations $\{(x_l, y_l, f_l = f(x_l, y_l)|l < k\}$

$f_\mathcal{M}(x_l, y_l) = f_l$.

If $k = (n + 1)(n + 2)/2$, this problem is equivalent to solve a linear system.

If $k < (n + 1)(n + 2)/2$, we solve the following problem

$$\begin{aligned} &\min_{\alpha,\beta,\gamma,\Gamma} ||\Gamma||_F^2 \\ &f_\mathcal{M}(x_l, y_l) = f_l \forall l \end{aligned} \tag{2.19}$$

by minimizing the Frobenius norm of the Hessian of $f_\mathcal{M} : ||\Gamma||_F^2 = \sum_{1\leq i,j\leq n} \Gamma_{i,j}^2$ to solve the under-determination.

2.3.5 Proposed algorithm: Derivative Free trust Region method for MINLPs

The proposed Derivative Free Trust Region algorithm for MINLP is the following:

- Initialization

  Choose the initial setup points, x and y, $\delta_{compare}$ and $\Delta_{max} > 0$, x_0 is the best initial point w.r.t. f. Choose Δ and related constants: $\epsilon_{good} = 0.9$ and $\epsilon_{ok} = 0.01$ Choose the stopping criteria (e.g., $\delta_{min} = 10e^{-5}$)

- Main Iteration

STEP 0. Build surrogate model $\mathcal{M}_0$.

STEP 1. Solve the trust region sub-problem for y fixed at the current value y_0. The bounds for the trust region sub-problem are determined by the intersection of the trust region (which is an l_1 sphere, i.e., simple bounds) and the simple bounds of the problem. There are 3 outcomes,

a) No significant movement but Δ is less than δ_{min},

b) No significant movement but Δ is greater than δ_{min},

c) Significant movement and trust region sub-problem has converged.

STEP 2. Compute $\rho = \frac{f(\tilde{x},\tilde{y})-f(x^k,y^k)}{f_{\mathcal{M}_k}(\tilde{x},\tilde{y})-f_{\mathcal{M}_k}(x^k,y^k)}$.

STEP 3. Trust region management.

a) if $\rho \geq \epsilon_{ok}$ and $\rho < \epsilon_{good}$: Not a very good ρ but true function did not increase. New x becomes the incumbent.

b) if not a good ρ: If the function improved, the new x becomes the best point.

c) If $\rho \geq \epsilon_{ok}$: Good or very good ρ: true function and model decreased. New x becomes the best point.

d) if $|\rho| \geq \epsilon_{ok}$: Reduce Δ by a factor of two (essentially zero movement). Update the surrogate model.

STEP 4. Solve the approximate problem for x and y within the trust region $\mathcal{R}$ and subject to local branching constraint.

If a new x becomes the best point after computing ρ we attempt to also improve y.

1) Build a new surrogate model.

2) Solve the new trust region sub-problem for both x and the relaxed y.

There are three possible outcomes,

a) y does not change in which case there is no need to continue since we have already solved that problem, or will in the next iteration,

b) y changes and the step is successful initially in which case we set Δ back to its original value because this is like a new problem,

c) y changes but the step is unsuccessful initially in which case we set Δ back to its prior value.

STEP 5. Stopping criteria.

If (x, y) barely changes and Δ is small enough, STOP.

If $\Delta < \delta_{min}$ STOP.

2.3.6 Basis for convergence of the algorithm

Salient features of the proposed algorithm are contained in STEP 0., STEP 2., STEP 3. for fixed y.

STEP 0. aims to do at least as well as (a fixed fraction of) the Cauchy point (the Cauchy point is the minimum of the model in the "steepest descent" direction within the trust region). This is a crucial point to guide the convergence of the trust region algorithms, see Conn et al. [26]. This model-based sub-problem can be solved by a classical derivative based nonlinear optimization method (here, we use IPOPT, an interior point solver, see Wächter [77])

STEP 3. appropriate management of the size of the trust region as described in Section 2.3.3.

Check consistency between $f_{\mathcal{M}}$ and f Section 2.3.2.

In STEP 4. solving the sub-problem in x and y starting from a point at least as good as the Cauchy point for fixed y should guarantee to obtain a better value for the model than the value for this Cauchy point, but maybe not for f. This model-based MIQP is solved by SCIP, see Achterberg [7].

Iterating and adding new simulations for (x, y) should improve the model and help to converge to a (local) solution. The next section describes how we force the method to continue the exploration after reaching a local optimum.

2.3.7 No-good cuts or how to avoid redundant space exploration

Suppose that the following situation occurs

- the model can be trusted in the current region,
- (x', y') is the current best point,
- but the current point cannot be improved.

In this case, the typical trust region method for continuous nonlinear problems would stop and provides the local minimum (x', y'). In the MINLP case, we propose a different approach. In particular, we want to force the algorithm to explore a different part of the y-space because there could be interesting unexplored parts. Thus, the local

constraint (2.18) is relaxed and the so-called no-good cut, i.e., a reverted local branching cut, is adjoined

$$\sum_{j:y_j^*=0} y_j + \sum_{j:y_j^*=1} (1 - y_j) \geq K^* + 1 \tag{2.20}$$

where $(y^*; K^*)$ represent the center and the radius of the region we consider as sufficiently explored. Note that, we will have a bunch of constraints (2.20), one for each sufficiently explored region.

As soon as a new current point is found, we can also restore back the local branching constraint (2.18).

Ideally we would like to use no-good cuts to mimic the pruning process of the branch-and-bound, i.e:

- we find the incumbent solution and we cannot improve it with the current trust region,
- the best solution we can end with the current trust region is worse than the incumbent,
- infeasibility: no feasible solution can be found with the current trust region (it can happen because of the presence of one or more no-good cuts).

Note that in all the cases above we suppose we trust the model. We end when the intersection of the no-good cuts is empty or a termination criterion is reached.

2.3.8 Application

2.3.8.1 Application on the 1D case

We first evaluate the method on a one dimensional case (see Section 1.4.1), on the heterogeneous case.

The MINLP formulation of the well placement is the following (we minimize -NPV which is equivalent to maximize NPV)

$$\begin{cases} \min -NPV(x, y) \\ x \in [x_L, x_U] \subseteq \mathbb{R}^{n+m} \\ y \in \{0, 1\}^{n+m} \\ g(x, y) \leq 0 \end{cases} \tag{2.21}$$

with the design variables $x \in \mathbb{R}^{n+m}$, and $y \in \{0, 1\}^{n+m}$ representing respectively the locations of the wells ($n = 2$ is the fixed number of producer wells and $m = 2$ the fixed number of injector wells) and their status (active or not), x_L, x_U are the bound vectors for $[1, 1000]^{n+m}$.

The considered NPV function is defined in Section 1.3 and its parameters are defined in Tab. 2.7.

Table 2.7: NPV function, parameters values.

Parameters	**Values**
C^{PO}	100 \$/m
C^{PW}	5 \$/m
C^{IW}	4 \$/m
C^{D}	1/6 \$

The resulting optimization problem deals with 4 continuous variables and 4 binary variables. Additional constraints are introduced in order to force the distance between wells and the minimal number of injector and producer well to be larger than 1.

We compare the results obtained with the Derivative Free Trust Region algorithm adapted for mixed variables with the results obtained with NOMAD using Mesh Adaptive Direct Search methods (see Section 2.1.2).

The results on the 1D well placement problem are presented in Fig. 2.34. The convergence obtained with the Derivative Free Trust Region method is faster than the one obtained with NOMAD with a similar objective function value after convergence. These preliminary results are encouraging despite the simplicity of the example. Even on such a simple example, the efficiency of the method has been illustrated.

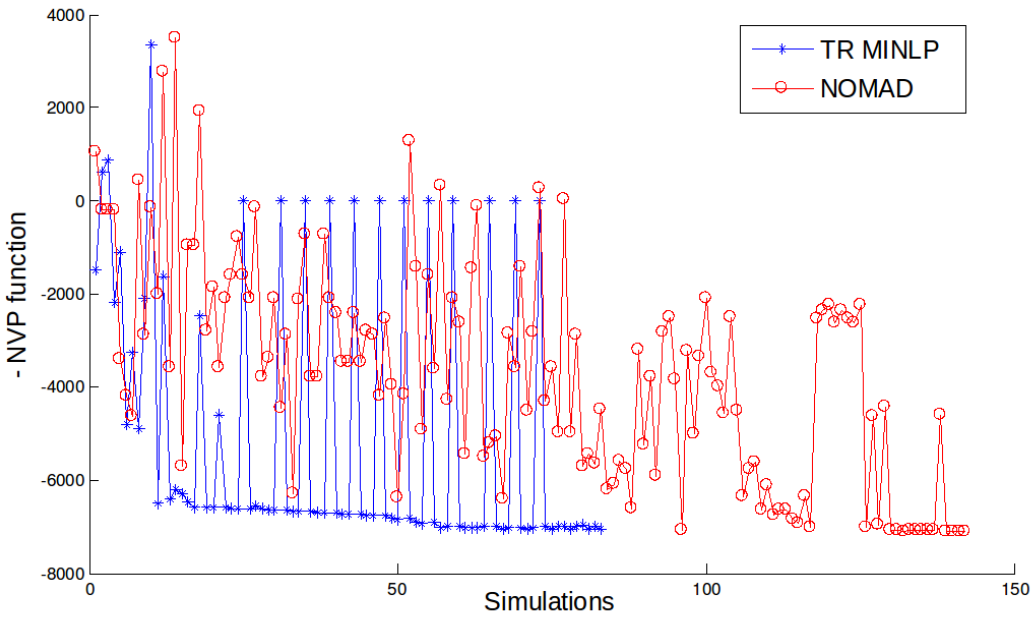

Figure 2.34: Comparison of the results obtained with the proposed method (Derivative Free Trust Region method for MINLP problems) and NOMAD (direct search method) for the simple well placement problem.

2.3.8.2 Application on the 3D case SPE10

The potential of the method has been evaluated by applying the method on the three dimensional case SPE10 (see Section 1.4.4). The reservoir simulator used is a MATLAB implementation (Lie [51], Toolbox [73]). We consider a production of 10 years with one injector well and between 1 to 4 producer wells. The objective of the study is to optimize the number of producer wells and their location and the location of the injector well to maximize the Net Present Value of the field in Section 1.2, or similarly to minimize its opposite value.

2.3.8.2.a Problem formulation

$$\begin{cases} \min -NPV(x,y) \\ x \in [x_L, x_U] \subseteq \mathbb{R}^p \\ y \in \{0,1\}^q \\ g(x,y) \leq 0 \end{cases} \tag{2.22}$$

With $x = \{x^1_{P1}, x^2_{P1}, x^1_{P2}, x^2_{P2}, x^1_{P3}, x^2_{P3}, x^1_{P4}, x^2_{P4}, x^1_I, x^2_I\}$, (x^1_{Pi}, x^2_{Pi}) are the two coordinates necessary to define vertical producer well $i = 1 \ldots 4$, and (x^1_I, x^2_I) are the coordinates of the injector well.

$y = \{y_1, y_2, y_3, y_4\}$ are binary values associated with the status of the producer well (0: inactive well, 1: active well), x_L and x_U are the bound vectors for x^1 and x^2. The resulting optimization problem deals with 10 continuous variables and 4 binary variables.

Additional constraints, $g(x, y)$ are introduced in order to force the distance between wells to be larger than a minimal distance (10 nonlinear constraints on x (and constrain the number of producer wells to be larger than 0 and smaller than 5 (2 linear constraints).

The objective function is the NPV function (see Section 1.2), and the parameters used are indicated in Tab. 2.8.

Table 2.8: NPV function, parameter values

Parameters	**Values**
C^{PO}	50 \$/bbl
C^{PW}	5 \$/bbl
C^{IW}	5 \$/bbl
C^F	45 \$/ft
C^D	1e6 \$
R	1

The results obtained with the Derivative Free Trust Region method are compared with results of optimization with NOMAD using MADS methods (see Section 2.1).

Different options of TR method are compared with NOMAD in Fig. 2.35:

- basic TR: without local branching (2.18) and without no good cut (2.20) for binary variables,
- with no good cut,
- with no good cut and local branching.

We observe for the 3 different options of the proposed TR algorithm a faster convergence than NOMAD towards a better solution. As expected, despite none configuration with 4 producer wells were evaluated, the no good cut constraint allows a better exploration than the basic algorithm. The combination of this option with the local branching gave the best result as shown in Fig. 2.35. The associated solution is a configuration with one producer well as shown in Fig. 2.36.

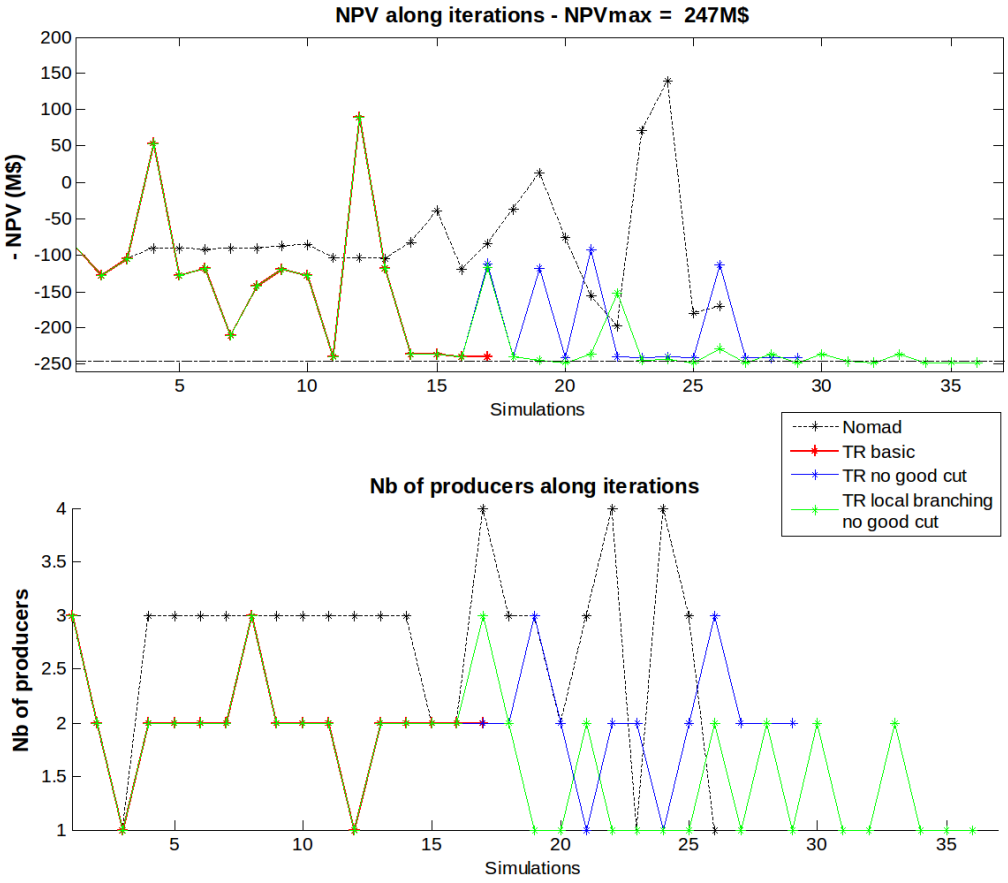

Figure 2.35: Evolution of objective function (top) and number of producer wells (bottom) along optimization processes. Comparison of NOMAD and our Derivative Free Trust Region method (run with 3 different tunings).

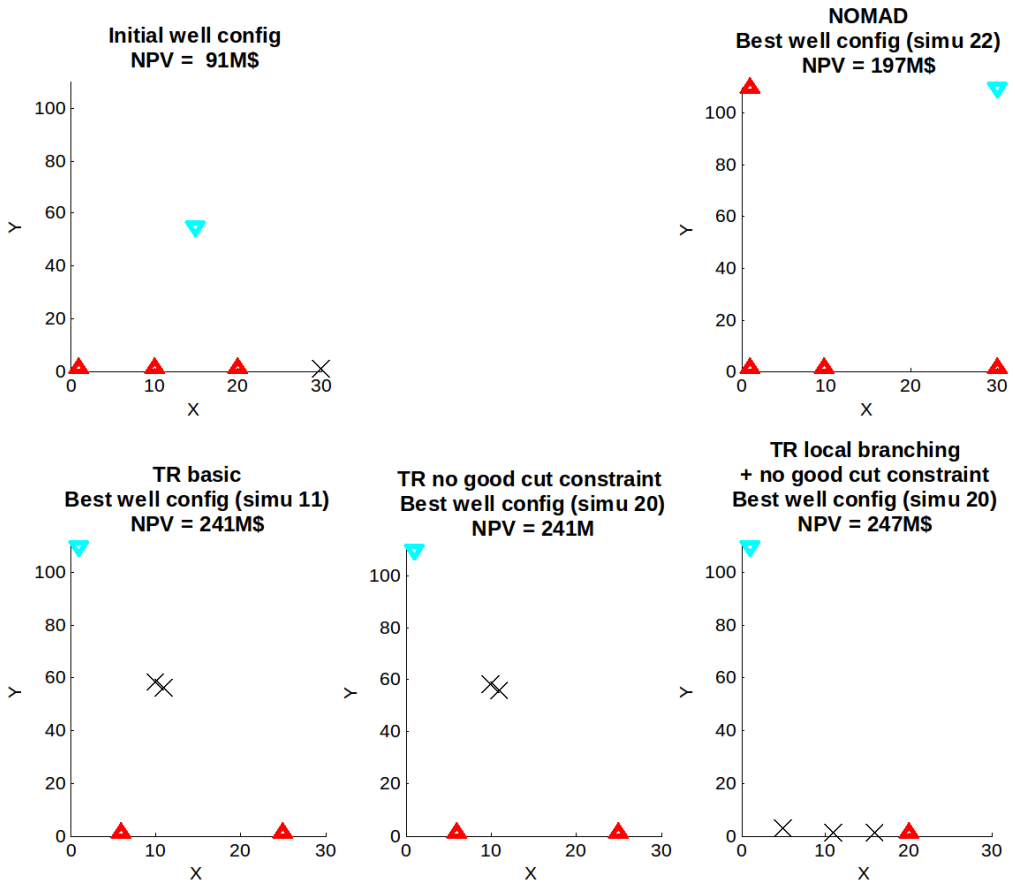

Figure 2.36: Comparison of solutions obtained with NOMAD and with our Derivative Free Trust Region method (run with 3 different tunings). The initial configuration is displayed in the top left hand corner.

2.3.8.3 Conclusion

We proposed and tested an approach to extend the Derivative Free Trust Region methods to mixed variables optimization problems. The method gave promising results on two test cases of reservoir models, however the method still need improvement to explore more the subset of binary variables. The project is still in study within the context of the collaboration with A. R. Conn (IBM Watson), a prototype with other options and the handling of constraints is in progress.

2.4 Discussion on the 1D case numerical results

In this section we discuss the results of each method presented in the chapter that were applied on the 1D case with the heterogeneous pattern (see Section 1.4.1). Tab. 2.9 indicates the maximum number of wells, the maximum number of well of each type, the best NPV function value, the number of wells in the solution configuration, and the number of evaluations for the NOMAD solver, the EGO algorithm used with the kriging model build from the 250 points model, and the Derivative Free Trust Region method.
We can see that the best NPV function is obtained with the NOMAD solver, however the Derivative Free Trust Region method obtains a slightly less good NPV function with a hundred less number of evaluations. The NPV results for the EGO algorithm with the kriging model are lower, but the maximum number of active wells for this method was of 3, and we can see that wells configuration solution for the two other methods have 4 active wells. This difference of number active wells can explain the lower NPV function for EGO with the kriging model. The method could be tested for a higher number of wells, but it would require additive simulations for the Kriging model construction sample. However, the EGO algorithm showed its effectiveness, since only 25 simulations of EGO (and 250 evaluations for the kriging sample) were necessary. Thus the number of evaluations for each combination of number of wells could be reduced for the kriging model construction sample.
In this 1D case, we can see that similar NPV results were obtained for the NOMAD solver and the Derivative Free Trust Region, but in terms of number of evaluations the Derivative Free Trust Region method is more effective. The EGO algorithm with the kriging method obtained lower NPV results, but with a different number maximum of active wells, and the method should be compared in same conditions of active wells.

Table 2.9: Optimization with NOMAD, EGO algorithm with the kriging model (250 points), and the Derivative Free Trust Region on the 1D case with the heterogeneous permeability (see Section 1.4.1).

	NOMAD	EGO & Kriging	DFTR
Maximum injector wells	2	2	2
Maximum producer wells	2	2	2
Maximum total of wells	4	3	4
Best NPV function value	7097 $	6775 $	7059 $
Number of active wells	4	3	4
Number of evaluations	300	275	200

2.5 Conclusions

In this chapter, we described three derivative free optimization methods, adapted to the well placement problem. We first described GPS and MADS methods, implemented in the NOMAD solver, and used it to optimize the number and location of vertical or rectilinear wells on several reservoir model cases, the simplified 1D case (see Section 1.4.1), and two 3D cases, the PUNQ-S3 (see Section 1.4.3), and the SPE10 test case (see Section 1.4.4) and obtain satisfying results. However the method requires numerous evaluations of the objective function which is costly in CPU time, thus it could be interesting to couple the algorithm with surrogates. In the second section we presented a kriging method for qualitative and quantitative variables function, and evaluate its accuracy by estimate the NPV function on the 1D test case. The estimations of the NPV functions present some irregularities on the boundary, and a large number of evaluations is required to have a precise estimation. However the model can be used to guide the optimization. We presented two optimization algorithms using the model, a basic approach and the EGO algorithm, and tested it on the NPV function. The basic algorithm did not give bad optimization results, but using EGO algorithm allowed to reach a better NPV function value than the best value found by the NOMAD solver on the same case. Similar results were obtained by using NOMAD or the EGO algorithm, this is encouraging to continue the exploitation of the quantitative and qualitative kriging method, as to combine methods, such as using the Expected Improvement as a search step in NOMAD, as it is done in Talgorn et al. [72]. In the last section we proposed a Derivative Free Trust Region adapted for Black-Box MINLP problems. We tested the method on the 1D case, and the 3D SPE10 case, and obtained promising results. However the method still need improvement to better explore the subset of integer variables, and to handle the constraints. The project is still in study within the context of the collaboration with A. Conn (IBM Watson).

Chapter 3

Well placement and trajectory optimization

In the well placement problem, optimizing the number of branches can be very costly in evaluations because of the variables used to describe the location and the status of the branches. In specific cases, the number of authorized simulations can be limited, and applying existing Black-Box MINLP method is then ineffective. Thus it is necessary to propose methods adapted to the well placement problem that account for this aspect. In this chapter, we propose a new methodology suited to the optimization of well placement and geometry. Starting from a vertical well configuration obtained from a first optimization, we aim at finding the potential branches that can be drilled from existing wells in order to increase oil production or the NPV function. This method is composed of two consecutive optimization problems. We first solve a Black-Box MINLP problem for vertical well placement. Then in an intermediate step, we analyze the obtained solution and define the inputs of a second convex MINLP optimization problem for well branching. In the second optimization problem, in order to reduce the number of Black-Box simulations, we use instead an estimation of the oil production. The method is investigated with three different models, then we apply the methodology to a 2D case, as presented in Lizon et al. [54, 55].

3.1 Optimization of vertical wells: type and location

The first step of the methodology consists in obtaining an acceptable and feasible solution for the well location problem. In the first optimization, we limit the number of optimized variables by considering a limited number of wells and only vertical trajectories. The optimized objective function is the Net Present Value (NPV) previously defined in Section 1.2.2.
Maximizing this objective function with respect to the number of active wells (producers and injectors), their location and their length, is a Black-Box MINLP optimization

problem coupled with a computationally expensive simulator, namely the reservoir fluid flow simulator. Since we use the NPV function, which is generally non-smooth and non-convex, and without available gradient information, minimization requires derivative-free global optimization methods ([17, 31, 42, 63]) as explained in the previous chapter in Section 2.1. This step provides a first optimized configuration with producer and injector vertical wells. The outputs of the reservoir fluid flow simulation obtained with this configuration are analyzed to determine the areas where oil is not produced.

3.2 Optimization of well trajectory: branching problem

In this second step we start from a vertical well configuration, given as the solution of the first optimization problem described in Section 3.1. The aim is to improve oil production, or NPV function, by adding branches to the existing vertical wells. To do so, we introduce a branch sub-problem: given a well configuration without branches, we aim at finding the potential branches that can be drilled from each well. We focus on branches from producer wells only. The vertical injection wells obtained in the first optimization solution are fixed and kept unchanged in this phase.

In order to obtain an efficient methodology and reduce computational time, we do not use reservoir fluid flow simulations. We consider instead an estimation of the oil production based on the quantity of oil remaining in the cells belonging to all areas connected to the producer wells by a branch. The objective function is computed from this estimation. An additional constraint is introduced to account for the branch construction cost, which in turn depends on the branch length, to avoid well configuration with an unacceptable drilling cost.

3.2.1 Input data: Analysis of optimized vertical well configuration

In practice, reservoir engineers make the decision to add wells or branches based on the analysis of the outputs of the fluid flow simulator. The commonly considered criteria are physical attributes such as oil saturation or oil thickness.
We combine this pragmatic approach with optimization in order to better determine the trajectory of wells with branches. We analyze the outputs of the simulator and attribute a score to distinguish regions of the field: a high score means that an area has good drilling potential. For simplicity, we focused on a single attribute called $H_u\phi S_o$ which is the product of the utility thickness (H_u), the porosity (ϕ), and the oil saturation (S_o). $H_u\phi S_o$ quantifies the volume of oil still not produced.
An example of a two-dimensional map, obtained with $H_u\phi S_o$ is displayed in Fig. 3.1. This map is obtained from the outputs of a reservoir fluid flow simulation after producing

the field for 4 years with one vertical well located at the center of the reservoir presented in Section 1.4.2. In this map, we observe that there is still unproduced oil.

The input data of the second MINLP optimization problem are the location of groups of cells where oil is not produced, and the quantities of the chosen attribute within the group of cells.

3.2.1.1 Physical attributes

The outputs of the fluid flow simulator are analyzed. They give an estimation of the quantity of attribute at a chosen time of the simulation on the whole domain: a value is assigned to each cell of the grid, as displayed in FIG. 3.2, and FIG. 3.1.

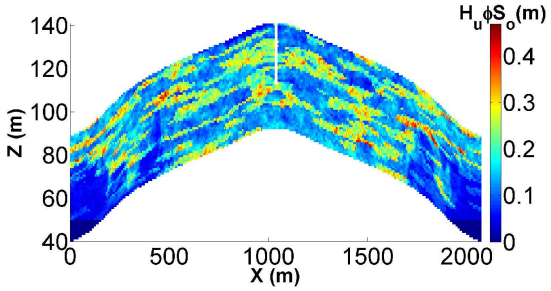

Figure 3.1: $H_u\phi S_o$ map associated with a single vertical well configuration on a 2D case (see Section 1.4.2).

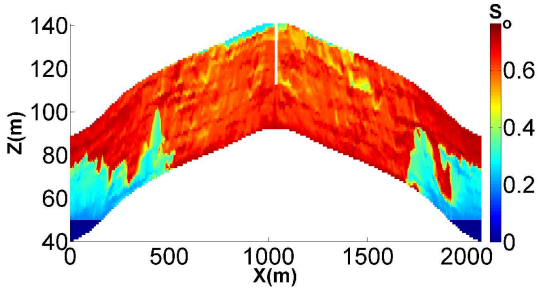

Figure 3.2: Oil saturation for initial vertical well configuration after 4 years of production.

3.2.1.2 Analysis

The first idea could be to use each cell with a value above a chosen amount, but we can see that some cells have a high $H_u\phi S_o$ but are spatially close to cells with low $H_u\phi S_o$ values. Clearly, such cells are not as interesting as zones with more homogeneous and relatively high $H_u\Phi S_o$ values. Moreover, taking all cells with high enough values would lead to a high complexity optimization problem. In order to reduce the number of variables, and to avoid isolated cells, scores are computed for groups of cells. Such groups of cells are called *areas* in the remaining of this chapter.

Another property that could be taken into account in the definition of high potential areas is the connectivity between cells. Based on connectivity consideration, we can introduce here *forbidden zones*, i.e., parts of the reservoir that cannot physically be drilled or that we do not want to drill because, in general, they are not well connected.

From this analysis, we determine the input data to be used within the following step of the proposed methodology, which is the branching optimization problem.

3.3 Mathematical models

In this section, we present several models used for the conception of branches. Different views for adding branches to wells were possible, and we tested several convex MINLP problems, i.e., MINLP problems with convex objective function and constraint functions. In the first model, (see Section 3.3.2), branches produced only oil located in cells closed to their extremities. In models 2 and 3, oil is produced along the whole branches. In Model 2 (see Section 3.3.3), branches are defined relatively to a well, but the model is valid in only two dimensions, whereas in Model 3 (see Section 3.3.4) their coordinates are defined in the whole domain. The definition and handling of budget and drilling constraints are also different in the three models.

3.3.1 Common notation

Notation commonly used in the proposed mathematical models show sets in Greek letters, input data and parameters in upper case, and variables in lower case. As variables and bounds depend on the models, we present in this section only sets and parameters:
Sets:

- Φ: set of wells in the initial solution.
- Π: set of areas of the reservoir representing the cells with a high score at the end of the he simulation with the initial well configuration.
- $\Delta = \{1, \ldots, D\}$: considered space dimensions ($D \leq 3$).

Indexes:

- i: index for area $i \in \Pi$.
- n: index for well $n \in \Phi$.
- d: index for domain dimensions $d \in \Delta$.

Parameters:

- D: number of space dimensions considered ($D \leq 3$).
- X_{id}: d^{th} coordinate of the center of oil area ($i \in \Pi, d \in \Delta$).
- O_i: volume of oil contained by oil area i ($i \in \Pi$).
- $[0, E_d]$: oil reservoir domain for dimension d ($d \in \Delta$). The domain is a rectangular cuboid of top corner (0), and E are the coordinates of the diagonally opposed corner.
- $\underline{L}$, $\overline{L}$, and H: minimum and maximum lengths of a branch, and maximum length of all branches built. The third parameter is necessary to limit the construction cost based on the target budget.

- R: maximum distance between oil area and branch end, i.e., maximum distance of a point in Π with respect to the branch reference point.
- V_{nd}: coordinates of the top of well n for dimension d ($d \in \Delta$, $n \in \Phi$).
- W_n: height of well n ($n \in \Phi$), we assume that there are vertical wells only.
- C: profit for a unit of oil produced.
- F: drilling cost per branch length.

In the next sections, some additional parameters and variables will be introduced for the different model formulations.

3.3.2 Model 1, a 3D model

In this section, we introduce the concept of *cluster* for modeling purposes. A cluster is a set of oil areas that are "close enough" to be reached by a single branch (close enough has to be understood in terms of connectivity). The mathematical model below can be used to aggregate areas into clusters and to determine how to reach a cluster with a branch from a well, with an acceptable construction cost. We define the end of a branch as a *cluster reference point*, i.e., a point that is close enough to each area of the cluster (the branch obtained can be expected to produce a significant amount of the oil in the cluster).

3.3.2.1 Sets

- $\Gamma = \{1, \ldots, K\}$: set of clusters.

Indexes:

- j: index for cluster $j \in \Gamma$.

Parameters:

- B: maximum number of branches per well.
- K: maximum number of clusters.

3.3.2.2 Variables

Continuous:

- z_{jd}: d-th coordinate of the reference point for cluster j ($j \in \Gamma$, $d \in \Delta$).
- w_{jn}: height of the junction point on well n connecting cluster j through a branch ($j \in \Gamma$, $n \in \Phi$).

- s_{jn}: square of the distance between the reference point of cluster j and the junction point of well n ($j \in \Gamma$, $n \in \Phi$).

Binary:

- q_{jn}: activation of connection between cluster j and well n ($j \in \Gamma$, $n \in \Phi$): [1 if active, 0 otherwise].

- x_{ij}: inclusion of oil area i in cluster j ($i \in \Pi$, $j \in \Gamma$): [1 if included, 0 otherwise].

- y_j: activation of cluster j ($j \in \Gamma$), an inactive cluster is not linked to any well: [1 if active, 0 otherwise].

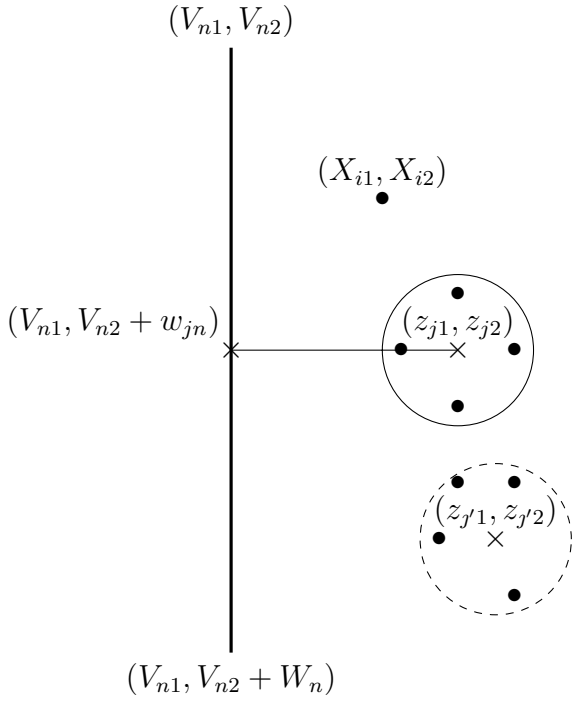

Figure 3.3: Branch and cluster modeling (in two dimensions) for Model 1.

In FIG. 3.3 we represent the cluster modeling. Well n represented by a vertical segment with its coordinates. Oil areas are represented by black dots, and are indexed in Π. One oil area, which coordinates are indicated on the figure, is not part of a cluster. The other oil areas are part of two clusters, Cluster j represented as a circle and Cluster j' represented as a dashed circle. Only Cluster j is linked by a branch to the well, which means only areas in Cluster j are produced in the represented configuration.

Some additional parameters and variables will be introduced later to formulate the optimization as a convex MINLP problem.

3.3.2.3 Simple Bounds

Simple bounds on continuous variables are defined to ensure that the well configurations obtained are specified within the physical domain considered.

$$0 \leq w_{jn} \leq W_n \qquad \forall j \in \Gamma, n \in \Phi, \tag{3.1}$$

$$0 \leq z_{jd} \leq E_d \qquad \forall j \in \Gamma, d \in \{1, 2\}, \tag{3.2}$$

$$0 \leq s_{jn} \leq ||E_d||_2^2 \qquad \forall j \in \Gamma, n \in \Phi. \tag{3.3}$$

3.3.2.4 Constraints

- Inequalities (3.4) represent a budget constraint. As anticipated at the beginning of this section, this constraint is introduced to limit the cost associated with the branches. The bilinear terms $q_{jn}s_{jn}$ will be reformulated at the end of the section to obtain a linear constraint plus auxiliary variables and linear constraints.

$$\sum_{j \in \Gamma, n \in \Phi} q_{jn} s_{jn} \leq H^2, \tag{3.4}$$

- Equations (3.5) represent bound inequalities on the branch length:

$$q_{jn}\underline{L}^2 \leq q_{jn}s_{jn} \leq q_{jn}\overline{L}^2 \quad \forall j \in \Gamma, \forall n \in \Phi, \tag{3.5}$$

- Each area i is assigned at most to one cluster:

$$\sum_{j \in \Gamma} x_{ij} \leq 1 \quad \forall i \in \Pi, \tag{3.6}$$

- We ensure that area i is assigned to cluster j only if this cluster is active:

$$x_i \leq y_j \quad \forall i \in \Pi, j \in \Gamma, \tag{3.7}$$

- Cluster j is not active if no area is assigned to it:

$$y_j \leq \sum_{i \in \Pi} x_{ij} \quad j \in \Gamma, \tag{3.8}$$

- Cluster j can be connected to well n only if this cluster is active:

$$q_{jn} \leq y_j \quad \forall n \in \Phi, j \in \Gamma, \tag{3.9}$$

- The maximum number of branches for each well is B:

$$\sum_{j \in \Gamma} q_{jn} \leq B \quad \forall n \in \Phi, \tag{3.10}$$

- If cluster j is active, it can be connected to at most one well:

$$\sum_{n \in \Phi} q_{jn} \leq y_j \quad \forall j \in \Gamma, \tag{3.11}$$

- As mentioned earlier, the oil-bearing areas have to be close enough to the reference point of a cluster to be assigned to it. In such a case, we can expect that a significant amount of the oil contained in these areas is produced through the branch corresponding to the cluster. Equations (3.12) ensure that, if area i is assigned to cluster j, the distance between the cluster reference point and the area cannot be larger than R:

$$\sum_{d\in\{1,2\}} (X_{id} - z_{jd})^2 \leq R^2 + M(1 - x_{ij}) \quad \forall j \in \Gamma, i \in \Pi, \tag{3.12}$$

- Here we represent an angle constraint between the well mainbore and a branch (if cluster j is connected to well n). Note that since all wells are vertical, forcing the height of the well junction point to be greater than the height of the cluster reference point is the same as setting a 90^o upper bound on the branch angle:

$$z_{j2} \leq V_{n2} + w_{jn} + (1 - q_{jn})E_2 \quad \forall j \in \Gamma, n \in \Phi, \tag{3.13}$$

- Equation (3.14) is related to the square distance between cluster reference point j and its junction point on well n:

$$s_{jn} \geq (V_{n1} - z_{j1})^2 + ((V_{n2} + w_{jn}) - z_{j2})^2 \quad \forall j \in \Gamma, n \in \Phi, \tag{3.14}$$

- Equations (3.15) establish symmetry breaking constraints, i.e., they impose an order for the lateral coordinates of the clusters. These symmetry breaking constraints do not remove any feasible solution (they keep only one configuration when several are equivalent). This limits the feasible space of the solution to be explored by the optimization algorithm considered:

$$z_{j1} \leq z_{k1} \quad \forall j < k \in \Gamma. \tag{3.15}$$

3.3.2.5 Objective function

The proposed model aims at maximizing the total amount of oil produced. Formally, this quantity can be estimated through the quadratic objective function:

$$\max_{x,q} \sum_{i\in\Pi, j\in\Gamma, n\in\Phi} x_{ij} q_{jn} O_i,$$

which accounts for the oil associated with areas belonging to a connected cluster.

3.3.2.6 Reformulations

We can simplify a number of constraints with the help of auxiliary variables: r_{jn} and u_{jni} replace the bilinear terms $q_{jn}s_{jn}$ ($j \in \Gamma$, $n \in \Phi$) and $x_{ij}q_{jn}$, ($j \in \Gamma$, $n \in \Phi$, $i \in \Pi$) respectively. The variable r_{jn} is the square of the distance between a connected cluster j and well n (if cluster j is not connected, $r_{jn} = 0$). The binary variable u_{jni} indicates

that area i belongs to cluster j which is connected to well n (if oil area i is not produced, $u_{jni} = 0$).

The relation between the original and the new variables is given by the following linear constraints:

$$r_{jn} \leq s_{jn} + M(1 - q_{jn}) \qquad \forall j \in \Gamma, n \in \Phi, \tag{3.16}$$
$$r_{jn} \geq s_{jn} - M(1 - q_{jn}) \qquad \forall j \in \Gamma, n \in \Phi, \tag{3.17}$$
$$r_{jn} \geq q_{jn}\overline{L}^2 \qquad \forall j \in \Gamma, n \in \Phi, \tag{3.18}$$
$$r_{jn} \leq q_{jn}\underline{L}^2 \qquad \forall j \in \Gamma, n \in \Phi, \tag{3.19}$$
$$u_{jni} \leq q_{jn} \qquad \forall n \in \Phi, j \in \Gamma, i \in \Pi, \tag{3.20}$$
$$u_{jni} \leq x_{ij} \qquad \forall n \in \Phi, j \in \Gamma, i \in \Pi, \tag{3.21}$$
$$u_{jni} \geq q_{jn} + x_{ij} - 1 \qquad \forall n \in \Phi, j \in \Gamma, i \in \Pi. \tag{3.22}$$

where M is defined as the square of the maximum distance between two points in the reservoir.

According to Equations (3.16), (3.17), (3.18) and (3.19), if cluster j is connected to well n (i.e., $q_{jn} = 1$), then the corresponding branch length cannot be larger than $\overline{L}$ and smaller than $\underline{L}$. If cluster j is not connected to well n (i.e., $q_{jn} = 0$), Equations (3.18) and (3.19) imply $r_{jn} = 0$, and (3.16) and (3.17) are satisfied.

In terms of the new variables, the budget constraint (3.4) can be rewritten as the linear constraint:

$$\sum_{j \in \Gamma, n \in \Phi} r_{jn} \leq H^2.$$

In addition, the objective function is now linear:

$$\max_{u} \sum_{i \in \Pi, j \in \Gamma, n \in \Phi} u_{jni} O_i.$$

Constraints (3.5) are also rewritten as linear constraints, which are constraints (3.18) and (3.19).

As a consequence, the optimization problem formulated in terms of the new variables is a convex MINLP.

It is important to note that in the above formulation we impose no restriction on the crossing or proximity of branches. These constraints are very important in practice but would complicate the problem considerably, and transform it into a non convex MINLP, which is much harder to solve.

This first model generates a lot of constraints, and cannot be applied in a case of large number of inputs. In the next section, two other models are presented, and the concept of cluster does not appear. Oil areas are not distributed into clusters before forming branches.

3.3.3 Model 2, a 2D model

To obtain an efficient model, we implement a second model restricted to $D = 2$ dimensions. In this model we add branches to wells considering that a branch produces an oil area, if this area is located in a given radius.

3.3.3.1 Sets

- $\Gamma = \{1, \ldots, K\}$: set of branches.

Indexes:

- j: index for branch $j \in \Gamma$.

Parameters:

- K: maximum number of branches allowed for each well.
- C: profit for a unit of oil produced.
- F: drilling cost per length of branch.
- P: distance between two branches of a same well.
- Q: minimal lateral distance between two branches of a different wells.
- h: identify if the total length of branches is part of the objective function or considered as a constraint [1 if objective function, 0 if constraint].

3.3.3.2 Variables

Continuous:

- w_{jn}: height of the j-th branch's junction of well n ($n \in \Phi$, $j \in \Gamma_n$).
- z_{jnd}: d-th coordinate of the ending point of the j-th branch of well n with respect to the junction given by (V_{nd}, w_{jn}) ($n \in \Phi$, $j \in \Gamma_n$, $d \in \Delta \setminus \{2\}$).
- g_{jnd}: absolute value of d-th coordinate of the ending point for j-th branch of well n with respect to the junction given by (V_{nd}, w_{jn}) ($n \in \Phi$, $j \in \Gamma_n$, $d \in \Delta$).
- s_{jn}: square of the distance between the starting and the ending point for j-th branch of well n ($j \in \Gamma_n$, $n \in \Phi$).
- $\delta^+_{jkn1}, \delta^-_{jkn1}, \delta^+_{jkn2}, \delta^-_{jkn2}$: slack variables ($n \in \Phi$, $j, k \in \Gamma_n, j > k$).

Binary:

- x_{ijn}: identify if i is reached (and produced) by j-th branch of well n [1 if produced, 0 otherwise] ($i \in \Pi$, $n \in \Phi$, $j \in \Gamma_n$).
- y_{jnd}: sign of variable z_{jnd} j [1 if positive, 0 otherwise] ($n \in \Phi$, $j \in \Gamma_n, d \in \Delta, d \neq 2$).
- u_{jkn}: identify if z_{jn1} and z_{kn1} have the same sign [1 if opposite, 0 otherwise] ($n \in \Phi$, $j, k \in \Gamma_n, j > k$).
- u_{jkn1}: identify the sign of $(g_{jn1} - g_{kn1})$ [1 if positive, 0 otherwise] ($n \in \Phi$, $j, k \in \Gamma_n, j > k$).

- u_{jkn2}: identify the sign of $((w_{jn} + z_{jn2}) - (w_{kn} + z_{kn2}))$ [1 if positive, 0 otherwise] $(n \in \Phi,\, j, k \in \Gamma_n, j > k)$.

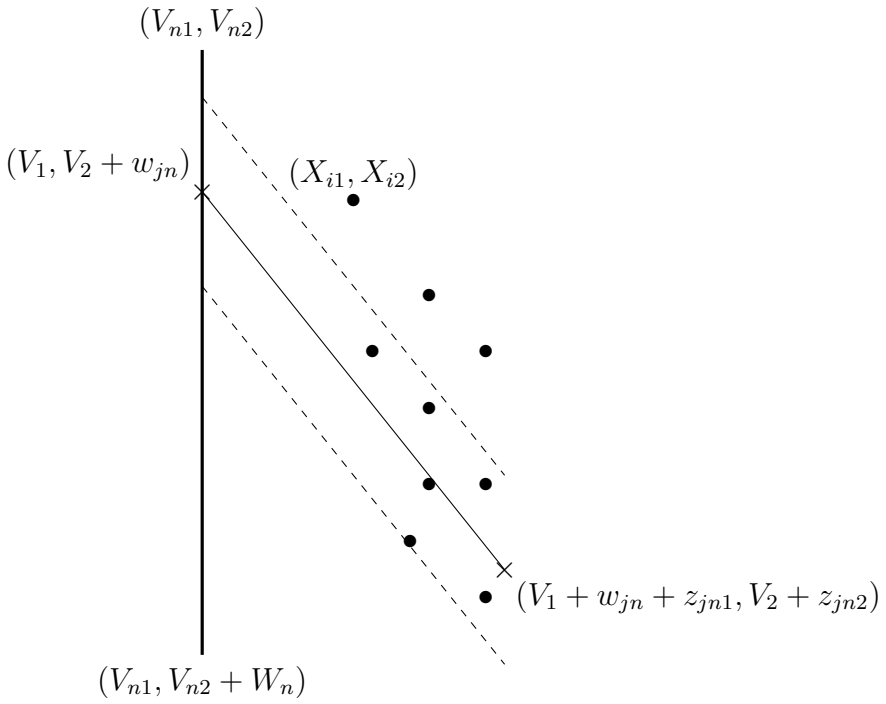

Figure 3.4: Branch modeling for Model 2.

In FIG. 3.4, we represent the branch modeling for the second proposed model. We can see that branch j on well n is represented by 3 coordinates, its height junction w_{jn}, z_{jn1} and z_{jn2}, its relative coordinates starting from the junction on well n. In this figure, branch j on well n is on the positive side of x axes, which means that the variable $y_{jn1} = 1$.

3.3.3.3 Simple Bounds

Simple bounds on continuous variables can be defined as follows:

$$0 \leq w_{jn} \leq W_n \qquad \forall j \in \Gamma_n, n \in \Phi, \tag{3.23}$$
$$-\overline{L} \leq z_{jnd} \leq \overline{L} \qquad \forall n \in \Phi, j \in \Gamma_n, d \in \Delta, d \neq 2, \tag{3.24}$$
$$0 \leq z_{jn2} \leq \overline{L} \qquad \forall n \in \Phi, j \in \Gamma_n, d \in \Delta, \tag{3.25}$$
$$0 \leq s_{jn} \leq 2\overline{L}^2 \qquad \forall j \in \Gamma_n, n \in \Phi. \tag{3.26}$$

3.3.3.4 Constraints

The set of constraints is presented in the following subsection. Variables g_{jnd}, u_{jnd}, u_{jknd} appear in equations above: their definition will be presented later in the section. We first present the equations directly linked to the branches and oil production:

- Equation (3.27) represents a budget constraint. As in the previous model, this constraint is introduced to limit the cost associated with the branches. If parameter $h = 0$, the constraint is active. The total sum of the length of each branch must be less than a fixed budget.

$$\sum_{n\in\Phi}\sum_{j\in\Gamma_n} s_{jn} \leq H^2(1-h) + Mh \;\forall n \in \phi, j \in \Gamma_n, \tag{3.27}$$

- By means of Equations (3.28) each area i is assigned at most to one branch.

$$\sum_{n\in\Phi:i\in\Pi_n}\sum_{j\in\Gamma_n} x_{ijn} \leq 1 \;\forall i \in \bigcup_{n\in\Phi}\Pi_n, \tag{3.28}$$

- Equation (3.29) forces the variable s_{jn} to have the same value of the length of branch j of well n.

$$s_{jn} = \sum_{d\in\Delta} z_{jnd}^2 \;\forall j \in \Gamma_n, n \in \Phi, \tag{3.29}$$

- Equations (3.30) ensure that the branch length satisfy bounds.

$$s_{jn} \leq L^2 \;\forall j \in \Gamma_n, n \in \Phi, \tag{3.30}$$

- Equations (3.31) establish symmetry breaking constraints, i.e., they impose an order for the coordinates $d = 2$ of the clusters.

$$w_{jn} \geq w_{kn} \;\forall n \in \Phi, j > k \in \Gamma_n, \tag{3.31}$$

- Equation (3.32) forces variable z_{jnd} to have an absolute value between L and R, and forces variable y_{jn} to have the value of 1 if coordinate z_{jn1} is positive, 0 if negative.

$$-L(1-y_{jn1}) + Ry_{jn1} \leq z_{jn1} \leq Ly_{jn1} - R(1-y_{jn1}) \;\forall n \in \Phi, j \in \Gamma_n, \tag{3.32}$$

- Equations (3.33) impose a distance between junctions of branches j and k if they are on the same side of well n.

$$w_{jn} - P + Mu_{jkn} \geq w_{kn} \;\forall n \in \Phi, j > k \in \Gamma_n, \tag{3.33}$$

- Constraints (3.34) check that oil area i is at a distance less than $R + z_{jn1}$ of well n and is active if z_{jn1} has a positive sign (that is when y_{jn1} is equal to 1). If not, x_{ijn} is equal to 0. $\forall n \in \Phi, j \in \Gamma_n, i \in \Pi_n$:

$$-M(1-y_{jn1}) + V_{n1}x_{ijn} \leq X_{i1}x_{ijn} \leq V_{n1}x_{ijn} + z_{jn1} + \frac{s_{jn}^{\frac{1}{2}}R}{z_{jn2}} + M(1-y_{jn1}) \tag{3.34}$$

- Constraints (3.35) check that oil area i is at a distance less than $R + z_{jn1}$ of well n and is active if z_{jn1} has a negative sign (that is when y_{jn1} is equal to 0). If not, x_{ijn} is equal to 0.

$$-My_{jn1} - \frac{s_{jn}^{\frac{1}{2}} R}{z_{jn2}} + V_{n1}x_{ijn} + z_{jn1} \leq X_{i1}x_{ijn} \leq V_{n1}x_{ijn} + My_{jn1} \; \forall n \in \Phi, j \in \Gamma_n, i \in \Pi_n, \tag{3.35}$$

- Equations (3.36) and (3.37) check that oil area i is in branch j of well n given the second coordinate. We project coordinate X_{i1} on branch j, and (see FIG. 3.5) compute coordinate $\tilde{X}_{i2}$ using the intercept theorem in elementary geometry: if a given line passes through the two sides of the given triangle and parallel to the third side, then it cuts the sides proportionally. The intercept theorem allows us to compute the value a and to subtract it the value b (which can be easily computed) to obtain the distance c. Equations (3.36) and (3.37) ensure that oil area i can be produced only if the difference $X_{i2} - \tilde{X}_{i2}$ (distance c in FIG. 3.5) is less than R.

  $\forall n \in \Phi, j \in \Gamma_n, i \in \Pi_n :$

$$z_{jn2}(X_{i1} - V_{n1}) + (w_{jn} + V_{n2})g_{jn1} - s_{jn}^{\frac{1}{2}} R - M(1 - x_{ijn})g_{jn1} \leq X_{i2}g_{jn1}, \tag{3.36}$$

$$X_{i2}g_{jn1} \leq z_{jn2}(X_{i1} - V_{n1}) + (w_{jn} + V_{n2})g_{jn1} + s_{jn}^{\frac{1}{2}} R + M(1 - x_{ijn})g_{jn1}. \tag{3.37}$$

- Equations (3.39), (3.40) (3.41) (3.42), check that the distance between branches of a same well is at least P. We use again the intercept theorem to compute the distance between projection of branches extremities. Variables u_{jknd} allow to differentiate the cases (see FIG. 3.6 to FIG. 3.9) that can appear when two branches are on a same side of a well. Equations (3.38) eliminate the case where two branches cross themselves (see case in FIG. 3.7). These constraints are active only if branches are on the same side of the well.

  $\forall n \in \Phi, j > k \in \Gamma_n :$

$$u_{jkn1} - u_{jkn2} \geq u_{jkn}, \tag{3.38}$$

$$w_{jn} + z_{jn2} - w_{kn} - z_{kn2} - \frac{z_{jn2}}{z_{jn1}}(z_{jn1} - g_{kn1}) \geq P - M(u_{jkn} + u_{jkn1} + u_{jkn2}), \tag{3.39}$$

$$(w_{jn} - w_{kn} - z_{jn2}) + g_{jn1}\frac{z_{kn2}}{g_{kn1}} \geq P - M(u_{jkn} + (1 - u_{jkn1})), \tag{3.40}$$

$$g_{kn1} - \frac{g_{jn1}}{z_{jn2}}(z_{kn2} + w_{kn} - w_{jn}) \geq P - M(u_{jkn} + u_{jkn2}), \tag{3.41}$$

$$\frac{g_{kn1}}{z_{kn2}}(z_{jn2} + w_{jn} - w_{kn}) - g_{jn1} \geq P - M(u_{jkn} + (1 - u_{jkn2})) \tag{3.42}$$

- Equation (3.43) keeps a minimal lateral distance of Q between branches of different wells.

$$(V_{m1} + z_{jm1}) - (V_{n1} + z_{kn1}) \geq Q \; \forall m = n + 1 \in \Phi, j \in \Gamma_m, k \in \Gamma_n. \tag{3.43}$$

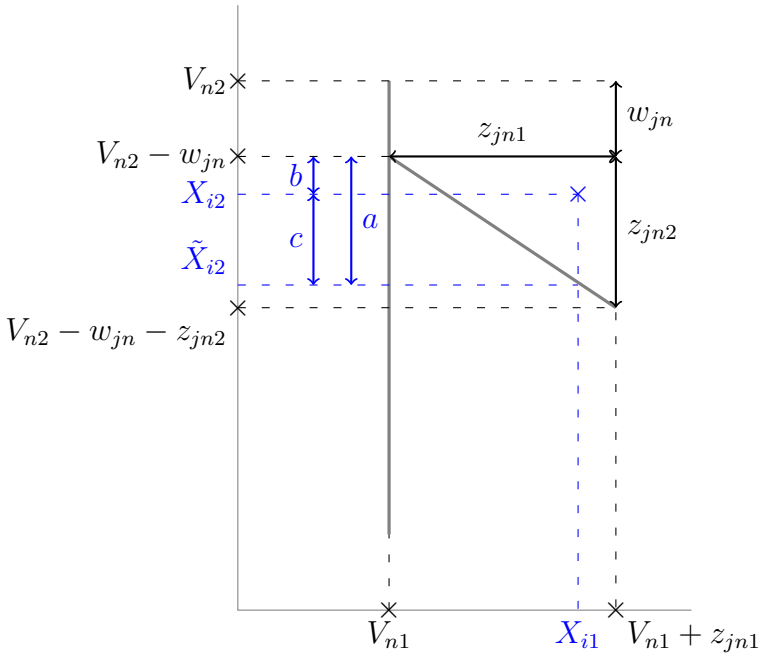

Figure 3.5: Production constraints.

Equations (3.44) and (3.45) gave to variable g_{jn1} the absolute value of z_{jn1}

$$-M(1-y_{jn1}) + z_{jn1} \leq g_{jn1} \leq z_{jn1} + M(1-y_{jn1}) \qquad \forall n \in \Phi, j \in \Gamma_n, \tag{3.44}$$

$$-My_{jn1} - z_{jn1} \leq g_{jn1} \leq -z_{jn1} + My_{jn1} \qquad \forall n \in \Phi, j \in \Gamma_n. \tag{3.45}$$

Equations (3.46),(3.47),(3.48) and (3.49) force variable u_{jkn} to be 1 if branches j and k are on the same side of well n.

$$y_{jn1} - y_{kn1} + u_{jkn} \geq 0 \qquad \forall n \in \Phi, j,k \in \Gamma_n, j > k, \tag{3.46}$$

$$y_{kn1} - y_{jn1} + u_{jkn} \geq 0 \qquad \forall n \in \Phi, j,k \in \Gamma_n, j > k, \tag{3.47}$$

$$y_{jn1} + y_{kn1} - u_{jkn} \geq 0 \qquad \forall n \in \Phi, j,k \in \Gamma_n, j < k, \tag{3.48}$$

$$y_{jn1} + y_{kn1} + u_{jkn} \leq 2 \qquad \forall n \in \Phi, j,k \in \Gamma_n, j > k. \tag{3.49}$$

In the next equations a new variable δ_{jknd}, $n \in \Phi j > k \in \Gamma$, $d \in \Delta$ provides a bound that allows to compute binary variables u_{jknd}.

$$g_{jn1} - g_{kn1} + \delta^+_{jkn1} - \delta^-_{jkn1} = 0 \qquad \forall n \in \Phi, j > k \in \Gamma_n, \tag{3.50}$$

$$0 \leq \delta^+_{jkn1} \leq Mu_{jkn1} \qquad \forall n \in \Phi, j > k \in \Gamma_n, \tag{3.51}$$

$$0 \leq \delta^-_{jkn1} \leq M(1-u_{jkn1}) \qquad \forall n \in \Phi, j > k \in \Gamma_n, \tag{3.52}$$

$$(w_{jn} + z_{jn2}) - (w_{jn} + z_{kn2}) + \delta^+_{jkn2} - \delta^-_{jkn2} = 0 \qquad \forall n \in \Phi, j > k \in \Gamma_n, \tag{3.53}$$

$$0 \leq \delta^+_{jkn2} \leq Mu_{jkn2} \qquad \forall n \in \Phi, j > k \in \Gamma_n, \tag{3.54}$$

$$0 \leq \delta^-_{jkn2} \leq M(1-u_{jkn2}) \qquad \forall n \in \Phi, j > k \in \Gamma_n. \tag{3.55}$$

Equations (3.50) (3.51) (3.52) force variable u_{jkn1} to be 1 if the absolute value of z_{jn1} is greater than z_{jkn1}. Equations (3.53) (3.54) (3.55) force variable u_{jkn2} to be 1 if the absolute value of z_{jn2} is greater than z_{jkn2}.

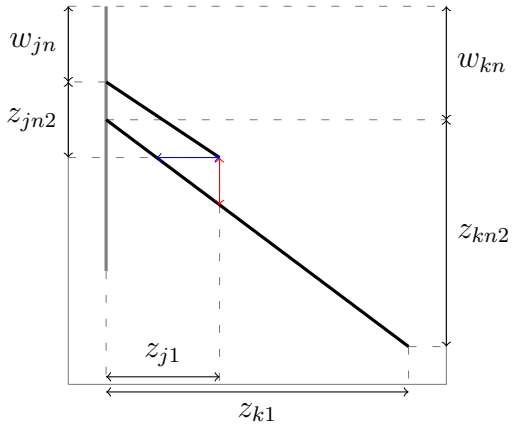

Figure 3.6: $u_{jk1} = 0$, $u_{jk2} = 0$.

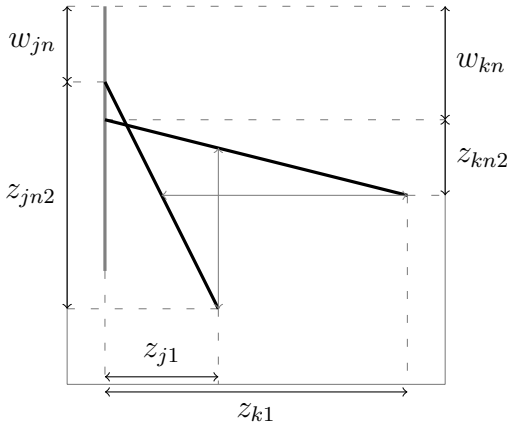

Figure 3.7: $u_{jk1} = 0$, $u_{jk2} = 1$.

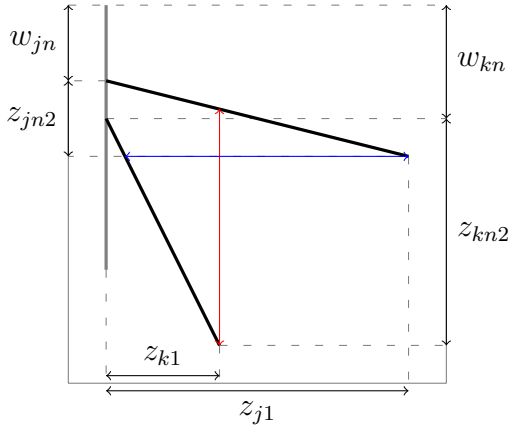

Figure 3.8: $u_{jk1} = 1$, $u_{jk2} = 0$.

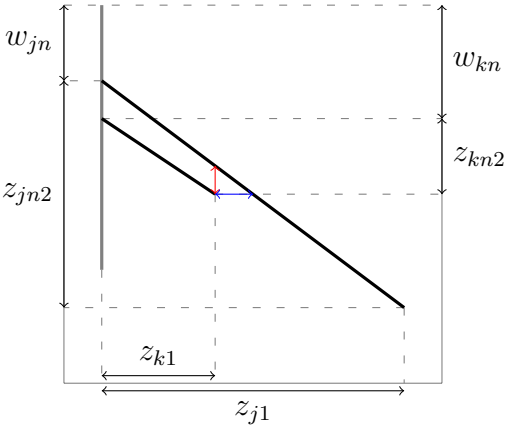

Figure 3.9: $u_{jk1} = 1$, $u_{jk2} = 1$.

3.3.3.5 Objective function

A "low" NPV function can be proposed if parameter h is fixed at 1. If $h = 0$, the model maximizes the total amount of oil produced. Formally, this quantity can be estimated through the linear objective function

$$\max_{x,s} \sum_{n\in\Phi, i\in\Pi, j\in\Gamma_n} x_{ijn} O_i C - h \sum_{n\in\Phi, j\in\Gamma_n} s_{jn} F$$

which accounts for the oil associated with produced areas, and the length of branches.

The model gives satisfying results in 2D as we can see in its application in Section 3.4.2), but cannot be generalized to the third dimension. Too many additive variables and constraints would be involved, in particular for the handling of crossing branches. In the next section, we propose another model that can be applied in 3D.

3.3.4 Model 3, a 3D model

In this section, we present a 3D model for the branching problem.

3.3.4.1 Sets

- $\Gamma = \{1, \ldots, K\}$: set of branches.

Indexes:

- j: index for branch $j \in \Gamma$.

Parameters:

- K: maximum number of branches allowed for each well.
- D: minimum distance between two junctions of the same well.
- P: minimum distance between two branches of two distinct wells.

3.3.4.2 Variables

Continuous:

- w_{jn}: the junction point of the j-th branch of well n with coordinates $(w_{jnd})_{d\in\Delta}$ ($n \in \Phi$, $j \in \Gamma_n$).
- z_{jn}: the ending point of the j-th branch of well n with coordinates $(z_{jnd})_{d\in\Delta}$ ($n \in \Phi$, $j \in \Gamma_n$).
- s_{jn}: square of the distance between the starting and the ending point for j-th branch of well n ($n \in \Phi$, $j \in \Gamma_n$).
- λ_{ijn}: used to define a point in $[w_{jn}, z_{jn}]$ ($n \in \Phi, j \in \Gamma_n$) and compute its distance to area i ($i \in \Pi_n$) (closest point to i).

Binary:

- x_{ijn}: identify if i is reached (and produced) by j-th branch of well n [1 if produced, 0 otherwise] ($i \in \Pi$, $n \in \Phi$, $j \in \Gamma_n$).
- a_{jn}: identify if the j-th branch of well n is active [1 if active, 0 otherwise]. A branch is active if its produce at least one area.

Some additional variables will be introduced later to reformulate the problem.

3.3.4.3 Simple Bounds

- $V_{n2} \leq w_{jn2} \leq W_n + V_{n2}, \quad \forall j \in \Gamma_n, n \in \Phi,$
- $w_{jnd} = V_{nd}, \quad \forall n \in \Phi, j \in \Gamma_n, d \in \Delta \setminus \{2\},$
- $V_{n2} \leq z_{jn2} \leq L + W_n + V_{n2}, \quad \forall n \in \Phi, j \in \Gamma_n,$
- $-L + V_{nd} \leq z_{jnd} \leq L + V_{nd}, \quad \forall n \in \Phi, j \in \Gamma_n, d \in \Delta \setminus \{2\},$

- $0 \leq s_{jn} \leq \min\{||E||_2^2,\ L^2\}, \quad \forall j \in \Gamma_n, n \in \Phi,$
- $\max\{0, V_{nd}\} \leq z_{jnd} \leq \min\{E_d,\ V_{nd} + L\}, \quad \forall j \in \Gamma_n, n \in \Phi, d \in \Delta \setminus \{2\},$
- $\max\{0, V_{n2} - L\} \leq z_{jn2} \leq \min\{E_2,\ V_{n2} + W[n] + L\}, \quad \forall j \in \Gamma_n, n \in \Phi,$
- $\lambda_{ijn} \in [0, 1], \quad \forall n \in \Phi, i \in \Pi_n, j \in \Gamma_n.$

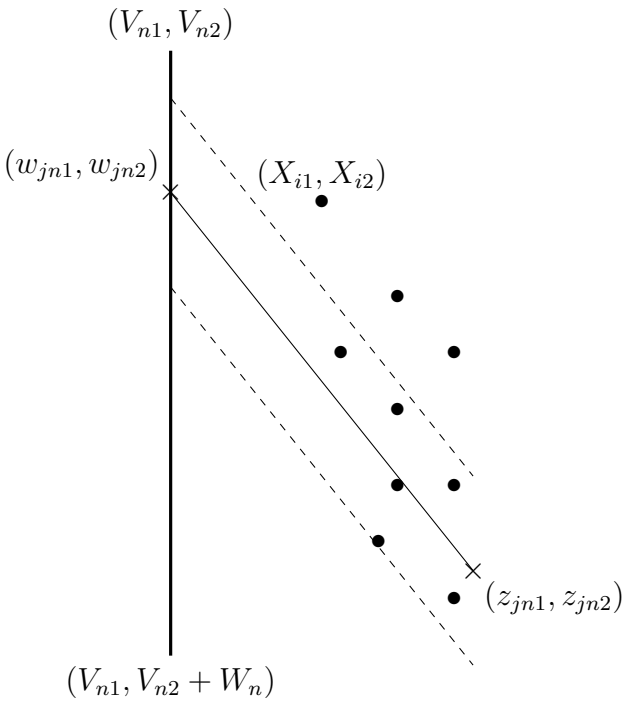

Figure 3.10: Branch modeling (in two dimensions) for Model 3.

In FIG. 3.10, we represent the branch modeling for the third proposed model. Branch j on well n is represented by the coordinates of its junction point w_{jnd} on well n, and its branch's extremity point z_{jnd}.

3.3.4.4 Constraints

- The total length of built branches must not exceed the maximum length H

$$\sum_{n \in \Phi} \sum_{j \in \Gamma_n} s_{jn} \leq H^2. \tag{3.56}$$

- The distance square s_{jn} is equal to the Euclidean distance square between w_{jn} and z_{jn} if branch j of well n is active (i.e., if $a_{jn} = 1$). If branch j is not active, s_{jn} is equal to 0.

$$\forall j \in \Gamma_n, n \in \Phi, \quad s_{jn} \leq a_{jn} L^2, \tag{3.57}$$

$$\forall j \in \Gamma_n, n \in \Phi, \quad s_{jn} \geq ||z_{jn} - w_{jn}||_2^2 - (1 - a_{jn}) L^2. \tag{3.58}$$

(We have replaced = by $\geq$ to have a convex formulation as we are minimizing s).

- Every (center) oil area is at most reached (and produced) by one branch

$$\forall i \in \bigcup_{n\in\Phi} \Pi_n, \quad \sum_{n\in\Phi: i\in\Pi_n} \sum_{j\in\Gamma_n} x_{ijn} \leq 1. \tag{3.59}$$

- The j-th branch of well n is active if it produce at least one point

$$\forall i \in \bigcup_{n\in\Phi} \Pi_n, \quad x_{ijn} \leq a_{jn}. \tag{3.60}$$

- The distance between a center oil area and its producing branch must not exceed R

$$\forall n \in \Phi, i \in \Pi_n, j \in \Gamma_n, \quad ||X_i - (\lambda_{ijn} w_{jn} + (1-\lambda_{ijn}) z_{jn})||_2^2 \leq R^2 x_{ijn} + M(1 - x_{ijn}). \tag{3.61}$$

- The distance between two branches of two different wells must be at least P if they are both active

$$\forall n, m \in \phi, m > n, j \in \Gamma_n, k \in \Gamma_m, \quad f(w_{jn}, z_{jn}, w_{km}, z_{km}) \geq P(-1 + a_{jn} + a_{km}), \tag{3.62}$$

where

$$f(w_{jn}, z_{jn}, w_{km}, z_{km}) = \begin{cases} \min \quad d_1 \\ d_1 = ||\alpha_1 w_{jn} + (1-\alpha_1) z_{jn} - (\alpha_2 w_{km} + (1-\alpha_2) z_{km})||_2^2 \\ d_1 \geq 0, \quad \alpha_1, \alpha_2 \in [0,1]. \end{cases} \tag{3.63}$$

- The distance between two branches of a same well must be at least D

$$\forall n \in \phi, j, k \in \Gamma_n, k > j, \quad f(w_{jn}, z_{jn}, w_{kn}, z_{kn}) \geq D(-1 + a_{jn} + a_{kn}), \tag{3.64}$$

where

$$f(w_{jn}, z_{jn}, w_{kn}, z_{kn}) = \begin{cases} \min \quad d_2 \\ d_2 = ||\beta_1 w_{jn} + (1-\beta_1) z_{jn} - (\beta_2 w_{kn} + (1-\beta_2) z_{km})||_2^2 \\ d_2 \geq 0, \quad \beta_1, \beta_2 \in [0,1]. \end{cases} \tag{3.65}$$

We also have symmetry breaking constraints.

- $j-1$-th branch of a well is active only if j-th branch is active

$$\forall j, k \in \Gamma_n, k = j - 1 \quad a_{jn} \geq a_{kn}. \tag{3.66}$$

- The 2-th coordinate of the junction point of branches of a same well are ordered

$$\forall j, k \in \Gamma_n, k = j - 1 \quad w_{jn2} \geq w_{kn2}. \tag{3.67}$$

- If point i is not produced, the value of λ is set to 0

$$\forall i \in \bigcup_{n\in\Phi} \Pi_n, \quad x_{ijn} \leq \lambda_{ijn}. \tag{3.68}$$

3.3.4.5 Objective function

The objective is to maximize the amount of produced oil by the additional built branches. This is given by:

$$\sum_{n\in\Phi,i\in\Pi,j\in\Gamma_n} x_{ijn}O_iC - \sum_{n\in\Phi,j\in\Gamma_n} s_{jn}F. \tag{3.69}$$

3.3.4.6 Reformulation

We can simplify some constraints, by introducing auxiliary variables. In Equations (3.59) we replace the term $(w_{jnd} - z_{jnd})$ by the variable r_{jnd} $(n \in \Phi, j \in \Gamma_n, d \in \Delta)$. Then the product $\lambda_{ijn}w_{jnd} + (1 - \lambda_{ijn})z_{jnd}$ becomes $\lambda_{ijn}r_{jnd} + z_{jnd}$. The product $\lambda_{ijn}r_{jnd}$ is isolated in variable u_{ijnd} $(n \in \Phi, i \in \Pi_n, j \in \Gamma_n, d \in \Delta)$, and we use variable q_{ijnd} $(n \in \Phi, i \in \Pi_n, j \in \Gamma_n, d \in \Delta)$ for the sum of the terms u_{ijnd} and z_{ijnd}. Theses reformulations correspond to the following constraints :

$$r_{jnd} = w_{jnd} - z_{jnd} \qquad n \in \Phi, j \in \Gamma_n, d \in \Delta, \tag{3.70}$$
$$u_{ijnd} = \lambda_{ijn}r_{jnd} \qquad n \in \Phi, i \in \Pi_n, j \in \Gamma_n, d \in \Delta, \tag{3.71}$$
$$q_{ijnd} = u_{ijnd} + z_{jnd} \qquad n \in \Phi, i \in \Pi_n, j \in \Gamma_n, d \in \Delta, \tag{3.72}$$
$$||X_i - q_{ijnd}||_2^2 \leq R^2x_{ijn} + M(1 - x_{ijn}) \qquad \forall n \in \Phi, i \in \Pi_n, j \in \Gamma_n. \tag{3.73}$$

Bounds can be provided for the auxiliary variables:

$$-L \leq u_{jnd} \leq L \qquad j \in \Gamma_n, d \in \Delta \setminus \{2\}, \tag{3.74}$$
$$-L \leq u_{jn2} \leq 0 \qquad j \in \Gamma_n. \tag{3.75}$$
$$-L \leq r_{jnd} \leq L \qquad j \in \Gamma_n, d \in \Delta \setminus \{2\}, \tag{3.76}$$
$$-L \leq r_{jn2} \leq 0 \qquad j \in \Gamma_n. \tag{3.77}$$
$$0 \leq q_{ijnd} \leq \max\{w_{jnd}, zjnd\} \qquad j \in \Gamma_n, d \in \Delta. \tag{3.78}$$

To help the optimization, in addition to the equality Constraints (3.71), we add McCormick approximation constraints (McCormick [58]).

$$u_{ijn2} \geq -L\lambda_{ijn} \qquad \forall i \in \bigcup_{n\in\Phi} \Pi_n, j \in \Gamma_n, \tag{3.79}$$
$$u_{ijn2} \geq r_{jn2} \qquad \forall i \in \bigcup_{n\in\Phi} \Pi_n, j \in \Gamma_n, \tag{3.80}$$
$$u_{ijn2} \leq r_{jn2} + \lambda_{ijn}L - L \qquad \forall i \in \bigcup_{n\in\Phi} \Pi_n, j \in \Gamma_n, \tag{3.81}$$
$$u_{ijn2} \leq 0 \qquad \forall i \in \bigcup_{n\in\Phi} \Pi_n, j \in \Gamma_n, \tag{3.82}$$
$$u_{ijnd} \geq -\lambda_{ijn}L \qquad \forall i \in \bigcup_{n\in\Phi} \Pi_n, j \in \Gamma_n, d \in \Delta \setminus \{2\}, \tag{3.83}$$
$$u_{ijnd} \geq r_{jnd}\lambda_{ijn}L - L \qquad \forall i \in \bigcup_{n\in\Phi} \Pi_n, j \in \Gamma_n, d \in \Delta \setminus \{2\}, \tag{3.84}$$
$$u_{ijnd} \leq \lambda_{ijnd}L \qquad \forall i \in \bigcup_{n\in\Phi} \Pi_n, j \in \Gamma_n, d \in \Delta \setminus \{2\}, \tag{3.85}$$
$$u_{ijnd} \leq r_{jnd} - \lambda_{ijn}L + L \qquad \forall i \in \bigcup_{n\in\Phi} \Pi_n, j \in \Gamma_n, d \in \Delta \setminus \{2\}. \tag{3.86}$$

3.3.4.7 Solving the problem

In this section, we aim at solving the optimization problem defined in the previous sections. We maximize the objective function (3.69) under several constraints.
Constraints(3.56) to (3.61) and constraints (3.66) to (3.86) are regrouped in the set of constraints $\mathcal{C}$. The problem can be written as follows:

$$\max_{x,s} \sum_{n\in\Phi, i\in \bigcup_{n\in\Phi}\Pi, j\in\Gamma_n} x_{ijn}O_iC - \sum_{n\in\Phi, j\in\Gamma_n} s_{jn}F,$$

$$s.t\ C$$

$$f(w_{jn}, z_{jn}, w_{km}, z_{km}) \geq P^2(-1 + a_{jn} + a_{km}) \quad \forall n, m \in \phi, m > n, j \in \Gamma_n, k \in \Gamma_m, \quad (3.87)$$

$$f(w_{jn}, z_{jn}, w_{kn}, z_{kn}) \geq D^2(-1 + a_{jn} + a_{kn}) \quad \forall n \in \phi, j, k \in \Gamma_n, k > j, \quad (3.88)$$

where

$$f(w_{jn}, z_{jn}, w_{km}, z_{km}) = \begin{cases} \min ||\alpha_1 w_{jn} + (1-\alpha_1)z_{jn} - (\alpha_2 w_{km} + (1-\alpha_2)z_{km})||_2^2 \\ \alpha_1, \alpha_2 \in [0,1], \end{cases}$$

$$f(w_{jn}, z_{jn}, w_{kn}, z_{kn}) = \begin{cases} \min ||\beta_1 w_{jn} + (1-\beta_1)z_{jn} - (\beta_2 w_{kn} + (1-\beta_2)z_{kn})||_2^2 \\ \beta_1, \beta_2 \in [0,1]. \end{cases}$$

To solve the optimization problem given by Model 3, we propose an iterative row generation algorithm. The approach is to solve iteratively a master problem $\mathcal{P}M$, and slave problems $\mathcal{P}s(w_{jn}, z_{jn}, w_{km}, z_{km})$ and $\mathcal{P}s(w_{jn}, z_{jn}, w_{kn}, z_{kn})$. We first define an initial master problem $\mathcal{P}M$ given by:

$$\begin{cases} \max & \sum_{n\in\Phi, i\in \bigcup_{n\in\Phi}\Pi, j\in\Gamma_n} x_{ijn}O_iC - \sum_{n\in\Phi, j\in\Gamma_n} s_{jn}F \\ s.t & C \end{cases} \quad (\mathcal{P}M)$$

Then we introduce the two slave problems and , and first we present the slave problem related to Equation (3.87). The problem as continuous variables α_1, and α_2, and its parameters are w_{jn}, z_{jn}, w_{km}, and z_{km}.

$$\begin{cases} \min\limits_{\alpha_1,\alpha_2} \quad f(w_{jn}, z_{jn}, w_{km}, z_{km}) \\ f(w_{jn}, z_{jn}, w_{km}, z_{km}) = ||\alpha_1 w_{jn} + (1-\alpha_1)z_{jn} - (\alpha_2 w_{km} + (1-\alpha_2)z_{km})||_2^2 \\ \alpha_1, \alpha_2 \in [0,1]. \end{cases}$$

$$(\mathcal{P}s(w_{jn}, z_{jn}, w_{km}, z_{km}))$$

The slave problem related to Equation (3.88), with continuous variables β_1 and β_2, and parameters w_{jn}, z_{jn}, w_{km}, and z_{km}, is written as follows:

$$\begin{cases} \min\limits_{\beta_1,\beta_2} \quad f(w_{jn}, z_{jn}, w_{kn}, z_{kn}) \\ f(w_{jn}, z_{jn}, w_{kn}, z_{kn}) = ||\beta_1 w_{jn} + (1-\beta_1)z_{jn} - (\beta_2 w_{kn} + (1-\beta_2)z_{kn})||_2^2 \\ \beta_1, \beta_2 \in [0,1]. \end{cases}$$

$$(\mathcal{P}s(w_{jn}, z_{jn}, w_{kn}, z_{kn}))$$

Initially, we solve the master problem $\mathcal{P}M$, and let (w, z, a) be its optimal solution. Then, for all active branch j of well n and active branch k of well m such that $n < m$,

we solve the problem $\mathcal{P}s(w_{jn}, z_{jn}, w_{km}, z_{km})$, and let $(\hat{\alpha}_1, \hat{\alpha}_2)$ be its solution, and f_1 its solution value. If $f_1 < P^2$ then we add to problem $\mathcal{P}M$ the constraint:

$$||\hat{\alpha_1} w_{jn} + (1 - \hat{\alpha_1}) z_{jn} - (\hat{\alpha_2} w_{km} + (1 - \hat{\alpha_2}) z_{km})||_2^2 > P^2(-1 + a_{jn} + a_{km}). \qquad (S_1)$$

Similarly, for all active branches j and k of well n such that $j < k$, we solve the problem $\mathcal{P}s(w_{jn}, z_{jn}, w_{kn}, z_{kn})$, and let $(\hat{\beta}_1, \hat{\beta}_2)$ be its solution, and f_2 its solution value. If $f_2 < D^2$ then we add to problem $\mathcal{P}M$ the constraint:

$$||\hat{\beta_1} w_{jn} + (1 - \hat{\beta_1}) z_{jn} - (\hat{\beta_2} w_{km} + (1 - \hat{\beta_2}) z_{km})||_2^2 > D^2(-1 + a_{jn} + a_{kn}). \qquad (S_2)$$

We repeat the process until the values f_1 and f_2 respectively generated by solving problems $\mathcal{P}s(w_{jn}, z_{jn}, w_{km}, z_{km})$ and $\mathcal{P}s(w_{jn}, z_{jn}, w_{km}, z_{km})$ are respectively greater than P^2 and D^2. The optimal solution of the whole optimization problem is then (w, z, a), solution of the master problem $\mathcal{P}M$.

The algorithm defined below presents this row generation method applied to our problem.

Algorithm

Initialization
Let P the minimum distance between two branches of a distinct wells, and D the minimum distance between two branches of a same well. Let $feas \leftarrow 0$

(1.) Solve master problem $\mathcal{P}M$ and let (w, z, a) be its solution. Let $feas \leftarrow 1$.

(2.) **For each** $(n, m) \in \Phi, n < m$ **do**

For each $j \in \Gamma_n, k \in \Gamma_m$, **do**

If $a_{jn} = 1$ and $a_{km} = 1$ Solve problem $\mathcal{P}s(w_{jn}, z_{jn}, w_{km}, z_{km})$ and let be $(\hat{\alpha}_1, \hat{\alpha}_2)$ its solution and f_1 its solution value

If $f_1 < P^2$, Let $feas \leftarrow 1$, Add to problem $\mathcal{P}M$ the constraint S_1
$||\hat{\alpha_1} w_{jn} + (1 - \hat{\alpha_1}) z_{jn} - (\hat{\alpha_2} w_{km} + (1 - \hat{\alpha_2}) z_{km})||_2^2 > P^2(-1 + a_{jn} + a_{km})$

(3.) **For each** $n \in \Phi$, **do**

For each $(j, k) \in \Gamma_n, j < k$ **do**

If $a_{jn} = 1$ and $a_{kn} = 1$, Solve problem $\mathcal{P}s(w_{jn}, z_{jn}, w_{kn}, z_{kn})$ and let be $(\hat{\beta}_1, \hat{\beta}_2)$ its solution and f_2 its solution value

If $f_2 < D^2$, Let $feas \leftarrow 0$, Add to problem $\mathcal{P}M$ the constraint S_2
$||\hat{\beta_1} w_{jn} + (1 - \hat{\beta_1}) z_{jn} - (\hat{\beta_2} w_{km} + (1 - \hat{\beta_2}) z_{km})||_2^2 > D^2(-1 + a_{jn} + a_{kn})$

(4.) **If** $feas = 1$ the solution of master problem $\mathcal{P}M$ is the solution of the optimization problem. **Else** Go to Step (1.).

We apply this algorithm in Section 3.4.3 on a small simple 3D case. Different software are used to solve the master problem and the slave problems.

3.4 Applications

In this section we test Model 1 and 2 (presented in Section 3.3.2 and 3.3.3) on the 2D case (see Section 1.4.2). Model 1 is used to optimize the trajectory of a single well, located in the center of the reservoir, the obtained results were presented in Lizon et al. [54]. The two steps of the methodology presented in the chapter are applied by optimizing the number and location of vertical producer wells with NOMAD. Then we use Model 2 to optimize the trajectory of the obtained wells. Then we test Model 3 on a small case with 2 vertical wells.

3.4.1 Optimization of well trajectory with Model 1

The initial configuration considered here consists in a single vertical producer located at the top of the anticline. FIG. 3.1 represents the $H_u\phi S_o$ values after a production time-frame of 4 years. The cumulative produced oil is 45.39 thousand cubic meters (285 Mbbl).

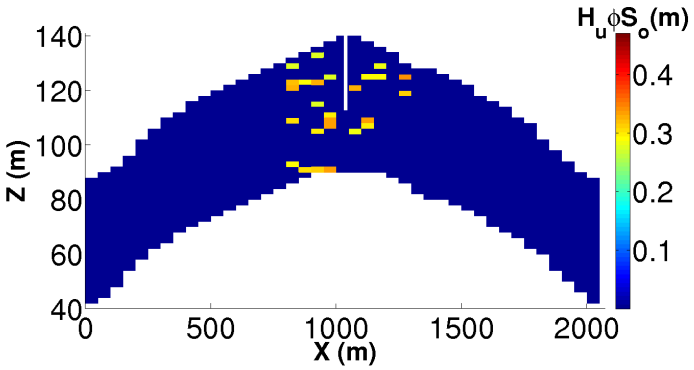

Figure 3.11: Map of high production potential areas.

In order to limit the number of variables in the model, we selected the high production potential areas, by using an approximation of $H_u\phi S_o$ based on a cubic spline interpolation using not-a-knot end conditions by using the interp2 MATLAB function with the "spline" method, for areas of 2×20 cells. We kept areas with an $H_u\phi S_o$ score above 2.75, these areas are represented in FIG. 3.11.

The mathematical model presented in Section 3.3.2 is written in AMPL, A Mathematical Programming Language, (Fourer et al. [37]), an algebraic modeling language widely used in the MINLP community. The input parameters of Model 1 are the maximum number of clusters, $K = 10$, the maximum length of any branch, $\overline{L} = 250$ m, and the maximum number of branches, $B = 5$. We solve the optimization problem with Basic Open-Source NonLinear Mixed Integer Programming (BONMIN), a MINLP solver (Bonami et al. [16]), using Branch-and-Bound method. We obtain a feasible solution in 2 hours.

FIG. 3.12 shows the optimized well trajectory where 5 branches were added. In FIG. 3.13 this solution is superposed to the $H_u\phi S_o$ values associated with the initial vertical well configuration. The oil saturation after 4 years of production for the initial

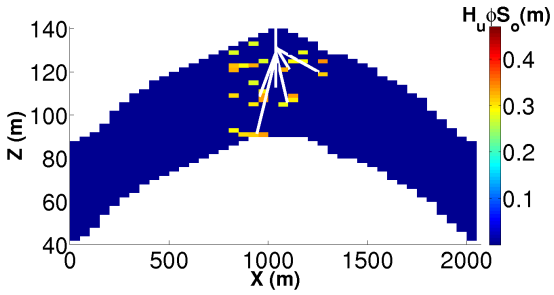

Figure 3.12: Optimized well trajectory with high production potential areas.

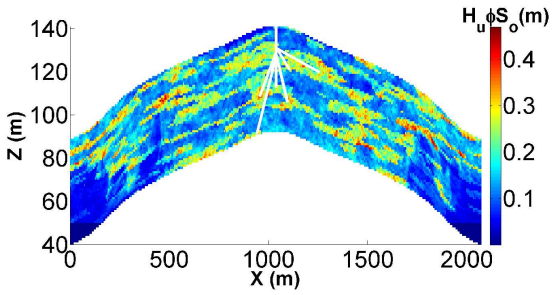

Figure 3.13: Optimized well trajectory with initial $H_u\phi S_o$ map.

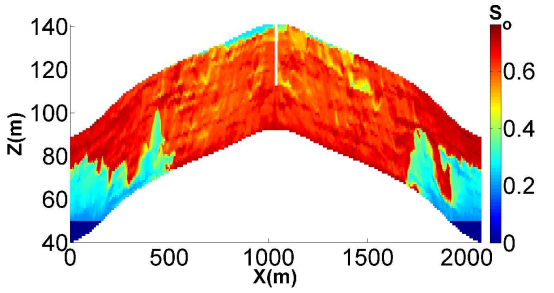

Figure 3.14: Oil saturation for initial vertical well configuration after 4 years of production.

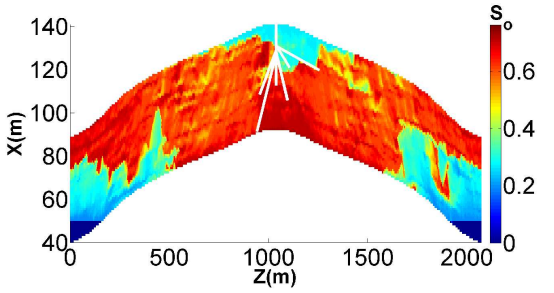

Figure 3.15: Oil saturation for optimized well trajectory with 5 branches after 4 years of production.

vertical well and for the optimized configuration with 5 branches are represented in Fig. 3.14 and 3.15. The improvement oil recovery is obvious.

The cumulative produced oil with the new configuration with 5 branches is equal to 51.59 thousand cubic meters (324 Mbbl). This means an improvement of 13.7% with respect to the initial configuration with one vertical well.

Despite the addition of branches, we can observe in Fig. 3.15, that there is still oil not produced in regions too far to be reached by a branch. Thus, we should revise the model and add the possibility to add vertical production wells when an area is too far from existing wells. Also in this model, oil is produced only at the extremities of the branch, which is not realistic since the production should be considered along the whole length of the branches.

3.4.2 Optimization with NOMAD + Model 2

In this section we apply the proposed methodology with Model 2 (see Section 3.3.3) on a 2D case presented in Section 1.4.2.

3.4.2.1 Optimization of vertical wells configuration with NOMAD

We first use NOMAD for the first step of the optimization of the location and the number of vertical producer wells. We optimization the location and the status active/inactive of 4 vertical producer wells, and we optimize the location of one vertical injector well. We use the NPV function 1.2, with parameters values given in TAB. 3.1. The production time-frame is of 6 years.

Table 3.1: NPV function, parameter values

Parameters	**Values**
C^{PO}	600 $\$/m^3$
C^{PW}	42 $\$/m^3$
C^{F}	1000 $\$/m$
C^{D}	1e6 \$
R	0.1

In FIG. 3.16 we can see the evolution of the value of the best NPV value at each simulation for the 100 first evaluations. It took 664 evaluations to reach convergence, with 5 days of execution time. In FIG. 3.17, we can see the $H_u\phi S_o$ map of solution configuration with 2 vertical wells, the configuration will be initial point of the second step of the methodology. The solution NPV value is of $74.6M\$$.

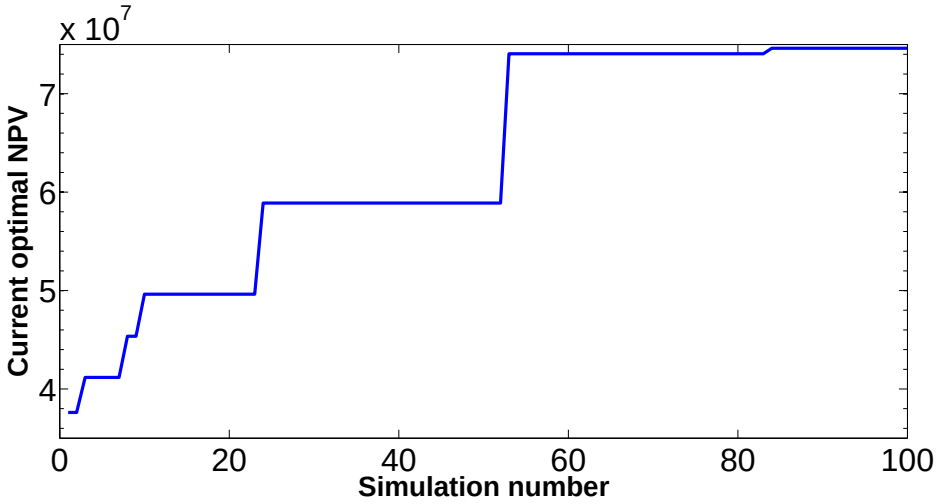

Figure 3.16: Best NPV during optimization

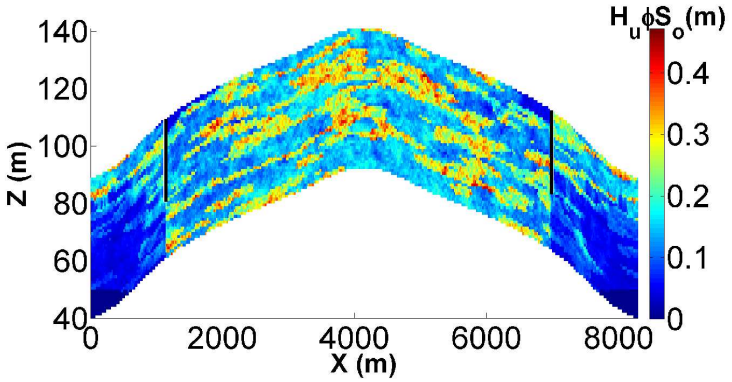

Figure 3.17: Solution of vertical well optimization $H_u\phi S_o$ map.

3.4.2.2 Optimization of wells trajectory with Model 2

For the second step of the optimization, we use the mathematical model presented in Section 3.3.3, written in AMPL. The input parameters of Model 2 are the maximum number of branches per well, $B = 4$, the maximum length of any branch, $\overline{L} = 700$ m. The optimization problem is solved with SCIP [70], and a feasible solution is obtained in 2 hours.

In order to limit the number of variables in the problem, we selected the high production potential areas, by using an approximation of $H_u\phi S_o$ (initial values are represented in FIG. 3.17) based on a cubic spline interpolation using not-a-knot end conditions by using the interp2 MATLAB function with the "spline" method, for areas of 2×20, cells and we kept areas with an $H_u\phi S_o$ score above 2.75. These areas are represented in FIG. 3.18. In FIG. 3.19 we show the optimized well trajectory where 5 branches were added. The oil saturation after 6 years of production for the initial vertical well and for the optimized configuration with 5 branches are represented in FIG. 3.20 and 3.21. The improvement oil recovery is obvious. The corresponding NPV value is of $78.4M\$$, which is an improvement of 5%. As in the optimization with Model 1 (see Section 3.4.1), we can observe in FIG. 3.21, that there is still oil not produced in regions too far to be reached by a branch.

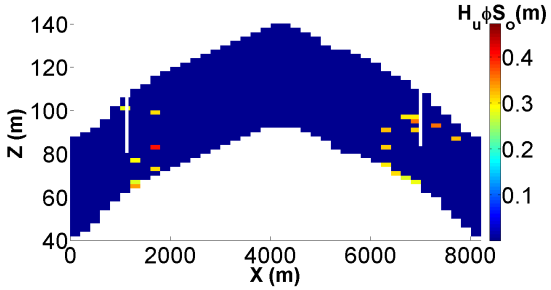

Figure 3.18: Map of high production potential areas.

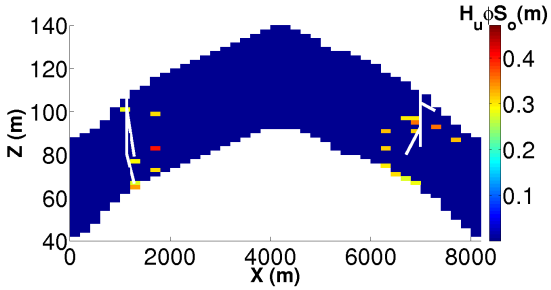

Figure 3.19: Optimized well trajectory with high production potential areas.

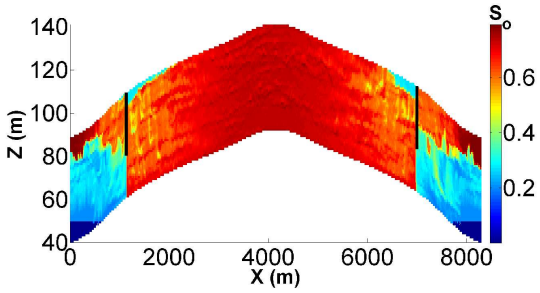

Figure 3.20: Oil saturation for initial vertical well configuration after 6 years of production.

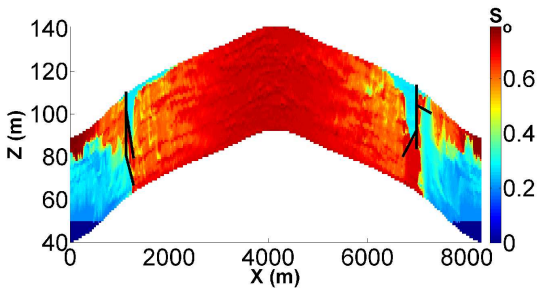

Figure 3.21: Oil saturation for optimized well trajectory with 5 branches after 6 years of production.

3.4.3 Optimization with Model 3

In this section we test Model 3 presented in Section 3.3.4 and the algorithm in Section 3.3.4.7. The model and the algorithm are written in AMPL (Fourer et al. [37]). We test Model 3 on a small case composed of 2 vertical wells, declined in 2D and 3D. The optimal solution is known (-16.61), and the problem is fast to solve. The input parameters of Model 3 are the maximum number of branches per well, $B = 2$, the maximum length of any branch, $\overline{L} = 700$, The minimum distance between branches of a same well $D = 11$, and between branches of distinct wells $P = 300$. In order to perform the algorithm presented in Section 3.3.4.7, we use SCIP, BONMIN and COUNNE Belotti et al. [13], to solve the master problem $\mathcal{P}M$, and the slave problems $(\mathcal{P}s(w_{jn}, z_{jn}, w_{km}, z_{km}))$ and $(\mathcal{P}s(w_{jn}, z_{jn}, w_{kn}, z_{kn}))$. For each problem the limit time of optimization is set at 1 hour.

The solution value f^* of the optimization problem, the optimization time, and the number of master problems that were solved are represented in Tab. 3.2 for each solver and each case (2D and 3D). We can see that SCIP was the most efficient to solve the 3D case. BONMIN was able to solve both cases with optimal solutions. BONMIN generated a larger number of master problems, when SCIP only solved 1 master problem. This means that the first master problem solution found by SCIP satisfied already the constraints and thus solving the slave problems did not generate new constraints. A non optimal solution was found for the 3D case by COUENNE in more than 1 hour. The first master problem was solved in 1 hour (which was the limit time), with a best objective value -16.770219 and a best possible value -16.770528. As these values were lower than the optimal objective function value (-16.61), solving the slave problems generated constraints. Then, COUENNE found an optimal solution for the second master problem. The solution value -16.22 is not the known optimal solution value. This is explained by one of the slave problems that was an infeasible problem for COUENNE, thus it may have generated too restrictive constraints.

From these runs, COUENNE is the least adapted to solve the problem with this row generation algorithm, SCIP is the fastest to solve the problem, but BONMIN is more reliable as it founds optimal solutions in both cases. Runs with COUENNE showed that caution must be taken when we solve the slave problems.

Table 3.2: Optimization with Model 3, using BONMIN, SCIP, and COUENNE

	BONMIN			SCIP			COUENNE		
Case	Time	$\mathcal{P}M$ solved	f^*	Time	$\mathcal{P}M$ solved	f^*	Time	$\mathcal{P}M$ solved	f^*
2D	4 min	2	-16.61	1 sec	/	error	12 min	3	error
3D	22 min	7	-16.61	1 sec	1	-16.61	1h05	4	-16.22

In order to test Model 3 on a case with a larger number of parameters and variables, we solved the problem defined in Section 3.4.2.2 and solved with Model 2 (presented in Section 3.3.4), but the solvers were not able to find a feasible result. To solve this problem efficiently we should design an algorithm dedicated to the problem. Having access to all feasible solutions of the master problem may be an help to solve the whole problem. Also we could generate a master problem for each well, and then solve the slave problems.

3.5 Conclusions

In this chapter we proposed an innovative methodology for well placement and well trajectory optimization by optimizing both the number of wells and the number of branches. In the relatively simple cases we considered, the methodology allowed to obtain a good improvement of the oil production and the NPV function.

Model 1, a 3D model presented in Section 3.3.2 forms cluster of oil before producing them at the extremities of the branches. However, forming clusters during the optimization can be costly in terms of constraints and variables. The model was tested on a 2D reservoir case, by proceeding only to the second step of the methodology. We obtain an improvement of 13 % of the oil production. We can note that the second step of the methodology can be applied alone if a good vertical wells configuration can be provided.

We presented a second 2D model, Model 2, in Section 3.3.3, which produces oil along the branches. We tested the whole methodology on the 2D reservoir model case (see Section 1.4.2). The second step of the methodology allowed to obtain an good improvement of 5 % of the NPV comparing to the NPV obtained in Step 1. We did not investigate a 3D model, because generalizing the model to the third dimension would have been too costly in constraints.

In Section 3.3.4 we presented a 3D model, Model 3, which produces oil along branches without considering oil clusters. The model was solved for a very simple 3D case, but its application to more realistic cases would require the design of a tailored algorithm. Computational time for solving the problem may be reduced by defining a problem for each well and solve then iteratively, or in parallel.

For the very simple 2D cases we considered, we analyzed the fluid flow simulator outputs corresponding to the end of the production time, however it may be interesting to compute scores at different production periods, e.g., when the oil production starts to

decline. Also for a better estimation of the oil production, we could solve the problem iteratively to improve the estimation of the radius of each branches.

We could also review the problem to as to have the possibility to add vertical production wells if remaining oil cannot be reached by a branch associated with an existing well. In this case much attention should be brought to the management of the number and relative location of branches for each well.

Another enhancement of the approach presented in this chapter could be related to the connectivity measure we considered (a distance criterion in this work). Alternative measures could be determined using the permeability field distribution, or a simplified/fast-to-evaluate fluid flow simulator.

Conclusions and outlooks

Conclusions

The aim of this thesis was to establish a methodology dedicated to the optimization of the number of wells and the trajectory of wells. Three methods were proposed for the optimization of the number and the location of rectilinear wells. One new method for the optimization of the number of wells and the number of branches was developed and tested.

- We first presented GPS and MADS methods, and the NOMAD solver in which these methods are implemented. The solver capabilities are evaluated for the optimization of vertical or deviated well number and position on different reservoir models, a 1D case and two 3D cases (the PUNQ-S3 case and the case derived from SPE10). The solver gives satisfactory results. However, depending on the optimization initialization, it can lead to local minima. The method remains costly in objective function evaluations, with numerous objective function evaluations being necessary for algorithm convergence. Moreover we also observed useless simulations during the optimization: the algorithm tested solutions with different coordinates for inactive wells.

- In the second method, we developed a substitution model appropriate for quantitative and qualitative variables, based on the kriging method (Zhou et al. [80]). The model predictability has been evaluated by estimating the NPV function value on a 1D reservoir model, for cases of different number and type of wells, and, even though the model has irregularities on the boundaries, we obtained estimations of sufficient quality to guide the optimization. To do so, we proposed two optimization algorithms:

  - a basic approach in which each iteration consists in minimizing the kriging model and updating it by adding the new point,
  - the EGO algorithm (Efficient Global Optimization, Jones et al. [44]), where the new point to be evaluate maximizes the expected improvement. This is a compromise between the model improvement and the minimization of the objective function.

  The two algorithms are tested for the optimization of the NPV function associated with the 1D case reservoir model by using two samples of different size (250 and 500

points) for the kriging model construction. The best results are obtained with the EGO algorithm, using the smallest construction sample. These results are slightly better than the one obtained with NOMAD, in terms of NPV function value and in terms of reservoir fluid flow simulator evaluations. Hence the kriging model is adapted to guide the well placement optimization without necessitating a sample of large size.

- The third method is a new Derivative Free Trust Region algorithm for Black-Box problems extended to binary variables. We developed an algorithm with different options for the handling of binary variables. The method is evaluated on the optimization of vertical wells on 1D case and a 3D case, and encouraging results have been obtained. The handling of binary variables still needs improvement since the binary variable set is not sufficiently explored. This is due to the model built during the optimization, which works better in continuous variables than in binary variables, thus a better point with different binary values is hard to find. The handling of constraints needs to be set up.

- Finally, a new two-step optimization method is proposed and developed, taking into account the characteristics of the Black-Box objective function, the result of a Black-Box simulator, and using classical MINLP methods. The first step consists in solving a Black-Box method that optimizes the placement and the number of vertical wells with the NOMAD solver (presented in the second chapter). In the second step we define a second MINLP problem from the previous step problem solution outputs. In this new MINLP problem, the hydrocarbon production is optimized by adding branches to vertical producer existing wells, the unknowns being the branches extremities position. The obtained configuration is evaluated by the Black-Box simulator, allowing to validate the problem solution. To perform this second step, three models of different MINLP problems for the branches conception are proposed.

  - A first 3D model, Model 1, identifies clusters of unproduced oil. Branches produce oil in clusters located at the branches extremities. This model is evaluated on a relatively simple 2D case without performing the first step of the methodology. The initial configuration with one vertical well located on the reservoir center is chosen, and solving the problem generated by the model allows to improve oil production by 13% with the addition of 5 branches.
  - The second model, Model 2, produces oil along the branches, which is more realistic, but was only designed for 2D cases. The proposed methodology is tested on the same previous 2D reservoir case: the first step, optimizing the number and location of vertical producer wells, allowed to obtain a 2 wells configuration. The second step, using the 2D model, adds a total of 5 branches to the 2 wells and allows an improvement of 5% of the NPV function. This model, and the above 3D model are effective, but their solution process is too long. Model 1 optimize the location of the center of clusters and the branch extremities, and Model 2 generates numerous constraints due to its design.

– A third 3D model, producing along the branches, is presented. An iterative row generation algorithm is proposed for its solving, and good results are obtained on a very simple non 3D case. However Model 3 requires a tailored algorithm to be applied efficiently to realistic cases.

Outlooks

Direct-search methods implemented in the NOMAD solver showed their effectiveness on the well placement problem, but still required too many objective function evaluations due to the algorithm convergence. However coupling the search step with substitution models could reduce the number of simulations. In order to perform this coupling, we propose to use the substitution model based on the kriging method extended to mixed variables (second chapter) to guide the optimization method. We could then set up our own search step in NOMAD, and use the substitution model and the EGO algorithm to identify new points to evaluate, as in Talgorn et al. [72]. The model would be built from points previously evaluated.

The MINLP problem solving methodology used in the third chapter could also be improved. It needs several (sometimes useless) iterations, particularly in the third proposed model. Designing a new algorithm for solving the third model will be the subject of future works, rather than directly using solvers such as BONMIN or SCIP. It could also be possible to proceed to several successive problem solving by varying the branch radius parameter and thus getting a good estimation of the production radius. We also observed unproduced oil that could not be reached by branches, hence adding new vertical wells in second step could improve the oil production.

The configuration output analysis obtained in first step could also be improved by accounting for the connectivity of the unproduced oil localization relating to the existing wells. As adding branches to wells can increase the water production, it is important to take this production into account in the second step. This could be done by adding water saturated areas to the problem but this approach has two main disadvantages: firstly this would increase the size of the problem, and secondly the oil production estimation we use is too uncertain to use another production parameter. Thus in order to account for water production, a method could be to generate several feasible solutions of the MINLP problem, and use a simplified simulator based for instance on the connectivity computation (simulation by stream lines), for their evaluation. The simplified simulator, with rapid execution, would enable to approximate the water generated by each configuration. The configuration with the best compromise between water and oil production would then be evaluated using the Black-Box simulator.

The second step of the methodology could be used alone for the optimization of well trajectory by letting experienced reservoir engineers choose the initial configuration, for instance in the case of reservoir models for which a fluid flow simulation takes several hours, and for which applying direct-search methods would be too costly in computational time.

Above research may find other applications in helping the decision making process,

for closing a producer well, opening an injector well or even turning a producer well which productivity is not profitable any more (very low oil production) into a injector well. This could be done by introducing an additive optimization variable corresponding to the period of closure/opening/conversion. The same kind of formulation could also allow the optimization of the drilling order, by obtaining intermediate producing configurations (with a reduced number of wells), before the final optimal configuration. Finally, the operating conditions (well pressure and rate) could be taken into account in the optimization.

Conclusions et perspectives en français

Conclusions

L'objectif de cette thèse était d'établir une méthodologie dédiée à l'optimisation du nombre de puits et de leur trajectoires. Nous avons proposé trois méthodes pour l'optimisation du nombre et de l'emplacement des puits verticaux ou déviés. Une nouvelle méthode pour l'optimisation du nombre de puits et le nombre de branches a été développée et évaluée.

- Nous avons présenté tout d'abord une méthode de Pattern-Search, méthode de recherche directe, ainsi que le logiciel NOMAD, dans lequel ces méthodes sont implémentées. Les possibilités du logiciel sont évaluées pour l'optimisation de la position et du nombre de puits verticaux ou déviés, sur différents modèles de réservoir, un cas 1D et et deux cas 3D (cas PUNQ-S3 et cas dérivé de SPE10). Le solveur donne des résultats satisfaisants, bien que, selon l'initialisation de l'optimisation, l'on puisse être bloqué dans des minima locaux. La méthode reste toutefois coûteuse en évaluation de la fonction objectif, de nombreuses évaluations de la fonction objectif étant nécessaire pour la convergence de l'algorithme ; de plus nous avons également observé des simulations inutiles au cours de l'optimisation : l'algorithme faisant varier les coordonnées d'un puits inactif.

- Pour la seconde méthode nous avons développé un modèle de substitution, adapté pour les variables quantitatives et qualitatives, basé sur le krigeage (Zhou et al. [80]). La prédictivité du modèle a été évaluée en estimant la valeur de la fonction NPV sur un modèle de réservoir 1D, pour des cas à nombre et type de puits variables, et bien que comportant des irrégularités sur les bords, nous avons obtenu des estimations de qualité suffisante pour guider l'optimisation. Pour cela, deux algorithmes d'optimisation utilisant le modèle de krigeage ont été proposés :

  - une approche basique dont chaque itération consiste à minimiser le modèle et le mettre à jour en y ajoutant le nouveau point obtenu,

  - l'algorithme EGO (Efficient Global Optimization, Jones et al. [44]), où le nouveau point à évaluer maximise l'espérance de gain, compromis entre l'amélioration du modèle et la minimisation de la fonction objectif.

  Les deux algorithmes sont testés pour l'optimisation de la fonction NPV associée au cas de modèle de réservoir 1D en utilisant deux échantillons de différente taille (250 et 500 points) pour la construction du modèle de krigeage. L'optimisation en utilisant le plus grand échantillon de points initiaux (nécessitant évidemment un grand nombre d'évaluations initiales) n'a pas mené à une meilleure solution, et a dans certains cas entrainé l'algorithme dans des minima locaux. L'approche d'optimisation basique permet d'obtenir des résultats intéressants, mais est aussi coûteuse en évaluations de la fonction objectif que le solveur NOMAD sur le cas, voire plus dans le cas du grand échantillon de points pour la construction du modèle initial. De plus il est possible d'être piégé dans un minimal local en minimisant le modèle de substitution. L'optimisation avec l'algorithme EGO, en utilisant l'échantillon de plus petite taille pour la construction du modèle, donne des résultats légèrement meilleurs que ceux obtenus avec NOMAD en terme de valeur de fonction NPV ainsi qu'en nombre total d'évaluations du simulateur d'écoulement. Le modèle de krigeage est ainsi adapté pour guider l'optimisation du placement de puits sans nécessiter un échantillon de très grande taille.

- Nous proposons également une nouvelle méthode d'optimisation par région de confiance pour les problèmes dit "boîtes-noires" étendue aux variables binaires, et avons développé un algorithme, avec différentes options pour la gestion des variables entières. La méthode est évaluée sur l'optimisation de la position de puits verticaux sur un cas 1D et un cas 3D, et des résultats encourageants sont obtenus. Il est nécessaire d'améliorer la gestion des variables entières en permettant de mieux explorer l'ensemble des possibilités. La gestion des contraintes est à mettre en place.

- Enfin, une nouvelle méthode d'optimisation en deux étapes est proposée et developpée, prenant en compte les caractéristiques de la fonction objectif, issue des résultats d'un simulateur "boîte-noire", tout en utilisant les méthodes MINLP classiques. La première étape consiste à résoudre un problème "boîte-noire" MINLP, optimisant le placement et le nombre de puits verticaux, avec le solver NOMAD, présenté dans le deuxième chapitre. Dans la seconde étape nous définissons un second problème MINLP à partir des données issues de la simulation de la configuration solution du problème de l'étape précédente. Dans ce nouveau problème MINLP, la production d'hydrocarbure est optimisée en ajoutant des branches aux puits producteurs verticaux existants, les inconnus étant la position des extrémités de ces branches. La configuration obtenue est évaluée par le simulateur "boîte-noire", permettant ainsi de valider la solution du problème. Pour réaliser cette deuxième étape, trois modèles de différents problèmes MINLP pour la conception des branches sont proposés.

  - Un premier modèle 3D, identifiant des clusters d'huile non produite et produisant ces clusters à l'extrémités de branches, est évalué sur le cas 2D inpiré

du cas SPE10. La configuration initiale comporte un puits vertical localisé arbitrairement au centre du réservoir, et la résolution du problème généré par le modèle permet d'améliorer la production d'huile de 13% en ajoutant 5 branches.

- Le deuxième modèle produit l'huile le long des branches ce qui est plus réaliste, mais a été conçu seulement pour les cas 2D. La méthodologie proposée est testée sur le même cas de réservoir 2D que précédemment : la première étape, optimisant le nombre et la position des puits producteurs verticaux avec NOMAD, a permis d'obtenir une configuration à 2 puits verticaux. La seconde étape, en utilisant le modèle 2D, ajoute un total de 5 branches aux 2 puits, et permet une augmentation de 5% de la fonction NPV. Ce modèle, et le modèle ci-dessus sont efficaces, mais leur résolution est trop longue, le premier modèle cherchant à la fois la position du centre des clusters et des extrémités des branches, et le second problème de par sa conception génère de nombreuses contraintes.
- Un troisième modèle, généralisé au cas 3D et produisant le long des branches, est présenté mais requiert un algorithme adapté à sa résolution pour pouvoir être appliqué efficacement à un cas réaliste.

Perspectives

Les méthodes de recherche directe de type Pattern-Search, implémentées dans le logiciel NOMAD, ont montré leur efficacité sur le problème de placement de puits, mais nécessite encore de trop nombreuses évaluations de la fonction objectif, dues à la convergence de l'algorithme : le couplage de l'étape de recherche avec des modèles de substitution pourrait permettre de diminuer le nombre de simulations. Pour réaliser ce couplage, nous proposons d'utiliser le modèle de substitution basé sur la méthode de krigeage étendue aux variables mixtes (deuxième chapitre) pour guider l'optimisation. Ainsi on pourrait mettre en place notre propre étape de recherche dans NOMAD, et utiliser le modèle de substitution et l'algorithme EGO pour identifier les nouveaux points à évaluer. Le modèle serait construit en utilisant les points déjà évalués.

Afin d'améliorer la méthodologie proposée dans le troisième chapitre, les possibilités seraient d'améliorer la résolution du problème d'optimisation de la seconde étape qui nécessite beaucoup d'itérations et parfois sans obtenir de résultats, notamment dans le cas du troisième modèle proposé. L'élaboration d'un algorithme pour la résolution du troisième modèle présenté dans le troisième chapitre fera l'objet de prochains travaux, plutôt que d'utiliser directement des solveurs tels que BONMIN ou SCIP. Il serait également possible de procéder à plusieurs résolutions successives du problème en faisant varier le paramêtre du rayon des branches afin d'avoir une bonne estimation de leur rayon de production.

L'analyse de la configuration obtenue en première étape pourrait egalement être améliorée en tenant en compte la connectivité de la localisation de l'huile non produite par rapport aux puits existant. L'ajout de branches aux puits pouvant augmenter la

quantité d'eau produite, il est important de prendre en compte les coûts liés à cette production dans la seconde étape. Cela pourrait se faire en incluant les zones saturées en eau aux données du problème d'optimisation MINLP, mais augmenterait d'une part la taille du problème, et d'autre part l'approximation de la production d'huile utilisée est trop incertaine pour y ajouter cette donnée. Ainsi pour pouvoir prendre en compte la production d'eau, une méthode pourrait être de générer plusieurs solutions admissibles du problème MINP, et d'utiliser un simulateur simplifié, basé par exemple sur le calcul des connectiviés (simulation par ligne de courant), pour leur évaluation. Le simulateur simplifié, à l'exécution rapide, permet d'obtenir une approximation de la production d'eau générée par chaque configuration, et de choisir d'évaluer avec le simulateur "boîte-noire" les configurations offrant le meilleur compromis entre production d'eau et huile.

La seconde étape de la méthodologie pourrait être utilisée seule pour l'optimisation de la trajectoire de puits, en choisissant la configuration de départ avec les connaissances de l'ingénieur de réservoir, par exemple pour des cas de modèle de réservoir dont la simulation d'écoulement prend plusieurs heures, et pour lesquels appliquer les méthodes de recherche directes seraient trop couteux en temps de calcul.

Concernant l'application, différents problèmes d'optimisation pourraient être envisagés par la suite, la décision de fermer un puits producteur ou d'ouvrir un puits injecteur, voir de transformer un puits producteur en un puits injecteur dans le cas d'une rentabilité moindre du puits (production très faible d'huile) pourrait être introduite en optimisant une variable supplémentaire correspondant à la date de fermeture/ouverture/conversion. On peut aussi envisager d'optimiser l'ordre de forage des puits, en obtenant des configurations opérationnelles intermédiaires (avec un nombre réduit de puits) avant la configuration optimale finale. Enfin, les conditions de fonctionnement des puits (débits et/ou pressions aux puits) pourraient également être prises en compte dans l'optimisation.

Nomenclature

List of symbols

ϕ Porosity

$H_u\phi S_o$ Oil utility thickness

IG Injected Gas (superscript)

IO Injected Oil (superscript)

IW Injected Water (superscript)

K Permeability

n index of well (subscript)

N_I Number of Injector wells

N_P Number of Producer wells

PG Produced Gas (superscript)

PO Produced Oil (superscript)

PW Produced Water (superscript)

S Saturation

t index of period of production (subscript)

List of acronyms

BB MINLP Black-Box Mixed Integer Non Linear Programming

BHFP Bottom Hole Fluid Pressure

BONMIN Basic Open-source Nonlinear Mixed INteger programming (software)

COUENNE Convex Over and Under ENvelopes for Nonlinear Estimation (software)

DFO Derivative Free Optimization

DFTR Derivative Free Trust Region (algorithm)

EGO Efficient Global Optimization (algorithm)

GPS Geralized Pattern Search (algorithm)

LHS Latin Hypercube Sample

MADS Mesh Adaptive Direct Search (algorithm)

MINLP Mixed Integer Non Linear Programming

NLP Non Linear Programming

NOMAD Nonlinear Optimization by Mesh Adaptive Direct Search (software)

NPV Net Present Value

RMSE Root Mean Square Error

SCIP Solving Constraint Integer Programs (software)

Appendix A

Optimization on a 1 dimensional case with continuous variables

In this section, we solve an NLP optimization problem on a 1 dimensional test case, presented in 1.4.1.

The Black-Box NLP optimization problem is written as:

$$\begin{cases} \min -NPV(x) \\ x \in [x_L, x_U] \subseteq \mathbb{R}^{n+m} \\ g(x) \leq 0 \end{cases} \tag{A.1}$$

with the design variables $x \in \mathbb{R}^{n+m}$. n is the fixed number of producer wells and m the fixed number of injector wells. x_L, x_U are the bound vectors for $[1, 1000]^{n+m}$. The resulting optimization problem deals with $n + m$ continuous variables. The considered NPV function is defined in Section 1.3, with binary variables fixed at 1 for each well, and its parameters are defined in Tab. A.1.

Table A.1: NPV function, parameter values.

Parameters	**Values**
C^{PO}	100 \$/m
C^{PW}	5 \$/m
C^{IW}	4 \$/m
C^{D}	0 \$

We use the fmincon MATLAB function from the optimization toolbox to optimize the NPV function for different values of $n \in \{1, 2\}$ and $m \in \{1, 2\}$. We use an interior

point method, and the gradient is computed with finite difference method studied in the rapport Lizon and Sinoquet [53]). As the number of wells is fixed, we do not take into account the drilling cost of a well C_{well}.

Optimization is done on two different domains, their permeabilities are represented in FIG. 1.1. For these two domains, different cases are runned:

- 1 injector well, 1 producer well (see FIG. A.1 for heterogeneous case and FIG. A.5 for homogeneous)
- 2 injector well, 1 producer well (see FIG. A.2 for heterogeneous case and FIG. A.6 for homogeneous)
- 1 injector well, 2 producer well (see FIG. A.3 for heterogeneous case and FIG. A.7 for homogeneous)
- 2 injector well, 2 producer well (see FIG. A.4 for heterogeneous case and FIG. A.8 for homogeneous)

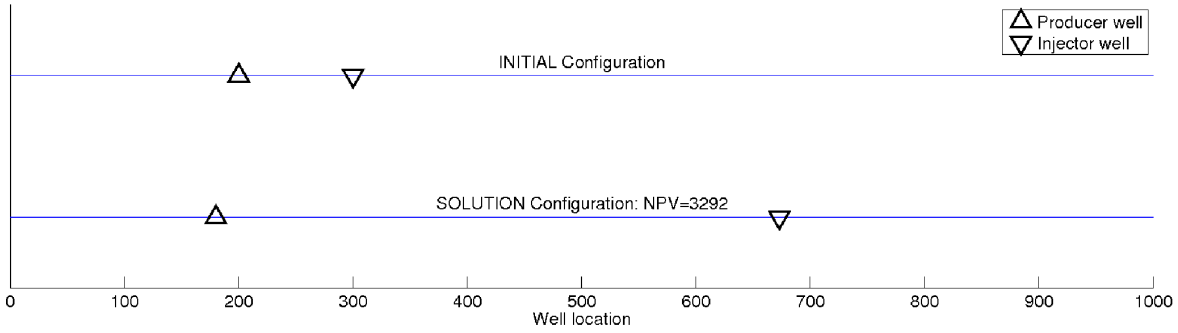

Figure A.1: Optimization with 1 producer well and 1 injector well, heterogeneous permeability.

By optimizing on the 1D domain with 1 injector well and 1 producer well, we would expect to obtain a solution configuration with one well at each extremity of the domain. We can see in FIG. A.1 that it is not the case. This can be explained by the important cost in water injected and produced associated to the one well at each extremity configuration. The permeability of the environment, non uniform, can also explain this result. The production time can also come into play: on lower period, the water production can be less high, and therefor the related costs can be less important.

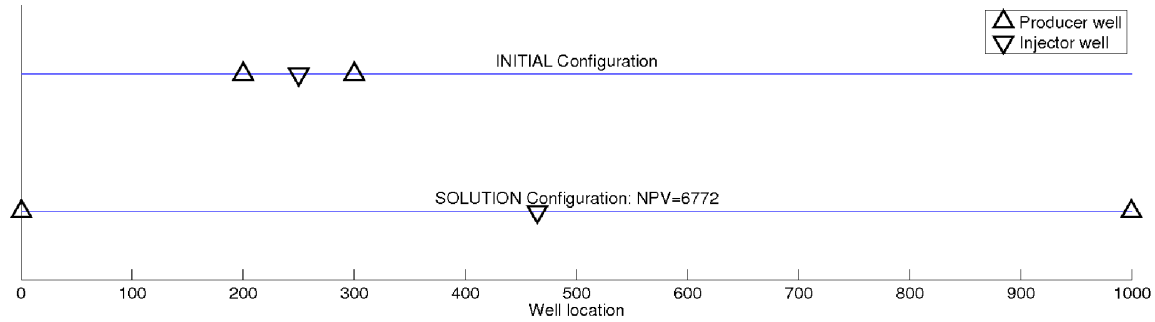

Figure A.2: Optimization with 2 producer well and 1 injector well, heterogeneous permeability.

In Fig. A.2, the optimization with 3 wells, 2 producers, 1 injector, give a solution that seem intuitively the best: both producer wells are at each extremity of the domain, and the injector well is at the center.

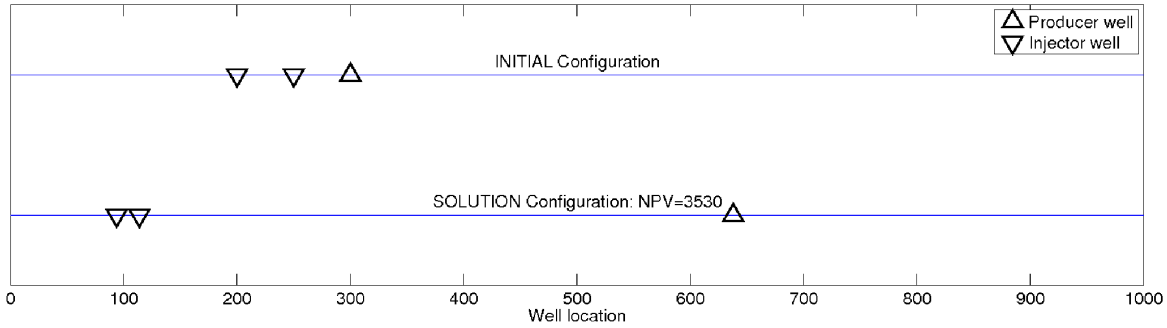

Figure A.3: Optimization with 1 producer well and 2 injector well, heterogeneous permeability.

In figure Fig. A.3, it seems that the optimization kept the initial order of the wells: the two injector wells are close, and the producer well is more far-off. The NPV function value is twice less important the one obtained for the optimization of two injector wells, and one injector well (see Fig. A.2). This configuration is in fact very costly in water. This solution is in all likelihood a local optimum.

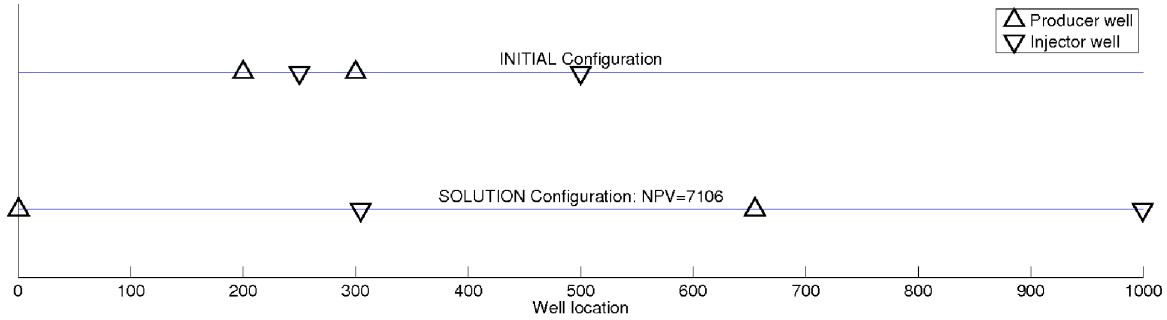

Figure A.4: Optimization with 2 producer well and 2 injector well, heterogeneous permeability.

We can observe in Fig. A.4 that the optimization of the position of two injector wells and two producer well give us a NPV function value lightly better than the one obtained for the optimization of two producers and two injectors (see Fig. A.2), but it requires the drilling of a additional well, which represent a cost that was not taken into account in that case.

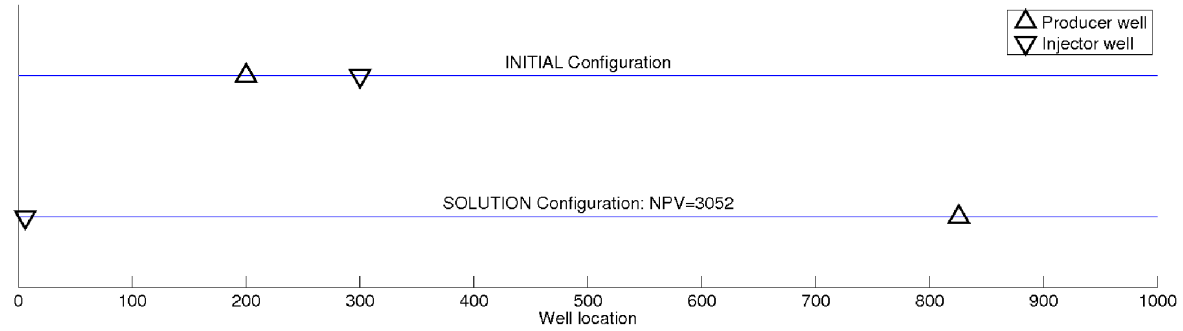

Figure A.5: Optimization with 1 producer well and 1 injector well, homogeneous permeability.

We now observe in Fig. A.5 the results of the optimization of the location of one injector well and one producer well. In this case the permeability of the domain is uniform. As in the heterogeneous permeability case, it is not the "intuitive" solution that is obtained.

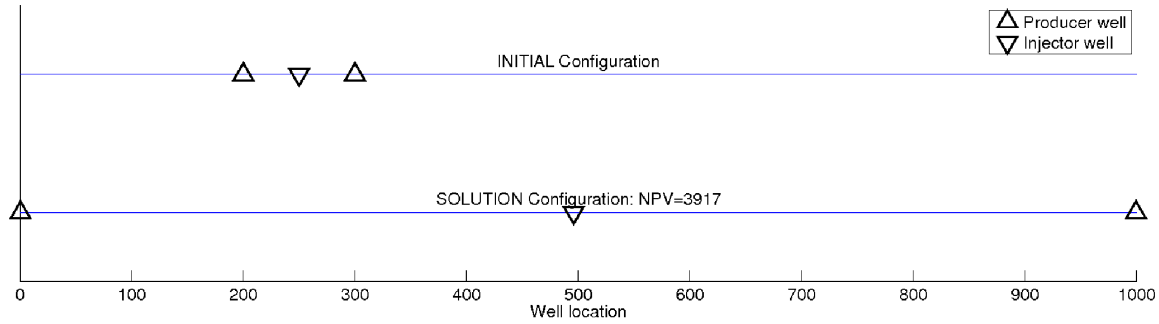

Figure A.6: Optimization with 2 producer well and 1 injector well, homogeneous permeability.

In Fig. A.6, we can see that result for the placement of two producer well, and one injector well. The configuration is similar to the one obtained in the heterogeneous case (see Fig. A.2), with different NPV values, which is not surprising seeing as geological properties of the domain are different.

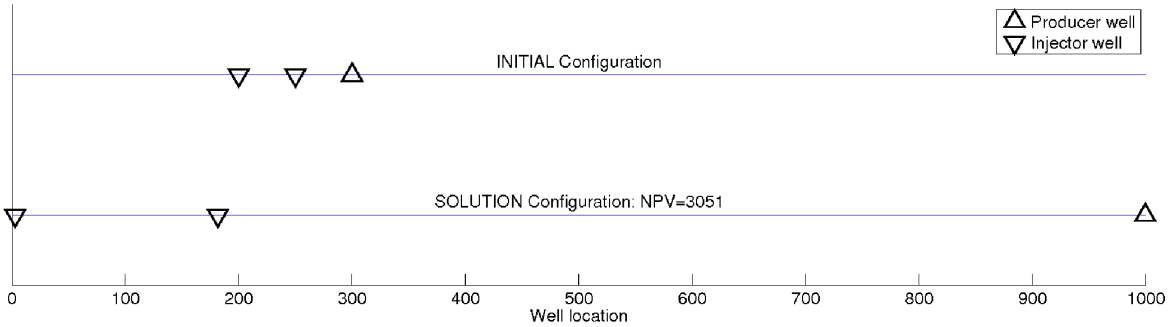

Figure A.7: Optimization with 1 producer well and 2 injector well, homogeneous permeability.

In Fig. A.7, the same problem as in Fig. A.3 is observed for the optimization of the placement of 2 injector wells, and 1 producer well: the order of the well remains unchanged, which could be a local optimum. Initial points has to be chosen carefully, or we can launch optimizations with several initial points.

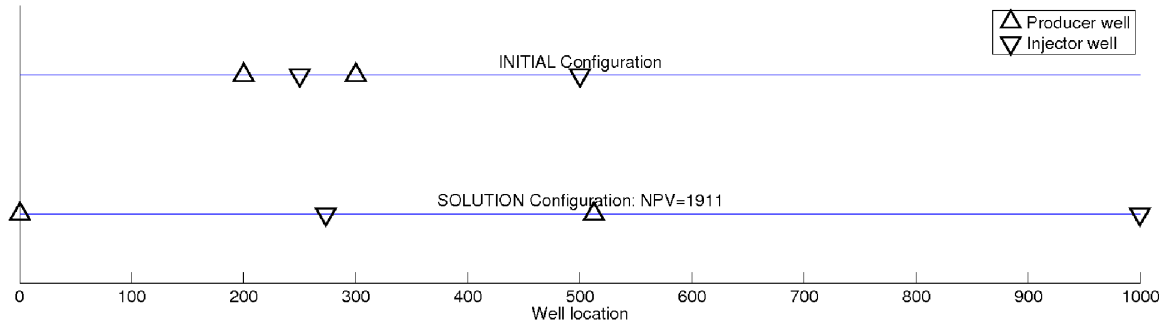

Figure A.8: Optimization with 2 producer well and 2 injector well, homogeneous permeability.

The optimization of two injector wells and two producer wells, in homogeneous environment (see Fig. A.8) does not obtain result as good (in terms of NPV function value) than with the two producer 1 injector well configuration (see Fig. A.6), as it was the case in the heterogeneous permeability environment.
These cases of optimization allow us to illustrate the importance of the choice of initial points, and the influence of the environment in which the wells are located. A method of type multi-start (launch of several local optimizations associated to different initial points) could avoid to be stocked in a wrong configuration because of a bad choice of initial point. We can also see the interest of optimization with integer value, since different solutions gave similar NPV values. Otherwise, fmincon is costly in number of evaluations, a hundred of evaluations is sometimes required for one iteration.

Appendix B

SPE10 case

In this section, we represent the Bottom Hole Fluid Pressure (BHFP), i.e., the well pressure for injector and producer wells for the starting and the solution configurations of the optimization from Section 2.1.3.3, and a reference configuration. Cumulative water injection quantities are also represented and are similar because the water injection rate was fixed (see Fig. B.3, Fig. B.7 and Fig. B.11).

B.1 Starting point

BHFP for injector wells and producers wells are displayed in Fig. B.1 and Fig. B.2, respectively. Cumulative water injection is displayed in Fig. B.3, and Fig. B.4 represents Water cut for producer wells. We can observe in Fig. B.3 that the water constraint was not satisfied for well $P3$ (in red) after 4 years, and for well $P2$ (in green) after 9 years of production. Thus the BHFP is constant for producer wells $P1$ (in blue) and $P4$ (in black) in Fig. B.9.

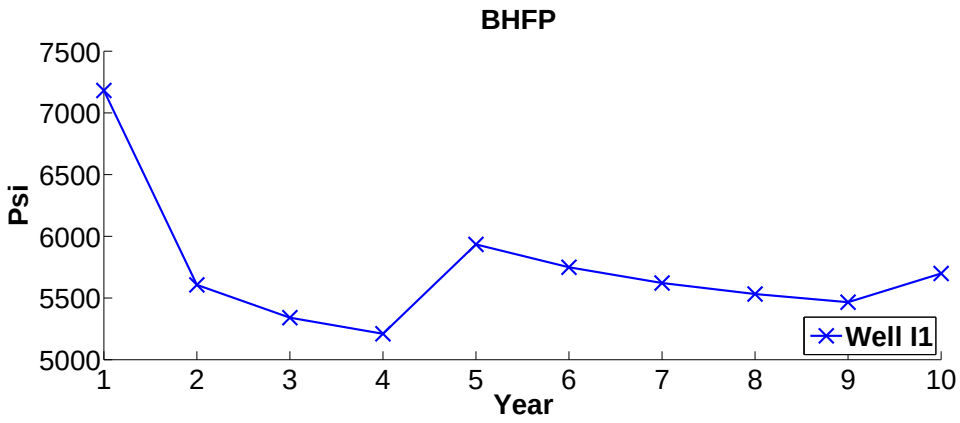

Figure B.1: Injection Bottom Hole Pressure of the starting point of the optimization (1 injector well and 4 producer wells).

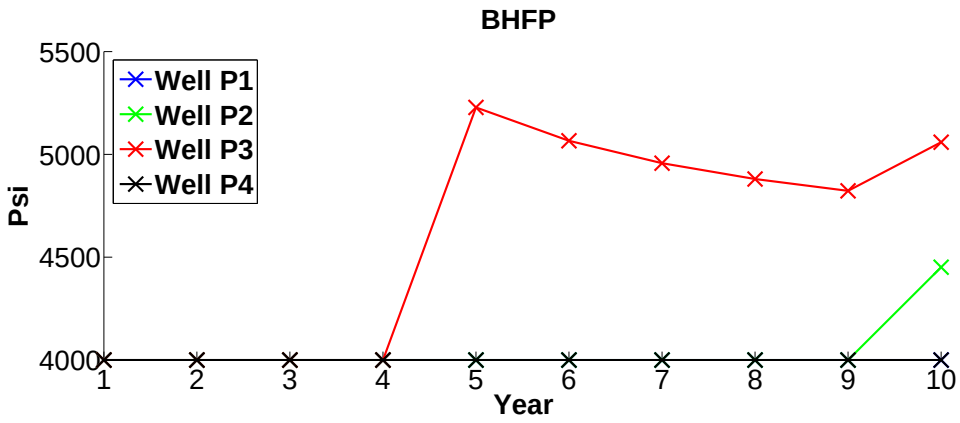

Figure B.2: Production bottom Hole pressure of the starting point of the optimization (1 injector well and 4 producer wells).

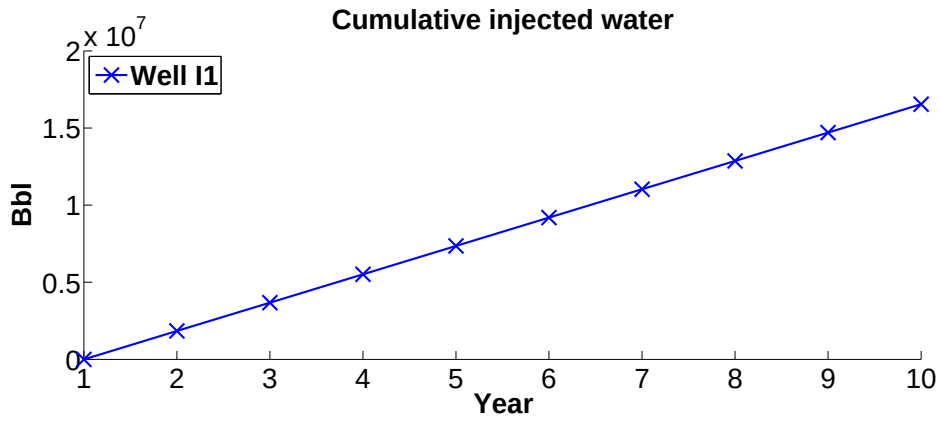

Figure B.3: Cumulative water injection of the starting point of the optimization (1 injector well and 4 producer wells).

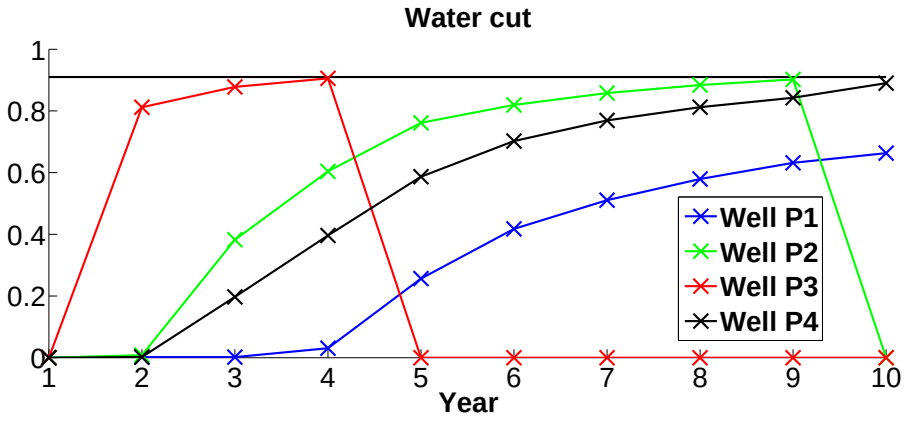

Figure B.4: Water cut for production wells of the starting point of the optimization (1 injector well and 4 producer wells).

B.2 Optimization solution

BHFP for injector wells and producers wells are respectively displayed in FIG. B.5 and FIG. B.6. Cumulative water injection is displayed in FIG. B.7. We can observe in FIG. B.8 that the water cut constraint was satisfied during the whole production time frame. Thus the BHFP is constant for the producer well in FIG. B.5.

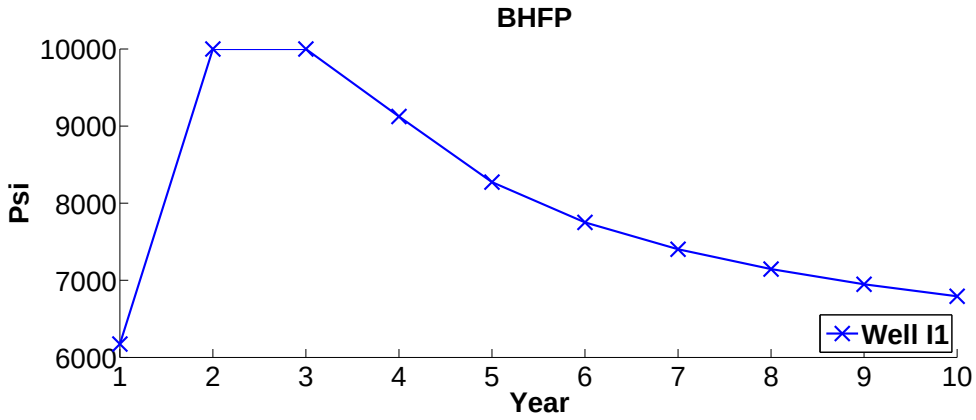

Figure B.5: Injection bottom Hole pressure of the solution configuration the optimization (1 injector well and 1 producer wells).

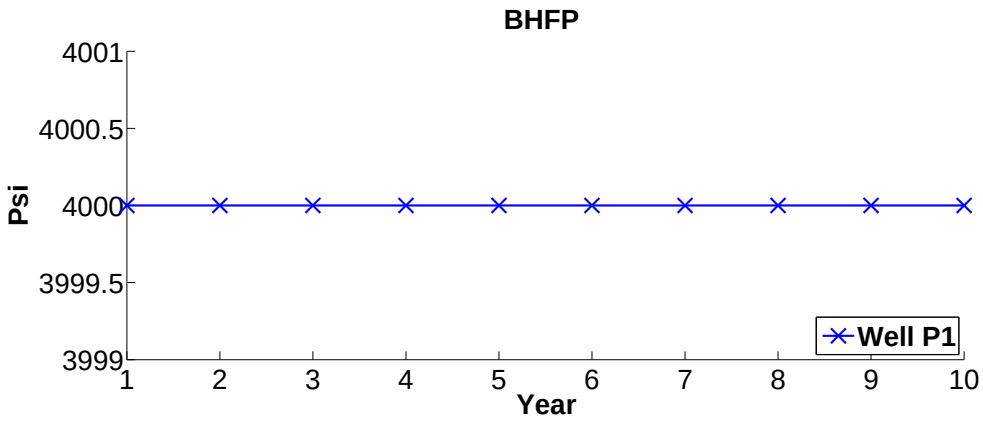

Figure B.6: Production bottom Hole pressure of the solution configuration the optimization (1 injector well and 1 producer wells).

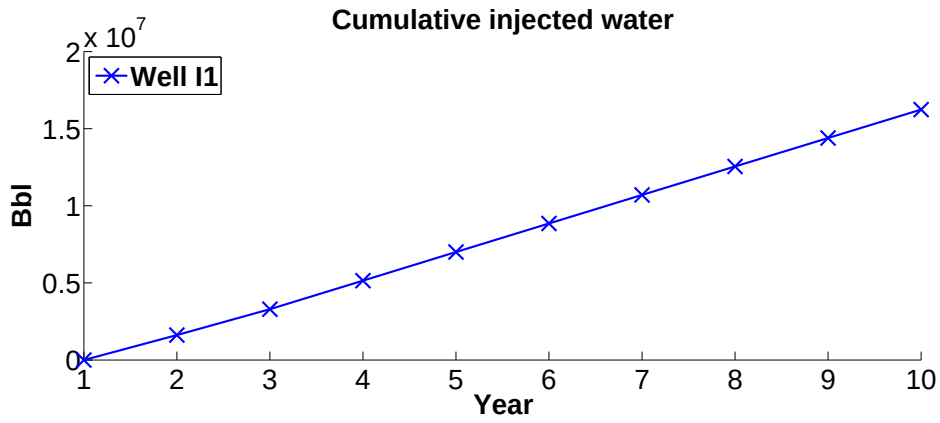

Figure B.7: Cumulative water injection of the solution configuration the optimization (1 injector well and 1 producer wells).

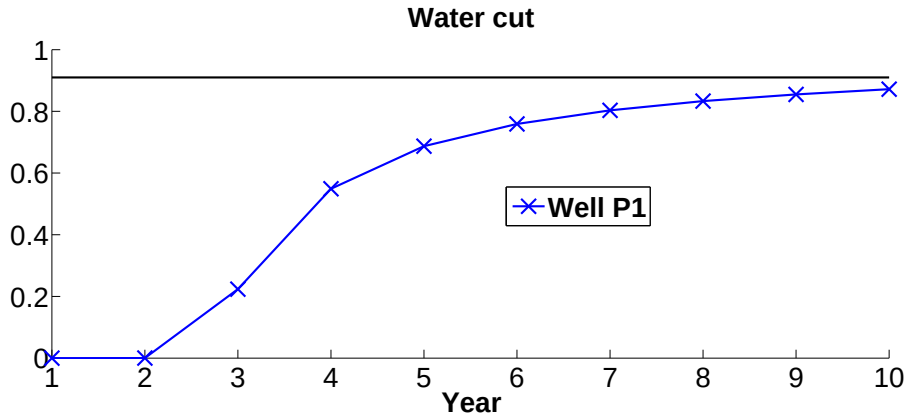

Figure B.8: Water cut for production wells of the solution configuration the optimization (1 injector well and 1 producer wells).

B.3 Reference configuration

BHFP for injector wells and producers wells are respectively displayed in FIG. B.9 and FIG. B.10. Cumulative water injection is displayed in FIG. B.11, and FIG. B.12 represents Water cut for producer wells. We can observe in FIG. B.11 that the water constraint was not satisfied for well $P4$ (in black) after 8 years, and for well $P2$ (in green) after 9 years of production. Thus the BHFP is constant for producer wells $P1$ (in blue) and $P3$ (in red) in FIG. B.9.

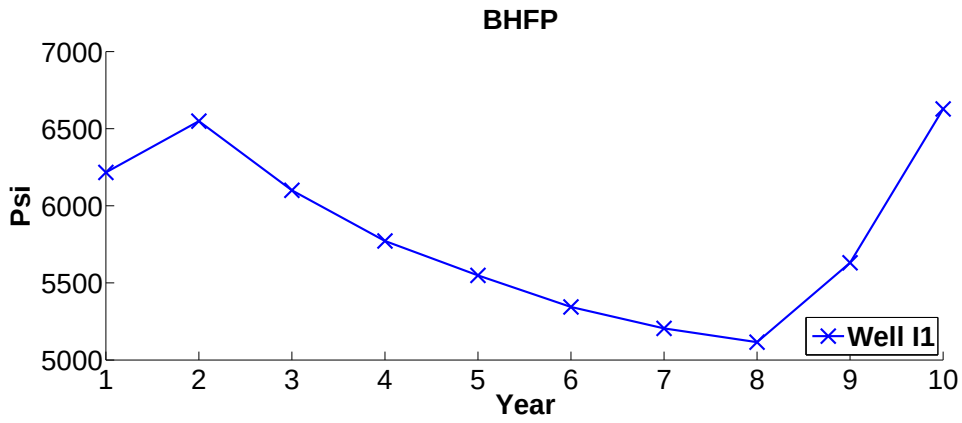

Figure B.9: Injection bottom Hole pressure of the reference configuration the optimization (1 injector well and 4 producer wells.)

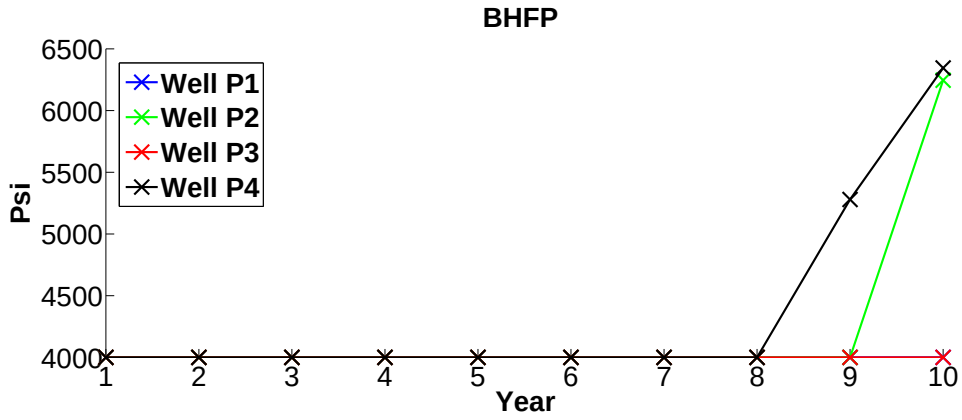

Figure B.10: Production bottom Hole pressure of the reference configuration the optimization (1 injector well and 4 producer wells).

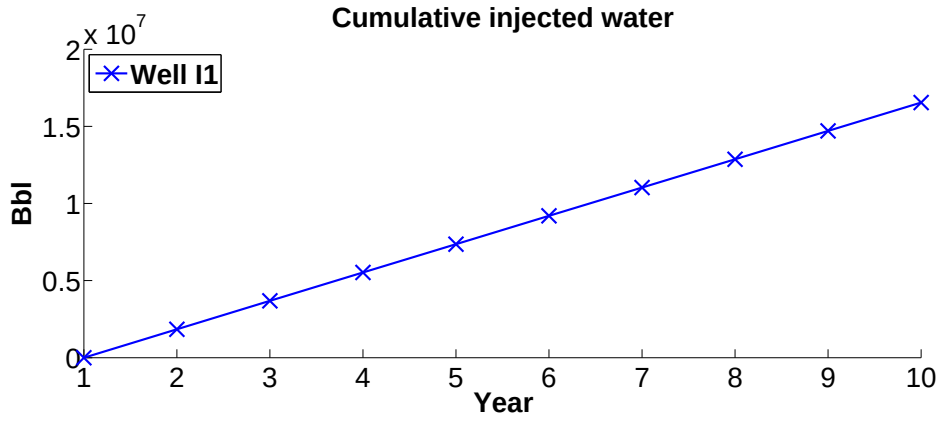

Figure B.11: Cumulative water injection of the reference configuration the optimization (1 injector well and 4 producer wells).

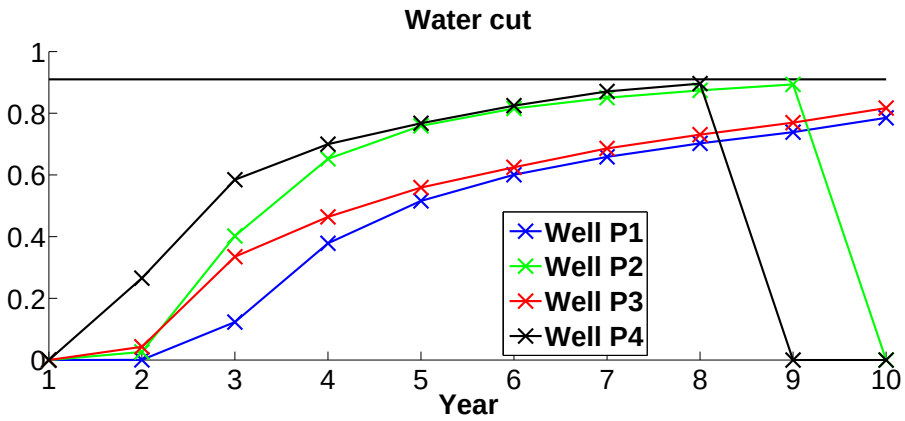

Figure B.12: Water cut for production wells of the reference configuration the optimization (1 injector well and 4 producer wells).

Appendix C

Kriging model evaluation on NPV function

C.1 Kriging model of the NPV function with respect to quantitative parameters

To evaluate the efficiency of the kriging method, we tested the method on the 1D reservoir model presented in Section 1.4.1 for different configurations in terms of number and type of wells. Models are built for 2 and 3 wells, with at least one injector well, and one producer well.

Models are built with Gaussian correlations, and constant regression functions. We used different sizes of samples to build the kriging models to evaluate the accuracy of the model depending of the sizes of the experimental plan. After building the samples for the models and the tests, to respect the constraint of distance between wells, we had to reject some points which do not respect this constraint. So, the actual size of the sample is not the initial size.

To evaluate the models, we use two values, Q2 and RMSE (Root Mean Square Error) defined in Section 2.2.2. We visualize errors by representing crossplots: we represent the values of the model and the observed points.

C.1.1 Test on the NPV function with 2 wells: 1 producer, 1 injector

In this section the kriging model presented in Section 2.2.1 is tested on the NPV function associated with 1D reservoir model with 2 wells, 1 producer well and 1 injector well. We built kriging model from LHS of size 50, 75 and 100, and evaluate their quality on a LHS of size 30 and a grid of size 10×10.

In Fig. C.1, Fig. C.3 and Fig. C.5 we represented crossplots and indicated the values of Q2 and RMSE for kriging model built from a LHS respectively of size 50, 75 and 100 and tested on LHS of size 30. In Fig. C.2, Fig. C.4 and Fig. C.6 we represented crossplots and indicated the values of Q2 and RMSE for kriging model built from a LHS respectively of size 50, 75 and 100 and tested on a grid of size 10×10. By observing the crossplots, we can see that the models estimations are more precise on the LHS than on the grid. This can be explained by the grid structure that give several configuration having wells with a small distance between them, and because of the discretization model, when two wells are closed some discontinuities in NPV function can be observed.

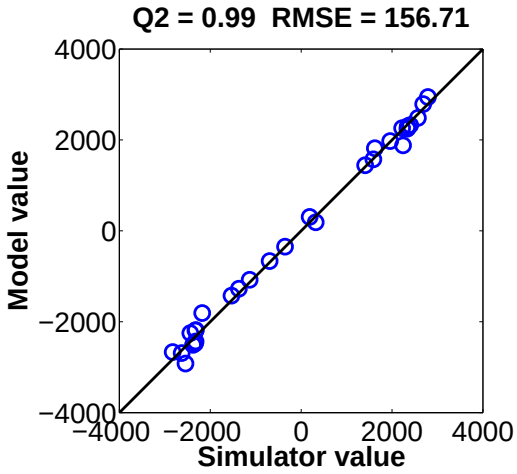

Figure C.1: Crossplot, model built with an LHS sample n = 50, tested on a LHS.

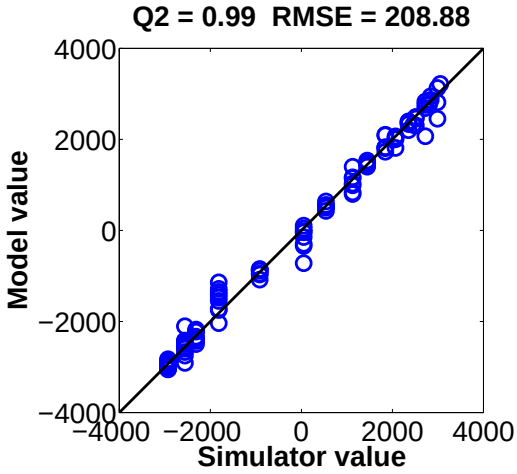

Figure C.2: Crossplot, model built with sample n = 50, tested on a grid.

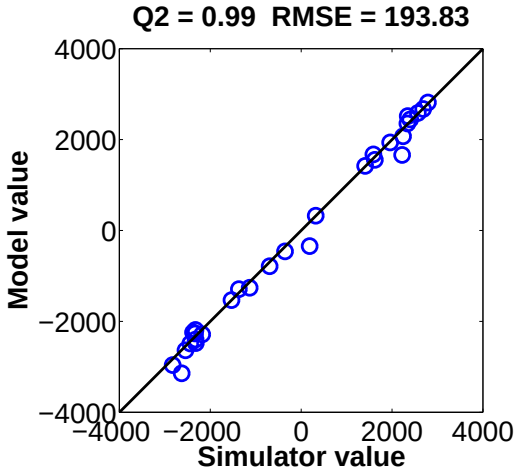

Figure C.3: Crossplot, model built with sample n = 75, tested on a LHS.

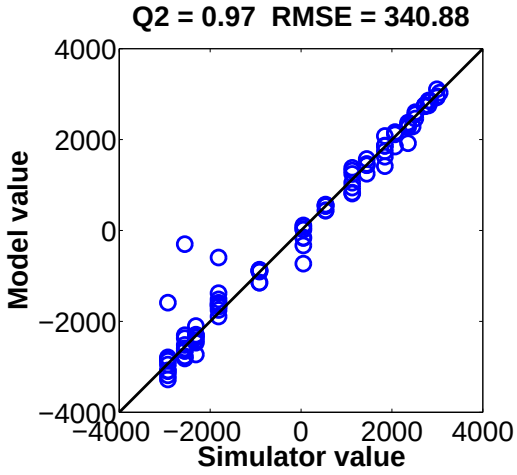

Figure C.4: Crossplot, model built with sample n = 75, tested on a grid.

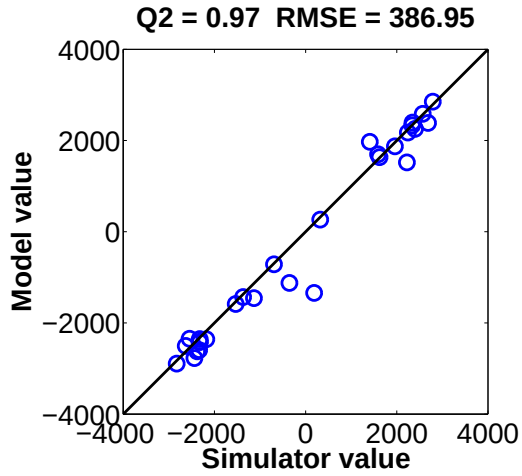

Figure C.5: Crossplot, model built with sample n = 100, tested on a LHS.

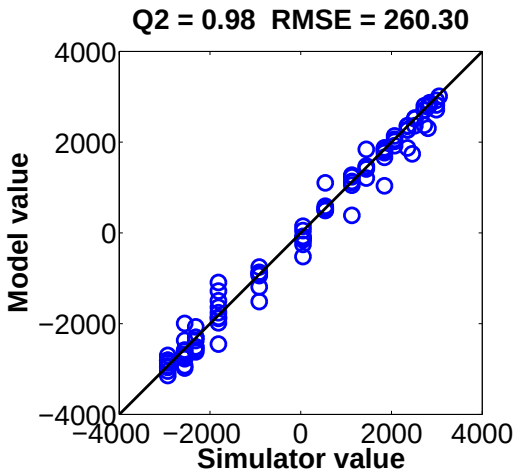

Figure C.6: Crossplot, model built with sample n = 100, tested on a grid.

In Fig. C.7, Fig. C.9 and Fig. C.11, we represented histograms of residual of kriging models built from LHS, respectively of size 50, 75 and 100, for each point of LHS of size 30. In Fig. C.8, Fig. C.10 and Fig. C.12, we represented residual of kriging models built from LHS, respectively of size 50, 75 and 100, for each point of grid of size 10×10.

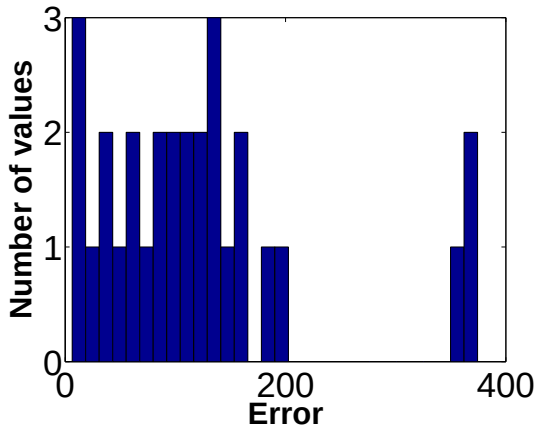

Figure C.7: Residual, model built with an LHS sample n = 50, tested on a LHS.

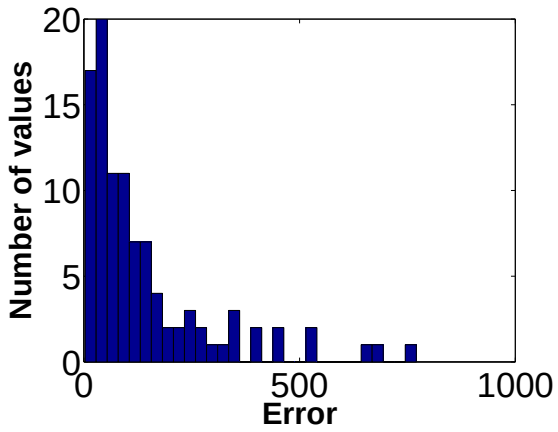

Figure C.8: Residual, model built with sample n = 50, tested on a grid.

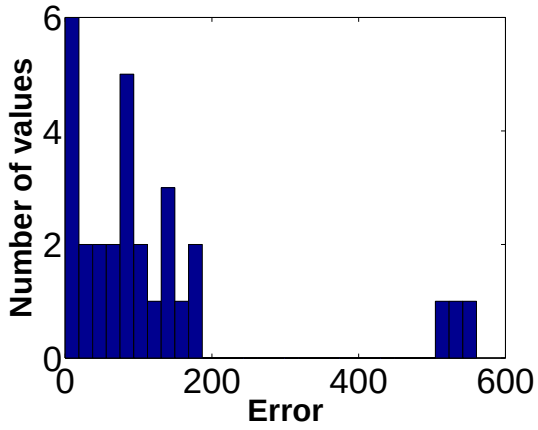

Figure C.9: Residual, model built with sample n = 75, tested on a LHS.

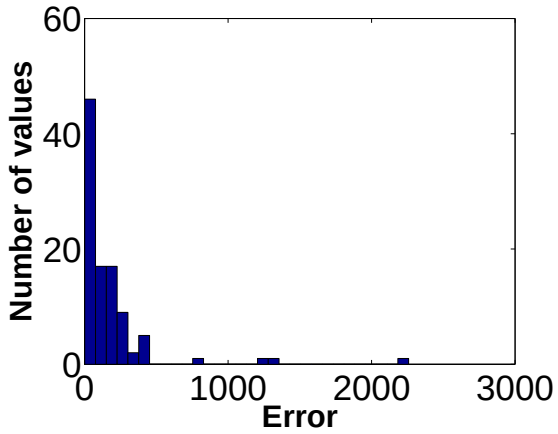

Figure C.10: Residual, model built with sample n = 75, tested on a grid.

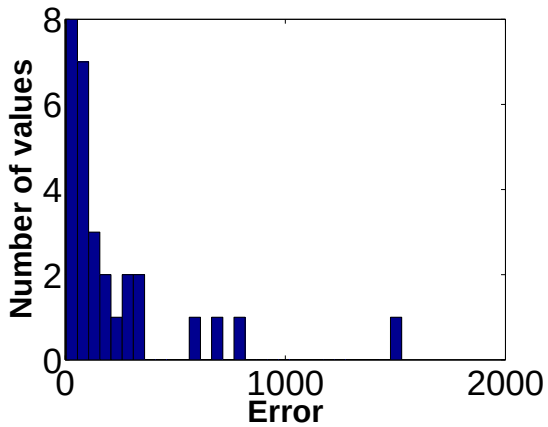

Figure C.11: Residual, model built with sample n = 100, tested on a LHS.

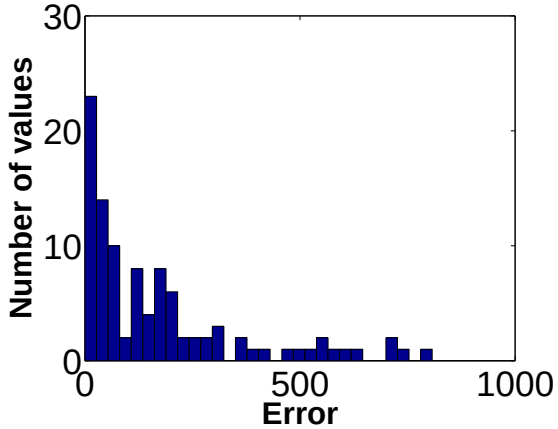

Figure C.12: Residual, model built with sample n = 100, tested on a grid.

Histograms show acceptable residuals, and we can see that, for models built from sample of size 50 and 75, the estimations on the LHS are better than estimations on the grid. Estimated values on the LHS and grid with the model built from a sample of 100 points have similar results, with most of the residuals smaller than 800.

Table C.1: Q2 and RMSE with respect to the size of the sample, test on a LHS

size of sample	50	75	100
Q2	0.994	0.991	0.965
RMSE	156	193	386

Table C.2: Q2 and RMSE with respect to the size of the sample, test on a grid.

size of sample	50	75	100
Q2	0.989	0.972	0.984
RMSE	208	340	260

In Tab. C.1 and Tab. C.2, we can see that using a large number of observed points does not give better results on the Q2 value. For the 3 models, Q2 results are good. In Tab. C.2, we can see that RMSE for the model built from 100 points is larger than the RMSE for the model built from 50 points, this is explained by the residuals of the models, we can see in Fig. C.7 that residuals of the model built from 50 points are smaller than the residuals of the model built from 100 points in Fig. C.11. Similarly, in Tab. C.2, model built from 75 points give an RMSE larger than the model built from 50 points.

In Fig. C.13, Fig. C.14 and Fig. C.15, we represented in green the NPV function of the kriging points (that were used to build the model). The models were tested on a LHS of size 30, and we represented in red the real NPV value of these points, and in blue the estimation of the model. We can see that there is more difference between the true function and the model when the two wells values are closed from the boundaries 0, due to boundary irregularities, and when the two values are similar due two the discontinuities of the NPV discretization model.

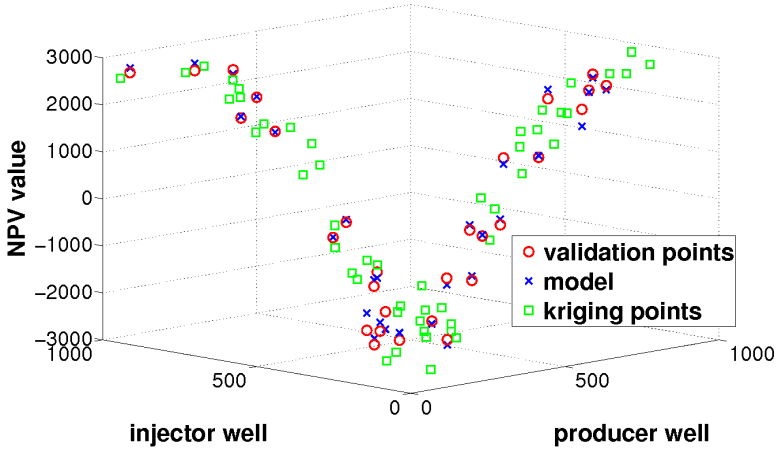

Figure C.13: Simulated and modeled points of LHS 30, and kriging points of LHS 50.

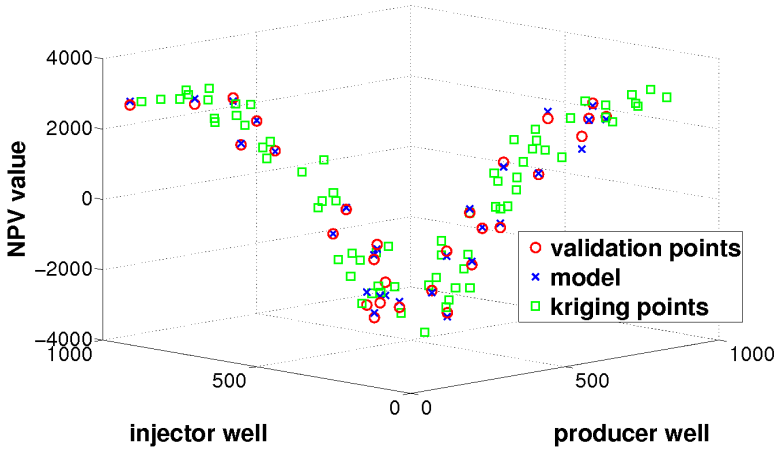

Figure C.14: Simulated and modeled points of LHS 30, and kriging points of LHS 75.

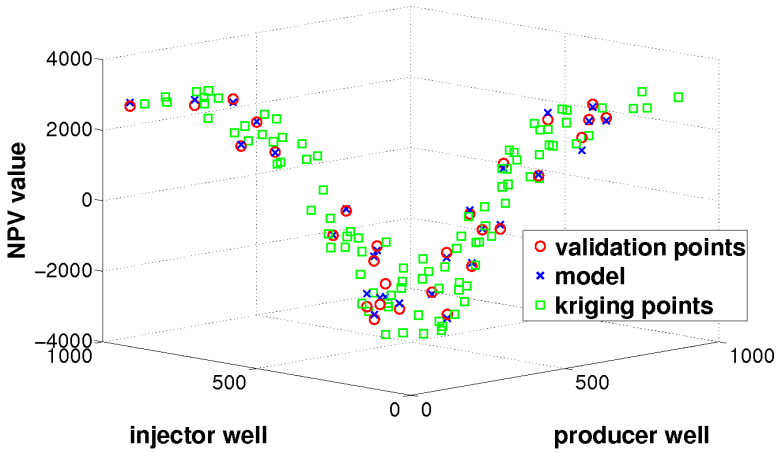

Figure C.15: Simulated and modeled points of LHS 30, and kriging points of LHS 100.

In Fig. C.16, Fig. C.17 and Fig. C.18, we represented in green the NPV function of the kriging points (that were used to build the model). The models were tested on a grid of size 10×10 and we represented in red the real NPV value of these points, and in blue the estimation of the model. In these three figures we can see that the red surface representing the true function and the blue one representing the estimation are of same shape, this show us that the model has a very similar trend to the NPV function. More irregularities are shown in Fig. C.18, probably due to errors of estimation on the boundaries. The 1D reservoir is considered as "closed" on boundaries, so the trend of the NPV function on boundaries is different.

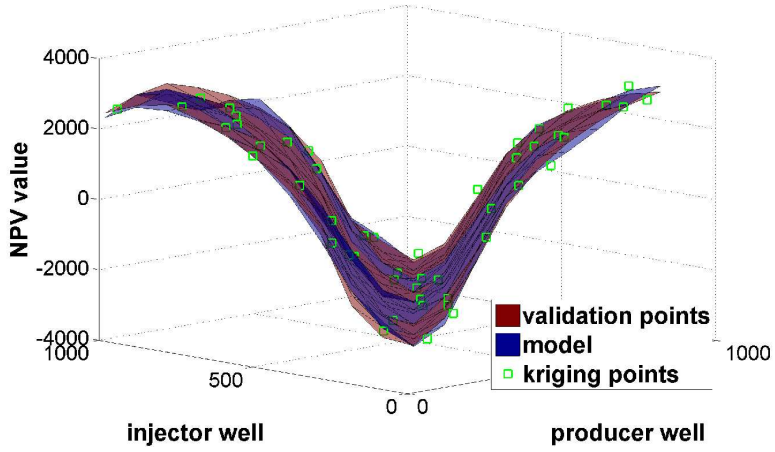

Figure C.16: Simulated and modeled points of grid 100, and kriging points of LHS 50.

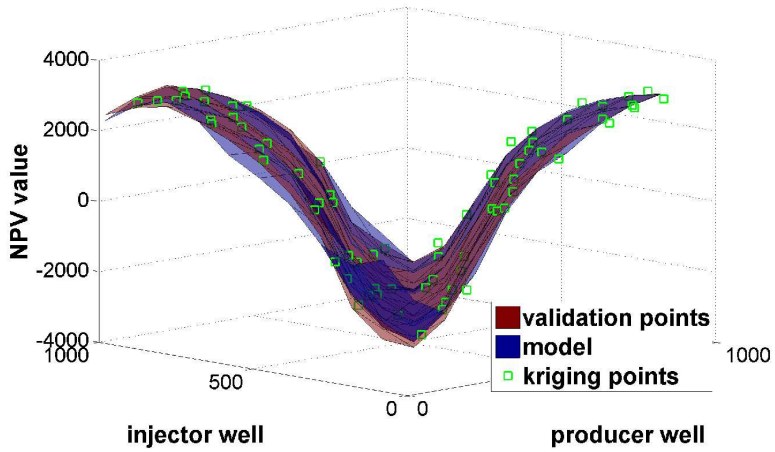

Figure C.17: Simulated and modeled points of grid 100, and kriging points of LHS 75.

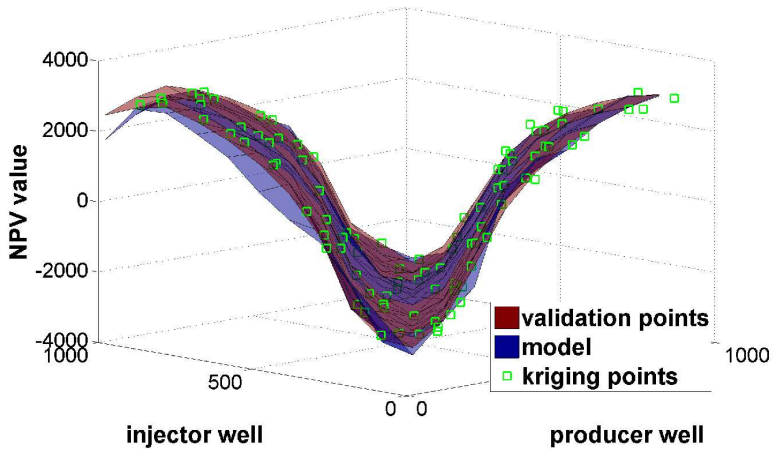

Figure C.18: Simulated and modeled points of grid 100, and kriging points of LHS 100.

Tests on the LHS and the grid gave good results, with some irregular points, but the

models built for two wells give a good indication of the behavior of the function, and a large number of evaluations is not necessary to obtain good results.

C.1.2 Test on the NPV function with 3 wells

In this section the kriging models presented in Section 2.2.1 are tested on the NPV function on reservoir model 1D with 3 wells. Two configurations of wells are tested, 1 producer well and 2 injector wells, and 2 producer wells and 1 injector well.

C.1.2.1 NPV function with 3 wells, 1 producer, 2 injector

In this section we test models for a well configuration of 1 producer well and 2 injector wells. We built kriging model from LHS of size 100, 150 and 200, and evaluate their quality on a LHS of size 30. In Fig. C.19, Fig. C.21 and Fig. C.23 we represented crossplots and indicated the values of Q2 and RMSE for kriging model built a LHS respectively of size 50, 75 and 100 and tested on LHS of size 30. Crossplots of models build with LHS of size 100 and 150 show us that these models are less precise than the model built with 200 points. In Fig. C.20, Fig. C.22 and Fig. C.24, we represented residual of kriging models built from LHS, respectively of size 50, 75 and 100, for each point of LHS of size 30. We can observe that the biggest error of estimation is more than 2000 for the model of 100 points, more than 1500 for model of 150 points, and less of 1000 for model of 200 points.

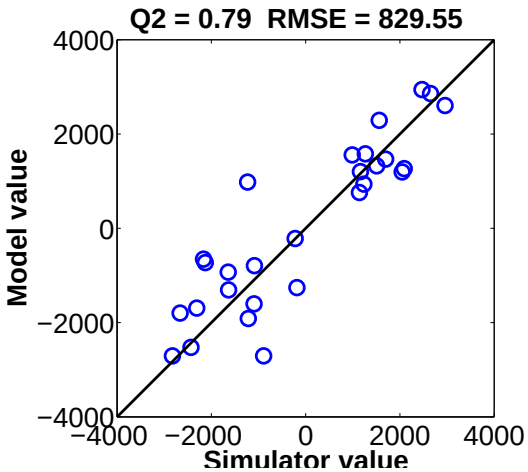

Figure C.19: Crossplot, model built with sample n = 100.

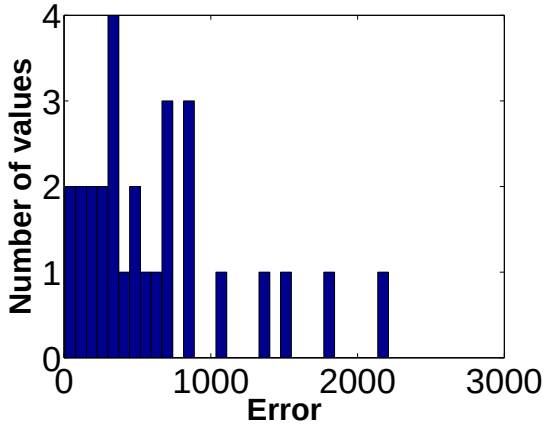

Figure C.20: Residual, model built with sample n = 100.

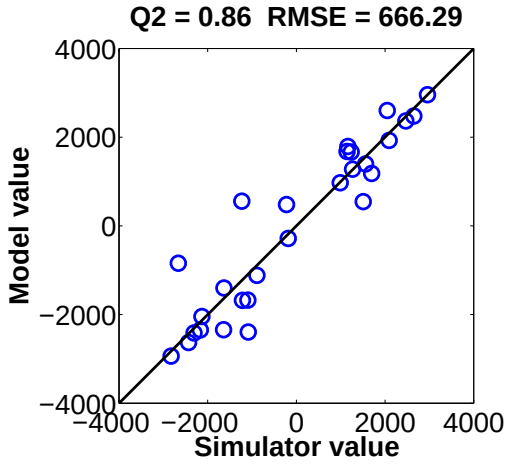

Figure C.21: Crossplot, model built with sample n = 150.

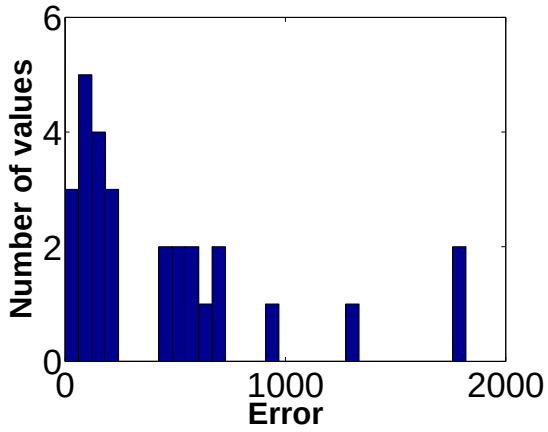

Figure C.22: Residual, model built with sample n = 150.

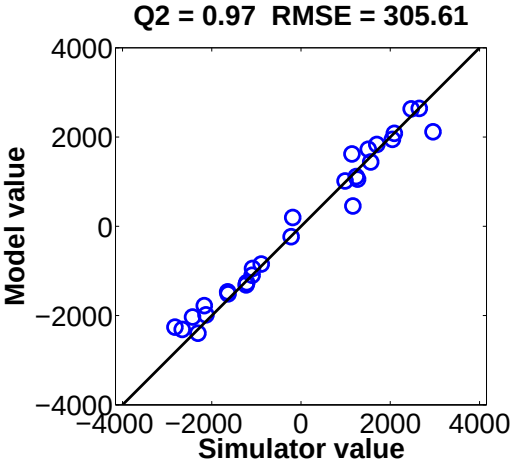

Figure C.23: Crossplot, model built with sample n = 200.

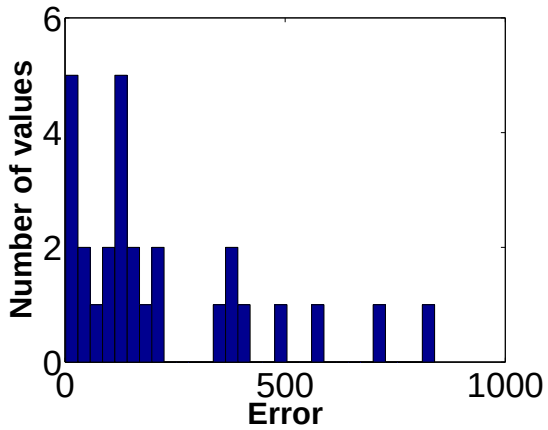

Figure C.24: Residual, model built with sample n = 200.

Table C.3: Q2 and RMSE with respect to of the size of the sample (1 producer, 2 injector).

size of sample	100	150	200
Q2	0.79	0.86	0.97
RMSE	829	666	305

We can see in Tab. C.3 the results of Q2 and RMSE. Q2 results are better when using a larger number of points, and residuals are smaller. The most accurate model is obtained with a sample of 200 points.

In Fig. C.25, Fig. C.26 and Fig. C.27 we tested model built from the LHS of size 100 on grid of size 10×10 corresponding to 1 injector well and 1 producer well, with respectively a fixed injector well of coordinates 30, 430 and 930. The red surface represents the true function, the blue surface represents the modeled points. Blue circles represent the error between the simulated and the modeled points, their size are proportional to the error. We can observe larger residuals in Fig. C.25 and Fig. C.27, it may be because of the coordinate of the fixed well, close to the boundaries (30 and 930).

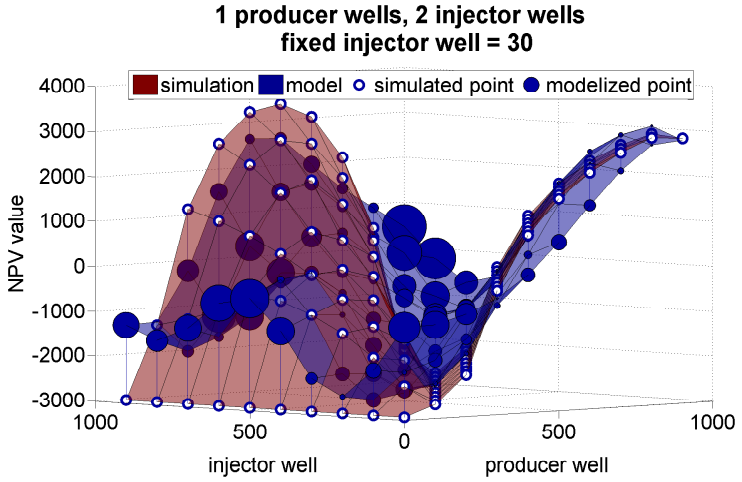

Figure C.25: Simulated and modeled points of grid 10×10, one fixed injector well with coordinate equal to 30.

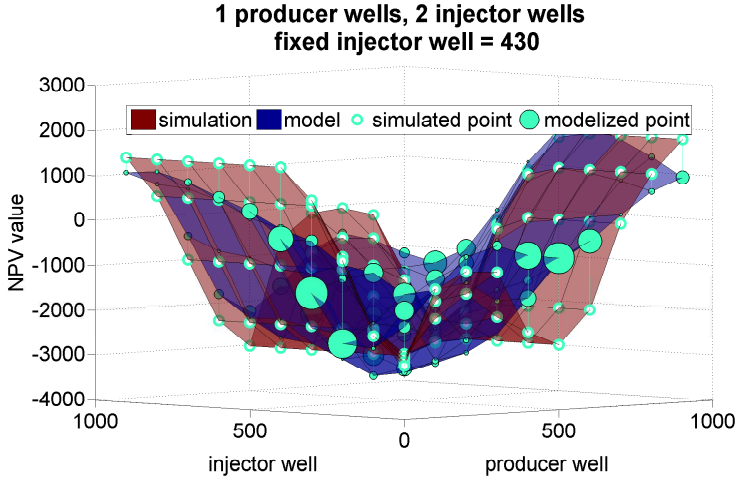

Figure C.26: Simulated and modeled points of grid 10×10, one fixed injector well with coordinate equal to 430.

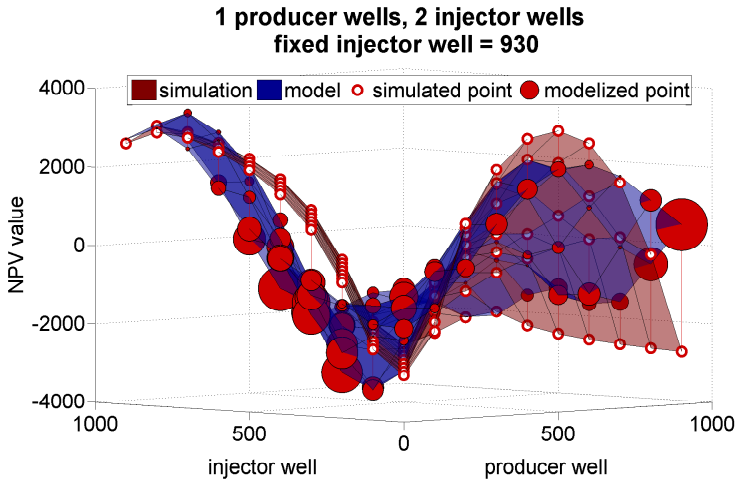

Figure C.27: Simulated and modeled points of grid 10×10, one fixed injector well with coordinate equal to 930.

In Fig. C.25, Fig. C.26 and Fig. C.27, coordinates of producer and injector vary, and we fixed the value of a second injector well, for 3 value: 30, 430 and 930. The red surface is the simulated NPV function, the blue surface is the approximated NPV function obtained with the model build with 100 points, and size of the approximated points depends of the value of the error between the two functions. We can see that they have globally the same shape, they do not fit at each point, particularly on extremities of the domain where we can observe most of largest residual.

C.1.2.2 NPV function with 3 wells, 2 producer, 1 injector

In this section we test models for a well configuration of 2 producer wells and 1 injector well. We built kriging model from LHS of size 100, 150 and 200, and evaluate their quality on a LHS of size 30. In Fig. C.28, Fig. C.30 and Fig. C.32 we represented crossplots and indicated the values of Q2 and RMSE for kriging model built a LHS respectively of size 50, 75 and 100 and tested on LHS of size 30. Crossplots show that models built with LHS of size 100 and 150 are more precise than the model built with 200 points. In Fig. C.29, Fig. C.31 and Fig. C.33, we represented residual of kriging models built from LHS, respectively of size 50, 75 and 100, for each point of LHS of size 30. We can observe that the largest error of estimation is about 4000 for the model of 100 points, about 1500 for model of 150 points. For model of 200 points, one point has a residual of 2100, and all other points have residuals smaller than 1000.

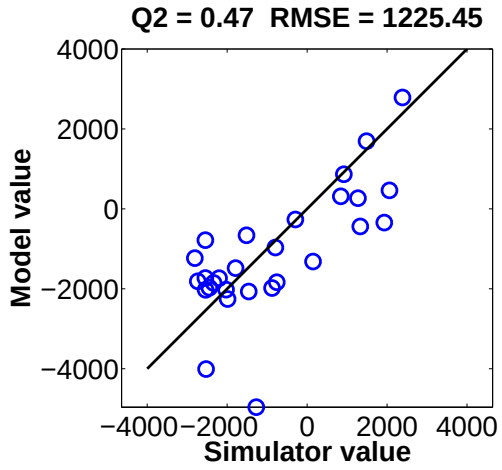

Figure C.28: Crossplot, model built with sample n = 100.

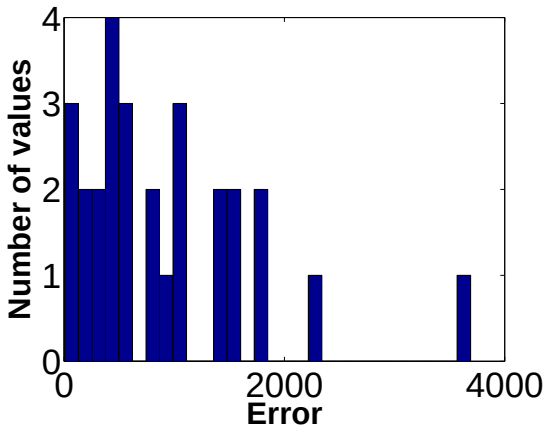

Figure C.29: Residual, model built with sample n = 100.

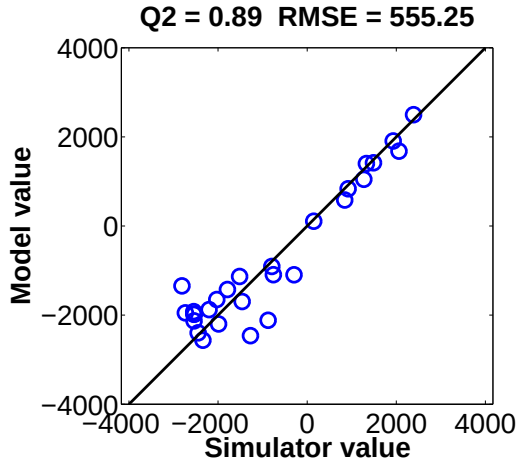

Figure C.30: Crossplot, model built with sample n = 150.

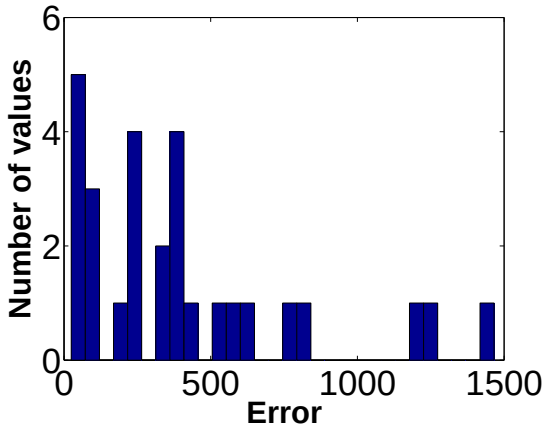

Figure C.31: Residual, model built with sample n = 150.

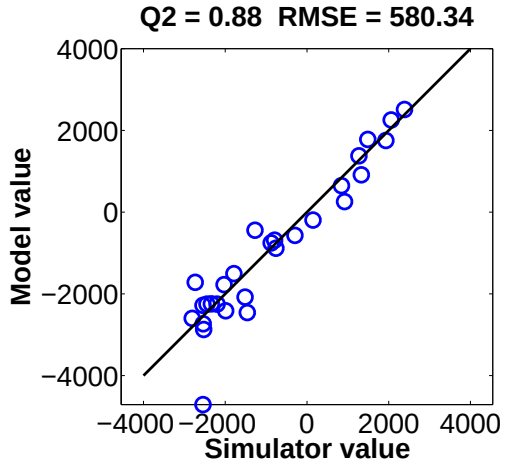

Figure C.32: Crossplot, model built with sample n = 200.

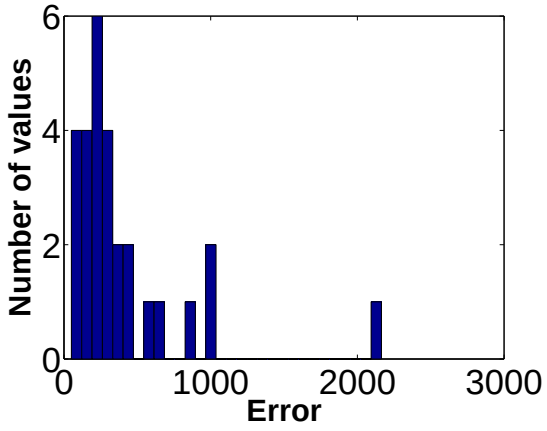

Figure C.33: Residual, model built with sample n = 200.

Table C.4: Q2 and RMSE with respect to the size of the sample (2 producer, 1 injector).

size of sample	100	150	200
Q2	0.46	0.89	0.88
RMSE	1225	555	580

We can see in Tab. C.4 the results of Q2 and RMSE. Q2 results are better when using larger number of points, but we can observe that we obtain similar results by using a sample of 150 and 200 points. By observing the residuals, we can say that the error is larger with models built from larger size. We have also see that residuals for configuration2 with 2 producer wells and 1 injector well were bigger than the residuals of the configurations with 2 injector wells and 1 producer well, this may be because of the difference of the effects of a type of well in the NPV function. colorred

C.1.3 Conclusion

In this section we evaluated the kriging model for quantitative variables NPV functions. For 2 wells, the kriging model is of good quality. In order to build a more accurate model for 3 wells configurations, we need a sample with a large number of wells, and we could see that residuals are larger when wells are on boundaries. We could also observe that the estimations were better for configuration with 2 producer wells and 1 injector well, than for configuration with 1 producer well and 2 injector wells. This is due to different effects that producer or injector wells can have on the NPV function. However large samples of points are not necessary to have a kriging model with a trend similar to the NPV function.

C.2 Kriging model of the NPV function with respect to quantitative and qualitative parameters

In this section we evaluate the efficiency of the kriging method on NPV function associated with the 1D reservoir model presented in Section 1.4.1 with qualitative variables. Two models built from sliced LHS sample of different size are used (see Section 2.2.3).

C.2.1 Sample of size $n_2 = 50$, and $n_3 = 100$

In this section, we build a model from sliced LHS sample composed of 3 concatenated LHS, one of size 50 for the configuration of two wells, and LHS of size 100 for the two configurations of 3 wells. The model is first tested on several LHS, first on the 2 wells configuration LHS, then on a 3 wells configurations LHS.

C.2.1.1 Test on a LHS associated with a 2 wells configuration

We test here a model built from 250 points on a LHS of size 30. In Fig. C.34 we represent the residuals of the estimated points. We can see in Fig. C.35 that the estimation value is not very distant from the true function.

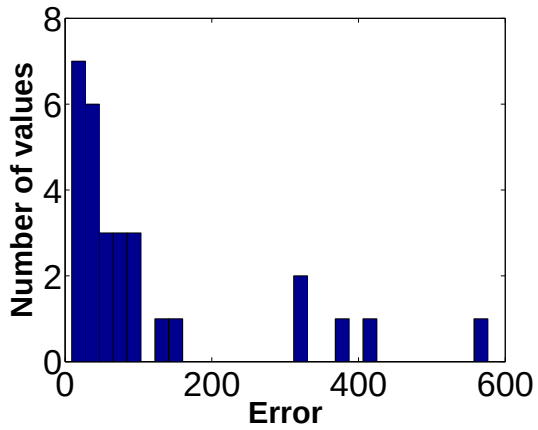

Figure C.34: Residual, model built with sample $n_2 = 50$, $n_3 = 100$, tested on the 2 wells LHS.

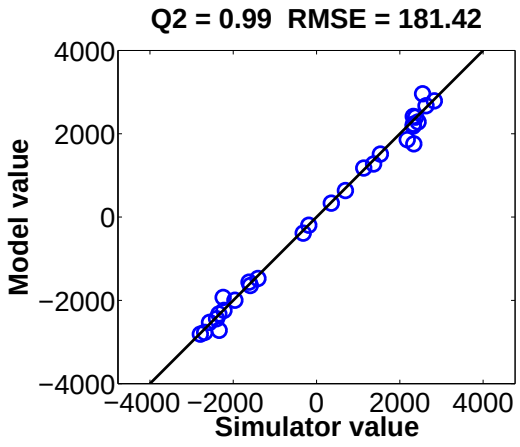

Figure C.35: Crossplot, model built with sample $n_2 = 50$, $n_3 = 100$, tested on the 2 wells LHS.

C.2.1.2 Test on a LHS associated with a 3 wells configuration (2 injector, 1 producer) and (2 producer, 1 injector)

We test here a model built from 250 points on LHS of size 75 for two configurations. In Fig. C.36 and Fig. C.38 we represent the residuals of the estimated points for re-

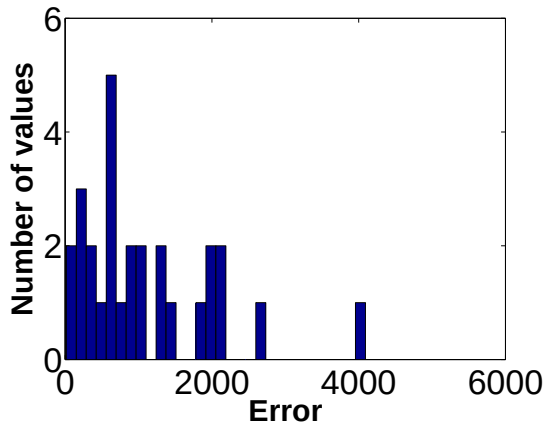

Figure C.38: Residual, model built with sample $n_2 = 50$, $n_3 = 100$, tested on the 3 wells (2 prod, 1 inj) LHS.

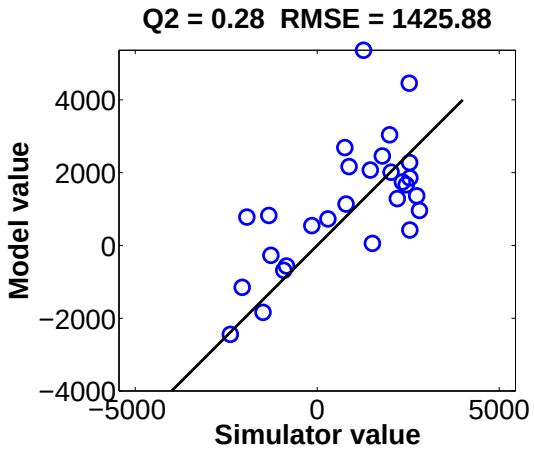

Figure C.39: Crossplot, model built with sample $n_2 = 50$, $n_3 = 100$, tested on the 3 wells (2 prod, 1 inj) LHS.

spectively configuration of two injector wells and one produced well, and configuration of one injector well and two producer wells. We can see in FIG. C.37 and FIG. C.37 that the estimation value is not very close to the true function. Errors are larger on the LHS with 2 producer wells.

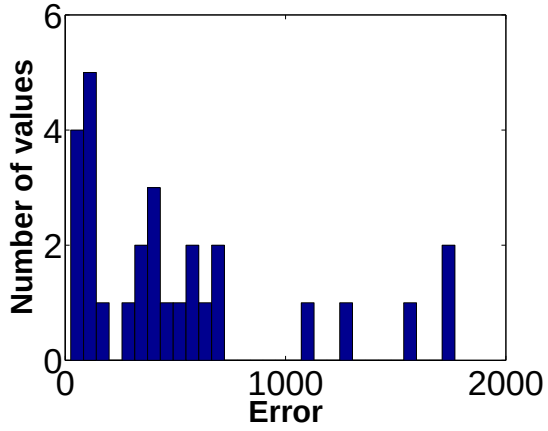

Figure C.36: Residual, model built with sample $n_2 = 50$, $n_3 = 100$, tested on the 3 wells (1 prod, 2 inj) LHS.

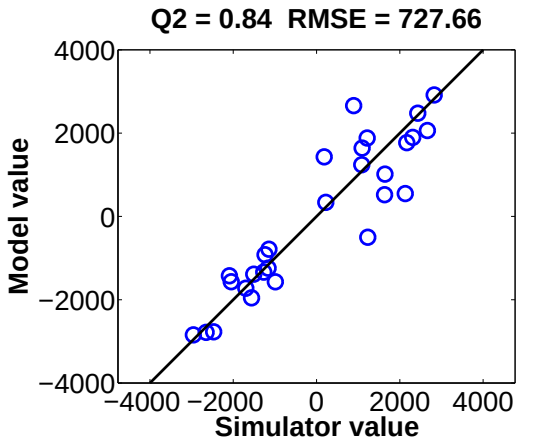

Figure C.37: Crossplot, model built with sample $n_2 = 50$, $n_3 = 100$, tested on the 3 wells (1 prod, 2 inj) LHS.

C.2.2 Sample of size $n_2 = 100$, and $n_3 = 200$

In this section, we build models from a sliced LHS sample composed of 3 concatenated LHS, one of size 100 for the configuration of two wells, and LHS of size 200 for the two configurations of 3 wells. The model is first tested on several LHS, first on the 2 wells configuration LHS, then on a 3 wells configurations LHS.

C.2.2.1 Test on a LHS associated with a 2 wells configuration

We test here a model built from 500 points on a LHS of size 30.

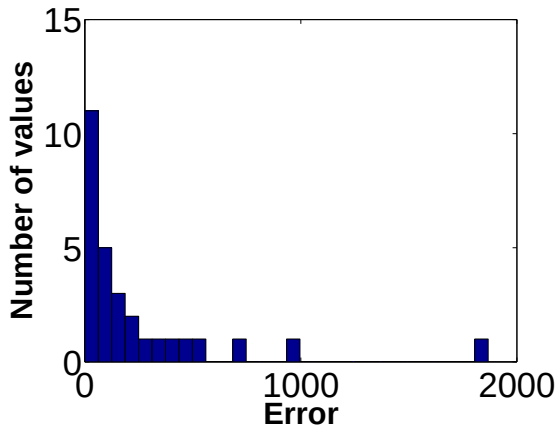

Figure C.40: Residual, model built with sample $n_2 = 100$, $n_3 = 200$, tested on the 2 wells LHS.

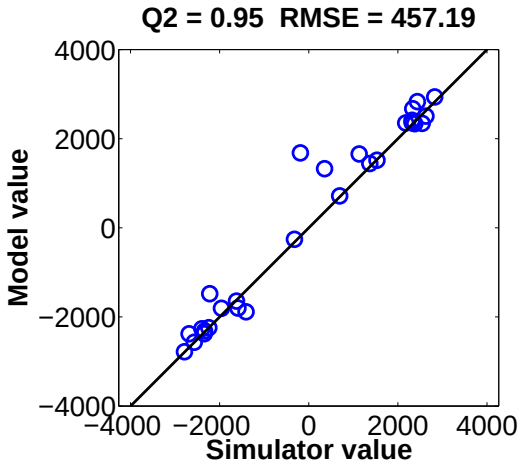

Figure C.41: Crossplot, model built with sample $n_2 = 100$, $n_3 = 200$, tested on the 2 wells LHS.

C.2.2.2 Test on a LHS associated with a 3 wells configuration (2 injector, 1 producer) and (2 producer, 1 injector)

In Fig. C.42 and Fig. C.44 we represent the residuals of the estimated points for respectively configuration of two injector wells and one produced well, and configuration of one injector well and two producer wells. We can see in Fig. C.43 and Fig. C.43 that the estimation value is not very close to the true function. Errors are bigger on the LHS with 2 producer wells.

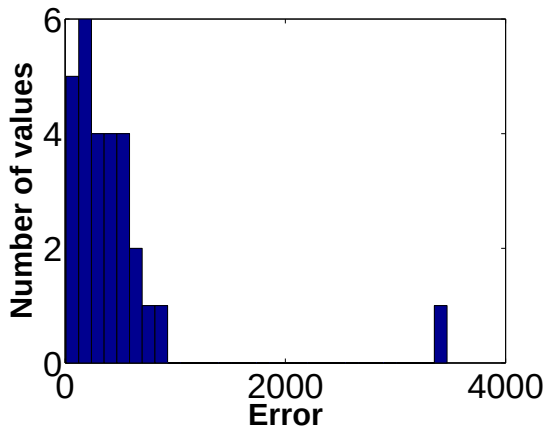

Figure C.42: Residual, model built with sample $n_2 = 100$, $n_3 = 200$, tested on the 3 wells (1 prod, 2 inj) LHS.

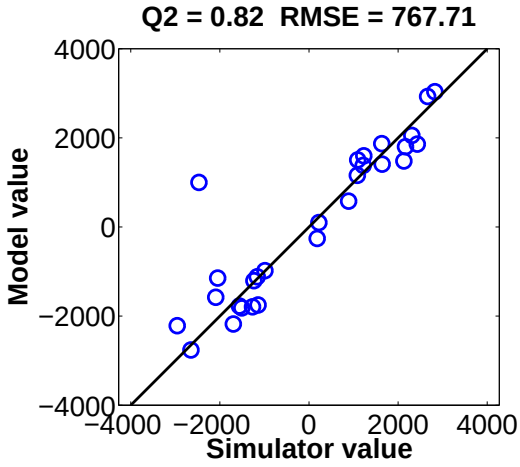

Figure C.43: Crossplot, model built with sample $n_2 = 100$, $n_3 = 200$, tested on the 3 wells (1 prod, 2 inj) LHS.

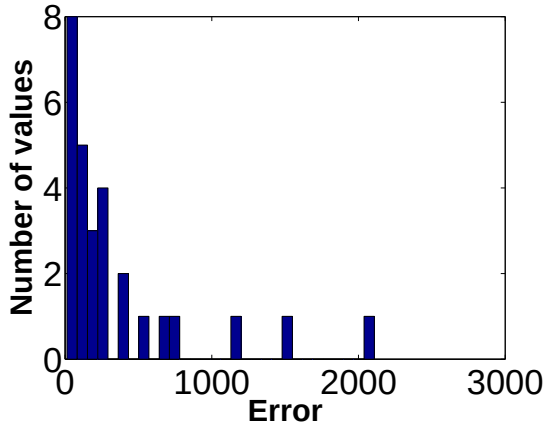

Figure C.44: Residual, model built with sample $n_2 = 100$, $n_3 = 200$, tested on the 3 wells (2 prod, 1 inj) LHS.

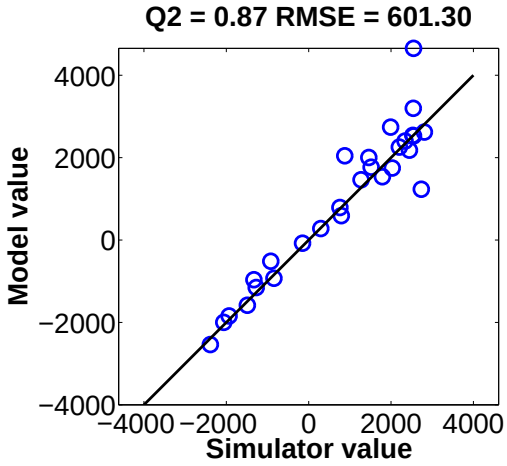

Figure C.45: Crossplot, model built with sample $n_2 = 100$, $n_3 = 200$, tested on the 3 wells (2 prod, 1 inj) LHS.

Bibliography

[1] J. E. Aarnes, T. Gimse, and K. Lie. An introduction to the numerics of flow in porous media using matlab. In Geir Hasle, Knut-Andreas Lie, and Ewald Quak, editors, *Geometric Modelling, Numerical Simulation, and Optimization*, pages 265–306. Springer Berlin Heidelberg, 2007.

[2] A. AbdulKarim, T. A. Al-Dhubaib, E. Elrafie, and M. O. Alamoudi. Overview of saudi aramco's intelligent field program. In *SPE Intelligent Energy Conference and Exhibition, Jaarbeurs, Utrecht, The Netherlands, March 23-25*. SPE, 2010.

[3] M. A. Abramson. *Pattern Search Algorithms for Mixed Variable General Constrained Optimization Problems*. PhD thesis, Rice University, 2002.

[4] M. A. Abramson. Mixed variable optimization of a load-bearing thermal insulation system using a filter pattern search algorithm. *Optimization and Engineering*, 5(2): 157–177, 2004.

[5] M. A. Abramson, C. Audet, and J. E. Dennis Jr. Filter pattern search algorithms for mixed variable constrained optimization problems. *Pacific Journal of Optimization*, 3(3):477–500, 2007.

[6] M. A. Abramson, C. Audet, J. W. Chrissis, and J. G. Walston. Mesh adaptive direct search algorithms for mixed variable optimization. *Optimization Letters*, 3 (1):35–47, 2009.

[7] T. Achterberg. Scip: solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[8] S. Alarie, C. Audet, V. Garnier, S. Le Digabel, and L. A. Leclaire. Snow water equivalent estimation using blackbox optimization. *Pacific Journal of Optimization*, 9(1):1–21, 2013.

[9] E. Aliyev and L. J. Durlofsky. Multilevel field-development optimization using a sequence of upscaled models. *Society of Petroleum Engineers Journal*, 2015.

[10] C. Audet and J. E. Dennis Jr. Pattern search algorithms for mixed variable programming. *Siam J. Optim.*, 11(3):573–594, 2001.

[11] C. Audet and J. E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *Society for Industrial and Applied Mathematics*, 17:188–217, 2006.

[12] M. C. Bellout, D. Echeverría Ciaurri, L. J. Durlofsky, B. Foss, and J. Kleppe. Joint optimization of oil well placement and controls. *Computational Geosciences*, 16(4): 1061–1079, 2012.

[13] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tighteningtechniques for non-convex minlp. *Optimization Methods Software*, 24(4-5): 597–634, August 2009.

[14] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 5 2013.

[15] G. Bilchev and I. C. Parmee. The ant colony metaphor for searching continuous design spaces. *Lecture Notes in Computer Science*, 993:25–39, 1995.

[16] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.

[17] Z. Bouzarkouna, D. Y. Ding, and A. Auger. Using evolution strategy with meta-models for well placement optimization. *CoRR*, abs/1011.5481, 2010.

[18] 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, 2012.

[19] S. Burer and A. N. Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97 – 106, 2012.

[20] D. Busby and E. Sergienko. Combining probabilistic inversion and multi-objective optimization for production development under uncertainty. In *ECMOR European Conference on the Mathematics of Oil Recovery,12th, Oxford, UK, 6-9 september 2010*, 2010.

[21] D. Busby, S. Da Veiga, and S. Touzani. A workflow for decision making under uncertainty. *Computational Geosciences*, 18(3-4):519–533, 2014.

[22] Y. Chang, Z. Bouzarkouna, and D. Devegowda. Multi-objective optimization for rapid and robust optimal oilfield development under geological uncertainty. *Computational Geosciences*, 19(4):933–950, 2015.

[23] J.-P. Chiles. Simulation of a nickel deposit: problems encountered and practical solutions. In *Geostatistics for natural resources characterisation, Part 2*, pages 1015–1039, 1984.

[24] A. Colorni, M. Dorigo, and V. Maniezzo. Distributed optimization by ant colonies. In *Proceedings of ECAL91 - European Conference on Artificial Life*. Elsevier Publishing, 1991.

[25] A. R. Conn, N. I. M. Gould, and P. L. Toint. Trust-region methods. mps-siam series on optimization. In *SIAM Philadelphia, 2000. Oxford, UK, 6-9 september 2010.*

[26] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization.* Society for Industrial and Applied Mathematics (MPS-SIAM Series on Optimization), Philadelphia, PA, USA, 2009.

[27] R. Cossé. *Techniques d'exploitation pétrolière. Le gisement.* 1988.

[28] C. D'Ambrosio and A. Lodi. Mixed integer nonlinear programming tools: an updated practical overview. *Annals of Operations Research*, 9:329–349, 2013.

[29] M. Dorigo. *Optimization, Learning and Natural Algorithms.* PhD thesis, Politecnico di Milano, Italie, 1992.

[30] D. Echeverría Ciaurri, O. J. Isebor, and L. J. Durlofsky. New approaches for generally constrained production optimization with an emphasis on derivative-free techniques. *CoRR*, 2010.

[31] D. Echeverría Ciaurri, O. J. Isebor, and L. J. Durlofsky. Application of derivative-free methodologies to generally constrained oil production optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 2(2): 134–161, 2011.

[32] A. I. Ermolaev and A. M. Kuvichko. HPC-based optimal well placement. In *ECMOR XIII-13th European Conference on the Mathematics of Oil Recovery*, 2012.

[33] R. Eymard, T. Gallouet, and R. Herbin. Finite volume methods. *Handbook of Numerical Analysis, Techniques of Scientific Computing*, 7:713–1020, 2006.

[34] M. Fischetti and A. Lodi. Local branching. *Math. Program.*, 98(1-3):23–47, 2003.

[35] 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.

[36] Fahim Forouzanfar and A.C. Reynolds. Joint optimization of number of wells, well locations and controls using a gradient-based algorithm. *Chemical Engineering Research and Design*, 92(7):1315 – 1328, 2014.

[37] R. Fourer, D. M. Gay, and Brian W. Kernighan. *AMPL: a modeling language for mathematical programming.* Pacific Grove, CA : Thomson/Brooks/Cole, 2nd edition, 2003.

[38] K. R. Fowler, J. P. Reese, C. E. Kees, J. E. Dennis Jr., C. T. Kelley, C. T.and Miller, C. Audet, A. J. Booker, G. Couture, R. W. Darwin, M. W. Farthing, D. E. Finkel, J. M. Gablonsky, G. Gray, and T. G. Kolda. Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems. *Advances in Water Resources*, 31(5):743–757, 2008.

[39] U. M. García-Palomares, E. Costa-Montenegro, R. Asorey-Cacheda, and F. J. González-Castaño. Adapting derivative free optimization methods to engineering models with discrete variables. *Optimization and Engineering*, 13(4):579–594, 2012.

[40] R. Hooke and T. A. Jeeves. "direct search" solution of numerical and statistical problems. *J. ACM*, 8(2):212–229, April 1961.

[41] T. D. Humphries and R. D. Haynes. Joint optimization of well placement and control for nonconventional well types. *Journal of Petroleum Science and Engineering*, 126: 242 – 253, 2015.

[42] O. J. Isebor, D. Echeverría Ciaurri, and L. J. Durlofsky. Generalized field development optimization using derivative-free procedures. *Society of Petroleum Engineers Journal*, 2014.

[43] Müller J. Miso: mixed-integer surrogate optimization framework. *Optimization and Engineering*, 2015. To appear.

[44] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[45] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.

[46] M. Laguna, F. Gortázar, M. Gallego, A. Duarte, and R. Martí. A black-box scatter search for optimization problems with integer variables. *Journal of Global Optimization*, 58(3):497–516, 2014.

[47] H. Langouët, F. Delbos, D. Sinoquet, and S. Da Veiga. A derivative free optimization method for reservoir characterization inverse problem. In *ECMOR European Conference on the Mathematics of Oil Recovery,12th, Oxford, UK, 6-9 september 2010*, 2010.

[48] S. Le Digabel. Algorithm 909: Nomad: Nonlinear optimization with the mads algorithm. *ACM Trans. Math. Softw.*, 37(4):44:1–44:15, 2011.

[49] M. Le Ravalec, B. Noetinger, and L. Y. Hu. The FFT moving average (FFT-MA) generator: An efficient numerical method for generating and conditioning gaussian simulations. *Mathematical Geology*, 32(6):701–723, 2000.

[50] M. Le Ravalec, E. Tillier, S. Da Veiga, G. Enchery, and V. Gervais. Advanced integrated workflows for incorporating both production and 4d seismic-related data into reservoir models. *Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles*, 67(2): 207–220, 2012.

[51] K. Lie. *An Introduction to Reservoir Simulation Using MATLAB - User Guide for the Matlab Reservoir Simulation Toolbox (MRST).* Sintef ICT, Department of Applied Mathematics, 2014.

[52] G. Liuzzi, S. Lucidi, and F. Rinaldi. Derivative-free methods for mixed-integer constrained optimization problems. *Journal of Optimization Theory and Applications*, 164(3):933–965, 2015.

[53] C. Lizon and D. Sinoquet. Optimisation non linéaire mixte en variables entières et réelles : application au problème de placement des puits en ingénierie de réservoir. Rapport 62863, IFPEN, 2013.

[54] C. Lizon, C. D'Ambrosio, M. Le Ravalec, L. Liberti, and D. Sinoquet. A mixed-integer nonlinear optimization approach for well placement and geometry. 2014.

[55] C. Lizon, C. D'Ambrosio, and L. Liberti. Méthode non linéaire mixte en variables entières et réelles pour le placement et la géométrie des puits en ingénierie de réservoir. 2015.

[56] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard. DACE - a matlab kriging toolbox, version 2.0. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2002.

[57] G. Matheron, H. Beucher, C. de Fouquet, and A. Galli. Conditional simulation of the geometry of fluvio-deltaic reservoirs. In *Ann. Techn. Conf. and Exhib., Dallas, TX, SPE 16753*, 1987.

[58] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I-convex underestimating problems. *Mathematical programming*, 10(1): 147–175, 1976.

[59] J. Müller, C. A. Shoemaker, and R. Piché. So-mi: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. *Computers & Operations Research*, 40(5):1383 – 1400, 2013.

[60] J. Müller, C. A. Shoemaker, and R. Piché. So-i: a surrogate model algorithm for expensive nonlinear integer programming problems including global optimization applications. *Journal of Global Optimization*, 59(4):865–889, 2014.

[61] E. Newby and M. M. Ali. A trust-region-based derivative free algorithm for mixed integer programming. *Computational Optimization and Applications*, 60(1):199–229, 2015.

[62] D. S. Oliver, A. C. Reynolds, and N. Liu. Inverse theory for petroleum reservoir characterization and history-matching. In *Cambridge University Press, New York, USA*, 2008.

[63] J. E. Onwunalu and L. J. Durlofsky. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, 14:183–198, January 2010.

[64] SPE Comparative Solution Project. URL `http://www.spe.org/web/csp/datasets/set02.htm`.

[65] PumaFlow^{TM}. *PumaFlow Reservoir Simulator Reference Manual*. BeicipFranlab, 2012.

[66] P. Z. G. Qian. Sliced latin hypercube designs. *Journal of the American Statistical Association*, 107(497):393–399, 2012.

[67] P. Z. G Qian, H. Wu, and C. F. J. Wu. Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics*, 50(3):383–396, 2008.

[68] R. Rebonato and P. Jaeckel. The most general methodology to create a valid correlation matrix for risk management and option pricing purposes. *Journal of risk*, 2(2):17–27, 1999.

[69] R. Schulze-Riegert and S. Ghedan. Modern techniques for history-matching. In *9th International Forum on Reservoir Simulation, Abu Dhabi, UEA, December 9-13*, 2007.

[70] SCIP. URL `http://scip.zib.de/scip.shtml`.

[71] M. G. Shirangi and L. J. Durlofsky. Closed-loop field development optimization under uncertainty. 2015.

[72] B. Talgorn, S. Le Digabel, and M. Kokkolaras. Statistical surrogate formulations for simulation-based design optimization. *Journal of Mechanical Design*, 137(2): 21405–1–021405–18, 2015.

[73] MATLAB Reservoir Simulation Toolbox. URL `http://www.sintef.no/projectweb/mrst/`.

[74] V. Torczon. On the convergence of pattern search algorithms. *SIAM J. Optim.*, 7 (1):1–25, 1997.

[75] F. van den Berg, R. K. Perrons, I. Moore, and G. Schut. Business value from intelligent fields. In *SPE Intelligent Energy Conference and Exhibition, Jaarbeurs, Utrecht, The Netherlands, March 23-25*. SPE, 2010.

[76] L. N. Vicente. Implicitly and densely discrete black-box optimization problems. *Optimization Letters*, 3(3):475–482, 2009.

[77] A. Wächter. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2002.

[78] H. Wang, D. E. Ciaurri, and L. J. Durlofsky. Optimal well placement under uncertainty using a retrospective optimization framework. *SPE Reservoir Simulation Symposium*, 2011.

[79] M. J. Zandvliet, M. Handels, G. M. Van Essen, D. R. Brouwer, and J.D. Jansen. Adjoint-based well-placement optimization under production constraints. *Society of Petroleum Engineers Journal*, 13(4):392–399, 2008.