

AMERICAN UNIVERSITY OF BEIRUT

WELL PLACEMENT OPTIMIZATION UNDER
GEOLOGICAL UNCERTAINTY

by
AHMAD MOHAMMAD HARB


A dissertation
submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
to the Department of Mechanical Engineering
of the Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
September 2021

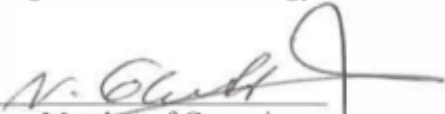
AMERICAN UNIVERSITY OF BEIRUT
WELL PLACEMENT OPTIMIZATION UNDER GEOLOGICAL
UNCERTAINTY

by
AHMAD MOHAMMAD HARB

Approved by:


Dr. Kassem Ghorayeb, Assistant Professor
Baha and Walid Bassatne Department of Chemical Engineering and Advanced Energy
American University of Beirut, Lebanon

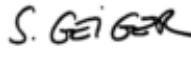
Advisor


Dr. Nissrine Ghadder, Professor
Department of Mechanical Engineering
American University of Beirut, Lebanon


Member of Committee


Dr. Mohammad Ahmad, Professor
Baha and Walid Bassatne Department of Chemical Engineering and Advanced Energy
American University of Beirut, Lebanon


Member of Committee


Dr. Sebastian Geiger, Professor
Institute for GeoEnergy Engineering
Heriot Watt University, United Kingdom


Member of Committee


Dr. Hussein Hoteit, Associate Professor
Department of Energy Resources and Petroleum Engineering
King Abdulla University of Science and Technology, Saudi Arabia

Member of Committee


Dr. Richard Torrens, Product Manager
Digital Planning Solutions
Schlumberger, United Kingdom

Member of Committee


Dr. Kamel Aboughali, Professor
Department of Mechanical Engineering
American University of Beirut, Lebanon

Chair of Committee

Date of dissertation defense: September 1, 2021

AMERICAN UNIVERSITY OF BEIRUT

DISSERTATION RELEASE FORM

Student Name: _____ Harb _____ Ahmad _____ Mohammad _____
Last First Middle

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of my dissertation; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes:

- As of the date of submission
- One year from the date of submission of my dissertation.
- Two years from the date of submission of my dissertation.
- Three years from the date of submission of my dissertation.

Ahmad Harb

September 17, 2021

Signature

ACKNOWLEDGMENTS

I was delighted to be supervised and trained by an inspiring advisor, Dr. Kassem Ghorayeb. My sincere appreciation and gratitude go to him for his incredible support and guidance throughout my doctoral study. It was a fruitful span of my life where I had a chance to develop and learn several skills including writing and communication, critical thinking, problem solving and professionalism.

I also would like to thank all Masters' students and research assistants who contributed significantly to different elements of this work. Hussein Kassem worked with me on the powerful BHPSO algorithm. Ahmad Moallem contributed significantly to the uncertainty management work. Tarek Naous and Hussein Hayek shared with me useful ideas on the work related to the well trajectory modelling and optimization.

My acknowledgements extend to all the committee members; Prof. Kamel Aboughali, Prof. Mohammad Ahmad, Prof. Sebastian Geiger, Prof. Hussein Hoteit, Prof. Nesreene Ghaddar, and Dr. Richard Torrens.

Last but not least, I would like to thank my family who gave me continuous emotional and psychological support through this period. I especially thank my mother, Nazmieh Beydoun, my father, Mohammad Harb, my siblings; Rida, Nour and Jinan, and my lovely wife and best friend Fatima.

ABSTRACT OF THE DISSERTATION OF

Ahmad Mohammad Harb for Doctor of Philosophy
Major: Mechanical Engineering

Title: WELL PLACEMENT OPTIMIZATION UNDER GEOLOGICAL
UNCERTAINTY

Investments in oil and gas projects are driven by critical field development decisions including well placement, which often significantly affect the projects' economics. Due to their typically high cost and inherently scarce data (especially in green fields, or fields in their early stage of development), managing uncertainty is critical when optimizing a field development plan. Consequently, robust field development plans require multiple geological realizations covering the range of uncertainty in reservoir properties and encompassing both multiple geological concepts and geostatistical properties distribution. On the other hand, the field development planning process involves the assessment of many development scenarios built through a combination of multiple decision parameters including, but not limited to, facility capacity, facility location, well count, well pattern, well location, well type, well trajectory, drilling schedule and operation constraints. These scenarios, when simulated over the full set of realizations, may lead to thousands of simulation runs, which are, often, practically infeasible to be conducted without an efficient automated optimization algorithm.

This work proposes a novel approach for well placement optimization under geological uncertainty. It was built on four main pillars that represent the main contributions of this work. First, we introduced a static map-based method, the black hole operator (BHO), that optimizes production and injection wells' placement based on a pre-defined well spacing (minimum distance between wells). The proposed method is systematically and thoroughly validated using a publicly available synthetic field (Olympus) that is inspired by a green oil field in the North Sea and developed for the purpose of a benchmark study for field development optimization. Results clearly illustrate the proposed method's capability of efficiently and robustly identifying optimal well placement in comprehensive scenarios including vertical and horizontal wells in pattern and peripheral water injection schemes.

In the second contribution, we introduced a new hybrid evolutionary optimization method; the black hole particle swarm optimization (BHPSO) for simultaneously optimizing well count, location, type, and trajectory. BHO was merged to the traditional particle swarm optimization (PSO) algorithm, where, for each particle in a BHPSO "iteration", the location of the first producer is identified by PSO based on a net hydrocarbon thickness map. The remaining wells (producers and injectors), whose

number is also potentially decided by PSO as an optimization parameter, are then automatically and optimally placed using BHO. The computational complexity of the proposed method is, thus, independent of the number of optimized wells. This drastically reduces the number of optimization parameters and, hence, the computational requirement to converge to an optimal solution. Validation results on the Olympus field show a systematically superior performance of the proposed BHPSO algorithm compared to the standard PSO.

In the third contribution, we proposed a new approach for managing uncertainty while employing the BHPSO algorithm. The statistical net hydrocarbon thickness (SNHCT) map was introduced to guide the BHPSO algorithm; and hence, pragmatically account for uncertainty in the process of well placement optimization. We optimize well placement on the realization corresponding to the minimum absolute difference between its NHCT map and the SNHCT map. The SNHCT combines the average and the P90 NHCT maps; hence, assuring that the selected sweet spots for well placement are statistically the best, with regard to the multiple subsurface realizations. The method is applied on the Olympus benchmark case and results are compared to two scenario reduction methods: RMfinder and k-means-k-medoids Clustering. Results show superior performance over the two methods in terms of optimality of well placement and the required computational load.

In the fourth contribution, we introduced, a novel machine learning based optimization algorithm for well trajectory design that achieved significant improvements in computational time compared to traditional optimization approaches. We used the Bézier curve to model the well trajectory then employed an optimization workflow to minimize the total well measured depth while honoring a dogleg severity constraint. We used the differential evolution (DE) optimizer to generate a large synthetic data that systematically, efficiently, and comprehensively cover the well trajectories. Three machine learning algorithms were then tested to train a model that predicts the well trajectory: artificial neural networks, support vector regression, and random forests. Using a machine learning model to design a well trajectory was three orders of magnitude faster than the differential evolution algorithm which, in turn, was the fastest among the different optimization algorithms that we have tested.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	1
ABSTRACT	2
ILLUSTRATIONS	7
TABLES	11
NOMENCLATURE	12
ABBREVIATIONS	14
INTRODUCTION	15
1.1 Literature review	18
1.1.1 Well Placement Optimization Approaches	18
1.1.2 Optimization Under Geological Uncertainty.....	31
1.2 Scope of Work	37
1.3 Dissertation Outline	39
WELL PLACEMENT OPTIMIZATION: THE BLACK HOLE OPERATOR (BHO)	41
2.1 The Black Hole Operator (BHO) Workflow	41
2.1.1 Quality Maps	41
<u>2.1.1.1 Net Hydrocarbon Thickness Map.....</u>	<u>41</u>
<u>2.1.1.2 Permeability Thickness (Kh) Map.....</u>	<u>43</u>
2.1.2 Well Placement Optimization Using the Black Hole Algorithm..	43
2.1.3 Well Placement Optimization in a Peripheral Water Injection Scheme	52
2.2 Results and Discussion – The “Olympus” Field.....	53
2.2.1 Field Description.	53
2.2.2 Pattern Water Injection.....	54
2.3 Summary	63

WELL PLACEMENT OPTIMIZATION: THE BLACK HOLE PARTICLE SWARM OPTIMIZATION (BHPSO)..... 65

3.1 BHPSO Workflow 65

 3.1.1 Particle Swarm Optimization (PSO) 65

 3.1.2 BHPSO – Black Hole Particle Swarm Optimization..... 68

3.2 Results..... 73

 3.2.1 Development Driving Value – The Optimization Objective Function
 73

 3.2.2 Validation of the Proposed Optimization Method – BHPSO vs. PSO
 74

3.3 Summary 80

WELL PLACEMENT OPTIMIZATION UNDER GEOLGOCIAL UNCERTAINTY 82

4.1 Methodology 82

 4.1.1 The Statistical NHCT map 84

 4.1.2 Workflow..... 86

 4.1.3 Rmfinder Scenario Reduction Method [12] 88

 4.1.4 k-Means – k-Medoids Clustering Scenario Reduction Method [14]. 91

4.2 Results and Discussion 93

 4.2.1 Application of the Proposed Method – The Statistical NHCT map.. 94

 4.2.2 Application of the Scenario Reduction Methods..... 97

 4.2.3 Optimization on the Selected Realizations and Comparative Analysis
 101

4.3. Summary..... 109

WELL TRAJECTORY OPTIMIZATION: A MACHINE LEARNING BASED APPROACH 111

5.1 Methodology 111

 5.1.1 Well Trajectory Modelling – The Bézier Curve..... 111

 5.1.2 Well Trajectory Optimization Using Non-Gradient Based Algorithms
 116

 5.1.3 Use Of Trained Machine Learning Models for Well Trajectory
 Design 120

5.2 Results And Discussion 125

5.2.1 Test case	125
5.2.2 Nongradient-Based Optimization Algorithms.....	126
5.2.3 Machine Learning (ML) Models	128
5.2.4 Comparison Of the ML Approach and The Optimization Approach	
135	
5.3 Summary	136
SUMMARY, CONCLUSIONS AND FUTURE WORK.....	138
6.1 Conclusions.....	138
6.2 Future Work.....	141
BIBLIOGRAPHY	143

ILLUSTRATIONS

Figure 1 Well subdivision to two sections: “Reservoir Section” from and “Well trajectory”: from drill center to reservoir section entry point (e.g., well heel in the case of a horizontal well).	19
Figure 2 NHCTmap used as main “Quality Map”	42
Figure 3 Kh map used as main “Quality Map” for peripheral injection	43
Figure 4 Illustration of well placement optimization of horizontal well azimuth - Producers.	45
Figure 5 Process through which wells drainage areas are eliminated when generating the production wells. The radius used in this elimination is $2 \times RD$	49
Figure 6 Process through which wells drainage areas are eliminated when generating the production wells in preparation for generating the injectors. The radius used in this elimination is $1 \times RD$	50
Figure 7 Illustration of the placement of the well’s horizontal section in the target layer based on the provided Horizon.	50
Figure 8 Process through which injectors are generated based on altered net-thickness map following the producers generation. The radius used in this elimination is $2 \times RD$	51
Figure 9 Illustration of the “Combined” set of two maps: NHCT map and Kh map used for well placement in a peripheral water injection scheme. Left: Olympus Upper and Right: Olympus Lower.	52
Figure 10 Olympus Upper, Pattern water injection, Realization 22. Sensitivity on the number of wells for two different horizontal section lengths.....	55
Figure 11 Olympus Upper, Pattern Injection, Realization 22, Sensitivity on Well Spacing, 12 Well, 500m	56
Figure 12 Olympus Upper, Peripheral water injection, Realization 22. Sensitivity on the number of wells for two different horizontal section lengths.....	57
Figure 13 Olympus Upper, Peripheral Injection, Realization 22, Sensitivity on Well Spacing, 12 Wells, 500m	58
Figure 14 Olympus Lower, Pattern water injection, Realization 6. Sensitivity on the number of wells for two different horizontal section lengths.....	60
Figure 15 Olympus Lower, Pattern Injection, Realization 6, Sensitivity on Well Spacing, 8 Wells, 500m	61

Figure 16 Olympus Lower, Peripheral water injection, Realization 6. Sensitivity on the number of wells for two different horizontal section lengths.....	61
Figure 17 Olympus Lower, Peripheral Injection, Realization 6, Sensitivity on Well Spacing, 8 Well, 500m	62
Figure 18 Graphical Illustration of the PSO particle velocity and position calculation. Extracted from [32]	67
Figure 19 Schematic of BHPSO workflow.....	68
Figure 20 NHCTmap and corresponding cutoff at a value of 2m.	71
Figure 21 Performance comparison between BHPSO and PSO with a swarm size of 10 particles on 2 different simulation cases (Problems 1 and 2): a) 6 vertical wells, and b) 12 horizontal wells. BHPSO performance is represented by the green curves while the standard PSO is represented by the red curves. The single simulation trials are depicted by dashed lines while their average is depicted by a bold line.	76
Figure 22 Performance comparison between BHPSO and PSO on Problem 1 and with two different swarm sizes: a) 15 particles and b) 20 particles. BHPSO performance is represented by the green curves while the standard PSO is represented by the red curves. The single simulation trials are depicted by the dashed lines while their average is depicted by the bold line.....	78
Figure 23 Performance comparison between BHPSO and PSO with a swarm size of 10 particles on Problem 1. BHPSO performance is represented by the green curves while the standard PSO is represented by the red curves.	79
Figure 24 Statistical NHCT map with $b = 1$ (a), $b = 0$ (b) and $b = 0.5$ (c).....	85
Figure 25 Summary of the proposed optimization under uncertainty workflow.....	87
Figure 26 Cross plots for a number of simulated realizations. Extracted from [36]	91
Figure 27 Good (left) vs. poor (right) selection of 10 representative realizations using the risk curve objective function. Extracted from [36].....	91
Figure 28 BHPSO performance on Olympus Upper for the selected realizations (left) and optimal well placement on different versions of the statistical NHCT map (right) – Part C. White and red well trajectories depict producers and water injectors, respectively. Green depicts the part of the trajectory starting from the platform position until the point of entry.....	97
Figure 29 NPV cumulative density function (Risk curve) of the three selected realizations by our proposed method.....	97
Figure 30 BHPSO performance (left) and optimal well placement (right) in Olympus Upper, Realization U12. White and red well trajectories depict producers and	

water injectors, respectively. Green depicts the part of the trajectory starting from the platform position until the point of entry.	99
Figure 31 10 best PSO optimization runs of the RMfinder method with a swarm size of 50 particles.	100
Figure 32 BHPSO performance on Olympus Upper for Group 1 realizations (left) and optimal well placement on NHCT map (right) – Part A. White and red well trajectories depict producers and water injectors, respectively. Green depicts the part of the trajectory starting from the platform position until the point of entry.	106
Figure 33 BHPSO performance on Olympus Upper for Group 2 realizations (left) and optimal well placement on NHCT map (right) – Part B. White and red well trajectories depict producers and water injectors, respectively. Green depicts the part of the trajectory starting from the platform position until the point of entry.	107
Figure 34 BHPSO performance on Olympus Upper for Group 3 realizations (left) and optimal well placement on NHCT map (right) – Part C. White and red well trajectories depict producers and water injectors, respectively. Green depicts the part of the trajectory starting from the platform position until the point of entry.	108
Figure 35 NPV cumulative density function (Risk curve) of the nine selected realizations by RMfinder and k-means – k-medoids methods.	108
Figure 36 NPV cumulative density function (Risk curve) of all twelve optimal scenarios: Nine from RMfinder and k-means – k-medoids methods and three from our proposed method (corresponding to $b = 0$, $b = 0.5$ and $b = 1$). Dashed and solid curves depict the results of the RMfinder and k-means – k-medoids methods and our proposed method, respectively.	108
Figure 37 Illustration of the input parameters for well trajectory design using the Bézier curve.	112
Figure 38 Third-order Bézier curves (a) with fixed C_E and varying C_S and (b) with varying C_E and fixed C_S . Extracted from [55].	113
Figure 39 3D, side and top views of a typical 3D well trajectory associated with Bézier curve input parameters.	114
Figure 40 Well trajectory optimization using nongradient – based algorithms.	116
Figure 41 Input and output features of the ML models used for the characterization of well trajectory.	123
Figure 42 3D, side, and top view of a typical 3D well trajectory associated with the machine learning input parameters.	123

Figure 43 Four horizontal wells generated at an equal Euclidean distance from center of the circle (drilling center). Illustration of the efficiency of using the combined angle as an input feature to the ML models.....	124
Figure 44 A side view of 500 horizontal wells generated randomly in 3D space.	126
Figure 45 A schematic of the synthetic wells that were generated to train the ML models.....	128
Figure 46 Architecture of the developed ANN model.....	129
Figure 47 dS (left) and dE (right) cross-plots for the training dataset.....	133
Figure 48 dS (left) and dE (right) cross-plots for the testing dataset.....	134

TABLES

Table 1 Input parameters used to calculate the objective function.	74
Table 2 Realizations ranking using our proposed method with three different values of the “b” variable: 0, 0.5 and 1.	96
Table 3 Decision variables and statistical measures (P90 and average NPV) of optimal scenarios obtained over the three selected realizations by our proposed method.	96
Table 4 10 best realizations’ combinations obtained using the RMfinder method. The best combination is marked in red.	100
Table 5 10 best realizations’ combinations obtained using the k-means – k-medoids clustering method. The best combination is marked in red.	100
Table 6 Decision variables and statistical measures (P90 and average NPV) of optimal scenarios obtained over the nine selected realizations by RMfinder and k-means – k-medoids methods. Note that Realization 15 is a common realization reducing the number of realizations from 10 to nine.	105
Table 7 Convergence criteria of the employed optimization algorithms.	127
Table 8 Numerical results achieved by the ML models for each output feature on the test set	134
Table 9 Computation time and percentage of feasibility wells achieved by the ML models. Application on a test dataset of 500 wells.	135
Table 10 Cumulative/total well length and computation time for optimizers and the combined ML and optimizer approach. Application on a test dataset of 500 wells.	136

NOMENCLATURE

Azimuth	=	Azimuth of the horizontal well
D_{\min}	=	Minimum distance between wells
H_{zL}	=	Horizontal wells with a predefined horizontal section length
IP_R	=	Injector to producer ratio
Kh	=	Permeability \times thickness
Kh_{map}	=	Kh Map
Kh_{ij}	=	Kh at Kh_{map} grid block ij
k_{ijk}	=	Permeability of grid block cell ijk
N_P	=	Number of producers
$N_{P_{\max}}$	=	Maximum number of producers
NHCT	=	Net Hydrocarbon Thickness
$NHCT_{\text{map}}$	=	<i>NHCT</i> Map
$SNHCT_{\text{map}}$	=	Statistical NHCT Map
$NHCT_{ij}$	=	NHCT at $NHCT_{\text{map}}$ grid block ij
NTG_{ijk}	=	Net-to-gross of grid block cell ijk.
N_{WI}	=	Number of water injectors
$Point_{\text{opt}}$	=	Optimal point on a map where a well (or well heel / toe) is placed
R_D	=	Well spacing (that is the radius of the disk around the wells)
SW_{ijk}	=	Water saturation and thickness of grid block cell ijk.
Z_{ijk}	=	Thickness of grid block cell ijk.
δ_{Azimuth}	=	Incremental change of the azimuth
φ_{ijk}	=	Porosity of grid block cell ijk
w	=	Inertia weight
c_p	=	Cognitive weight
c_g	=	Social weight

$P_{best(i,j)}$	=	Particle local best solution
$g_{best(i,j)}$	=	Swarm global best solution
C_S	=	Control point at starting point S
C_E	=	Control point at starting point E
\vec{t}	=	Tangent vector at any point on the trajectory
β	=	Azimuth at a point on well trajectory
ϕ	=	Inclination at a point on well trajectory
d	=	Scalar multiplier used to define the position of the control points
α	=	Incremental angle
Ψ	=	Angle at drilling center
Ω	=	Angle between drilling direction and well direction

ABBREVIATIONS

FDP	=	Field Development Plan
PSO	=	Particle Swarm Optimization
BHPSO	=	Black Hole Particle Swarm Optimization
GA	=	Genetic Algorithms
DE	=	Differential Evolution
BHO	=	Black Hole Operator
ANN	=	Artificial Neural Network
SVR	=	Support Vector Regression
RF	=	Recovery Factor
CDF	=	Cumulative Density Function
NPV	=	Net Present Value
COP	=	Cumulative Oil Production
Kh map	=	Permeability Thickness map
DLS	=	Dogleg Severity
BF	=	Brute Force Algorithm
ML	=	Machine Learning
ADAM	=	Adaptive Moment Estimation algorithm
ReLU	=	Rectified Linear Unit

CHAPTER 1

INTRODUCTION

Geoscientists and engineers target hydrocarbon reservoirs lying thousands of meters beneath the earth / seabed surface relying on a relatively limited set of data. Subsurface data include seismic data, log data, core data, fluid data, well test data and production data. Reservoir models consolidate the up-to-date available knowledge (data and interpretation) about the reservoir and are, nowadays, the single most important tool used in field development planning and reservoir management. The “one dimensional” log data—the foundation of property distribution in a reservoir model—is limited to the point of entry and requires sophisticated geological correlations and geostatistical methods for predicting the spatial distribution of the geological properties through building the “static model” of the reservoir. From the static model, a dynamic model is typically generated by incorporating “dynamic” data (e.g., well test, production, PVT, SCAL, monitoring and surveillance), calibrating the model to this data, and potentially, upscaling the grid to be pragmatically usable in the decision-making process. The dynamic model constitutes the basis for selecting the “optimal” field development plan (FDP), which maximizes the value of the project according to pre-defined driving values among multiple potentially viable development scenarios. Given the associated subsurface uncertainty, multiple subsurface realizations (models) emerge, which take into account, to a certain extent, the available static and dynamic data. To account for subsurface uncertainty, a set of equiprobable (or with varying probabilities of occurrence) geological realizations of the reservoir model are generated deterministically, or by using geo-statistical tools to account for a wide range of

possible (static and dynamic) reservoir properties distributions. The selection of the “optimal” FDP cannot, therefore, rely on a single “base-case” reservoir simulation model. It is, therefore, crucial to account for subsurface uncertainty in the field development planning process. Otherwise, decisions are likely to be sub-optimal with the associated risks and missed opportunities.

On the other hand, the field development planning process involves the assessment of many development scenarios built through a combination of decision parameters including, but not limited to facility capacity, facility location, well count, well pattern, well location, well type, well trajectory, drilling schedule and operation constraints. These scenarios, when simulated over the full set of realizations, may lead to thousands of simulation runs, which is, often, practically infeasible to conduct due to time and computational resources constraints. Hence, optimization is considered in the industry as an efficient solution for field development decision making. An optimization problem is typically characterized by an objective function that is minimized or maximized, a number of decision parameters, and a number of constraints. However, these optimization algorithms are computationally exhausting, especially with large reservoir models. To handle this prohibitively complex issue, several research efforts focused on boosting the overall field development optimization workflow. One major research theme focused on improving reservoir simulation speed (e.g., streamline-based simulation [1–3], upscaling techniques [4–6], parallel processing [7–9] and proxy modelling[10]). Other research efforts focused on developing and engaging faster optimization algorithms [11].Some other researchers worked on producing efficient uncertainty management workflows in which optimization algorithms are run on a representative subset of the geological realizations instead of the full set of realizations

[12–14]. Our ultimate objective was to develop a fast practical well placement optimizer that functions within an efficient uncertainty management workflow.

Following this objective, at the first stage of this research, we introduced a map-based method that optimizes production and injection wells' placement based on a pre-defined well spacing (minimum distance between wells). In other words, wells are automatically and optimally placed using primarily a net hydrocarbon thickness (*NHCT*) map. The $NHCT_{map}$ is updated after every well placement by eliminating a disk (black hole) of a radius defined by the well spacing. Different radii are used to account for various producers and injectors. For horizontal wells, once the heel/toe of the well is placed the method identifies the azimuth corresponding to a maximum cumulative *NHCT* value, which is typically a highly productive zone. At the second stage, we developed a novel and drastically improved evolutionary algorithm, the black hole particle swarm optimization (BHPSO) that was successful in simultaneously optimizing the well count, location, and type. The proposed algorithm couples PSO with the black hole operator (BHO). BHO undertakes a major part of the optimization load by optimizing some of the decision variables, significantly reducing the computation load through making its computational complexity independent of the number of wells involved. The third stage of this research relates to allowing this optimizer to be used through an efficient uncertainty management workflow. To achieve this, we introduced the so-called Statistical Net Hydrocarbon thickness (SNHCT) map, which was deployed for selecting a representative geological realization and for guiding the black hole particle swarm optimization (BHPSO) in the optimization process. The rationale behind employing the SNHCT map is making it possible to achieve a robust well placement configuration while running over a single representative realization. This drastically

reduces the computational effort needed to obtain an optimal result and enables the method to be feasible in real field development planning projects. The fourth and final contribution focused on introducing an efficient well trajectory optimizer that can be utilized within an integrated field development optimization framework. We employed the Bézier curves for modeling the 3D well trajectory section then run it through an optimization workflow that minimizes the length of the curve while honouring the dogleg severity constraint. We took the study one step further and formulated the problem in a novel way to fit a machine learning (ML) optimization approach which contributed to drastically reducing the required computational time for obtaining an optimal result.

1.1 Literature review

In this section, we run through the relevant work in the areas of optimization of well and well trajectory placement optimization as well as subsurface uncertainty management methods used for the selection of a representative subset of geological realizations.

1.1.1 Well Placement Optimization Approaches

Well placement is a critical aspect in field development planning as it marks a major cost component of oil and gas projects and is associated with high technical risks, especially in the deep and ultra-deep water drilling. Wells can be generally divided into two main sections as shown in **Figure 1**. The “Reservoir Section” extends from the reservoir entry point (well heel in the case of a horizontal well) to the total depth or the well’s end point, while the “Well Trajectory” extends from the drilling center to the

reservoir entry point. Generally, reservoir engineers first define the number, location and type of the reservoir sections using indepth reservoir simulation studies.

Accordingly, drilling engineers optimize the location of the drilling centers and the design of well trajectories. In most studies, these two sections are optimized sequentially. In the following sections, we review the literature of each of those sections on its own since they are thought of as separate optimization problems.

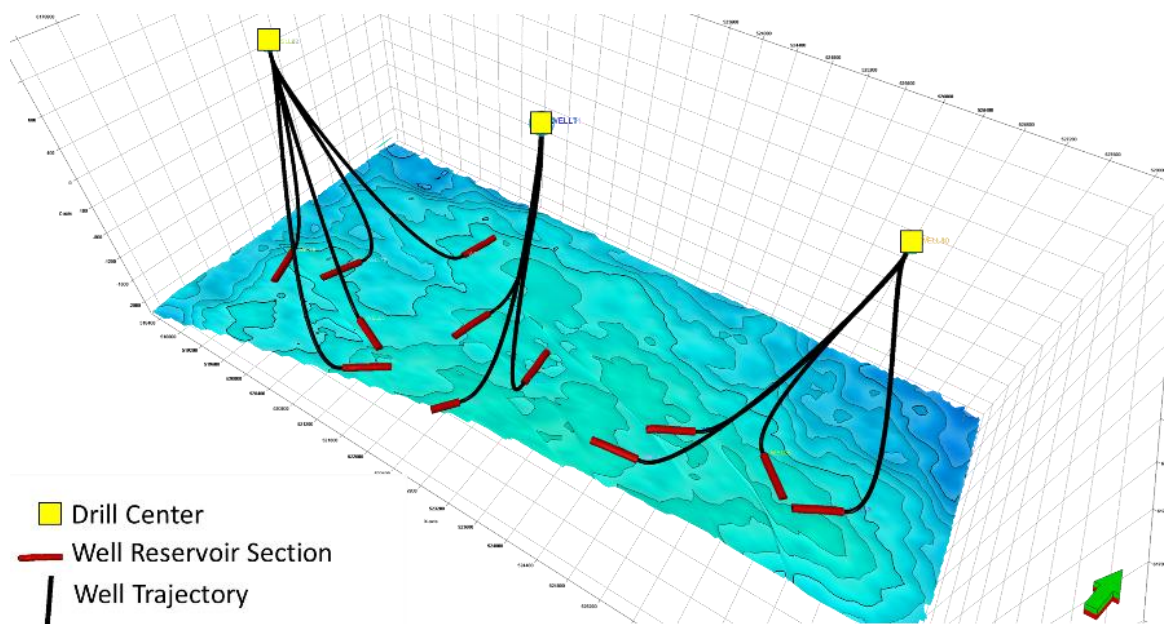


Figure 1 Well subdivision to two sections: “Reservoir Section” from and “Well trajectory”: from drill center to reservoir section entry point (e.g., well heel in the case of a horizontal well).

1.1.1.1 Optimization of Well Reservoir Section Placement – Well Placement

Optimization

The optimization of the well reservoir section placement involves a number of decision parameters including well count, type (horizontal vs vertical), location and control. Optimizing these parameters manually is often a challenging and a time demanding task. Hence, researchers have focused on developing and engaging new automated and computationally efficient optimization procedures. Nevertheless, the

computational demands of such procedures/algorithms are massive and require continuous research and development as the convergence of these algorithms may require hundreds of equivalent simulations (or generations). The number of equivalent simulations depends on the size of the search space, the number of optimization parameters, as well as the type of the algorithm used [15]. Hence, it is crucial to employ a robust automation algorithm with a fast “convergence factor” towards an efficient and comprehensive field development planning process.

Automated optimization algorithms can be generally categorized into two main types: gradient-based and gradient-free algorithms. Gradient-based optimization algorithms e.g. the conjugate gradient method and Newton’s method require the computation of gradients of the objective function. The gradients can be established either by a numerical finite difference or the adjoint method and require access to the model equations which lacks the flexibility especially when using commercial simulation software. Furthermore, such type of algorithms often fails in non-smooth problems such as well placement, where they get stuck in local optima making their solution highly dependent on their initial guess [16]. On the other hand, gradient-free algorithms, such as particle swarm optimization (PSO) [17], simulated annealing (SA) [18] and genetic algorithms (GA) [19], are global search, stochastic algorithms that operate without the computation of the objective function derivatives. These algorithms are equipped with internal random operators that reduce the likelihood of trapping in local optima, as is witnessed in gradient-based algorithms. Gradient-free algorithms are of a self-learning nature, tend to learn and adapt through their simulation experience, and, therefore, require a large number of function evaluations. Furthermore, their performance significantly depends on their algorithmic parameters which, most of the

cases, need to be tuned. Although several successful research studies related to gradient-based algorithms were conducted [20–23], gradient-free algorithms are still significantly more widely used in field development optimization, especially in well placement due to their higher efficiency in handling discrete problems [18], ease of use and their inherent adherence to commercial software.

Gradient-free algorithms were employed in various field development planning applications. Sen et al. [18] applied GA and two versions of SA algorithms to stochastic reservoir modelling, compared their performance, and showed that SA outperforms GA in terms of computational efficiency. Hardling et al. [24] applied GA for optimizing production scheduling of a group of connected oil and gas fields and demonstrated its superiority over SA. Anderson [25] combined GA with artificial neural networks (ANNs) for optimizing well control of the Brugge case and showed advantage of using such combination for solving well control optimization problems. Rameznpour and Siavashi [26] combined PSO with a local search optimizer – the pattern search (PS) – and used the hybrid method for optimizing well operation parameters in nano-fluid flooding enhanced oil recovery (EOR). This hybridization deemed effective in terms of convergence speed and efficiency in overcoming local optima. Siavashi and Yazdani [27] performed an extensive comparative study on PSO and GA for water flooding optimization in an EOR project. The authors coupled PSO with GA (PSO-GA), the newton method with PSO (NPSO) and NPSO with GA (NPSO-GA), and subsequently compared their performance. They showed that PSO performs better than GA, that both are inferior to the PSO-GA combination and that NPSO and NPSO-GA perform better than the PSO-GA combination.

In this work, we aim to significantly enhance the performance of gradient-free optimization algorithms in well placement (reservoir section) optimization. In this specific area of research, these algorithms have received substantial attention and were deemed proficient in several research studies. Bittencourt and Horne [28] introduced a hybrid algorithm by crossbreeding GA, polytope search and Tabu search for optimizing well location and trajectory in a fractured reservoirs. The algorithm was applied to a real field case and results showed a more profitable development strategy compared to that selected based on reservoir engineering best practices. Guyaguler et al. [29] introduced a hybrid genetic algorithm by coupling GA to a polytope algorithm, a kriging algorithm and neural networks, and applied it for determining the optimal location and injection rates of different numbers of vertical injectors. This hybridization showed excellence over GA in local search ability and convergence speed. Yeten et al. [30] hybridized GA with acceleration routines that include artificial neural networks, hill climbing method, and near well upscaling techniques and successfully employed it for optimizing the location, trajectory, and type of multilateral wells. The authors concluded that the optimization process is highly dependent on reservoir heterogeneity, geological uncertainty and the objective function considered. Emerick et al. [31] introduced a GA based technique for simultaneously optimizing the number, location and trajectory of producers and injectors. Population initialization was based on the quality map technique [35] and reservoir engineering best practices. They were able to obtain 20 to 30% NPV enhancement over manual optimization based on reservoir engineering best practices.

PSO was first introduced to well placement optimization by Onwunalu and Durlofsky [32] in 2010. In their work, they compared the performance of PSO to that of

GA in optimizing well location and type. Their results showed that PSO outperforms GA on average. However, they stated that each method is highly case-dependent and further work was suggested for real and large-scale problems. In another study, Onwunalu and Durlofsky [15] introduced a new well pattern optimization (WPO) procedure for sequentially optimizing well count and location in large-scale development problems. The WPO workflow comprises of two optimization steps: well-pattern-description (WPD) and well-by-well perturbation (WWP). WPD encrypts potential development solutions in terms of pattern type (e.g. nine spot, line drive) and operator. Once well pattern and count are selected, WWP entails a local perturbation of well location toward the optimal well placement. PSO was used as the core optimizer at both phases. The authors concluded that WPO results in significant improvements to NPV compared to those obtained by WPD and WWP, when considered individually. Nwankwor et al. [33] considered the hybridization of PSO with another evolutionary approach known as differential evolution (DE) for vertical well placement optimization. The hybrid particle swarm differential evolution approach (HPSDE) yielded better results than PSO and DE when considered individually. Isebor et al. [34] hybridized PSO with mesh adaptive direct search (MADS) – a local pattern search method – and applied the resulting hybrid method to optimize well count, type and drilling sequence. The approach benefits from the deterministic “local convergence” of MADS and the global optimization characteristics of the PSO and outperforms both PSO and MADS when considered separately. Ding et al. [35] introduced a modified PSO (MPSO) by adjusting the velocity equation. The selected parameters of MPSO showed a faster convergence compared to the standard PSO. The MPSO method used the quality map in order to initialize the optimization parameters. Ding et al. [36] added

one extra optimization parameter, well trajectory (vertical, horizontal, or deviated) to MPSO. Similar to their previous study (Ding et al. [40]) results showed better performance than the standard PSO. Hamida et al. [37] applied genetic similarity algorithm (GSA), a modified version of GA with an additional operator named the similarity operator, for optimizing vertical well location. GSA was applied to two synthetic reservoir models, PUNQ-S3 and Brugge, and results also showed that the proposed algorithm outperforms GA in terms of convergence rate and solution robustness. Alrashidi and Sayyazadeh [38] introduced Evolution Strategy (ES) to well placement and control optimization and compared its performance with four different algorithms: GA, star PSO, random PSO and covariance matrix adaption evolution strategy (CMA-ES). The authors concluded that ES outperforms the four mentioned algorithms in all tested cases except for the one with low dimensional control optimization problems. They also concluded that CMA-ES and PSO outperform GA in all tested cases.

The abovementioned gradient-free algorithms require running a large number of simulation models and may, in real field cases with large number of planned wells and under typically tight project timeframes, be prohibitively expensive. Engineers often need rapid methods that may not necessarily provide the most optimal solution but are capable of scanning a large set of potential development scenarios [39]. For this reason, some researchers worked on the so-called “quality maps” that indicate the high quality reservoir regions that should be targeted for optimal well placement. Da Cruz et al. [40] introduced the “quality map” which is a 2D representation of reservoir responses and their uncertainty. This map was successfully employed for ranking realizations, comparing reservoirs and managing uncertainty. Molina et al. [41]

implemented the opportunity index (OI) formula that combines three main variables; the mobile hydrocarbon volume, flow capacity in the rock and the reservoir pressure gradient. 3D OI maps were generated to identify sweet spots change with time to define good well locations for unproduced reserves. Karim et al. [42] introduced a modified version of OI, the simulation opportunity index (SOI) which is represented by the geometric average of measures of flow capacity, movable oil, and oil volume. The resulting map was successfully employed to optimize the development strategy of the tested field case. Abdy et al. [43] compared SOI to OI and concluded that both indexes were successful in identifying sweet spots and that their results were comparable. Al-Khazraji et al. [44] successfully employed OI for well placement in a heterogeneous brownfield. Chandra [45] proposed a fast map-based well placement optimizer and used a new quality factor that can be generated using reservoir's properties including porosity, permeability, pressure, hydrocarbon saturation and distance of perforation from water-oil contact and gas-oil contact. He introduced two indices that are coupled with Lagrange multiplier to produce an optimum index for well placement optimization.

Based on this extensive literature review, we conclude that quality maps are practical tools for fast scanning of potential development scenarios. We can also conclude that gradient-free algorithms are particularly suited for solving well placement optimization problems. Among gradient-free algorithms, PSO and GA are the most popular in the field. Nevertheless, these algorithms typically require an extensive number of reservoir simulation runs especially for large-scale field development plans that may involve hundreds of wells. Therefore, the employment of these methods will remain infeasible in practical field development projects. The computational load of these algorithms is mainly driven by the number of optimization parameters involved as

most conventional well placement optimization methods treat every well position as a separate decision variable.

1.1.1.2 Well Trajectory Modeling and Optimization

Traditionally, vertical wells were the only type of wells used for extracting hydrocarbon. However, as time progressed, oil and gas companies started exploring and unlocking deeper and tougher resources, which increased the drilling complexity. With the advancement in drilling technology and knowledge, engineers and researchers were able to reduce these complexities through directional drilling and horizontal and multilateral well profiles [46]. An important advantage of these non-traditional wells is that they were able to minimize surface footprint and drilling costs while maximizing hydrocarbon recovery. Such wells have an extended reach and target larger production zones which allows them to substitute numerous vertical wells considering their higher production efficiency per cost value. Despite their high production efficiency, the design of such wells is a challenging task as it involves finding out several optimum directional well design parameters. One important parameter is the maximum allowable degree of bending or what is known by Dogleg Severity (DLS). Drilling engineers must take into consideration that the higher the DLS, the larger the deviation and the smaller the well length required to intersect the target [47]. However, having high levels of DLS may lead to more frequent drilling complications including drilling string failure [46]. Therefore, having the highest possible DLS is not always the best option. A second important parameter is the kick-off point, which is the point where the vertical section of the well starts deviating to intersect the target [48]. The depth of the kick-off point is normally dependent on the geological layering of the drilling zone. For example, a

shallow kick-off point is selected when large inclinations are needed to intersect the target. In other cases, a deep kick-off point is selected in order to avoid the near-surface “soft formations” [48]. Similarly, the usage of the different well profiles present in directional and horizontal wells may be necessary in case of prohibited surface areas above the drilling target or subsurface complexities [49]. Hence, the optimization task of a well trajectory requires two main inputs (1) a mathematical model used to design the well trajectory and (2) an optimization algorithm to minimize the length/cost of the curve while honoring a predefined set of constraints e.g., DLS, torques, strain energy etc.

Various models have been presented in the literature for designing well trajectories including the tangential method, balanced tangential method, mercury method, angle averaging method, and the radius of curvature. The main difference between these models is that some adopt straight line approximations for the trajectory while others assume the well runs along a curved line and thus use curve segments approximations [50]. Amorin and Broni-Bediako [50] investigated the performance of some of these models in designing 3D well trajectories. Results showed that there is small difference in the length computation between the average angle, balanced tangential, radius of curvature and minimum curvature models. However, the authors noted that the balanced tangential, mercury, and angle averaging models can be only applied to wellbore trajectories which follow straight line course while the radius curvature can be applied to a wellbore trajectory which follows a curved segment. The main drawback of all methods mentioned above except for the radius of curvature is that they do not tackle possible conditions (azimuth and inclination) for the final end of the trajectory (the well trajectory section) [51]. Hence, some of these methods fail to

solve the problem when the direction of the reservoir well section is not in line with the drilling direction, which is highly probable in case of drilling multi wells from a single drilling center or a platform. Moreover, despite its ability to solve this problem successfully, the radius of curvature application includes a large number of decision variables [52], which makes it a complex mathematical problem.

Aiming to solve these challenges, some researchers investigated using spline and polynomial curves for representing 3D well trajectories. Scholes [53] proposed a method to generate a 3D well trajectory by representing the coordinates of each point on the trajectory by a cubic polynomial equation. The solution is obtained by establishing the coefficients of the cubic polynomial equation from the coordinates of the starting point, the azimuth and inclination of the starting point and the azimuth and inclination of the end point. Following the same path, Sampaio et al. [51] expended the cubic solution by adding different conditions such as the free-end, set-end, set-inclination and free-azimuth, and free-inclination and set-azimuth. However, the problem with applying the cubic function is that there is no control to both the curvature and length of the well trajectory [54] which leads to a single solution which may be unfeasible (i.e. not honoring of DLS or other constraints). Towards addressing this issue, Sampaio [54] introduced the spline-in-tension curves to represent well trajectories. These curves are defined in terms of hyperbolic functions which engage a new parameter called tension. The tension parameter provides some control over both the length and curvature of the planned well trajectory. To increase such control and well trajectory flexibility, Sampaio et al. [55] introduced the Bézier curves for representing 3D well trajectories which require solving a single simple parametric equation and provide two degrees of freedom due to the presence of the control points (“ C_S ” and “ C_E ”) at the starting point

and the end point of the trajectory. The mathematical formulation of these curves provides higher flexibility allowing for smooth intersection with the end target (reservoir section). Furthermore, there is no need to perform numerical approximations for obtaining the coordinates of the points constituting the well trajectory. Moreover, the Bézier curves build the well trajectory by using a unified simple mathematical expression that accounts for changes in three-dimensional space, hence making the optimization application an easier process.

The minimization of well trajectory length through mathematical optimization has also gained high interest in literature as a way to improve the economics of the drilling operation. In most cases, non-gradient based optimization algorithms were employed as they outperform gradient based algorithms in a typical non-smooth problem like the well trajectory optimization [56]. Helmy et al. [57] employed non-linear optimization theory for minimizing the length of a 2D well trajectory. They used a sequential unconstrained minimization to optimize the kick-off point, the build up rate, drop off rate, inclination, and casing depth. Targeting a faster approach, Shokir et al. [52] successfully employed Genetic Algorithms (GA) and Atashnezhad et al. [58] applied Particle Swarm Optimization (PSO) to minimize the Total Measured Depth (TMD). PSO was faster than GA, however both algorithms were susceptible to local optima which often lead to sub-optimal solutions. Mansouri et al. [59] presented a Multi-Objective Genetic Algorithm (MOGA) for minimizing two conflicting objective functions; the length of the curve and the torque on the string. Wood et al. [60] modified the basic bat algorithm by applying additional metaheuristics and applied it to a well-studied complex wellbore example. Sha et al. [61] proposed a Fibonacci Sequence based self-adjustment Quantum Genetic Algorithm (FSQGA) for 3D wellbore trajectory

optimization problems. The performance of the presented algorithm was compared to a number of modified versions of PSO and GA algorithms and showed an advantage in terms of parallel processing and fast global optima searching. Mansouri et al. [59] ran a performance comparative study between Ant Colony Optimization (ACO), GA, Artificial Bee Colony (ABC), and harmony search (HS) on optimizing well length. Results showed that ACO outperformed GA and ABC in obtaining global optima [59]. Biswas et al. [62] proposed a new hybridization of Cellular Automata (CA) technique with Grey Wolf optimization (GWO) and PSO algorithms for solving a multi-objective well trajectory optimization problem aiming to minimize the TMD and torque and strain energy. Several parametric tests (Inverted Generational Distance (IGD), Spacing metric (SP), Maximum Spread (MS)) were conducted to investigate the efficiency of the proposed hybridization and were compared to that of three other metaheuristic methods. Results showed that the proposed method obtained 50 – 60 % increase in IGD compared to other tested methods and had the minimum spacing metric and the maximum spread.

In all the above well trajectory design optimization applications, the length of the curve was generated using the Radius Curvature Method (RCM) which was proposed by [52]. This model involves minimizing the length of the well trajectory while respecting a predefined DLS constraint. Few researchers [63] focused on the application of the Bézier curve since it was presented in 2017 by [55]. One can argue that the RCM model may lead to a reduced length of curve compared to the Bézier curve approach due to the large number of control points it holds. However, when considering an integrated optimization approach, which includes multi-well and facility

placement, optimizing such complex model that hosts a large number of decision variables and constraints, might be computationally unfeasible.

1.1.2 Optimization Under Geological Uncertainty

Uncertainty can be generally classified into 1) subsurface uncertainty related to incomplete knowledge of reservoir geometry and spatial distribution of petrophysical properties, 2) operational uncertainty related to infrastructure and facility constraints, and 3) economic uncertainty related to the cost, price and market variability. We focus here on subsurface uncertainty, which results in an uncertain simulation response (e.g., production and injection rates, cumulative production and injection, and recovery factors.) and has a significant effect on the placement of wells in a reservoir. One of the most popular methods in handling uncertainty and the one we focused on is what is known in the industry by “Scenario Reduction”.

Scenario reduction is a process in which a subset of geological realizations is selected to carry out reservoir simulation in a way that its forecasted response is deemed, to a certain extent and within a prescribed tolerance, statistically representative to that obtained when simulating over the full set of realizations [64]. The cumulative density functions of both reduced and original sets have similar mean: P10, P50 and P90 values of the simulated response (e.g., NPV, COP).

Scenario reduction methods can be classified into three main categories: 1) ranking methods, 2) clustering methods and 3) optimization methods. In ranking methods, geological realizations are ranked according to predefined static or dynamic criteria, based on which “low,” “mid” and “high” realizations are selected to represent the low, mid and high response of the full set of realizations. Ranking methods were

first introduced by Balin [65] who constructed fast simulators (FS) using tracer simulation to represent the response of the comprehensive flow simulators (CS) on a water flooding example. The author ranked realizations based on the FS responses (e.g., NPV and COP), generated a cumulative density function (CDF) and selected the relevant realizations corresponding to P10, P50 and P90 of the specified objective function. While this method provided a 90% reduction in computation time, results were based on a single development scenario. Ranking realizations using the responses of a single development scenario can be misleading since different well placement scenarios likely have different responses on different geological realizations, especially when the spatial distribution of reservoir properties is highly uncertain. Deutsch [66,67] proposed ranking realizations based on geologically connected objects “Geo-Objects.” The number of geo-objects, their sizes and their tortuosity are employed for ranking and obtaining realizations with low, median, and high connectivity. McLennan and Deutsch [66] employed several static measures for ranking realizations. The authors characterized these measures into 1) statistical static measures, which are based on an average of pre-defined geological parameters, 2) fractional static measures based on a good net cells fraction of the reservoir, and 3) volumetric static measures, which are based on the calculated hydrocarbon volume. Among these measures, the ranking based on local and global connectivity was found to have the highest correlation with the ranking of the dynamic results. Fenik et al. [68] ranked and selected realizations based on connected hydrocarbon volume (CHV). CHV is a geostatistical method that calculates the connected cells within a local window around a given well for a SAGD application. Li et al. [69] introduced a modified version of the CHV measure including the calculation of the distance from each cell to the nearest production well and the

geometric average permeability of the cells between the cell and the nearest production well. The authors presented a workflow to calculate a quality measure from three-dimensional realizations of cell volume, porosity, permeability and water saturation. The modified CHV version by Li et al. [69] is more favourable than the one proposed by [68] for well placement optimization since it accounts for all wells in the field. However, the workflow of both CHV-based methods is highly dependent on prescribed well locations, which in reality is untrue as these vary as part of the optimization process. Steagal and Schiozer [70] proposed to select seven representative realizations: two realizations with responses near the P10 NPV, three realizations with responses near the P50 NPV and two realizations with responses near the P90 NPV. Results showed a good representation of the reduced set of realizations with regards to the production responses and the stock tank oil in place (STOIP). In an extension to the Steagal and Schiozer method, Schiozer [71] proposed to select similar realizations based not only on NPV estimates but also on COP, cumulative water production (CWP) and oil recovery factor (RF) estimates. The disadvantages of such an approach are that the selection is manual and time consuming, and the result of the realizations reduction is, again, based on a single development scenario. Da Cruz et al. [40] introduced the “quality map,” which is a 2D representation of the 3D reservoir model, and employed it for ranking and selecting realizations. The map is generated by running a flow simulator with a single producer and changing its position to get a representative coverage on the entire grid. The quality of a realization is calculated as the sum of the cumulative oil production from all tested locations and is used for ranking. An approximate density function is generated after ranking in which a subset of realizations is selected.

A major drawback of the ranking-based approach is that the selected realizations are allocated with equal probability [72]; that may not be a reliable solution especially when the chosen realizations “represent” a varying number of realizations within the full set. To address this issue, some researchers employed clustering techniques where a dissimilarity matrix between realizations is computed based on a pre-defined dynamic or static criterion and used to divide realizations into clusters of similar characteristics. Consequently, a probability value is assigned to each cluster based on the number of realizations corresponding to each cluster. Scheidt and Caer [73] introduced kernel k-means clustering to reduce the number of realizations. They computed the dissimilarity matrix using streamline simulation results after which multi-dimensional scaling (MDS) and Gaussian kernels were used to transform the matrix into a reduced dimensional data in Euclidean space on which k-means clustering is applied. Quantile estimations were deemed better than those estimated using traditional ranking. However, here again, the dissimilarity matrix was generated from the results of a single development scenario. Singh et al. [74] applied the kernel k-means clustering workflow to reduce the number of history-matched models in a real field application. The method was successful in reducing the original set of history-matched models to five. Wang et al. [75] introduced retrospective optimization (RO) from a well placement optimization problem. The method involves increasing the number of realizations used as the optimization run progresses. At different stages of the optimization run, k-means clustering is applied on a dissimilarity matrix based on a set of dynamic and static criteria. RO provided better results compared to the random-selection approach and a drastic decrease in the computational cost compared to optimization over the full set of realizations. Shirangi and Mukerji [76] applied RO with kernel k-medoids clustering or

well control optimization problems. In contrast to k-means clustering, k-medoids selects one element (subsurface realization in the context of this dissertation) within the cluster to represent the whole cluster, while in k-means, the cluster's centre may not be one of the elements within the cluster. Shirangi et al. [14] coupled k-means clustering to group realizations and k-medoids clustering to select the closest realization to all other realizations in the cluster. This method was applied to both well placement and control setting problems with different dissimilarity criteria, static (permeability based), dynamic (flow based) and combined. Realizations were clustered using the flow simulation results of a base case scenario and tested on a number of different development scenarios. For well control settings, the purely flow-based criterion was deemed the best, while for the well placement settings, all three combinations performed nearly the same. The disadvantage of k-means and k-medoids clustering is that they both require the number of clusters to be predefined, hence it's not possible to identify the optimal number of realizations. To overcome this problem, Liu and Forouzanfar [77] used hierarchal clustering for scenario reduction in optimizing smart wells control. Hierarchal clustering marks different level of clusters without the need of predefining the number of clusters. However, once hierarchal clustering is completed, a certain criterion needs to be selected to choose the most optimal clustering level, or in other means, the representative number of clusters. Another solution was proposed by Barros et al. [78]. The authors used the Kolmogorov-Smirnov test for defining the optimal number of clusters to be considered. A divergence factor is calculated while changing the number of clusters after which a knee-point is identified to specify the optimal number of clusters to be considered.

Apart from clustering and ranking techniques, recent studies have also focused on optimization methods to reduce the number of geological realizations. Sarma et al. [79] introduced a minmax optimization algorithm, which can concurrently select a few reservoir models from a large ensemble of models by matching target percentile of multiple output responses (e.g., matching P10, P50 and P90 of OPC, WPC and OOIP) while also obtaining maximally different models in the input uncertainty space. The algorithm was fit for purpose; however, it matched the results of a single scenario and failed to assign a specific probability value for different models. In addition, it is practically unfeasible when there is high uncertainty in the spatial distribution of properties where it is not possible to select different input models through picking low, mid and high input parameters. Rahim et al. [64] introduced a mixed-integer linear optimization-based method to select a subset of realizations through minimizing the Kantorovich distance between distributions. The proposed method considers multiple static measures and geological properties to quantify dissimilarity between realizations and uses the Kantorovich distance to quantify the probability distance between the superset and the subset of realizations. The proposed method outperformed various ranking and kernel k-means clustering method to obtain good correlation with the dynamic responses of the full set of realizations. Furthermore, Meira et al. [80] introduced the RMfinder, an optimization based approach for reducing the number of realizations. The method resembles the method proposed by [70] but instead of running the workflow manually, a greedy algorithm was introduced. Further modifications are presented in RMfinder in which an additional operator was incorporated to compute the probability for the selected realizations.

From extensive research, we conclude that clustering and optimization techniques are more favourable than ranking methods as they can be associated with probability assignments for selected realizations. In all methods, however, there is a major issue related to the choice of the type of measures. Static measures are computationally cheap but perform poorly in capturing a reliable match between the subset and the full set of realizations due to the nonlinearities between the static and dynamic properties on one hand and the dynamic response on the other hand. As for dynamic measures, they lead to a better representation but are computationally prohibitively expensive especially when it comes to real field development planning problems with large reservoir models. Furthermore, most dynamic measure-based methods are limited to the results of one development scenario, mainly the P50 scenario, which might likely lead to unreliable results especially if there is a high uncertainty in the spatial distribution in the geological properties. Almost all methods fail to find a clear relation between the dynamic and static measures, which created some discrepancies as reported in the literature.

1.2 Scope of Work

The ultimate objective of our work is to develop an efficient well placement optimizer that can be used for practical field development planning projects as we see many well placement optimization efforts in literature that failed to move into practical applications due to their high computational requirement. Towards achieving this objective, our research was divided into two main themes: one focused on generating a fast and robust optimizer for well placement and another focused on introducing an efficient uncertainty management workflow through which a subset of subsurface

geological realizations is used instead of the full set within the optimization workflow. We first focused on developing an efficient optimizer, the black hole particle swarm optimizer (BHPSO) that combined two pillars: the black hole operator (BHO) and the particle swarm optimization (PSO) algorithm. BHPSO was very efficient in determining well location, count and type. However, at this first stage, the optimizer was evaluated over a single geological realization among multiple potential realizations which may typically lead to a suboptimal development scenario. Aiming to solve this issue, the second task focused on introducing a new uncertainty management methodology for defining a subset of representative subsurface realizations instead of the full set, in which the developed optimizer can be utilized and obtain statistically viable results. At a later stage, our focus was on integrating the developed well placement optimizer with a facility placement optimizer (developed by another PhD Candidate, Haytham Dbouk) towards a modular, efficient and comprehensive field development planning framework. The link between the decision parameters of these two algorithms is the well trajectory section (**Figure 1**). Such integration involves numerous well trajectory design iterations which require a uniquely robust and fast algorithm. Hence, our third task focused on introducing an efficient algorithm for modeling and optimizing well trajectory. In line with the discussion above, the key research objectives of this work are as follows:

1. Develop a robust method for optimizing reservoir well location, count, and type.

This algorithm should have a fast and robust convergence which enables it to be used in practical applications. It should also be flexible and extensible to accommodate further developments including hosting additional decision variables such as well control and production layout design.

2. Study different uncertainty management workflows adopted in the literature and develop a robust workflow that is compatible with BHPSO that was introduced in Research Task 1. As was mentioned in Section 1.1, field development plans should be robust to uncertainty represented by multiple geological realizations. Therefore, in this task, we focus on reducing the number of the assessed geological models necessary to be accounted for while preserving the usefulness of the achieved optimal results.
3. Generate an efficient well trajectory optimizer. Well trajectory is the link between reservoir well placement and facility placement. Since the integration would host several optimizers that work back-to-back, there is need for a robust and efficient well trajectory optimizer towards an integrated and computationally feasible solution.

1.3 Dissertation Outline

In Chapter 2, we describe the BHO workflow and showcase its workability and efficiency by testing it on the Olympus Field case. We first introduce the net hydrocarbon thickness (NHCT) map then we run through the details of the method's workflow. Finally, we demonstrate the BHO performance by generating several development scenarios.

In Chapter 3, we describe the BHPSO workflow and apply it on the Olympus optimization challenge. We first introduce the workflow of the traditional PSO optimizer, and then describe how it was combined with BHO to form the BHPSO algorithm. Next, we validate our methodology by comparing its performance with that of the PSO algorithm on three different optimization problems.

In Chapter 4, we present a novel method for selecting a representative subset of geological realizations to be used within the BHPSO workflow as a pragmatic way to manage uncertainty for practical field development planning projects. We first introduce the statistical map along with the workflow that was considered for selecting a representative subset of realizations. Next, we apply the presented methodology on the Olympus challenge and compare its results with two popular scenario reduction methods in the literature: RMfinder and Kmeans – kmedioids methods

In Chapter 5, we discuss the work done on the integrated field development optimization framework then we present the well trajectory optimizer considered. We first describe the Bézier curve formulation along with the developed optimization workflow. We, then, elaborate on the machine learning models used in this work as an alternative to conventional evolutionary optimization algorithms along with their training process. Next, we compare the results of optimizing Bézier curves using the traditional optimizers with those of the proposed machine learning models.

Chapter 6 comprises of a summary of the research highlighting conclusions, ongoing work, as well as recommendations for future work related to well placement optimization under uncertainty.

The new BHPSO algorithm (described in Chapter 2), has been published in the Computational Geosciences journal [81] and is currently under a patent application [82]. The statistical NHCT map approach for selecting a subset of realization, described in Chapter 3, has also been published in Computational Geosciences journal [83]. Our new methodology for optimizing well trajectory (Chapter 4) has been reviewed by our industrial sponsor (Schlumberger) and is currently under review at the Journal of Petroleum Science and Engineering.

CHAPTER 2

WELL PLACEMENT OPTIMIZATION: THE BLACK HOLE OPERATOR (BHO)

In this chapter, we introduce the black hole operator (BHO) that optimizes production and injection wells' placement based on a pre-defined well spacing (minimum distance between wells). We start by describing the quality maps that guides BHO then we run through the details of the optimizer's workflow. Next, we conduct comprehensive validation using a two conceptually different reservoirs belonging to the publicly available synthetic Olympus field [84]. Finally, we present a summary of the method and the results achieved.

2.1 The Black Hole Operator (BHO) Workflow

In this following section, we describe the two quality maps that were deployed within the BHO workflow then we run through the details of the workflow in both peripheral and pattern injection.

2.1.1 Quality Maps

2.1.1.1 Net Hydrocarbon Thickness Map

In this work, we used the net hydrocarbon thickness (NHCT) map as the main “quality map” for pattern well placement. The $NHCT_{map}$ is a two dimensional “condensed” version of the three-dimensional spatial reservoir properties. It consolidates the information from porosity, permeability, and hydrocarbon initially in

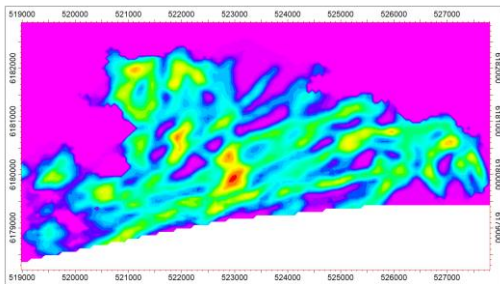
place. For each cell in the aerial direction of a three-dimensional reservoir simulation model (i, j coordinates), the NHCT value is calculated as following:

$$NHCT_{ij} = \sum_{k=1}^{N_k} \varphi_{ijk} NTG_{ijk} (1 - Sw_{ijk}) Z_{ijk} \quad (1)$$

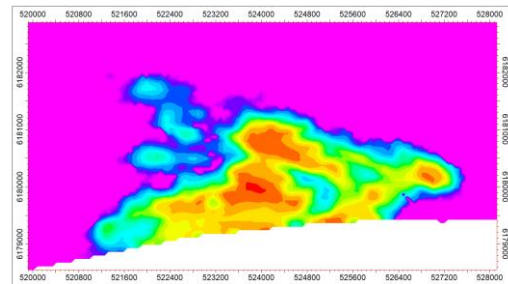
where φ_{ijk} , NTG_{ijk} , Sw_{ijk} and Z_{ijk} denote the porosity, net-to-gross, water saturation and thickness of grid block cell ijk . Since φ_{ijk} , NTG_{ijk} and Sw_{ijk} are dimensionless, $NHCT_{ij}$ has the same unit as Z_{ijk} , that is meter (feet). $NHCT_{ij}$ is, generally, proportional to the following properties:

- Porosity and permeability. These petrophysical properties intervene directly in the formula and indirectly through the dependency of the water saturation function on porosity, permeability and rock quality index, in general.
- Height above free water level. Here also, Sw_{ijk} equals 1 below the free water level (FWL), the irreducible water saturation (Sw_{iijk}) high enough above the FWL and a value between 1 and Sw_{iijk} in the transition zone.

Figure 2 depicts an example of two NHCT maps. The red color indicates high NHCT map that is typically the best location to place a producer. On the other hand, the purple color depicts areas below the FWL or impermeable area (non-reservoir facies).



(a) Olympus Upper (Realization 22)



(b) Olympus Lower (Realization 6)

Figure 2 $NHCT_{map}$ used as main “Quality Map”.

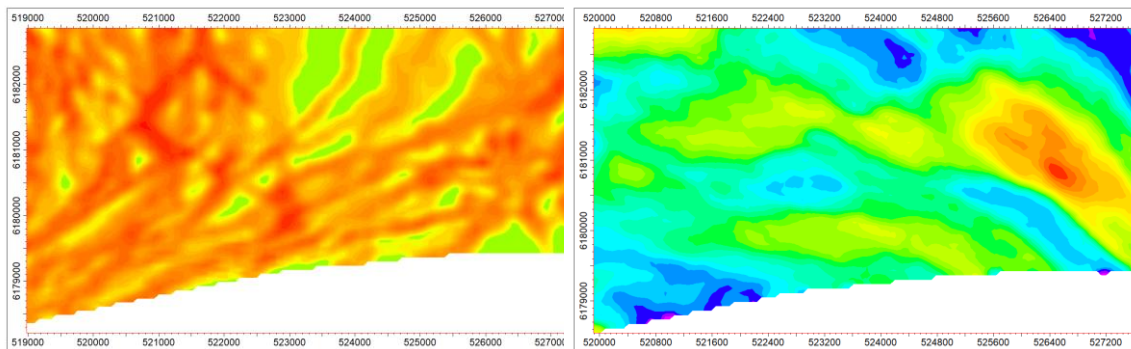
The use of $NHCT_{map}$ is very efficient for handling uncertainty. Different realizations due to uncertainty in reservoir properties can be represented by their corresponding NHCT maps which enables a practically feasible workflow to generate, when needed, well placement scenarios corresponding to different realizations in the process of a field development planning.

2.1.1.2 Permeability Thickness (Kh) Map.

While the NHCT is ideal for producers' or injectors' placement in a pattern-based water injection scheme, the "Kh" map is more suited for water injector placement in a peripheral water injection process. For each cell in the aerial direction of a three-dimensional reservoir simulation model, the Kh is calculated as following:

$$Kh_{ij} = \sum_{k=1}^{N_k} k_{ijk} Z_{ijk} \quad (2)$$

where k_{ijk} denotes the (aerial) permeability of grid block cell ijk . **Figure 3** depicts examples Kh maps; those corresponding to the same reservoirs as **Figure 2**.



(a) Olympus Upper (Realization 22)

(b) Olympus Lower (Realization 6)

Figure 3 Kh map used as main "Quality Map" for peripheral injection.

2.1.2 Well Placement Optimization Using the Black Hole Algorithm

In this section, we describe a typical development concept with two main ingredients:

1. Horizontal wells with a predefined horizontal section length (H_{zL}).
2. Pattern water injection with an injector to producer ratio of one (IP_R).

Note the following:

- The case of vertical well is a particular and simpler case of horizontal wells in the context of this work.
- The proposed algorithm can address any value of IP_R . The value of “one” selected here is for illustration purposes only.
- The case of peripheral water injection is addressed separately as it requires the simultaneous use of two different maps.

The algorithm generates ready to be used well trajectories typically for reservoir simulation purposes. Below are the input data and parameters to the algorithm:

- ($NHCT_{map}$). This is typically built using a pre-processor. The $NHCT_{map}$ may be generated from the reservoir model; fine (geocellular) grid or upscaled dynamic grid.
- A structured map ($Horizon_{map}$) representing the horizon / depth in which the horizontal section to be placed. Typically, the layer containing the largest quantity of hydrocarbon initially in place could be selected to generate the horizon. However, different Horizon maps can be used for placing the horizontal sections of producers and injectors. Furthermore, in a sensitivity analysis process, different depth (and hence Horizon maps) can be tested to select the optimal one, e.g., that resulting in the highest cumulative oil.
- Number of producers and number of injectors, N_P and N_{WI} , respectively.
- Horizontal well section length, H_{zL} .

- Well spacing (that is the radius of the disk around the wells as illustrated in **Figure 4**), R_D . The well spacing controls the maximum number of wells that can be placed based on a given map. In such a case, N_P and N_{WI} become maximum number of wells that cannot be exceeded. This is not the case in the combined BHPSO as we will see in chapter 3 where we introduce the Technical Well Spacing (TWS) as optimization parameter, which is calculated based on the number of wells involved and, hence, ensuring there will always be a space for any prescribed number of wells to be involved.

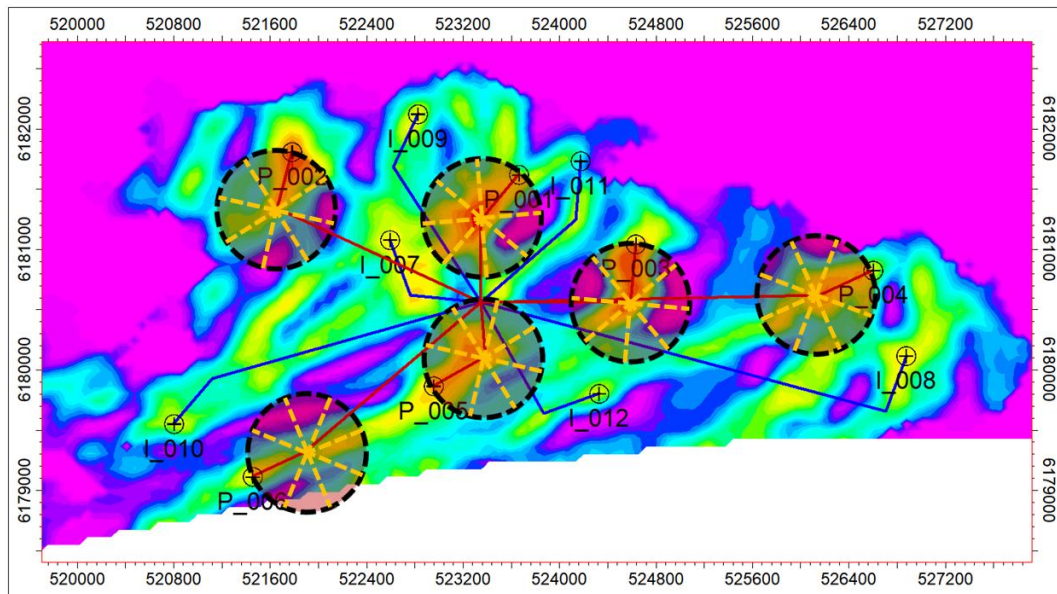


Figure 4 Illustration of well placement optimization of horizontal well azimuth - Producers.

- Increment to be used in optimizing the well azimuth, $\delta_{Azimuth}$. Minimum and maximum radius limits can be provided to restrict the well azimuth (e.g., based on geomechanics considerations and or major fault orientation).
- Minimum distance between wells, D_{min} : A parameter that controls the optimization of the well azimuth and, potentially the total number of wells. For

instance, in case $R_D = H_{zL}$, this parameter limits the well azimuth optimization and keeps the wells toes from potentially colliding through keeping the distance between them larger or equal to D_{\min} .

From a programming point of view, the map is list of points each having the following items; i_x , j_y , x , y and value based on a corner-point grid. Below is a step-by-step description of the algorithm as applied for pattern water injection case.

Producers Placement Using the Black Hole Operator. The algorithm starts by first placing the N_p ($p = 1, \dots, N_p$) producers

1. Copy $NHCT_{\text{map}}$ in a temporary map c
2. Copy $NHCT_{\text{map}}$ in a temporary map $NHCT_{\text{map_temp_inj}}$
3. Creating the producers' heels / toes: For $p = 1, N_p$
 - a. Find the point with the highest $NHCT$ value in $NHCT_{\text{map_temp_prod}}$. Let $\text{Point}_{\text{opt}}$ denote the identified point. In case $NHCT_{\text{map_temp_prod}}$ does not have, anymore, points with a value $> \text{value_cutoff}$, exit the loop.
 - b. Place producer p at $\text{Point}_{\text{opt}}$. This may represent the heel or the toe of the well (depending, for instance, on the platform location in an offshore development planning problem).
 - c. Eliminate a disk of centre $\text{Point}_{\text{opt}}$ and radius $2 \times R_D$ from $NHCT_{\text{map_temp_prod}}$. Practically, that is equivalent at setting $\text{value} = 0$ for every point within this disk.
 - d. Eliminate a disk of centre $\text{Point}_{\text{opt}}$ and radius $1 \times R_D$ from $NHCT_{\text{map_temp_inj}}$.
 - e. $N_{p_max} = p$

4. Generating the Horizontal sections and optimizing their azimuth: For $p = 1$,

N_{P_max}

a. Create the horizontal section of p based on any predefined Azimuth.

This section is made of a large number of “connections” to enable the horizontal section to smoothly follow the $Horizon_{map}$. Let $N_{Connections}$ denote the total number of connections per well.

b. Calculate the cumulative NHCT value, $NHCT_{cumulative}$ for well p by summing up the NHCT value for every “cell” in the map crossing one of the well’s connections.

c. $Azimuth_{opt} = Azimuth$

d. Angle = 0. While Angle < 360°

i. $Azimuth_{temp} = Azimuth + \delta_{Azimuth}$

ii. Calculate the cumulative NHCT value, $NHCT_{cumulative_temp}$, corresponding to $Azimuth_{temp}$.

iii. If the two following conditions are True:

1. the entire well falls within the map, that is no connections of the well correspond to undefined map points for

$Azimuth_{temp}$

2. $NHCT_{cumulative_temp} > NHCT_{cumulative}$

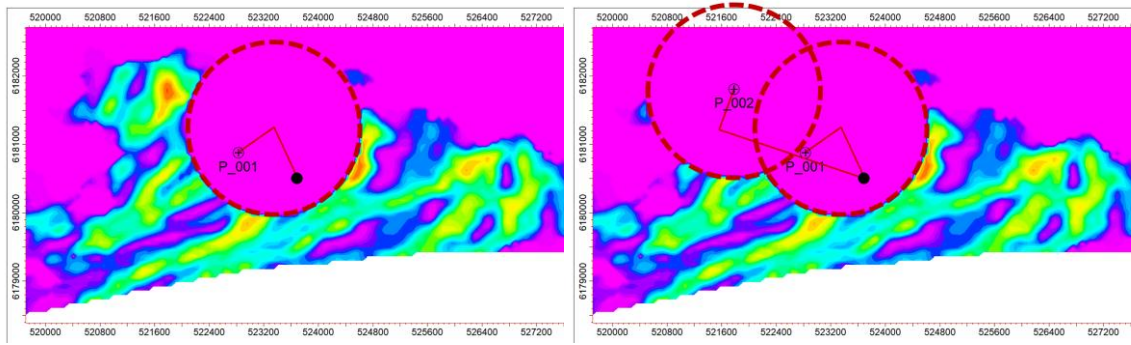
Then: $Azimuth_{opt} = Azimuth_{temp}$.

Note: In the special case of producers only (no injectors), $Azimuth_{opt} = Azimuth_{temp}$ only if the distance between any connection of well p for $Azimuth_{temp}$ and any

connection of any other placed well is larger than D_{\min} . That is on top of the two above conditions.

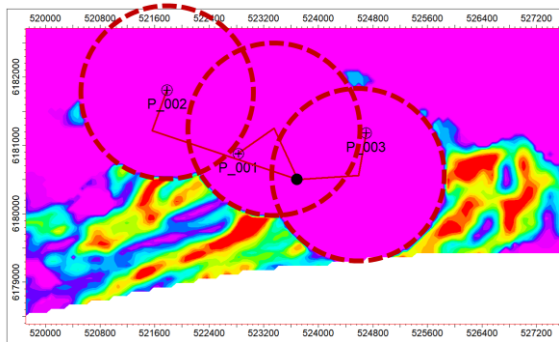
- iv. $\text{Angle} = \text{Angle} + \delta_{\text{Azimuth}}$
 - e. Re-create the horizontal section using $\text{Azimuth}_{\text{opt}}$. When it comes for the depth at which the horizontal section is landed, any default value can be given at this stage.
5. Landing the producers' horizontal sections in the right depth. For $p = 1$, $N_{P_{\max}}$
- a. For connection $c = 1$, $N_{\text{Connections}}$ of well p
 - i. Locate the “cell” in $\text{Horizon}_{\text{map}}$ that corresponds to this connection.
 - ii. Change the depth of connection c to correspond to the cell's centre.

The process through which wells' drainage areas (Disks) are eliminated from $\text{NHCT}_{\text{map_temp_prod}}$ when generating the production wells is illustrated in **Figure 5**. Similarly, **Figure 6** illustrates the process through which wells' drainage areas are eliminated from $\text{NHCT}_{\text{map_temp_inj}}$ when generating the production wells in preparation for generating the water injectors. The resulting final well placement after placing the well along the provided $\text{Horizon}_{\text{map}}$ is illustrated in **Figure 7** for one of the wells. Note that existing wells may be accounted for through “zeroing” the NHCT_{map} around these wells prior to starting the well placement process.

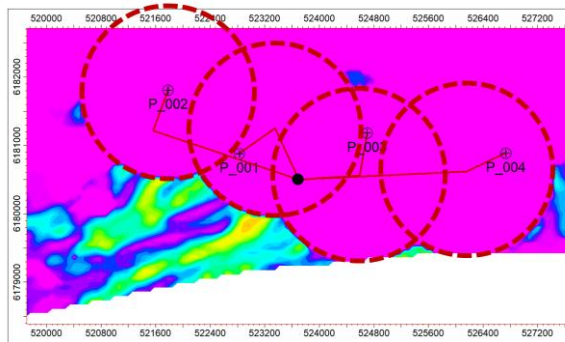


(a) Producer 1

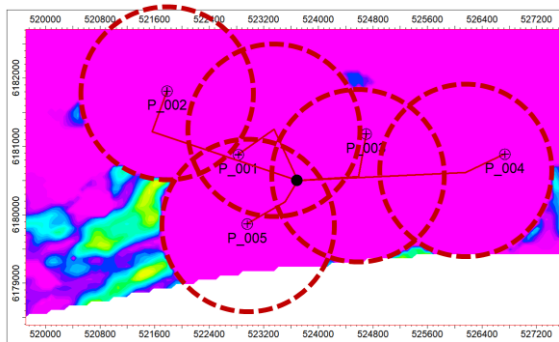
(b) Producer 2



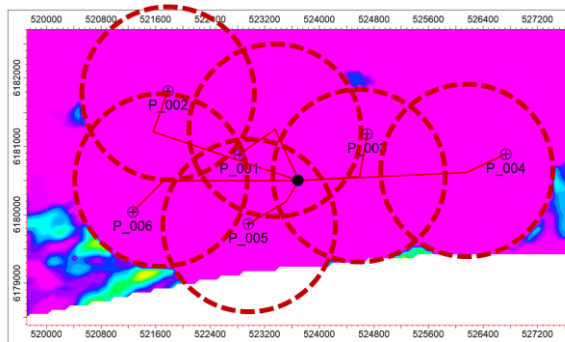
(c) Producer 3



(d) Producer 4

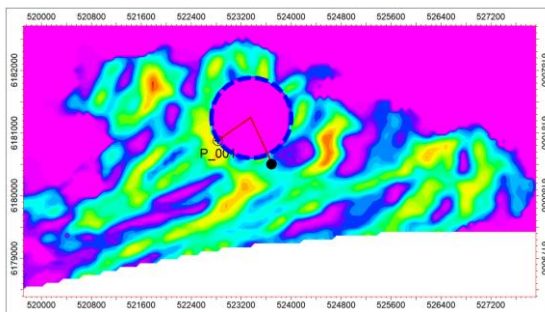


(e) Producer 5

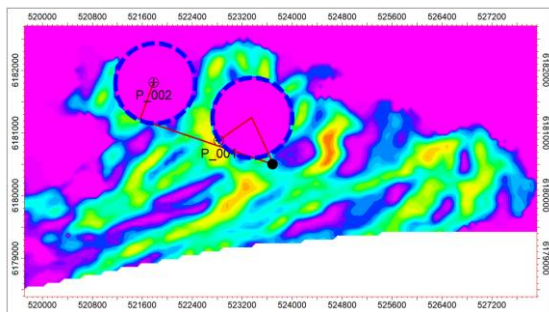


(f) Producer 6

Figure 5 Process through which wells drainage areas are eliminated when generating the production wells. The radius used in this elimination is $2 \times R_D$.



(a) Producer 1



(b) Producer 2

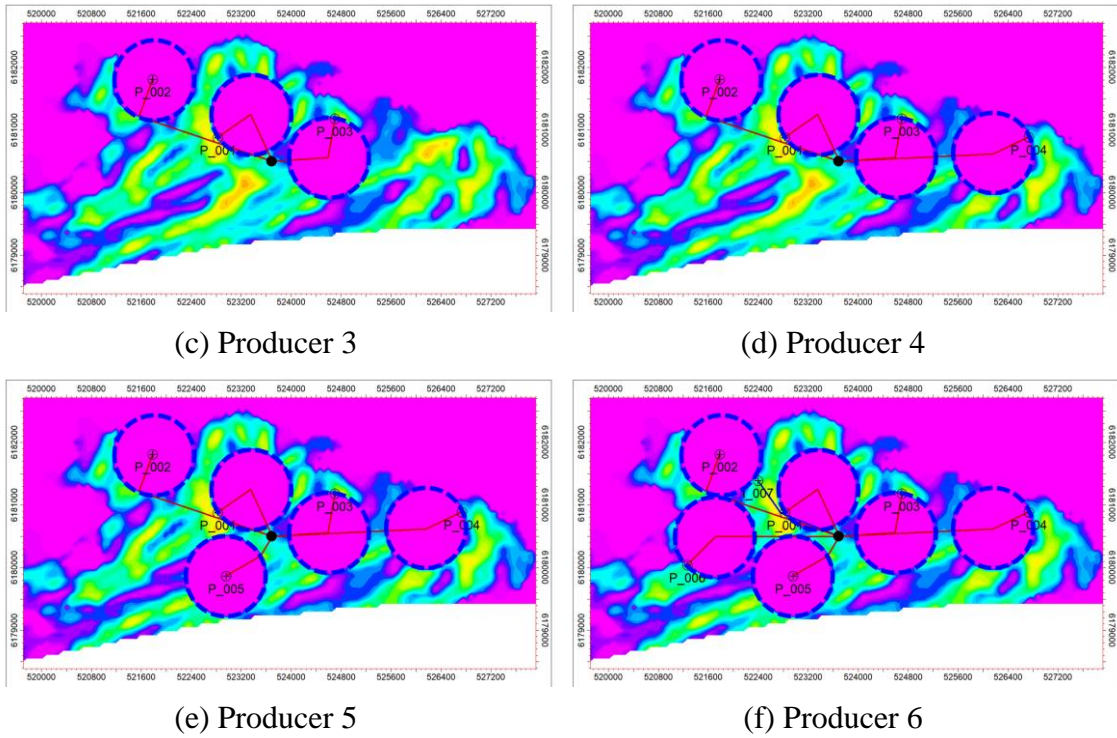


Figure 6 Process through which wells drainage areas are eliminated when generating the production wells in preparation for generating the injectors. The radius used in this elimination is $1 \times R_D$.

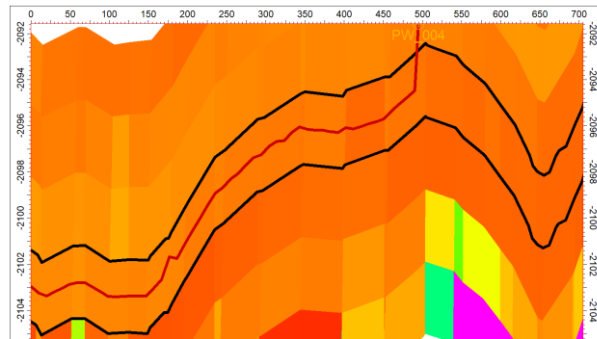


Figure 7 Illustration of the placement of the well's horizontal section in the target layer based on the provided Horizon.

Water Injectors Placement Using the Black Hole Operator. The N_{WI} water injectors ($wi = 1, \dots, N_{WI}$) placements follows. The process through which water injectors are placed is identical to that described in Steps 3, 4 and 5 with the following differences:

- $NHCT_{map_temp_inj}$ resulting from the above Step 3 is used as a starting point.
That is, disks of radius $1 \times R_D$ around all producers have been already removed.
- In Step 4.d.iii, $Azimuth_{opt} = Azimuth_{temp}$ only if the distance between any connection of well p for $Azimuth_{temp}$ and any connection of any other placed well is larger than D_{min} .

The above-described process for water injectors is depicted in **Figure 8**.

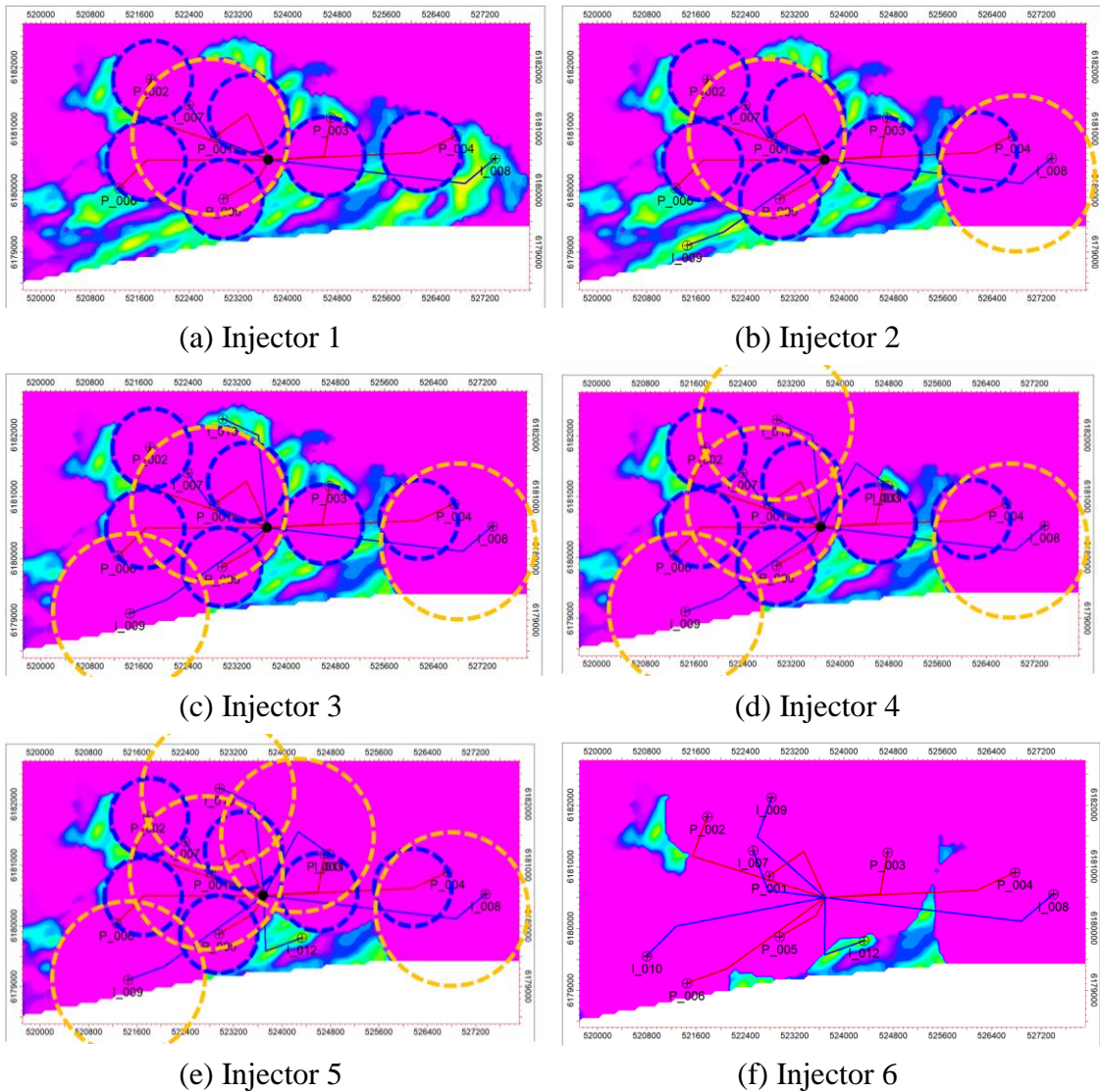


Figure 8 Process through which injectors are generated based on altered net-thickness map following the producers generation. The radius used in this elimination is $2 \times R_D$.

2.1.3 Well Placement Optimization in a Peripheral Water Injection Scheme

While the $NHCT_{map}$ is ideal for producers' or injectors' placements in a pattern-based water injection scheme, the “Kh” map is more suited for water injector placement in a peripheral water injection process. Below is a description of the algorithm used to generate wells in this scenario with a focus on the differences compared to pattern water injection. Two maps are used for the purpose: the $NHCT_{map}$ is used to generate the producers while the Kh map is used to generate the injectors as illustrated in **Figure 9**.

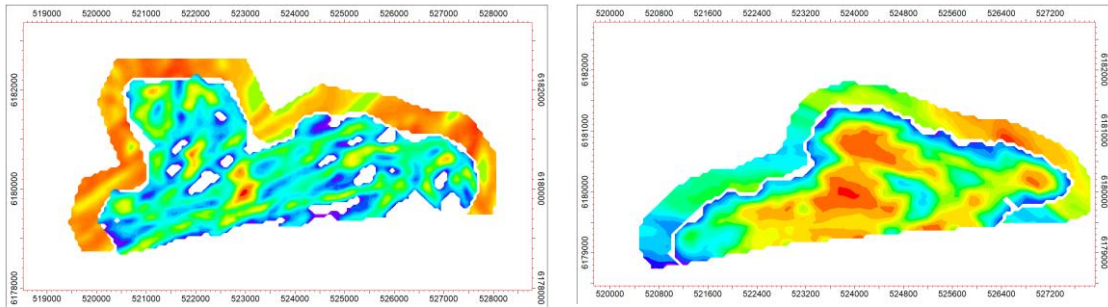


Figure 9 Illustration of the “Combined” set of two maps: $NHCT$ map and Kh map used for well placement in a peripheral water injection scheme. Left: Olympus Upper and Right: Olympus Lower.

Producers Placement. The process through which producers are placed in this case is identical to that described in Steps 3, 4 and 5 for producers' placement in a pattern injection scheme with the following differences:

- $NHCT_{map_temp_inj}$ is not used.
- The disk eliminated from $NHCT_{map_temp_prod}$ in Step 3.c is of radius $1 \times R_D$
- In Step 4.d.iii, $Azimuth_{opt} = Azimuth_{temp}$ only if the distance between any connection of well p for $Azimuth_{temp}$ and any connection of any other placed well is larger than D_{min} .

Water Injectors Placement. The Kh map, Kh_{map} used in this part is a “ring” through which injectors will be placed in a similar way they are placed in the case of pattern water injection. What dictates their placement are R_D , D_{min} and δ_{Azimuth} . The same drivers used in the case of pattern water injection will ensure that well are optimally distributed following Kh_{map} . This is presented in the following section.

2.2 Results and Discussion – The “Olympus” Field

In this section, we demonstrated that efficiency of BHO by employing it on a number of realizations of the Olympus field.

2.2.1 Field Description.

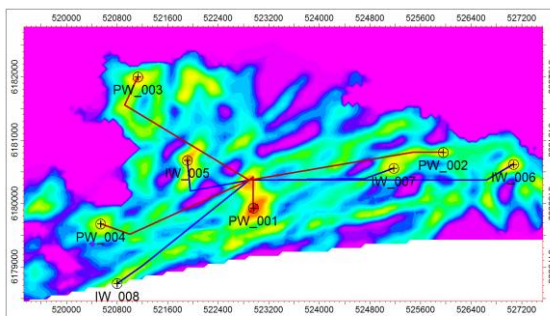
The Olympus challenge aims at testing novel optimization methodologies for field development planning under uncertainty. A synthetic reservoir model, the Olympus, was developed as a benchmark study for the Olympus challenge. The model was inspired by a green field in the North Sea and includes several geological aspects that increase the complexity of the optimization problem such as faults, fault throws, barriers, channels, multiple geological realization as well as coarsening/tightening [85].

The Olympus field has an aerial extent of 9 km by 3 km and a thickness of 50m for which 16 layers have been modelled. The reservoir comprises of two zones, upper and lower zones, separated by an impermeable shale layer. The upper zone (Olympus Upper) involves fluvial channel sands embedded in floodplain shales. The lower zone (Olympus Lower) comprises alternating layers of coarse, medium and fine sands that resembles a clinoformal stratigraphic sequence.

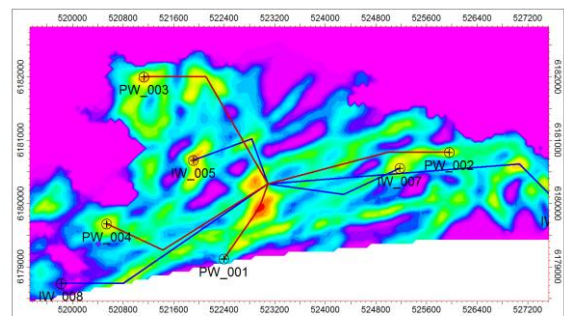
Fifty sub-surface realizations were generated to account for the ranges of uncertainty on porosity, permeability, net-to-gross and irreducible water saturation. The orientation and number of channels varies in the top reservoir section while in the bottom reservoir section the clinoformal stratigraphic sequence is varied to generate these realizations. Realizations 22 and 6 are used as representative realizations for Olympus Upper and Olympus Lower (**Figure 2**).

2.2.2 Pattern Water Injection.

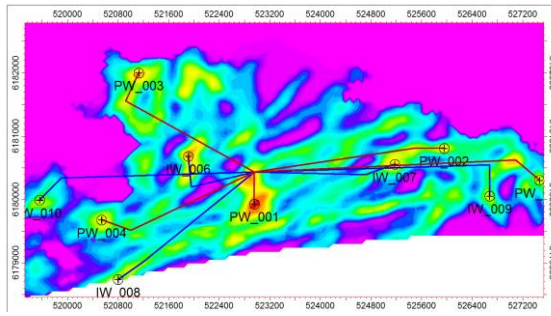
Figure 10 shows the trajectory of the generated wells for Olympus Upper, Realization 22. A sensitivity analysis on the number of wells is performed for two different horizontal well section length under a pattern water injection scenario. Producers are located in the thickest *NHCT* areas accounting for the “disk” radius dictated by a fixed $2 \times R_D$ and D_{min} . While increasing the number of wells gradually increases the covered areas in the *NHCT*_{map}, longer horizontal wells cause a more substantial change on the location of horizontal wells. In both ways, injectors will find their way to be optimally placed to support surrounding producers.



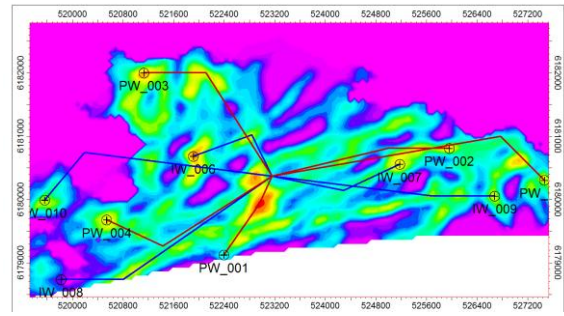
8 Horizontal Wells; 500m



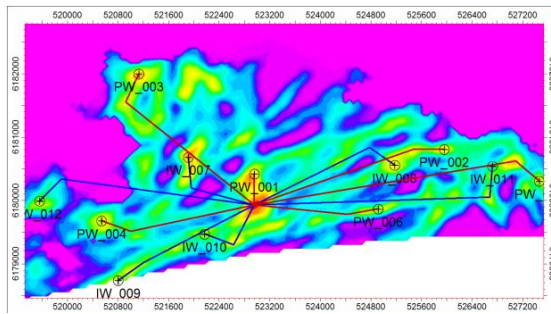
(b) 8 Horizontal Wells; 1000m



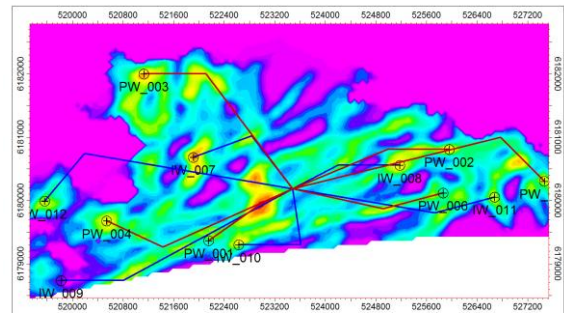
(c) 10 Horizontal Wells; 500m



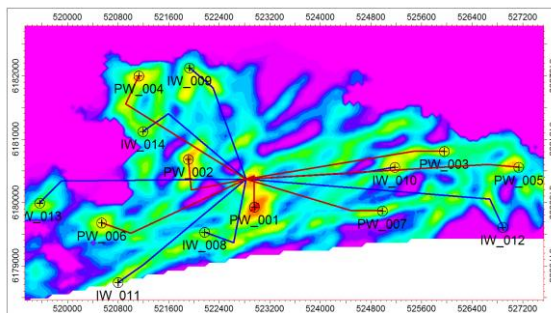
(d) 10 Horizontal Wells; 1000m



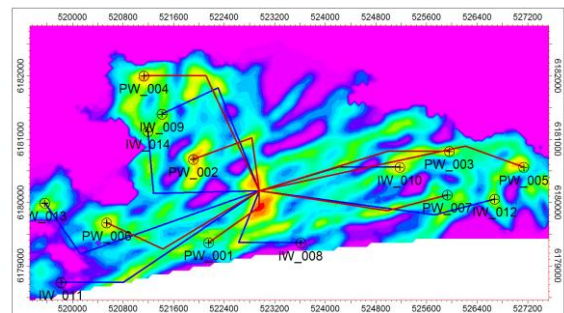
(e) 12 Horizontal Wells; 500m



(f) 12 Horizontal Wells; 1000m



(g) 14 Horizontal Wells; 500m

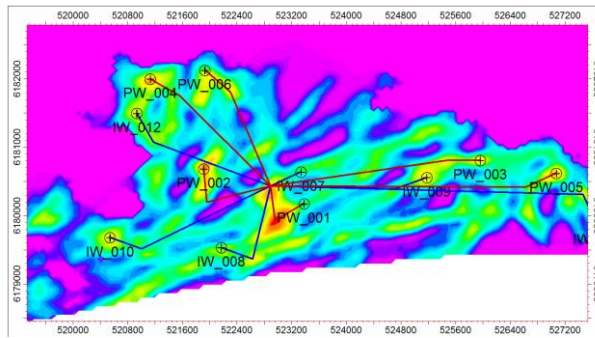


(h) 14 Horizontal Wells; 1000m

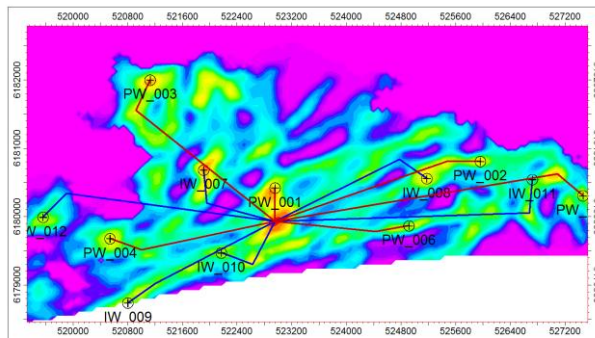
Figure 10 Olympus Upper, Pattern water injection, Realization 22. Sensitivity on the number of wells for two different horizontal section lengths.

A different type of sensitivity is shown in **Figure 11** where the number of wells is fixed, so is the horizontal section length; what varies in these scenarios, is R_D .

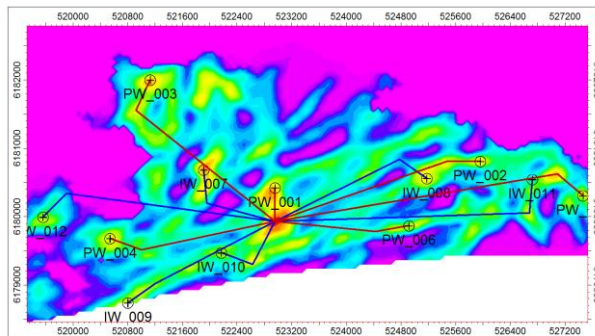
Substantial impact may be observed when increasing $2 \times R_D$ and D_{min} ; that varies from 1300 m to 1400.



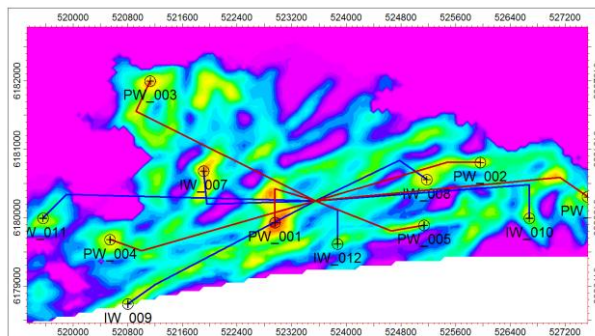
Well spacing = 1300m



(b) Well spacing = 1400m



(e) Well spacing = 1500m



(g) Well spacing = 1600m

Figure 11 Olympus Upper, Pattern Injection, Realization 22, Sensitivity on Well Spacing, 12 Well, 500m

Peripheral Water Injection. The robustness of the proposed method was assessed for peripheral water injection pattern. Results are shown in **Figure 12** and **Figure 13**. Injectors are optimally placed in the highest permeability zones using Kh map.

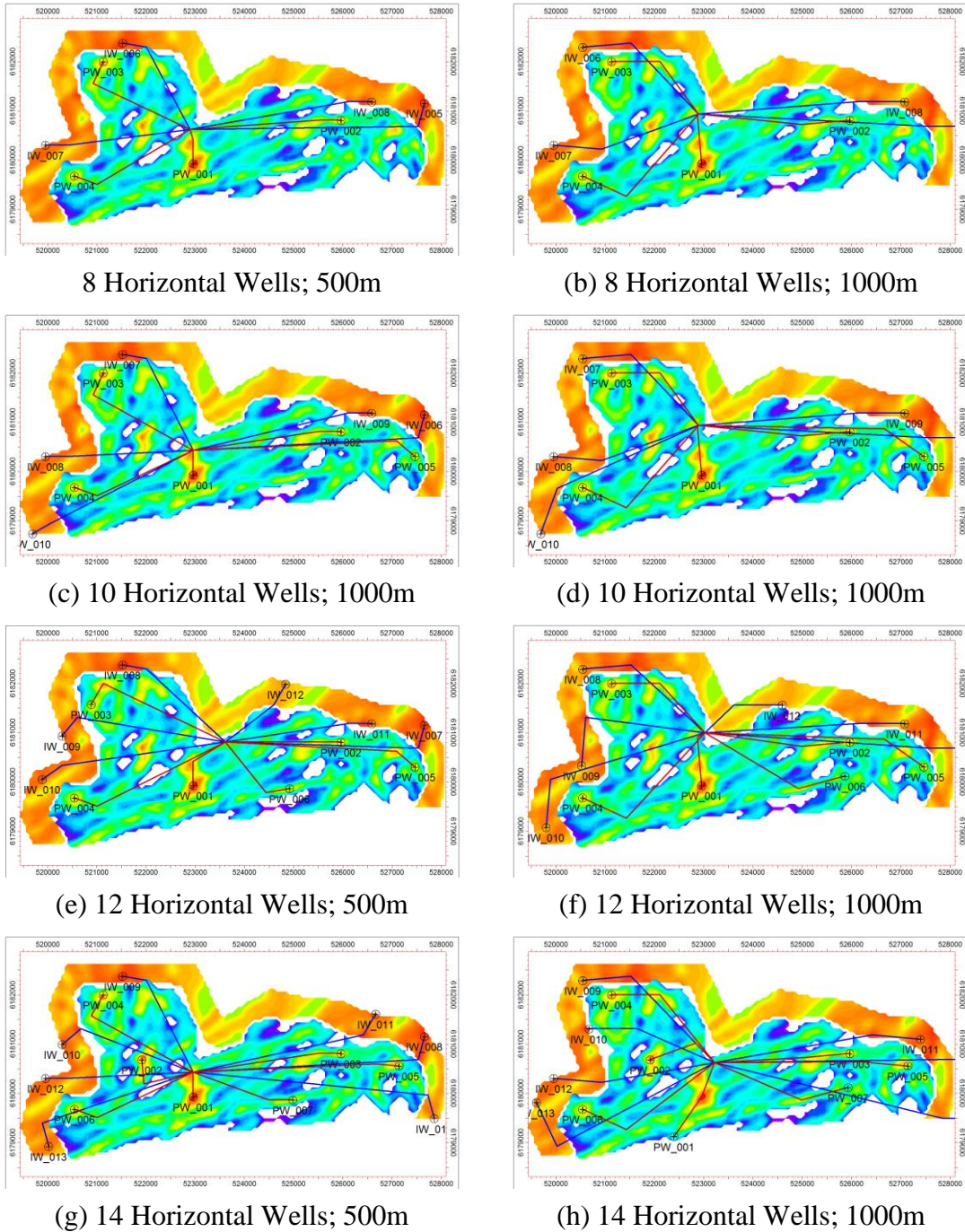
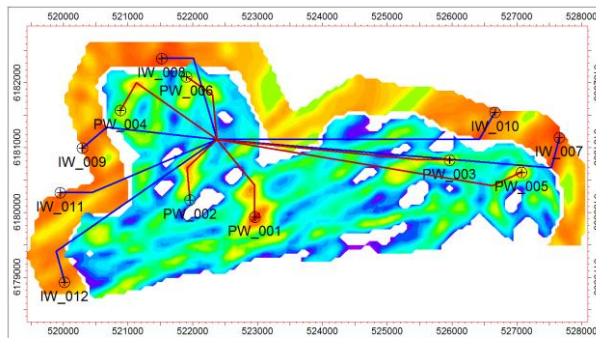
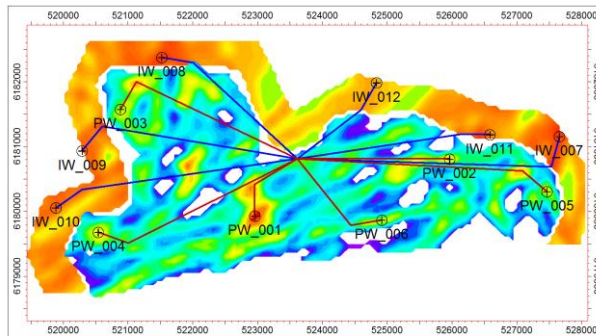


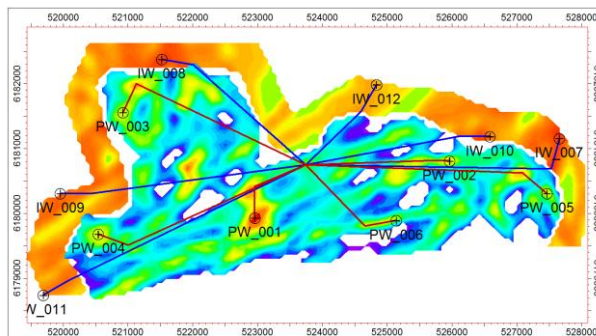
Figure 12 Olympus Upper, Peripheral water injection, Realization 22. Sensitivity on the number of wells for two different horizontal section lengths.



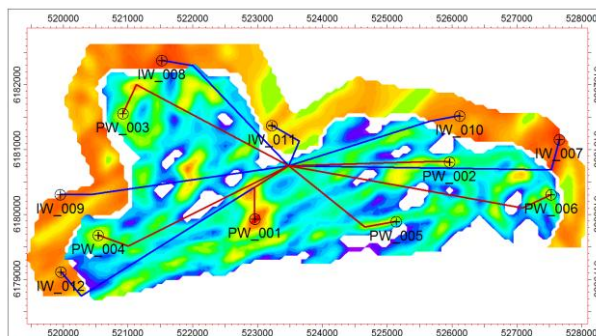
Well spacing = 1300m



(b) Well spacing = 1400m



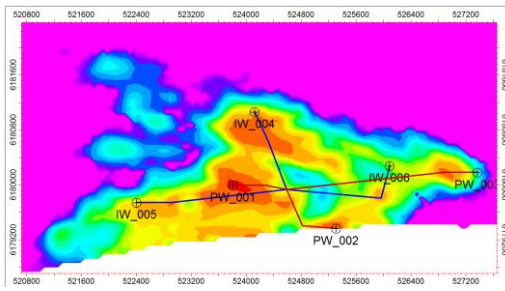
(e) Well spacing = 1500m



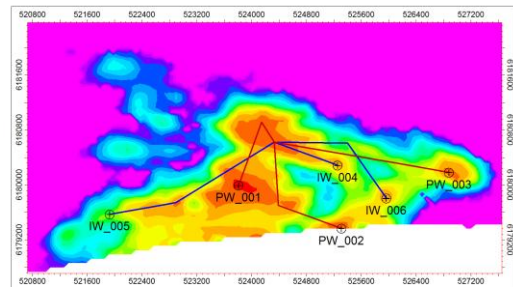
(g) Well spacing = 1600m

Figure 13 Olympus Upper, Peripheral Injection, Realization 22, Sensitivity on Well Spacing, 12 Wells, 500m

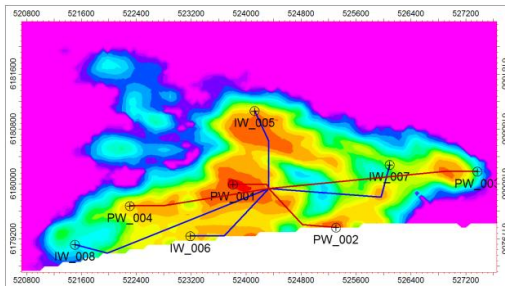
Olympus Lower. Similar sensitivities were performed for Olympus Lower, Realization 6 as shown in **Figure 14**, **Figure 15**, **Figure 16** and **Figure 17**. This illustrates the robustness of the proposed method for addressing different types of geology. Note that sensitivities were also run on different subsurface realizations for both Olympus Upper and Lower with systematic and consistent performance.



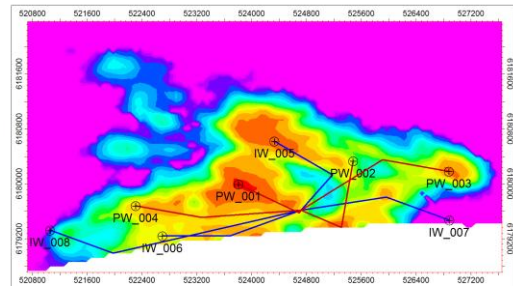
6 Horizontal Wells; 500m



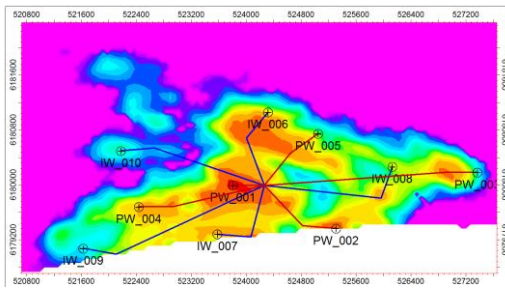
(b) 6 Horizontal Wells; 1000m



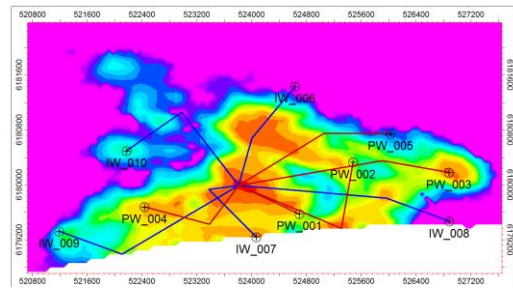
(c) 8 Horizontal Wells; 500m



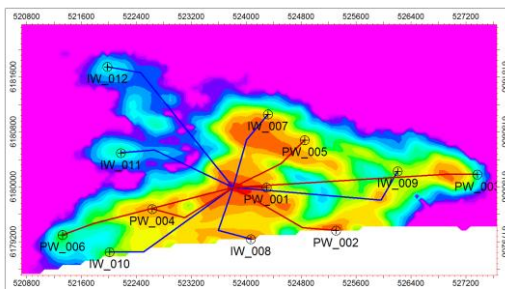
(d) 8 Horizontal Wells; 1000m



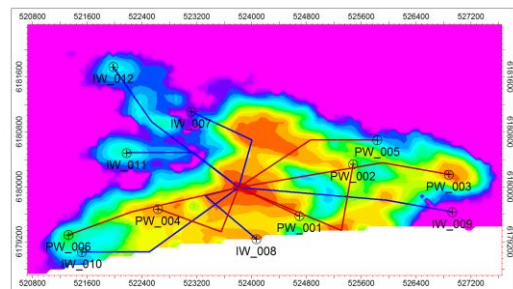
(e) 10 Horizontal Wells; 500m



(f) 10 Horizontal Wells; 1000m

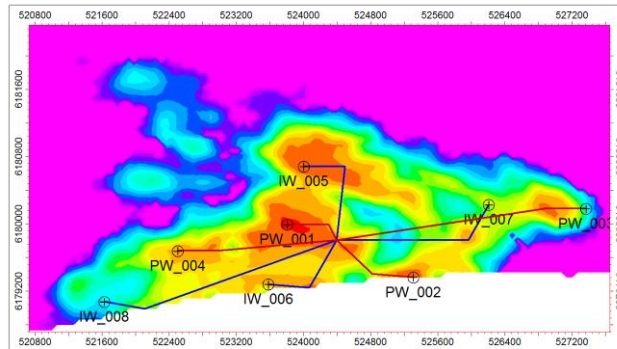


(g) 12 Horizontal Wells; 500m

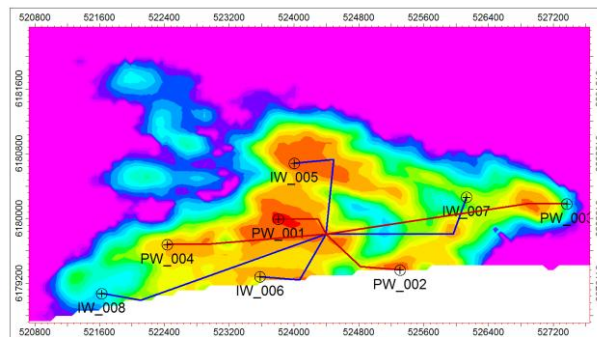


(h) 12 Horizontal Wells; 1000m

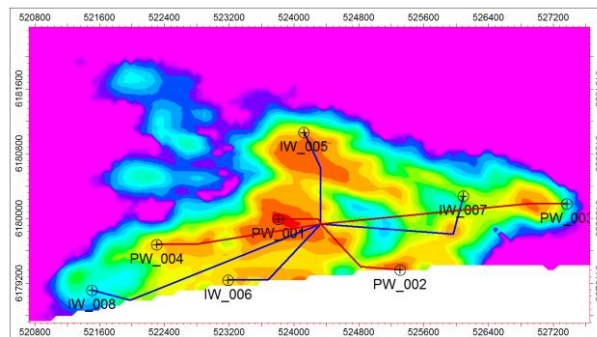
Figure 14 Olympus Lower, Pattern water injection, Realization 6. Sensitivity on the number of wells for two different horizontal section lengths.



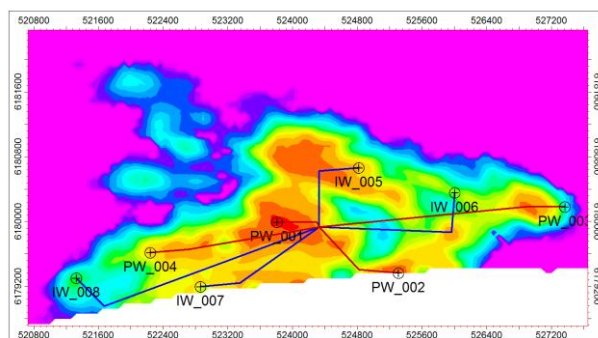
Well spacing = 1300m



(b) Well spacing = 1400m

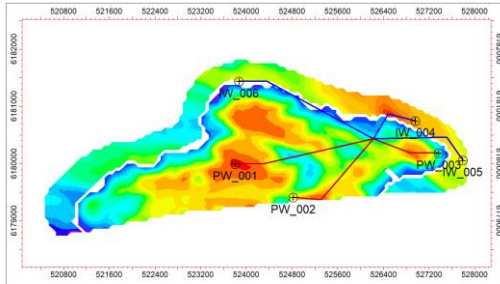


(e) Well spacing = 1500m

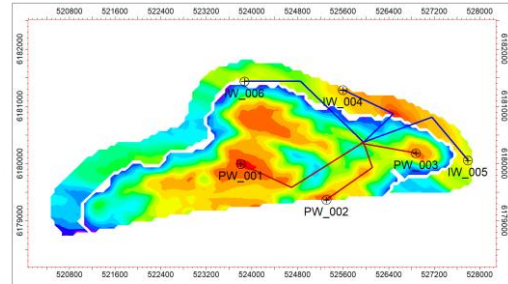


(g) Well spacing = 1600m

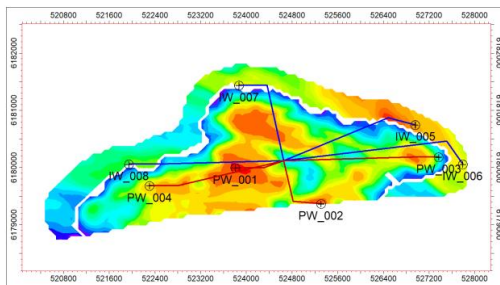
Figure 15 Olympus Lower, Pattern Injection, Realization 6, Sensitivity on Well Spacing, 8 Wells, 500m



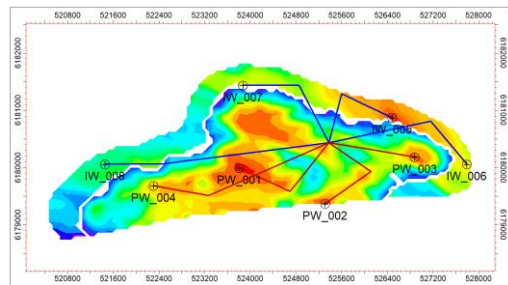
6 Horizontal Wells; 500m



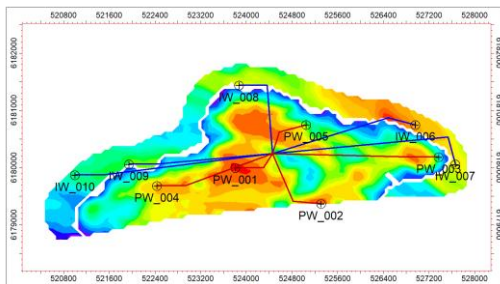
(b) 6 Horizontal Wells; 1000m



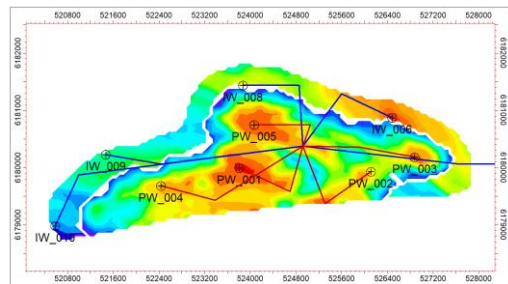
(c) 8 Horizontal Wells; 1000m



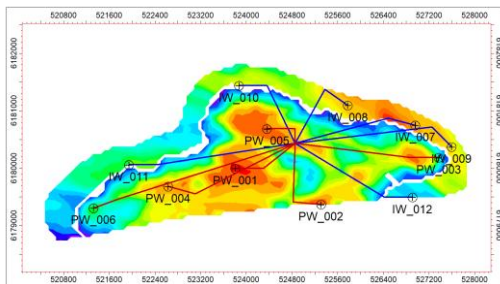
(d) 8 Horizontal Wells; 1000m



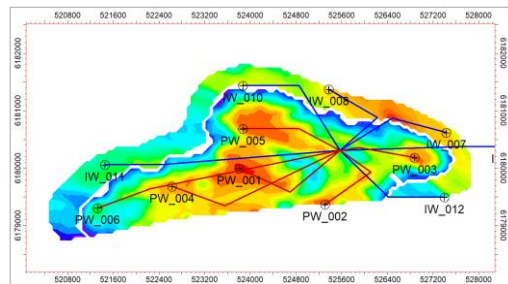
(e) 10 Horizontal Wells; 500m



(f) 10 Horizontal Wells; 1000m

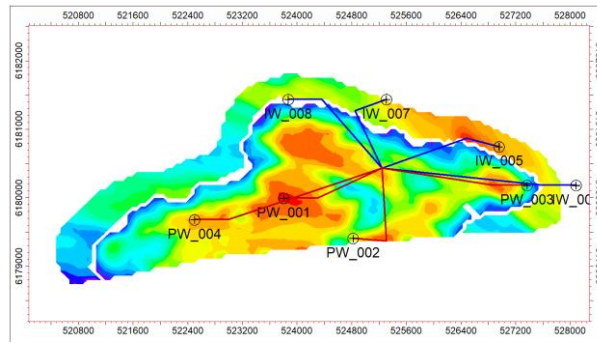


(g) 12 Horizontal Wells; 500m

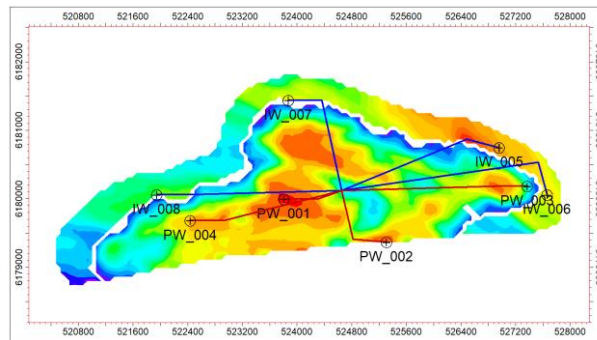


(h) 12 Horizontal Wells; 1000m

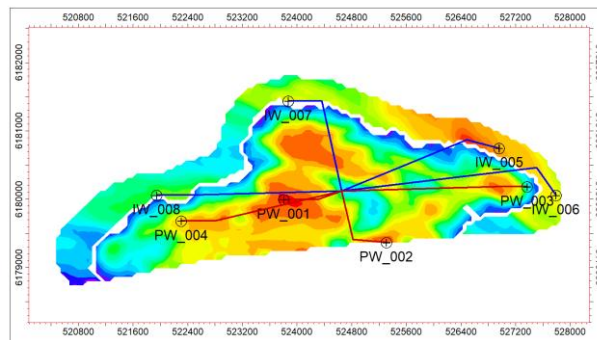
Figure 16 Olympus Lower, Peripheral water injection, Realization 6. Sensitivity on the number of wells for two different horizontal section lengths.



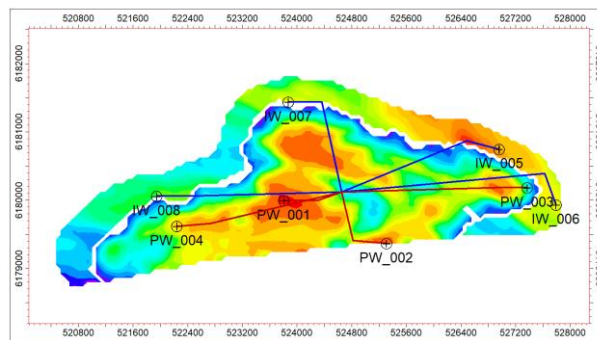
Well spacing = 1300m



(b) Well spacing = 1400m



(e) Well spacing = 1500m



(g) Well spacing = 1600m

Figure 17 Olympus Lower, Peripheral Injection, Realization 6, Sensitivity on Well Spacing, 8 Well, 500m

2.3 Summary

Engineers often need rapid methods that may not necessarily provide the most optimal solution but are capable of scanning a large set of potential development scenarios, running sensitivities on these scenarios, and accounting for large uncertainty on reservoir properties during the process. “Static”, map-based, optimizers are employed for that purpose. These optimizers use 2D “quality maps” that indicate the good regions to be targeted for optimal well placement. Selection of the quality map, the suitability of the well placement algorithm to address a wide range of development scenarios, the robustness of the process and its capability to be automated to address a large number of potential scenarios with their associated sensitivities are all challenges that need to be addressed in static optimization methods.

We propose a map-based method that optimizes production and injection wells’ placement based on a pre-defined well spacing (minimum distance between wells). Wells are automatically and optimally placed using primarily a net hydrocarbon thickness (NHCT) map. The $NHCT_{map}$ is updated after every well placement by eliminating a disk (Black Hole) of a radius defined by the well spacing. Different radii are used to accommodate producers and injectors. For horizontal wells, once the heel/toe of the well is placed the method identifies the azimuth corresponding to a maximum cumulative NHCT value.

The proposed method is systematically and thoroughly validated using a publicly available synthetic field (Olympus) that is inspired by a green oil field in the North Sea and developed for the purpose of a benchmark study for field development optimization. Results clearly illustrate the proposed method’s capability of efficiently

and robustly identifying optimal well placement in comprehensive scenarios including vertical and horizontal wells in pattern and peripheral water injection schemes.

CHAPTER 3

WELL PLACEMENT OPTIMIZATION: THE BLACK HOLE PARTICLE SWARM OPTIMIZATION (BHPSO)

In this chapter, we describe the BHPSO methodology and apply it to the Olympus optimization challenge. The BHPSO is an automated optimization algorithm that optimizes some of the input parameters of BHO (Chapter 2). We first introduce the workflow of the traditional PSO optimizer, and then describe how it was merged with BHO to form the BHPSO algorithm. Next, we validate our methodology by comparing its performance with that of the PSO algorithm on three different optimization problems on the Olympus field.

3.1 BHPSO Workflow

Two algorithms were combined to form the black hole particle swarm optimization algorithm (BHPSO): 1) PSO and 2) BHO (Chapter 2). In the following, we briefly describe both algorithms before detailing the process through which they are used to form the efficient well placement optimization method presented in this work.

3.1.1 Particle Swarm Optimization (PSO)

PSO is an evolutionary algorithm developed by Kennedy and Eberhart in 1995 [86]. It is inspired by a close behavioral review of schools of fish and flocks of birds. Normally, fish and birds travel in groups without colliding with one another by following the group and adjusting to its position and velocity based on the information provided by the group itself. Each “particle” represents a solution of the objective

function, and a “swarm” depicts the group of particles involved in the optimization workflow. These particles could be further grouped into subgroups (neighborhood topologies) allowing the exchange of information with other particles in addition to their own.

The position of each particle in the search space is driven by a calculated velocity parameter based on previous iteration results. The velocity is updated by a mathematical formulation between the prior particle velocity, its distance to the position where the particle achieved its local best and its distance from the particle that achieved the global best. Each particle memorizes the best position or “solution” it attains during the entire optimization process (local best). The algorithm also memorizes the best position attained by any of its particles (global best) [87]. The velocity is truncated by a maximum value defined by the boundary of the search space. The position and the velocity relationship are obtained by the formula below and illustrated by **Figure 18** :

$$x_{i,j}(k + 1) = x_{i,j}(k) + v_{i,j}(k + 1) \quad (3)$$

$$v_{i,j}(k + 1) = w \times v_{i,j}(k) + c_p r_1 (p_{best(i,j)} - x_{i,j}(k)) + c_g r_2 (g_{best(i,j)} - x_{i,j}(k)) \quad (4)$$

Where, i refers to the particle, j refers to the optimization variable and k refers to the current iteration.

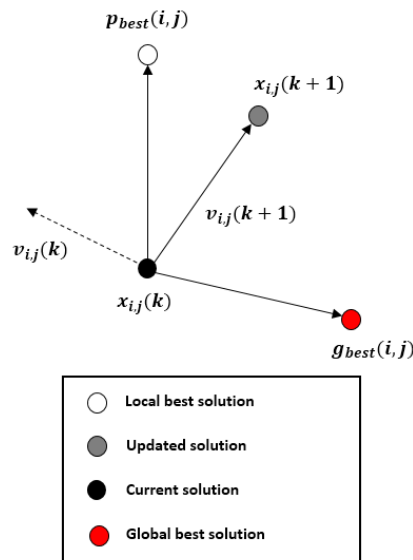


Figure 18 Graphical Illustration of the PSO particle velocity and position calculation. Extracted from [32]

The velocity equation involves three main parameters: the inertia weight (w), the cognitive weight (c_p) and the social weight (c_g). The inertia weight defines the impact of the trend towards the previous particle velocity, the cognitive weight defines the impact of the trend towards the particle local best ($p_{best(i,j)}$) and the social weight defines the impact of the trend towards the swarm global best ($g_{best(i,j)}$). PSO also involves two independent uniform random variables r_1 and r_2 between 0 and 1. The main purpose of these variables is to make the overall process stochastically-dependent which helps the optimizer avoiding local optima traps.

Many theoretical analyses were conducted in order to gain deeper insight into the mechanism of PSO. Some researchers assessed the convergence rate of PSO by running a sensitivity analysis on its algorithmic parameters [88–90]. Others introduced different neighborhood topologies e.g. star topologies [17], random topologies [91] and several other topologies. In this manuscript, we employed the star PSO that considers

the swarm as a single neighborhood moving toward the local best of every particle and the global best of the whole swarm along with the algorithmic parameter used in [87] ($w = 0.721, c_g = c_p = 1.1931$). The sensitivity analysis on algorithmic parameters and neighbourhood topologies of PSO was kept for future investigations as the main objective of this manuscript is to show the advantage of merging BHO with the PSO algorithm.

3.1.2 BHPSO – Black Hole Particle Swarm Optimization

The BHPSO workflow is summarized in **Figure 19**. Below we describe the different building blocks of the algorithm as applied to well placement in a pattern water injection scheme. Injectors’ placement in a peripheral water injection scheme may use a similar process. However, this part was left for future work.

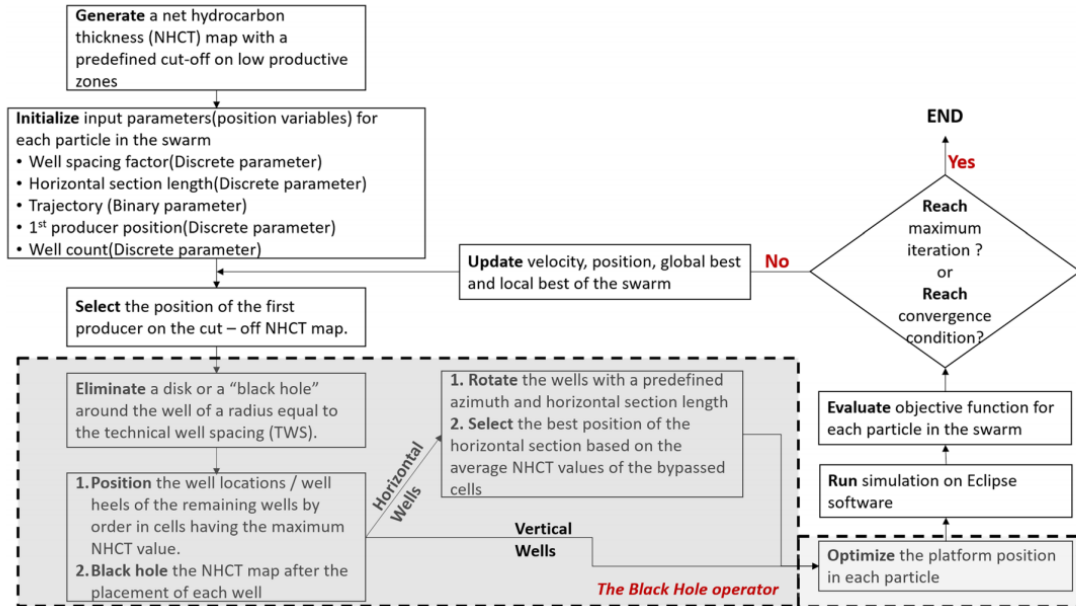


Figure 19 Schematic of BHPSO workflow.

3.1.2.1 Convergence Criteria

It is a challenging task to set convergence criteria for evolutionary algorithms, as these criteria cannot be mathematically proven. In this work, convergence of BHPSO is declared when one of the following two conditions is met:

A maximum number of iterations, $Iter_{max}$ is reached. In this work, $Iter_{max} = 30$. No further improvement of results is observed after a prescribed number of iterations, $Iter_{conv}$. These conditions are set due to limitations in computational power, simulation time and observations from the runs specific to the Olympus Challenge. Further investigations are needed to define more robust convergence criteria for BHPSO.

Initialization of the Optimization Parameters

The PSO algorithm initializes the solution vector of the black hole operator. The solution vector includes five parameters that can be categorized into:

- Discrete parameters
 - Row index of the i and j coordinates of the first well on the $NHCT_{map_prod}$ in the case of a vertical well.
 - Row index of the i and j coordinates of the first well toe/heel on the $NHCT_{map_prod}$ in the case of a horizontal well
 - Well count: A number between a predefined minimum and maximum.
 - Horizontal section length (in the case of a horizontal well), L_{Hz} : A number between a predefined minimum and a maximum.
- Continuous parameter
 - Well spacing factor (F): Dimensionless parameter between 1 and 2.
- Binary parameter

- Well trajectory: “0” represents a vertical trajectory and “1” represents a horizontal trajectory.

While the convergence criteria defined in the above section is not met, the well placement optimization parameters are updated through the following steps.

3.1.2.2 Placement of the First Well

This is the only well that is placed by PSO which makes the whole process independent of the number of wells and hence is the drastic efficiency of the proposed method. PSO selects the well location (i and j coordinates) based on results from previous iterations. Once selected, BHO eliminates a disk “black hole” from the $NHCT_{map_prod}$ around the selected location with a radius equals to $2 \times TWS$, where TWS is the Technical Well Spacing. TWS is used for BHPSO exclusively to ensure a place for all wells on the $NHCT_{map_prod}$ and is defined by:

$$TWS = F \sqrt{\frac{A_r}{2\pi N_w}} \quad (5)$$

where A_r , N_w and F denote, the total $NHCT_{map_cutoff}$ area, the number of wells, and the “well spacing factor”, respectively. The $NHCT_{map_cutoff}$ is a cut-off $NHCT_{map}$ at a value of 2m to reduce the search area for the optimization algorithm (**Figure 20**). Note that the $NHCT_{map_cutoff}$ is only used for the placing the first well which is always a producer. F is a multiplier introduced to the technical well spacing formula to enable variability and flexibility in the selection of the optimal horizontal well length and distance between wells. Simultaneously, BHO eliminates a “black hole” from the $NHCT_{map_inj}$ with a radius equal to TWS to enable optimal well placement of injectors.

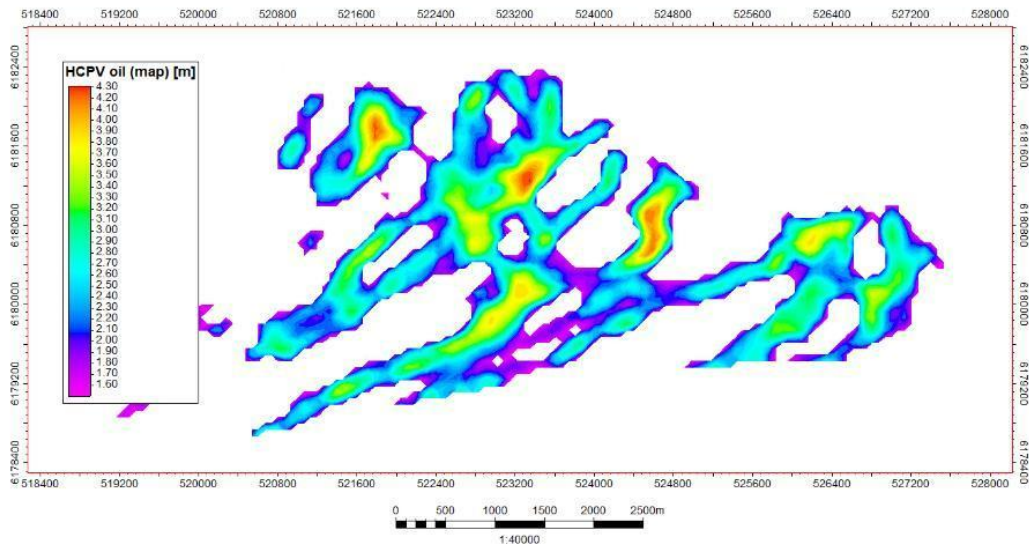


Figure 20 $NHCT_{map}$ and corresponding cutoff at a value of 2m.

3.1.2.3 Placement of the Remaining Wells

The remaining wells (whose number is decided by PSO) are placed by BHO, sequentially, using the $NHCT_{maps}$. The operator picks the location of the next producer or injector based on the maximum $NHCT$ value on the relevant black-holed maps, ($NHCT_{map_{prod}}$) for producers and $NHCT_{map_{inj}}$ for injectors. First, the black hole operator places the producers, one by one, while blacking the $NHCT_{map_{prod}}$ with a radius of $2XTWS$ and the $NHCT_{map_{inj}}$ with a radius $1XTWS$ around the location of the placed producers. Then, after placing all the producers, the injectors are placed, one by one, while blackholing again the $NHCT_{map_{inj}}$ with a radius of $2XTWS$ around the location of the placed injectors. This ensures a pattern – like injection scheme where there is at least an injector optimally placed between any two producers. Note that producers are placed first to give them priority on areas with highest $NHCT$ values.

3.1.2.4 Case of Horizontal Wells – Well Azimuth Optimization

BHO provides the core algorithm (PSO) with the option to alter the trajectory type (horizontal vs. vertical wells). The trajectory option is a binary variable where all wells are either horizontal or vertical. For horizontal trajectory, after locating the toe (same as vertical trajectory discussed above), BHO assesses the NHCT corresponding to different azimuths as illustrated in **Figure 4** and selects the azimuth crossing through the cells having the maximum cumulative NHCT. Note that the horizontal wells are placed in specified vertical layers and their length is an optimization parameter that is altered by PSO. In this work, we have chosen an increment of 45 degrees. However, this number is a user-set parameter that can be changed to any other value. As for vertical trajectory, wells are chosen to be fully penetrating all the layers.

3.1.2.5 Platform Placement Optimization

NPV optimization involves optimal platform optimization. In this work, a single platform is used; the optimizer search for the cell on the $NHCT_{map}$ yielding to the lowest cumulative well's trajectory (from platform to toe). Note that the platform position defines the heel/toe of the well. Among the two end points of the horizontal section, the heel is the closest point of the well to the platform while the toe is furthest point of the well to the platform.

3.1.2.6 Swarm Evaluation and Algorithmic Parameters Update

Each PSO iteration holds several particles or “simulation cases” in which their optimization parameters are updated at each iteration. The simulation results are fed into an operator that updates the position and velocity of each particle. The updated position

of each particles defines the values of the parameters of the next iteration in the optimization process. The process re-iterates until convergence criteria is met.

3.2 Results

In this section, we briefly introduce the Olympus challenge and the associated reservoir model. Then, we validate BHPSO by comparing its performance to that of the standard PSO.

3.2.1 Development Driving Value – The Optimization Objective Function

BHPSO was tested on the Olympus optimization challenge, which aims at assessing novel algorithms for field development optimization under uncertainty. This challenge mainly revolves around three main optimization parts: Optimization of 1) Well Control (WC), 2) Well Placement (WP) and 3) Joint WP + WC. In this work, we focus on the WP exercise where the number, location, type, and trajectories of wells are to be optimized under subsurface uncertainty represented by 50 geological realizations.

The field development plan is evaluated by the expected Net Present Value (NPV) over a fixed project duration of 20 years. **Table 1** lists the input parameters required for NPV calculation. The NPV for each of the development scenarios is defined by (US \$):

$$NPV = \sum_{i=1}^{N_t} \frac{R(t_i) \times \Delta k_i}{(1+d)^{t_i/\tau}} \quad (6)$$

where, $\Delta k_i = k_i - k_{i-1}$, the time period between k_i and k_{i-1} , in days, d is the discount factor, τ is number of days in a year and $R(t_i)$ is the sum of all costs and revenues acquired within the time period Δt_i . The cost term $R(t_i)$ in US\$ and is calculated using the following formula:

$$R(t_i) = Q_{op}(t_i) \cdot rop - Q_{wp}(t_i) \cdot rwp - Q_{wi}(t_i) \cdot rwi - P - D(t_i) \quad (7)$$

Where:

Q_{op} , Q_{wp} and Q_{wi} are the cumulative oil production, water production and water injection volumes over Δt_i , respectively.

rop , rwp and rwi are the oil revenue (price), water production handling cost and water injection handling cost, respectively; all in \$ per unit volume,

P is the platform cost

$D(t_i)$ is the total well drilling and completion costs incurred during the time period Δt_i

The time period (in days) to drill a well is denoted by the following:

$$\Delta t_D = 0.015 \cdot \Delta Z + 0.02 \cdot |\Delta XY| \quad (8)$$

Where, ΔZ and $|\Delta XY| = \sqrt{(\Delta X)^2 + (\Delta Y)^2}$ are the vertical and horizontal step out of the well end point from the Platform location, respectively.

Table 1 Input parameters used to calculate the objective function.

Contribution	Value	Units
Platform cost	500	Million \$
Drilling & well completion	$500 \cdot \Delta Z + 10000 \cdot \Delta XY$	\$, ΔZ and ΔXY in m
Oil price	45	\$ per bbl
Water production cost	6	\$ per bbl
Water injection cost	2	\$ per bbl
discount factor (Annual)	0.08	Dimensionless

3.2.2 Validation of the Proposed Optimization Method – BHPSO vs. PSO

A comparative study was conducted to demonstrate the advantages of combining the BH operator with evolutionary algorithms; PSO in this work. This study entails a three-dimensional sensitivity analysis for comparing the performance of PSO and BHPSO algorithms in relation to problem complexity, swarm size and stochastic

nature of the core algorithm. Thus, the size of the search space, and the amount of computational load vary for the different sensitivities.

In all sensitivities, the same algorithmic parameters [87] and neighborhood topology (star topology) were used for the core optimizer (PSO algorithm) of both algorithms. Furthermore, both algorithms were initialized in each sensitivity with the same initial solution using the BH operator and run for a fixed number of iterations (30 iterations). The NPV was used as the objective function and computed using Equation 6. The reservoir of interest for the validation purpose is Olympus Upper, Realization 22. It is worth noting that some of the optimization parameters were excluded in this study since it was practically infeasible to optimize them using the standard PSO; which is not the case of BHPSO.

3.2.2.1 Sensitivity 1 – Problem complexity

BHPSO and PSO were applied to two optimization problems with different levels of complexity. Problem 1 entails the optimization of the location of 6 vertical wells. In this problem, the PSO is optimizing 6 decision variables (e.g., 6 row indices of x, y coordinates) while the BHPSO is optimizing only two decision variables (e.g., first well index of x, y coordinates and well spacing factor (F)). On the other hand, Problem 2 involves the optimization of the location and azimuth of 12 horizontal wells of a fixed horizontal section length (500 m). PSO, in Problem 2, is optimizing 24 decision variables (e.g., 12 row indices of x, y coordinates and 12 well azimuths) while BHPSO on the other hand, is optimizing only two decision variables (e.g., first well index of x, y coordinates and well spacing factor (F)). This reduction in the number of decision variables is induced by the BH operator which is taking a major part of the optimization

load. Furthermore, in both problems, the optimization run of each algorithm was repeated three times and average NPV was generated for comparison. **Figure 21** illustrates the progression of NPV for both algorithms in the selected simulation problems.

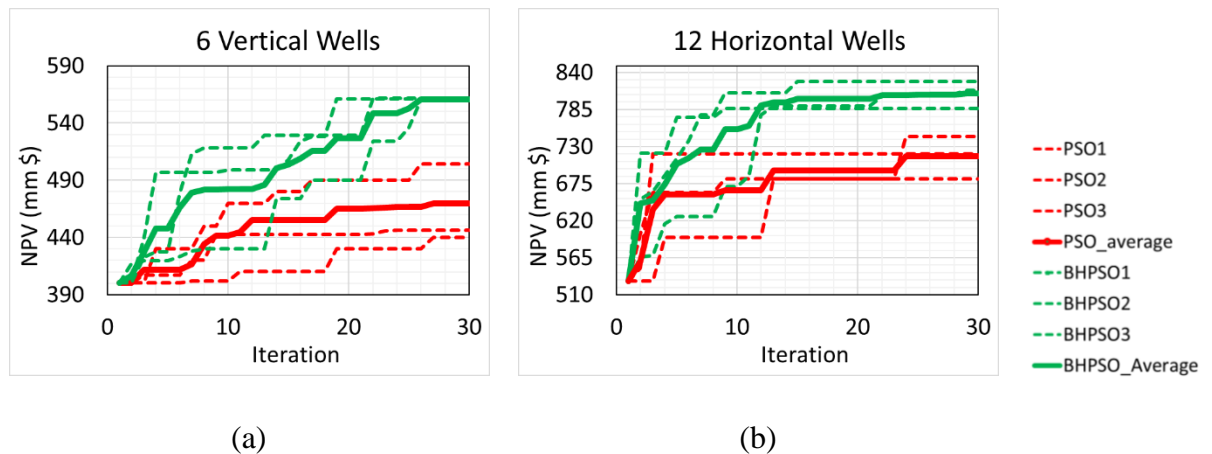


Figure 21 Performance comparison between BHP SO and PSO with a swarm size of 10 particles on 2 different simulation cases (Problems 1 and 2): a) 6 vertical wells, and b) 12 horizontal wells. BHP SO performance is represented by the green curves while the standard PSO is represented by the red curves. The single simulation trials are depicted by dashed lines while their average is depicted by a bold line.

In Problem 1 (**Figure 21a**), BHP SO systematically outperforms the standard PSO in all three trials. BHP SO shows a better tendency in avoiding local optima traps making its progression more efficient than that of the PSO. With BHP SO, NPV reaches an average of 561 million US dollars (MM US\$) while the average for PSO is 470 MM US\$; 19 % less than the result of the BHP SO. On average, BHP SO achieved a 40 % improvement over its first trial while PSO has achieved a 17.5 % improvement over the same first trial. Furthermore, unlike PSO, BHP SO has a consistent performance over the 3 trials where there is a minimal difference in the reached optimal values compared to those reached by PSO.

Problem 2 (**Figure 21b**) involves a new type of decision variable, the azimuth of the horizontal section of the well, which made it a more complex problem than Problems 1. As expected, BHPSO is drastically outperforming the PSO in this problem. Similar to Problems 1, trials of BHPSO are converging faster than those with PSO. BHPSO reaches an average optimal value of 813 MM US\$ that is higher than that reached by PSO (720 MM US\$). On average, BHPSO achieved a 53.3 % improvement vs. its initial trial while PSO achieved a 36.5%. The complexity of this problem has slightly affected the consistency of BHPSO performance as we see a difference between the optimal values of different trials; however, this difference is relatively minimal (5 %). The convergence factor of BHPSO will be tackled in future work on neighborhood topologies and algorithmic parameters of the core optimizer (PSO).

3.2.2.2 Sensitivity 2 – Number of Particles

Considering the difference in the number of decision variables between the two algorithms, it was decided to investigate the effect of the swarm size on the algorithm's performance. Thereby, in this sensitivity, BHPSO and PSO were employed for Problem 1 with two different swarm sizes: 15 and 20 particles. At both swarm sizes, both algorithms were initialized with same initial solution that was used for Problem 1 in sensitivity 1 and their optimization run was repeated three times. **Figure 22** illustrates the progression of NPV for both algorithms at different swarm sizes.

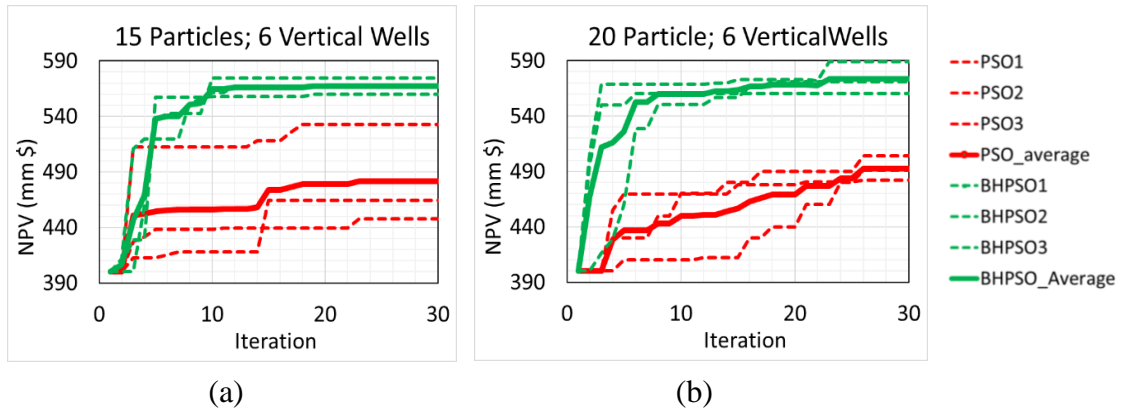


Figure 22 Performance comparison between BHP SO and PSO on Problem 1 and with two different swarm sizes: a) 15 particles and b) 20 particles. BHP SO performance is represented by the green curves while the standard PSO is represented by the red curves. The single simulation trials are depicted by the dashed lines while their average is depicted by the bold line.

Similar to observations in sensitivity 1, BHP SO is outperforming PSO in all optimization runs. With 15 particles (**Figure 22a**), NPV with BHP SO reaches an average of 567 million US dollars (MM US\$) while the average for PSO is 482 MM US\$; 17.6 % less than the result of the BHP SO. On the other hand, with 20 particles (**Figure 22b**), NPV with BHP SO reaches an average of 574 million US dollars (MM US\$) while the average for PSO is 492 MM US\$; 16.6 % less than the result of the BHP SO. Relative to results of Problem 1 in sensitivity 1, the convergence rate of BHP SO is drastically improving. With 10 particles in Sensitivity 1 (**Figure 21a**), BHP SO is reaching a near optimal value in 20 iterations whereas with 15 and 20 particles in sensitivity 2 (**Figure 22a, b**), BHP SO is nearly converging at 10 – 12 iterations. Furthermore, observing the final results for 10 particles in Sensitivity 1 (**Figure 21a**) and for 15 and 20 particles (**Figure 22a, b**) in Sensitivity 2, we see a minimal improvement of PSO relative to BHP SO. With further increase in the swarm size, this improvement may grow however, according to results, it will require a large

swarm size to meet the performance of BHPSO. More difficulty arises when the problem is more complex and include a larger number of decision variables.

3.2.2.3 Sensitivity 3 – Stochastic Nature

In sensitivities 1 and 2, optimization runs were repeated only three times and was deemed sufficient for the purpose of each sensitivity, however such number could be considered insufficient to compare both algorithms in relation to the stochastic nature of the core algorithm. Thereby, in this sensitivity, Problem 1 was repeated ten times and the average NPV was used for comparison between the two algorithms. **Figure 23** illustrates the progression of the NPV of 10 optimization runs for both algorithms along with their average.

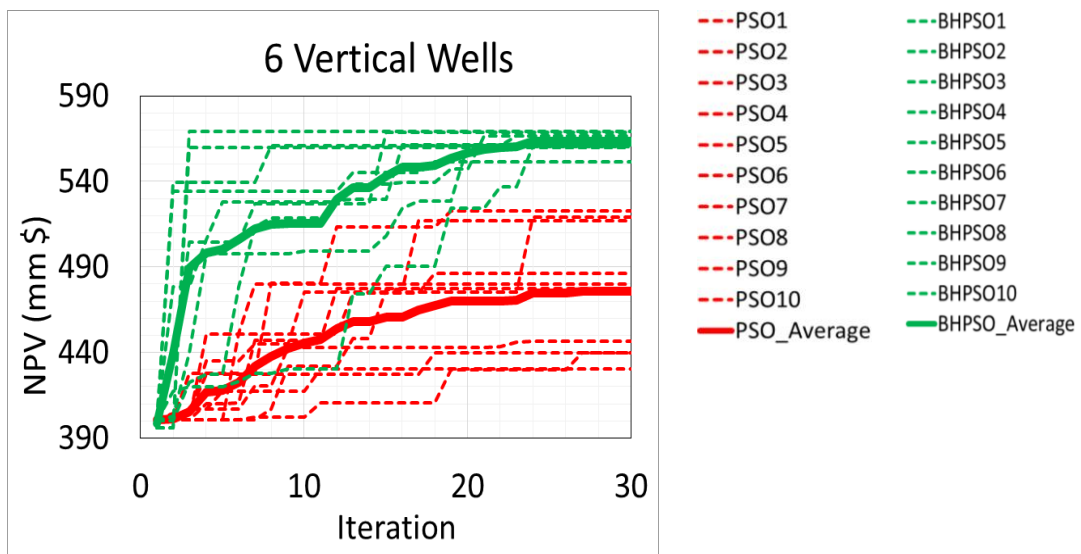


Figure 23 Performance comparison between BHPSO and PSO with a swarm size of 10 particles on Problem 1. BHPSO performance is represented by the green curves while the standard PSO is represented by the red curves.

As expected, BHPSO is also outperforming PSO in all optimization runs.

Similar to the results attained in Problem 1, NPV with BHPSO reaches an average of

561 million US dollars (MM US\$) while the average for PSO is 475 MM US\$; 18.5 % less than the result of the BHPSO. On average, BHPSO achieved a 40 % improvement over its first trial while PSO has achieved a 17.5 % improvement over the same first trial. Once again, BHPSO shows consistency in the final results as there is a minimal difference (3 %) between the maximum and the minimum trials unlike PSO where the difference reached 18 %.

3.3 Summary

Well placement optimization is a very challenging task in field development planning as it involves a large number of optimization variables resulting from the multidimensional space of well parameters. Manual assessment of the permutation of these variables yields an excessively large number of scenarios and, hence, is practically infeasible in the process of field development planning.

In this chapter, we introduce a new hybrid evolutionary optimization method; the Black Hole Particle Swarm Optimization (BHPSO) for simultaneously optimizing well count, location, type, and trajectory. For each particle in a BHPSO “iteration”, the location of the first producer is identified using Particle Swarm Optimization (PSO) based on the NHCT map. The remaining wells (producers and injectors), whose number is also potentially decided by PSO as an optimization parameter, are then automatically and optimally placed using BHO where wells are automatically and optimally placed using primarily a net hydrocarbon thickness (NHCT) map. The NHCT map is updated after every well placement by eliminating a disk (black hole) of a radius defined by the well spacing. Different radii are used to accommodate producers and injectors. For

horizontal wells, once the heel/toe of the well is placed the method identifies the azimuth corresponding to a maximum cumulative NHCT value.

The computational complexity of the proposed method is, thus, independent of the number of optimized wells. This drastically reduces the number of optimization parameters and, hence, the computational requirement to converge to an optimal solution. The proposed method is systematically and thoroughly validated using the publicly available synthetic field (Olympus) that is inspired by a green oil field in the North Sea and developed for the purpose of a benchmark study for field development optimization. Results show a systematically superior performance of the proposed BHPSO algorithm compared to the standard PSO.

CHAPTER 4

WELL PLACEMENT OPTIMIZATION UNDER GEOLOGICAL UNCERTAINTY

In this chapter, we present a novel method for selecting a representative subset of geological realizations to be used within the BHPSO workflow. We first introduce the Statistical map along with the workflow that was considered for selecting a representative subset of realizations. Next, we apply the presented methodology on the Olympus challenge and compare its results with two popular scenario reduction methods: the RMfinder and the Kmeans – kmediods methods

4.1 Methodology

A “robust” field development plan can be generally obtained via two different optimization approaches: (1) “Scenario-Based Optimization” or (2) “Robust Optimization”. Both approaches require the selection of a subset of subsurface realizations to reduce the computational load of the optimization process. Nevertheless, the main difference between the two approaches lies in the technicalities of the optimization workflow. Scenario-based optimization entails optimizing well placement over each selected realization on its own and thereby resulting in a number of optimal development scenarios. On the other hand, “robust” optimization involves a simultaneous optimization of well placement over all selected realizations, hence resulting with a single optimal development scenario. Each of the above approaches has its pros and cons [92]. However, in this work, the scenario-based optimization approach was adopted as it fits the nature of BHPSO: the core optimization algorithm.

Additionally, scenario-based optimization results with several optimal scenarios allow assessment of several statistical measures (e.g., P90 NPV, average NPV and standard deviation) in the comparative analysis.

The ultimate objective of a reservoir engineer/geoscientist is to reach an optimal development scenario that is robust to uncertainty within a project timeframe. The subsurface uncertainties translate into a set of reservoir simulation models (realizations) that can be derived geostatistically or based on multiple geological concepts. Selecting a development plan based on one subsurface realization (e.g., “base case”) may be sub-optimal and may imply unbearable risks on the project economics especially for capital intensive projects in an era of oil price volatility. Consequently, the project “driving value” (e.g., NPV and COP) should meet/exceed specific prescribed targets not only according to a “base case” subsurface realization, but also on a majority of subsurface realization, e.g., the P90 realization; that is there is a 90% that the proposed development plan meets its objectives. All should be done within a project decision timeframe, a typical challenge in real field development planning projects.

We addressed this challenging objective through a “pragmatic” optimization workflow that combines the BHPSO algorithm and the SNHCT map. The main contribution of this work is the introduction of the statistical NHCT map and its combination with BHPSO as a drastically efficient method to address uncertainty in well placement optimization. In the following, we describe the workflow of the proposed SNHCT map and elaborate on the process through which BHPSO and SHNCT are combined to meet the defined objectives.

4.1.1 The Statistical NHCT map

The NHCT map is a 2D map that delineates the “sweet spots” of the reservoir hence making it an efficient tool for placing wells. The *NHCT* value is calculated for each cell in the aerial direction of the simulation model (i, j coordinates), as follows:

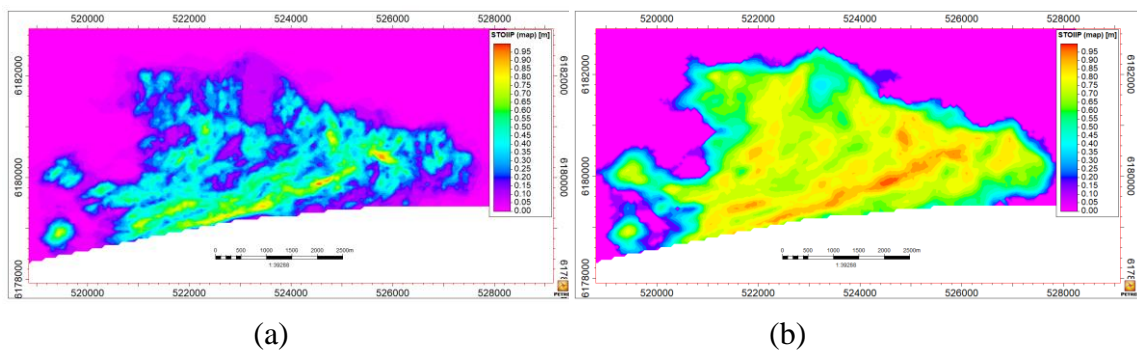
$$NHCT_{ij} = \sum_{k=1}^{N_k} \varphi_{ijk} NTG_{ijk} (1 - Sw_{ijk}) Z_{ijk} \quad (9)$$

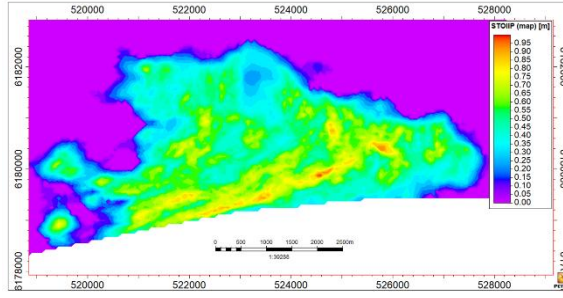
Where Z_{ijk} , Sw_{ijk} , φ_{ijk} and NTG_{ijk} , denote the thickness, water saturation, porosity and net-to-gross of the grid block cell ijk . $NHCT_{ij}$ has the same unit as Z_{ijk} , that is meter.

$NHCT_{ij}$ is indirectly related to the following properties:

1. Height above free water level.
2. Porosity and permeability.

To delineate statistical sweet spots among all defined geological realizations, we combine all maps into **Error! Reference source not found.** the so-called statistical $NHCT_{map}$, $SNHCT_{map}$ depicted in **Figure 24**. This concept stands on two pillars: 1) the average $NHCT_{map}$ and the P90 $NHCT_{map}$.





(c)

Figure 24 Statistical NHCT map with $b = 1$ (a), $b = 0$ (b) and $b = 0.5$ (c).

The NHCT for each cell on each of these maps is calculated as follows:

$$\text{Average NHCT}_{ij} = \frac{\sum_{r=1}^R (\text{NHCT}_{ij})_r}{r} \quad (10)$$

where r is the index of each geological realization and R is the total number of geological realizations.

P90 NHCT_{ij} is a NHCT value of a particular ij cell defined in such a way that there is 90% chance that the assigned value is exceeded within the set of NHCT values of all geological realizations in this particular cell.

The average NHCT_{map} defines areas with high average NHCT while P90 NHCT_{map} identifies areas with high 90th percentile value; that is areas with more than 90% chance of occurrence of a sweet spot. Combining the two maps marks well placement areas with high likelihood of success. For generating the $\text{SNHCT}_{\text{map}}$, both the average and the P90 NHCT maps are normalized and a user-defined variable b between 0 and 1 is incorporated to different map combinations. Hence, the NHCT for each cell on the $\text{SNHCT}_{\text{map}}$, is calculated as follows:

$$\text{SNHCT}_{ij} = b \times (\text{P90 NHCT}_{ij})_{\text{Norm}} + (1 - b) \times (\text{Average NHCT}_{ij})_{\text{Norm}} \quad (11)$$

Therefore, $b = 1$, $b = 0.5$ and $b = 0$ represent the P90 NHCT map, a combined version of the P90 and average NHCT map and the average NHCT map, respectively.

4.1.2 Workflow

The workflow of the scenario reduction method workflow implemented in this work is described in **Figure 25**. In the proposed method, we aim to maximize the average and P90 NPV of the field using a single geological realization (reservoir model). BHPSO is a map guided optimization algorithm, which requires a quality map in addition to a representative simulation model. The $SNHCT_{map}$ is used as the main quality map for the optimization process while the representative simulation model is selected using the following workflow:

1. Generate the $NHCT_{map}$ for each defined geological realization.
2. Generate the average $NHCT_{map}$ and the P90 $NHCT_{map}$ from all geological realizations.
3. Calculate the “Absolute Difference” map for each realization for both the P90 and the average $NHCT_{map}$. The “Absolute Difference” map is the difference between the $NHCT$ of a realization and the relevant $NHCT_{map}$. Each cell on the Absolute Difference $NHCT$ map of a particular realization r is calculated as follows:

$$(\text{Absolute difference } NHCT_{ij})_{Average,r} = \left| \text{Average } NHCT_{ij} - NHCT_{ij,r} \right| \quad (12)$$

$$(\text{Absolute difference } NHCT_{ij})_{P90,r} = \left| P90 \text{ } NHCT_{ij} - NHCT_{ij,r} \right| \quad (13)$$

4. Calculate the total absolute difference ($t_{P90,r}$, $t_{Average,r}$) for each realization relative to the P90 and average $NHCT_{map}$ respectively:

$$t_{Average,r} = \sum_{i=1}^{I_{max}} \sum_{j=1}^{J_{max}} (\text{Absolute difference } NHCT_{ij})_{Average,r} \quad (14)$$

$$t_{P90,r} = \sum_{i=1}^{I_{max}} \sum_{j=1}^{J_{max}} (\text{Absolute difference } NHCT_{ij})_{P90,r} \quad (15)$$

Where I_{\max} and J_{\max} denote the total number of cells in the I and J directions, respectively.

5. Normalize the total absolute difference ($t_{P90,r}$, $t_{Average,r}$)

$$(t_{Average,r})_{\text{normalized}} = \frac{t_{Average,r} - \min(t_{Average,r}, r \in R)}{\max(t_{Average,r \in R}) - \min(t_{Average,r}, r \in R)} \quad (16)$$

$$(t_{P90,r})_{\text{normalized}} = \frac{t_{P90,r} - \min(t_{P90,r}, r \in R)}{\max(t_{P90,r \in R}) - \min(t_{P90,r}, r \in R)} \quad (17)$$

6. Calculate the combined absolute difference ($t_{c,r}$) for each realization

$$t_{c,r} = b \times (t_{Average,r})_{\text{normalized}} + (1 - b) \times (t_{P90,r})_{\text{normalized}} \quad (18)$$

7. Select the subsurface realization with the lowest $t_{c,r}$.

8. Run BHPSO using the reservoir model corresponding to the selected subsurface realization along with the generated

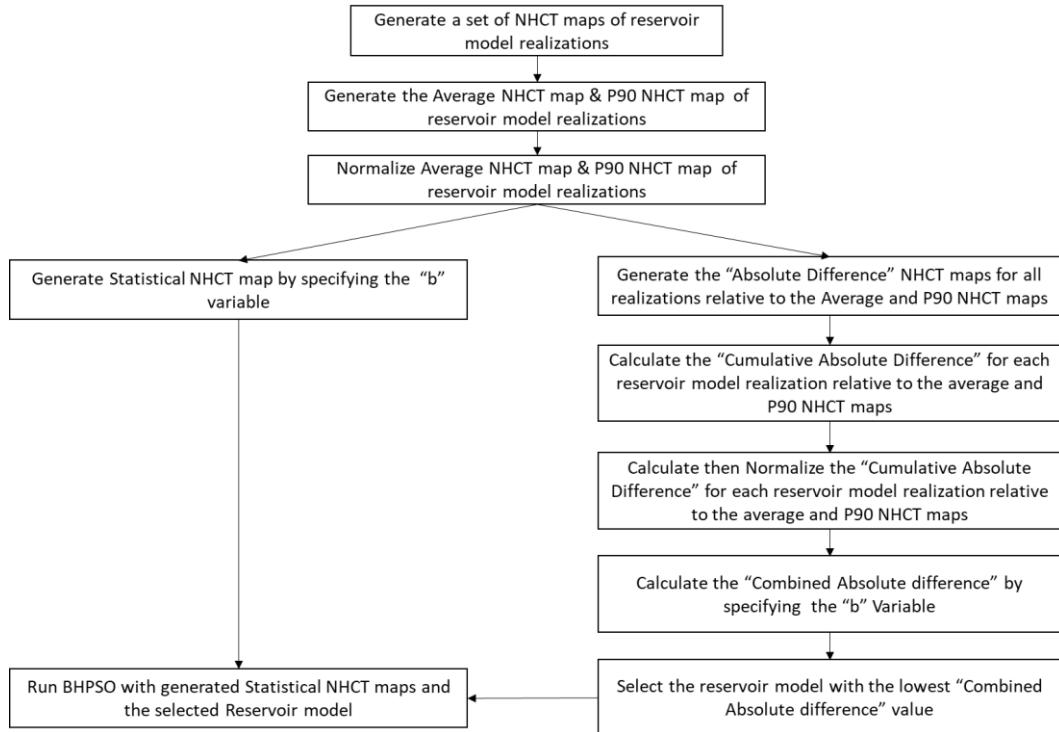


Figure 25 Summary of the proposed optimization under uncertainty workflow

4.1.3 Rmfinder Scenario Reduction Method [12]

RMfinder presents a mathematical model developed to model the representativeness of a subset of realizations with respect to the full set of realisations. The model is optimized using a greedy algorithm towards identifying the representative realizations. The objective function¹ of the model is a combination between three main criteria that when minimized, leads to high quality representative realizations.

$$F_{\text{tot}} = q_1 F_{\text{cross}} + q_2 \cdot F_{\text{risk}} + q_3 F_{\text{atr}} \quad (19)$$

Where, F_{cross} , F_{risk} and F_{atr} are the cross-plot objective function, the risk curve objective function, and the attribute – level objective function and q_1 , q_2 and q_3 are constants representing the weighting factor of each objective function.

The Cross Plot Objective Function (F_{cross}) assesses the representativeness of the selected realizations on the cross plots of four simulation variables as illustrated in **Figure 26**. These variables are the Net Present Value (NPV), Oil cumulative production (Np), Water cumulative production (Wp) and Oil recovery Factor (ORF) of a development scenario optimized to a base case (P50) reservoir model. At each iteration, the optimization algorithm selects certain realizations, and allocate to each of the selected realizations, a cluster of realizations based on the closeness in terms of Euclidian distance to the relevant realization. After allocating a cluster of realizations to each selected realization, the cross-plot function of a certain projection is calculated as the sum of the Euclidian distances between the cross plots properties of the selected realization and those of its allocated realizations. This is conducted for all six projections and the results are summed to represent the total cross plot function as follows:

$$F_{cross}^{total} = q_4 \cdot F_{NPV.N_p}^{cross} + q_5 \cdot F_{NPV.W_p}^{cross} + q_6 \cdot F_{N_p.W_p}^{cross} + q_7 \cdot F_{ORF.W_p}^{cross} + q_8 \cdot F_{N_p.ORF}^{cross} + q_9 \cdot F_{NPV.ORF}^{cross} \quad (20)$$

Consequently, the subset of realizations with less F_{cross}^{total} is deemed more representative. Note that the weights ($q_4 \dots q_9$) assigned for each component of the overall objective function allow the user to lend an importance to any aspect of the optimization procedure.

The Risk Curve Objective Function (F_{risk}) assesses the spread in of the representative realizations on the cumulative probability curve of the whole set and is thus applied on a variable-to-variable basis, with an adequate representative model showing equal distancing between its comprising realizations in the risk curve. F_{risk} of indicator a is calculated as follows:

$$F_{risk}^a = \sum_{i=0}^n d_i^2 \quad (21)$$

Where d_i is the Euclidian distance between two consecutive realizations on the cumulative curve function and n is number of selected realizations. Hence, with less F_{risk}^a , the difference between distances decreases, allowing for more representative realizations to be selected (**Figure 27**). The risk curve objective function is the sum of F_{risk} of the main indicators:

$$F_{risk}^{total} = q_{10} \cdot F_{NPV}^{risk} + q_{11} \cdot F_{W_i}^{risk} + q_{12} \cdot F_{W_p}^{risk} + q_{13} \cdot F_{N_p}^{risk} \quad (22)$$

Where F_{NPV}^{risk} , $F_{W_i}^{risk}$, $F_{W_p}^{risk}$ and $F_{N_p}^{risk}$ respectively represents the risk curve function of the Net Present Value, Cumulative Water Injection, Cumulative Water Production and Cumulative Oil Production. Note that additional indicators could be incorporated based on user knowledge and experience. The weights assigned for each

component of the overall objective function allow the user to lend an importance to any aspect of the optimization procedure.

The Attribute - level Objective Function (F_{atr}) assesses the relative frequency of the attributes within both the RM and the original set, in other words, the probability of an attribute value lying within a level must ideally be the same for the RM as it is for the original set. This aspect is represented by the use of F_{dist} , the distribution function, and a corresponding penalty is assigned via $F_{penalty}$ in case an attribute level is not present within the RM, as that would surely mean the representative set is not as close to the original set as can be. As such, the attribute function is obtained by the following:

$$F_{atr} = \left(1 - F_{penalty}(R)\right) \cdot F_{dist} \quad (23)$$

With the distribution function as:

$$F_{dist} = \sum_{a \in \{l_1, l_2, \dots, l_n(a)\}} (\text{relfreq}(a, l, RM) - \text{relfreq}(a, l, R))^2 \quad (24)$$

And the penalty function as:

$F_{penalty}(R) = 0$. If all attribute-level pairs occur at least one time in the RM for every scenario in R

$F_{penalty}(R) = 1$: All other cases

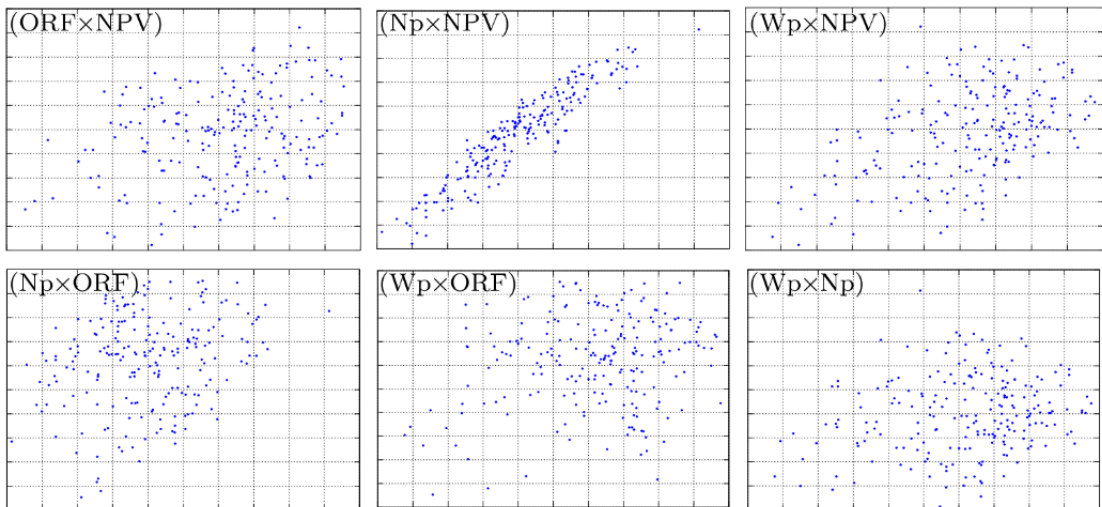


Figure 26 Cross plots for a number of simulated realizations. Extracted from [36]

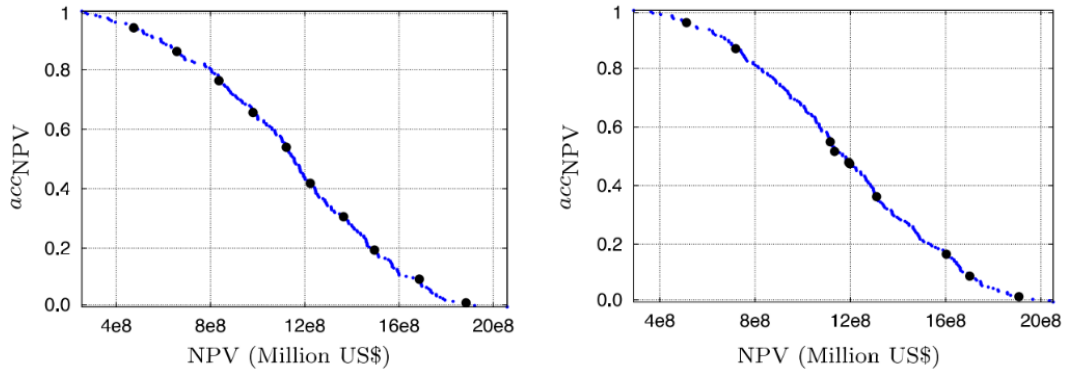


Figure 27 Good (left) vs. poor (right) selection of 10 representative realizations using the risk curve objective function. Extracted from [36].

4.1.4 k-Means – k-Medoids Clustering Scenario Reduction Method [14]

This method combines two clustering methods; k-means and k-medoids that work back-to-back for selecting a representative subset of realizations. k-means aims to partition n realizations into k clusters in which each of the realizations belongs to the cluster with the nearest mean. Following the k-means application, k-medoids marks one realization from each cluster as the cluster representative. The nominated realization shall have the least sum of the Euclidean distances between its measuring criteria and those of the remaining realizations.

The main contribution of this work does not lie in the clustering method itself but in the measuring criteria that is used. In this work, purely flow-based criteria were used since, as was concluded by Shirangi [31], is a good representation for different well setting configurations and is easy to compute. As such, the first step is the creation of the flow response vectors, where each vector is representative of the flows within a specific realization. The flow response vector is computed from running the P50 optimal scenario over the full set of realizations. The response vector for realization m_j ,

part of the original set $M_{\text{full}} = [m_1, m_2, m_3, \dots, m_{N_R}]$. In this instance, N_R is the total number of realizations in the original set (with $n_r \ll N_R$) and x is the well parameter vector, which defines variables such as BHP.

The low dimensional vector is defined as follows:

$$r_j(x, m_j) = [q_j^1, q_j^2, \dots, q_j^{N_w}]^T \quad (25)$$

Where, N_w is the number of wells. As such, for each well, depending on its type, the flow response vector can be represented as follows for a producer well k using the cumulative amounts of water production of water and oil:

$$q_j^k = [Q_{o,1}^k, Q_{o,2}^k, \dots, Q_{o,n_t}^k, Q_{w,1}^k, Q_{w,2}^k, \dots, Q_{w,n_t}^k] \quad (26)$$

Where, n_t is the number of time intervals chosen for the response vectors whereas $Q_{o,i}^k$ and $Q_{w,i}^k$ represent the amount of oil and water produced within the i^{th} time interval.

For a producer well however, only the injected water is considered:

$$q_j^k = [Q_{w,1}^k, Q_{w,2}^k, \dots, Q_{w,n_t}^k] \quad (27)$$

Having obtained the flow response vectors, the low dimensional vector $r_j(x, m_j)$ is obtained for all realizations, and are then added together to form the flow-based feature matrix Z_f as follows:

$$Z_f = \begin{pmatrix} r_1(x_0^1) & \cdots & r_{N_r}(x_0^1) \\ \vdots & \ddots & \vdots \\ r_1(x_0^b) & \cdots & r_{N_r}(x_0^b) \end{pmatrix} \quad (28)$$

With b is the number of different well parameter vectors used. In this work, b was defined to be one as we are only using the base case for selecting the subset of realizations same as in [31].

Having obtained Z_f , the matrix is normalized by the number of rows and then each row is normalized by the standard deviation. Afterwards, the clustering methods are applied, in this instance k-means clustering is used in order to subdivide the total set

of realizations into k clusters, then k-medoids clustering is used to extract the k medoids, in other words, extract one realization per cluster as being representative.

Below steps summarize the method:

1. The creation of flow response vectors for each realization.
2. The use of these flow response vectors for the creation of the Z_f matrix.
3. The normalization of the matrix.
4. Start of the iteration.
5. The use of k-means clustering to obtain n clusters of realizations.
6. The use of k-medoids clustering in order to obtain one realization per cluster.
7. Calculation of the difference between the original set and the smaller selected set.

4.2 Results and Discussion

First, we employ the proposed method and conduct a sensitivity analysis on the “b” parameter. Then, we apply the two scenario reduction methods (k-means – k-medoids clustering method [14] and RMfinder method [12]) and perform a comparative analysis between our proposed method and the results obtained with these scenario reduction methods. The comparative analysis is conducted on Olympus Upper of the Olympus field (Olympus Upper) since it involves is associated with high uncertainty in the spatial distribution of reservoir properties. A description of the Olympus field is provided in Chapter 2.

4.2.1 Application of the Proposed Method – The Statistical NHCT map

SNHCT_{map} guides BHPSO in optimizing over a “single” representative realization, resulting with a “statistically” robust development scenario over the full set of realizations. This is a novel concept that eliminates any prior simulation work and drastically reduces the simulation requirements needed to converge towards an optimal solution.

For selecting a “representative” realization, realizations are ranked according to their combined absolute difference value ($t_{c,r}$). Realizations with low $t_{c,r}$ are deemed similar to the corresponding SNHCT maps in terms of the spatial distributions of the NHCT values. Three different cases of the “b” parameter: $b = 0$, $b = 0.5$ and $b = 1$ were tested. Realization U50 achieved the lowest $t_{c,r}$ and ranked first with $b = 1$ (P90 NHCT map). However, U22 realization achieved the lowest $t_{Average,r}$ and a relatively low $t_{P90,r}$, and thereby ranked first for both $b = 0$ and $b = 0.5$. Table 1 illustrates the ranking of realizations with the three considered “b” parameters.

BHPSO was run on the selected realizations, each using its dedicated SNHCT_{map}. Results are presented in Table 3 and **Figure 28**. The algorithm converged after 30 iterations (240 simulations) on U50 ($b = 1$) realization, 35 iterations (280 equivalent simulations) on U22 ($b = 0$) realization and 24 iterations (192 simulations) on U22 ($b = 0$) realization (**Figure 28a, b and c**). Three different optimized well placement solutions were, consequently, obtained using the three cases. Each of these solutions was, then, tested on the 50 realizations and the cumulative density function for the NPV was built for each of the three cases as illustrated in **Figure 29**.

The three optimization runs resulted on an NPV that is close to the statistical measures obtained over the full set of realizations. The optimization runs with statistical maps on U22 converged on 0.98 and 0.95 billion USD while obtaining an average NPV of 0.98 and 0.99 billion USD when run over the full set of realizations. Furthermore, the U50 ($b = 1$) optimization run converged at 0.82 billion USD while obtaining a P90 NPV of 0.75 billion USD (**Figure 29**). This indicates that the selection methodology is fit for purpose since the dynamic behavior of the selected realizations is similar to that of the whole set of realizations on average.

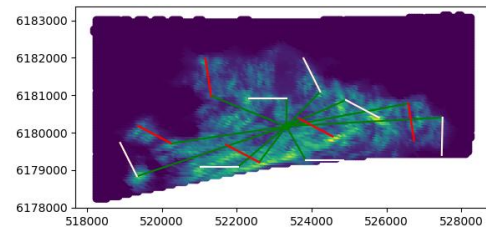
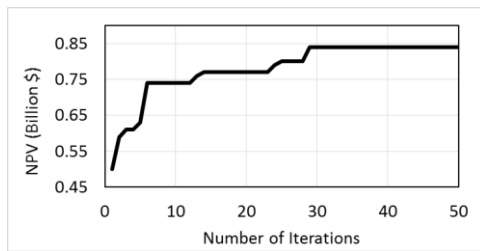
The optimal scenario obtained over realization U50 was statistically dominant over the other two-optimization runs, resulting with a higher average NPV and P90 NPV. This scenario concluded with highest well count (12 wells), and the largest horizontal length of 1000 m. Both factors induced a high-pressure support, hence accelerating production. Despite the large horizontal length, the obtained well spacing, as shown in **Figure 28a**, was large enough to slow down early water production concluding with a high NPV among low and high permeable realizations. The optimal scenarios obtained over realization U22 were close. The U22 ($b = 0.5$) scenario had 1% higher average NPV and 2% lower P90 NPV than U22 ($b = 0$). U22 ($b = 0$) had three wells less than U22 ($b = 0.5$) scenario, which gave it an advantage in terms of cost and disadvantage in terms of pressure support and early production. Although the U50 ($b = 0$) scenario is outperforming the U22 scenarios ($b = 1$, $b = 0.5$), it is still early to take stand on that since the work was applied to a single field case. Further work is needed to test different “b” values on other field cases.

Table 2 Realizations ranking using our proposed method with three different values of the “b” variable: 0, 0.5 and 1.

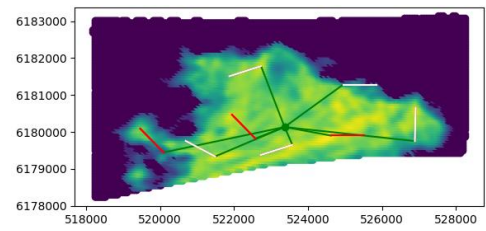
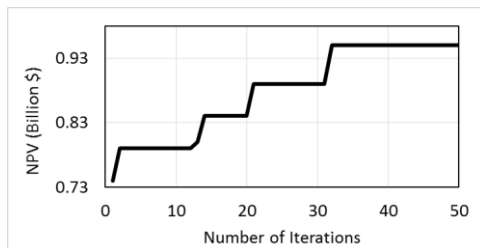
Ranked Realizations (b=0)	$t_{c,r}(b=0)$	Ranked Realizations (b=0.5)	$t_{c,r}(b=0.5)$	Ranked Realizations (b=1)	$t_{c,r}(b=1)$
22	0.00	22	0.14	50	0.00
9	0.07	50	0.24	1	0.07
14	0.08	1	0.27	41	0.15
19	0.13	40	0.28	45	0.21
2	0.16	5	0.28	39	0.22

Table 3 Decision variables and statistical measures (P90 and average NPV) of optimal scenarios obtained over the three selected realizations by our proposed method.

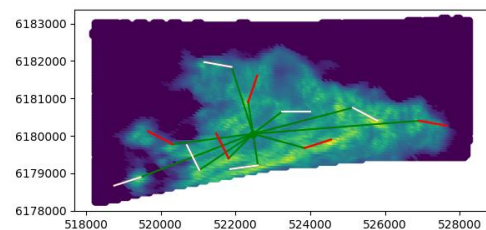
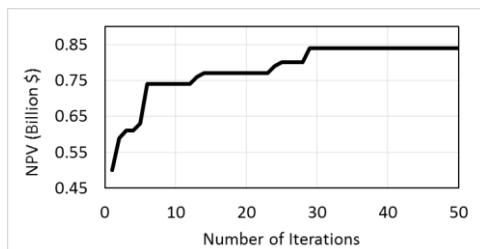
Selected Realization	Well count	Producer Count	Injector Count	Horizontal Length	Average NPV (Billion \$)	P90 NPV (Billion \$)
U22 (b = 0)	8	5	3	901	0.99	0.6
U22 (b =0.5)	11	6	5	746	0.98	0.61
U50 (b=1)	12	7	5	1000	1.08	0.75



Statistical *NHCT* ($b = 0$); Realization U50



Statistical *NHCT* ($b = 1$); Realization U22



Statistical *NHCT* ($b = 0.5$); Realization U22

Figure 28 BHPSO performance on Olympus Upper for the selected realizations (left) and optimal well placement on different versions of the statistical NHCT map (right) – Part C. White and red well trajectories depict producers and water injectors, respectively. Green depicts the part of the trajectory starting from the platform position until the point of entry

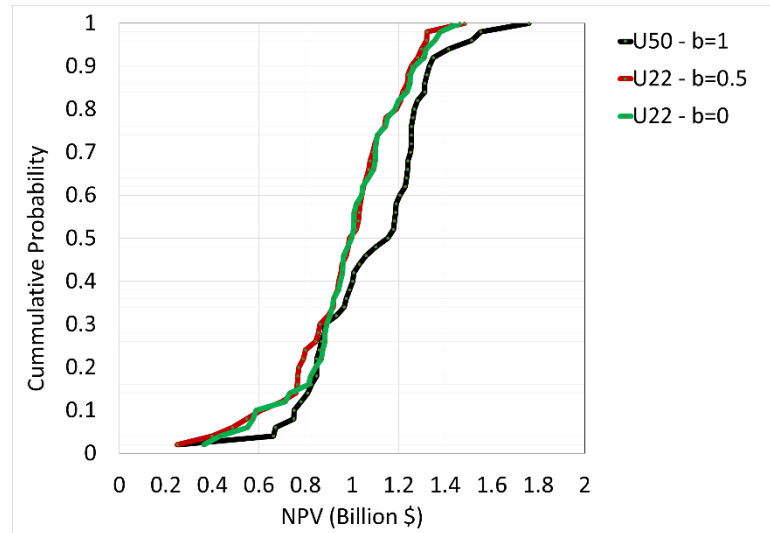


Figure 29 NPV cumulative density function (Risk curve) of the three selected realizations by our proposed method.

4.2.2 Application of the Scenario Reduction Methods

Here we demonstrate that our proposed method yields superior results compared to two scenario reduction methods despite the drastically lower computational requirements by our proposed method: k-means – k-medoids clustering method [14] and RMfinder method [12].

Ten realizations were selected using the RMfinder and k-means – k-medoids methods (5 realizations each) and were optimized using BHPSO, each with its own $SNHCT_{map}$ resulting with 10 optimal development scenarios. First, the optimization work of the base case of the Olympus Upper will be described. Then, the selection of the subset of realizations of the two methods is discussed and finally, the results of BHPSO optimization over the 10 selected realizations is analyzed and compared.

4.2.2.1 BHPSO Optimization of the P50 Realization

In contrast with the proposed method, both RMfinder and k-means-k-medoids clustering require an optimized base case development scenario in their application. The optimal scenario obtained is, then, run over the full set of realizations and relevant flow characteristics are extracted to be used as an input for applying both methodologies. For this, BHPSO was employed to optimize realization 12 of Olympus Upper (U12) corresponding to the P50 of STOIP with a fixed number of particles (8 particles) and a fixed number of iterations (50 iterations). These algorithmic parameters were used throughout all simulation work in this work since they were deemed sufficient for BHPSO to converge or provide a relatively high NPV. As can be shown in **Figure 30**, the algorithm converged after 22 iterations (176 equivalent simulations) reaching a maximum NPV of 0.99 billion USD. The optimal scenario included 11 wells (6 producers and 5 injectors) with an equal horizontal length of ~450 m (**Figure 30**). The resulting optimal development scenario was, then, run over the 50 realizations and the relevant results were generated to serve the requirement of the considered methods.

4.2.2.2 Selection of Realizations – RMfinder

Meira et al. [93] employed the RMfinder to the full Olympus benchmark case hence, some of their assumptions were considered in our application. The results of the optimized U12 described in the above section were used for applying the RMfinder. Since Olympus has no uncertain variables other than the 50 subsurface realizations built on different distributions of reservoir properties, the uncertain attribute levels, F_{penalty} and F_{atr} were omitted from the optimization process. Furthermore, we considered four

output variables in $F_{\text{cross}}(\tau')$: NPV, N_p , ORF and W_p and 15 variables in $F_{\text{risk}}(\tau')$: 4 field variables (NPV, N_p , W_p , W_{inj}), 6 injector productivity indices and 5 producer productivity indices. The weights were assigned in the same manner as in [93].

PSO was used as the core algorithm to minimize $F_{\text{tot}}(\tau')$. Due to the stochastic nature of PSO, the run was repeated 50 times with algorithmic parameters of 50 particles and 500 iterations. The results of the best 10 runs are illustrated in Table 4 and **Figure 31**. Combination of U2, U8, U15, U24 and U41 yielded the lowest $F_{\text{cross}}(\tau')$ marked is red in and were considered for running the Scenario based optimization workflow.

4.2.2.3 Selection of Realizations – k-means–k-medoids Clustering

The k-means – k-medoids clustering technique was applied to the flow measures obtained from realization U12 optimal scenario. A MATLAB k-means built-in algorithm was combined with an in-house iterative algorithm (k-medoids) for selecting 5 realizations. Since this algorithm often gets stuck in local optima, the run was performed for 500 iterations and repeated for 50 times. The combination with the minimum Euclidian distance was selected for running the scenario-based optimization workflow. The results of the best 10 runs are shown in

Table 5 . The final solution corresponded to U4, U5, U7, U9 and U15.

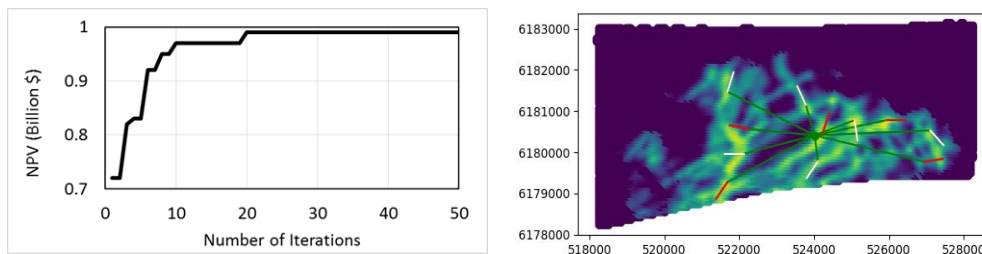


Figure 30 BHP SO performance (left) and optimal well placement (right) in Olympus Upper, Realization U12. White and red well trajectories depict producers and water

injectors, respectively. Green depicts the part of the trajectory starting from the platform position until the point of entry.

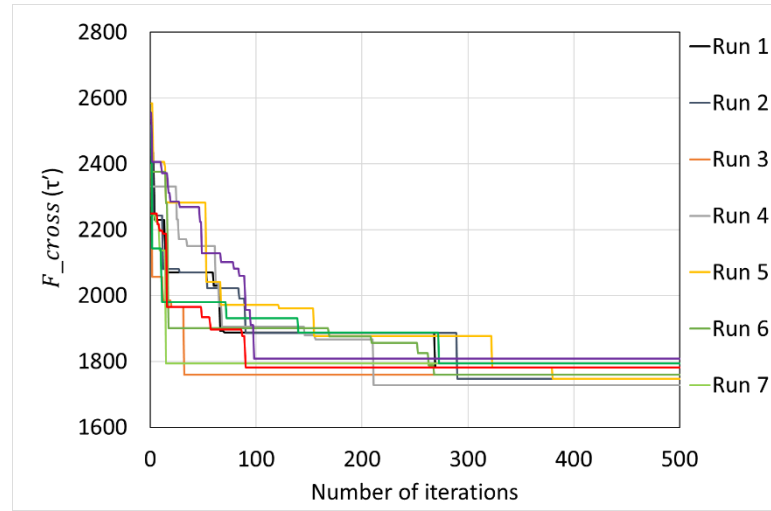


Figure 31 10 best PSO optimization runs of the RMfinder method with a swarm size of 50 particles.

Table 4 10 best realizations' combinations obtained using the RMfinder method. The best combination is marked in red.

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run7	Run8	Run9	Run 10
Realization 1	45	44	8	41	41	8	44	25	26	45
Realization 2	44	6	6	24	24	6	43	21	44	44
Realization 3	21	40	45	8	8	41	48	48	21	42
Realization 4	40	21	44	15	15	45	25	43	43	32
Realization 5	35	45	41	2	2	44	21	44	48	35
F(TOTAL)	1781	1747	1759	1727	1747	1759	1794	1794	1808	1782

Table 5 10 best realizations' combinations obtained using the k-means – k-medoids clustering method. The best combination is marked in red.

	Run1	Run2	Run3	Run4	Run5	Run6	Run7	Run8	Run9	Run10
Realization 1	2	4	4	4	1	4	4	4	1	4
Realization 2	5	9	6	5	5	5	6	5	4	5
Realization 3	8	11	8	7	7	7	8	7	7	8
Realization 4	13	15	9	9	8	8	9	11	10	9
Realization 5	7	20	10	15	14	12	10	32	32	23
Dissimilarity	0.88	0.84	0.93	0.79	0.96	0.95	0.93	0.80	0.91	0.87

4.2.3 Optimization on the Selected Realizations and Comparative Analysis

Since realization U15 is common for both methods, nine realizations were selected to for the scenario-based optimization workflow in the following section: U2, U4, U5, U7, U8, U9, U15, U24 and U41. BHPSO optimizes several variables including well count, producer – injector (P–I) ratio, location, horizontal length, and the spatial spread of wells that effect the distance from the platform to the point of entry. Nine optimal development scenarios were obtained using BHPSO, each was run using its corresponding $NHCT_{map}$. Then, these optimal development scenarios were run over the full set of realizations. Results are depicted in **Figure 31**, **Figure 32**, **Figure 33** and **Figure 34**. For each of the subsurface realizations, the left-hand figure shows the NPV progression as a function of the number of PSO iterations while the right-hand figure depicts the placement of the producers (in white) and injectors (in green) for the selected realization on its own $NHCT_{map}$. Most optimization runs converged in less than 20 iterations (160 equivalent simulations) with an NPV improvement over the first iteration ranging from 30 to 50%, which confirms the efficiency of BHPSO. Note that BHPSO is equipped with the black hole operator, which leads to a relatively good, engineering guided, well placement at every iteration hence, the first iteration will often obtain high value, which drastically affects the optimization run time. The optimal NPV reached varied between realizations due to differences in STOIP and differences in the obtained optimal well placement, which leads to different revenues and cost figures, respectively.

Although all the scenarios were optimized over their selected realizations, it was crucial to assess their performance over the full set of realizations to analyze and compare their statistical measures (P90 NPV and average NPV). **Figure 35** combines

the results via a cumulative density function of the obtained NPV over all realizations. Results summarized in Table 6 show that the statistical measures (P90 and average NPV) were not inline in ranking realizations. Some development scenarios obtained a higher average NPV but relatively lower P90 NPV than others, and here lies the role of the decision maker attitude towards risk taking in selecting the development scenario. The average NPV ranged between 0.63 and 1.01 billion US \$ while the P90 NPV ranged between 0.31 and 0.68 billion US\$ (**Table 6**). Although well count was varied between 8 and 14, the maximum number of feasible well count obtained was 12. This is in line with what was concluded in the pre-optimization work. In all scenarios, the producer – injector (P-I) ratio was equal or more than 1 except for the two scenarios with the lowest average NPV obtained using Realizations 15 and 5. As per the horizontal length, it ranged between 500 and 1000 m, with only 2 scenarios (U15 and U8) on the boundaries. Optimal scenario obtained using U2, was deemed the best scenario, yielding the same average NPV (1.01 billion \$) as that obtained by U7 but with 10% higher in the P90 NPV. The worst scenario was the one obtained using realization 15 resulting with an average NPV of 0.63 billion USD and P90 NPV of 3.1 billion USD (**Table 6**).

Towards a systematic analysis, the optimal scenarios were grouped based on their well count for evaluation purposes. In addition, we further analyzed the first and second ranked scenarios since the results were close with different well counts. Group 1, which includes U2, U4, U5 and U15, obtained a well count of 12 wells. U2 (**Figure 32b**) and U4 (**Figure 32c**) obtained a better statistical result than U5 (**Figure 32d**) and U15 (**Figure 32a**) with a higher P-I ratio. The field was simulated using equal production and injection rates in all scenarios. However, the number of injectors seemed

to boost the water breakthrough, especially in high permeable realizations, hence affecting the overall results. That is especially observed in realization U15 characterized by the lowest P-I ratio and low well spacing which, combined with a high permeability accelerates water breakthrough resulting with the worst outcome. Optimal scenario obtained using U2 had a 23% higher average NPV and 65% higher P90 NPV than the second ranked optimal scenario (U4) of this group. U2 had a higher P-I ratio with a greater horizontal length of 884, 174 m more than that obtained in U4. A larger horizontal length induces high pressure but speeds up water breakthrough, while a greater P-I ratio slows water breakthrough. This balance gave U2 an advantage over U4 in the overall result. On the other hand, the optimal scenario obtained with U5 had a lower statistical result than U4. In addition to the lower P-I ratio, we observe a larger spread of wells in U5 which reduced pressure support even though U5 had a larger horizontal length.

Group 2 (11 wells) included optimal scenario over realizations U41 (**Figure 33a**) and U24 (**Figure 33b**). Optimal scenario with U24 obtained 11.5% higher average NPV and 57% higher P90 NPV compared to U41. This result shows that the optimal scenario obtained using U24 is statistically dominant over that obtained using U41 (**Figure 33a**). In addition to the higher P-I ratio, U24 optimal scenario had a smaller horizontal length, which gave it an advantage in terms of the speed of the water breakthrough.

Group 3 (10 wells) included optimal scenarios over realizations U8 (**Figure 34b**) and U9 (**Figure 34a**). Optimal scenario on U8 obtained 2% higher average NPV but 18.8% lower P90 NPV than U9. This result shows that optimal scenario obtained on U8 is favoring the high permeable realizations, however, the U9 scenario is favoring the

low permeable realizations. This is justified by the difference in horizontal length between the two scenarios. U8 scenario had wells with relatively very low horizontal length of 500 m, ~380 m less than that obtained using U9. This favors high permeable realizations in terms of slowing water breakthrough but perform poorly in low permeable realizations that require a high-pressure support.

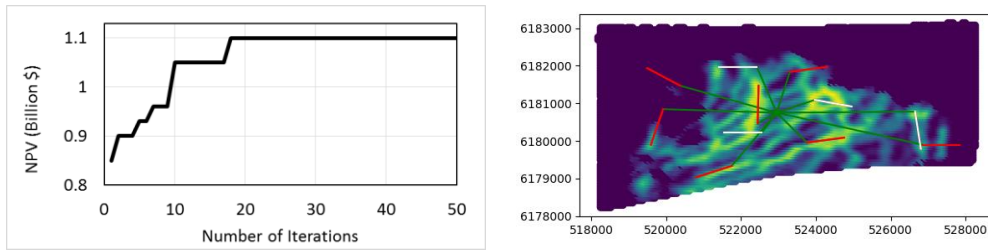
In comparing the first and second ranked scenarios (U2 and U7), we observed that U7 (**Figure 34c**) concluded with 9 wells, 3 wells less than that obtained in the top ranked scenario (U2). The low number of wells drastically decreased the cost of this scenario. However, the revenues associated to U7 were not high enough to outperform the top-ranked scenario corresponding to U2, but high enough to outperform the remaining scenarios, which also have a higher number of wells. This “contradiction” is mainly driven by the well location. A productive well in a given realization may be shut down in another realization due to different spatial distribution of properties between realizations. Hence, a well placement of 12 wells realization may be effectively lower for some of the realizations. It was very challenging to analyze this factor as the statistical quality of the location is defined by 49 other realizations and here lies the main contribution of our work.

While, on average, RMFinder provided better results in the demonstrated scenario-based optimization workflow compared to the k-means–k-medoids clustering technique, our proposed method outperformed both methods in terms of the optimality of the result and number of simulation runs. **Figure 36** combines the results of all generated optimal scenarios via cumulative density functions. The cumulative density function curves corresponding to the proposed methods lie on the rightmost side of the graph, denoting statistical dominance over other optimization runs. The best result

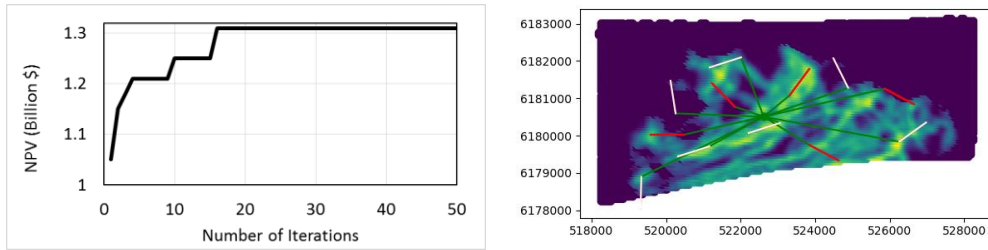
(U50) resulting from the proposed method obtained 7% higher average NPV and 10% higher P90 NPV compared to the best result (U2) among the nine selected scenarios. U2 development scenario obtained 2% and 2.5% higher average NPV and 11% and 10% higher P90 NPV than the U22 (b = 1) and U22 (b = 0.5) scenarios, respectively. Nevertheless, if when observing the 12, 14, 16 and 18 percentiles NPV (P88, P86, P84 and P82) on **Figure 36**, it can be noted that both, U22 (b = 1) and U22 (b = 0.5), have better NPV values than the U2 development scenario. This shows that even the second and third ranked development scenarios obtained with the proposed method, are likely more suitable for a risk averse decision maker or at least comparable to the best run obtained among all nine selected realizations.

Table 6 Decision variables and statistical measures (P90 and average NPV) of optimal scenarios obtained over the nine selected realizations by RMfinder and k-means – k-medoids methods. Note that Realization 15 is a common realization reducing the number of realizations from 10 to nine.

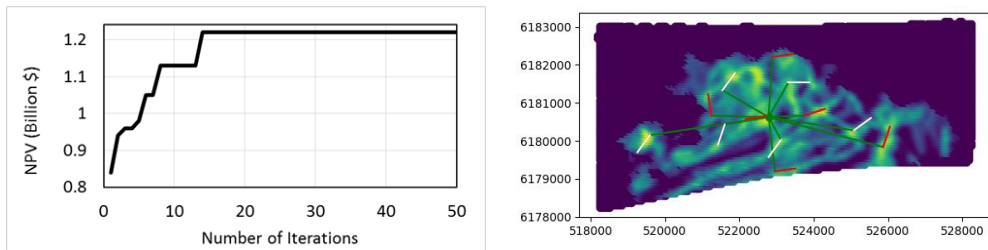
	Well Count	Producer Count	Injector Count	Horizontal Length	Average NPV (Billion \$)	P90 NPV (Billion \$)
U2	12	7	5	884	1.01	0.68
U8	10	5	5	500	0.92	0.48
U15	12	4	8	1000	0.63	0.31
U24	11	7	4	710	0.87	0.55
U41	11	6	5	897	0.78	0.35
U4	12	6	6	561	0.79	0.41
U5	12	5	7	635	0.75	0.41
U7	9	6	5	788	1.01	0.62
U9	10	6	4	879	0.9	0.57



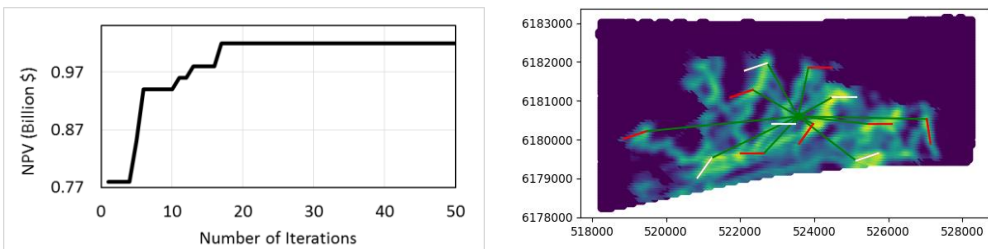
(a) – U15



(b) – U2

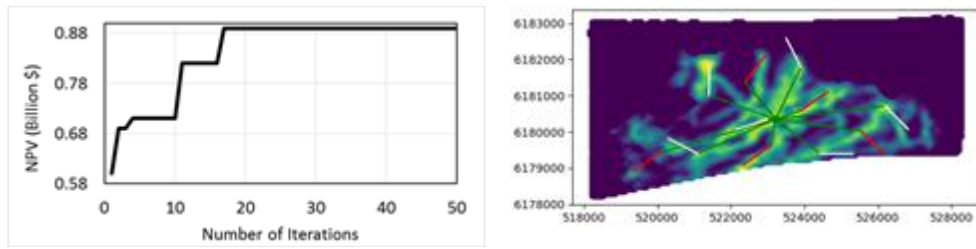


(c) – U4

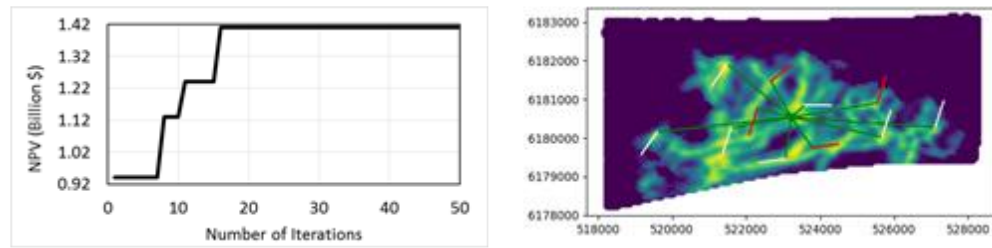


(d) – U5

Figure 32 BHPSO performance on Olympus Upper for Group 1 realizations (left) and optimal well placement on NHCT map (right) – Part A. White and red well trajectories depict producers and water injectors, respectively. Green depicts the part of the trajectory starting from the platform position until the point of entry

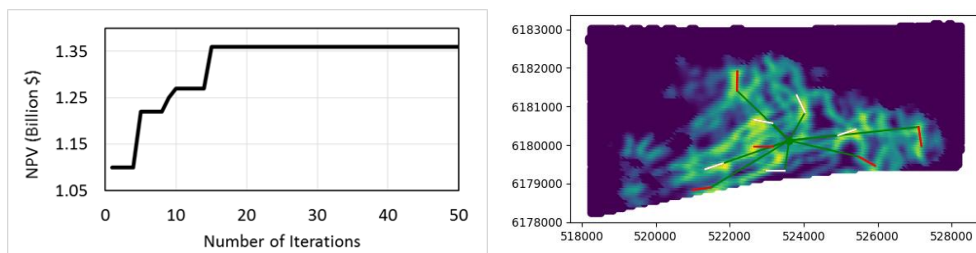


(a) – U41

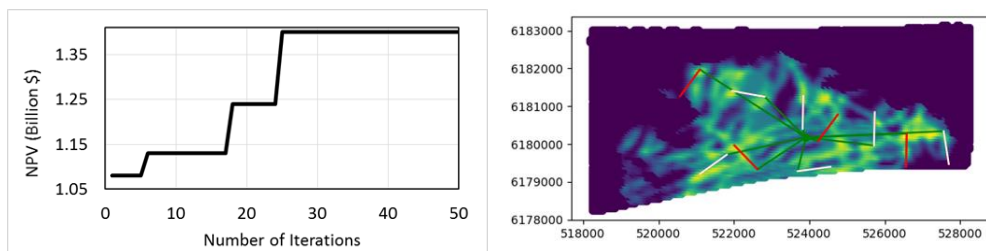


(b) – U24

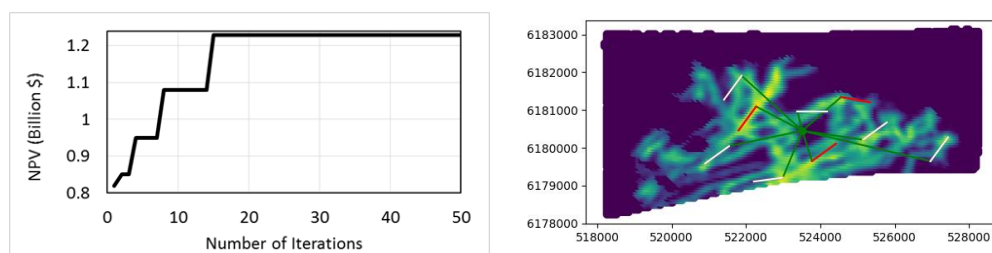
Figure 33 BHPSO performance on Olympus Upper for Group 2 realizations (left) and optimal well placement on NHCT map (right) – Part B. White and red well trajectories depict producers and water injectors, respectively. Green depicts the part of the trajectory starting from the platform position until the point of entry.



(a) – U8



(b) – U9



(c) – U7

Figure 34 BHPSO performance on Olympus Upper for Group 3 realizations (left) and optimal well placement on NHCT map (right) – Part C. White and red well trajectories depict producers and water injectors, respectively. Green depicts the part of the trajectory starting from the platform position until the point of entry.

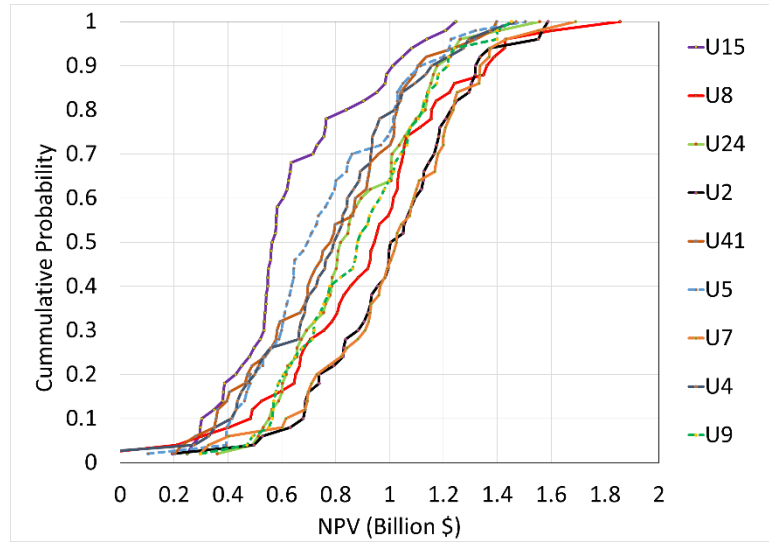


Figure 35 NPV cumulative density function (Risk curve) of the nine selected realizations by RMfinder and k-means – k-medoids methods.

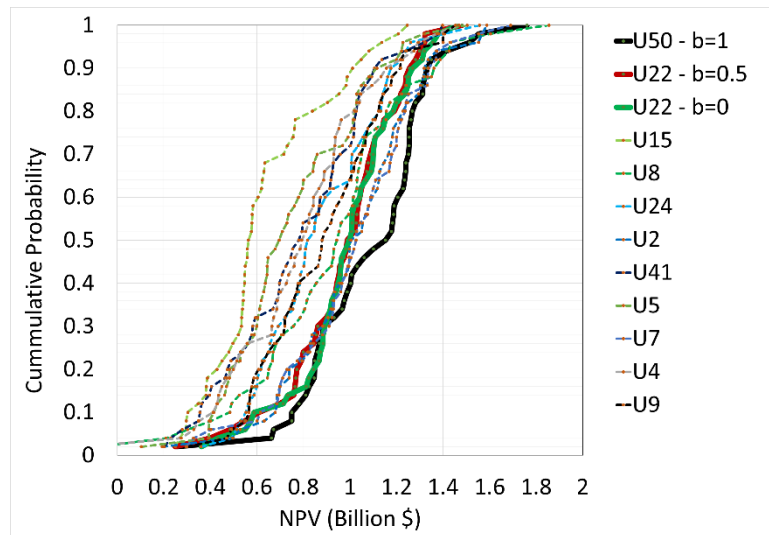


Figure 36 NPV cumulative density function (Risk curve) of all twelve optimal scenarios: Nine from RMfinder and k-means – k-medoids methods and three from our proposed method (corresponding to $b = 0$, $b = 0.5$ and $b = 1$). Dashed and solid curves depict the results of the RMfinder and k-means – k-medoids methods and our proposed method, respectively.

4.3. Summary

Investments in oil and gas projects are driven by critical field development decisions including well placement, which often significantly affect the projects' economics. Due to their typically high cost and inherently insufficient data (especially in green fields, or fields in their early stage of development), managing uncertainty is critical when optimizing a field development plan. The use of a single deterministic base case for hydrocarbon, both in place assessment and production forecasting is often misleading and leads to sub-optimal decisions. Consequently, robust field development plans require multiple geological realizations covering the range of uncertainty in reservoir properties and encompassing both multiple geological concepts and geostatistical properties distribution.

Typically, an objective function such as the average net present value (NPV) or the average cumulative oil production (COP) is optimized in order to select an optimal development scenario. Nevertheless, such an assessment can be computationally prohibitive, especially when using optimization methods require hundreds, often thousands of costly simulations over a single realization, a number that significantly increases when multiple realizations are involved. This study proposes a new method for well placement optimization under uncertainty, building on map-based evolutionary optimization technique: the black hole particle swarm optimization (BHPSO). The statistical net hydrocarbon thickness (SNHCT) map is introduced to guide the BHPSO algorithm; and hence, pragmatically account for uncertainty in the process of well placement optimization. We optimize well placement on the realization corresponding to the minimum difference between its NHCT map and the SNHCT map.

The SNHCT combines the average and the P90 NHCT maps; hence, assuring that the selected sweet spots for well placement are statistically the best with regard to the multiple subsurface realizations. The method is applied on the Olympus benchmark case and results are compared to two scenario reduction methods: RMfinder and k-means-k-medoids Clustering. Results show superior performance over the two methods in terms of optimality of the result and the required computational load.

CHAPTER 5

WELL TRAJECTORY OPTIMIZATION: A MACHINE LEARNING BASED APPROACH

In this chapter, we discuss the methodology used for optimizing well trajectory optimization then we study the results for the presented method compared to that of the traditional optimizers. We first describe the Bézier curve formulation along with the optimization workflow that was considered. Then we elaborate on the machine learning models used in this work along with their training process. Next, we compare the results of optimizing Bézier curves using the traditional optimizers with that of the proposed machine learning models.

5.1 Methodology

In this section, we first describe the Bézier curve methodology and then elaborate on the well trajectory optimization workflow, in addition to the optimization algorithms used. Finally, we present the ML problem formulation for well trajectory optimization.

5.1.1 Well Trajectory Modelling – The Bézier Curve

Bézier curves were introduced by Paul de Faget de Casteljay in 1959, a mathematician at Citroen, and patented by Etienne Bézier, an engineer at Renault who leveraged the Bézier curves to design automobile bodies. In 2017, Sampaio [55] was the first to introduce Bézier curves to the oil industry and employ the method for designing 3D well trajectories.

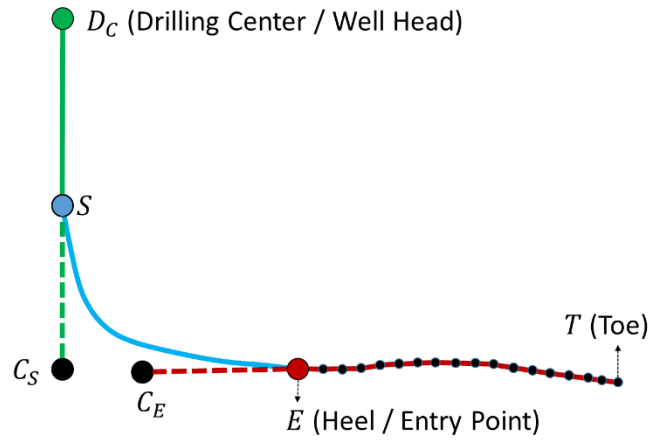


Figure 37 Illustration of the input parameters for well trajectory design using the Bézier curve.

Bézier curves can be used to embody special planar (2D) and nonplanar (3D) trajectories. A third order Bézier curve is illustrated in **Figure 38**. It is generated using the following mathematical formulation:

$$B(u) = (1 - u)^3 S + 3(1 - u)^2 u C_S + 3(1 - u) u^2 C_E + u^3 E \quad (29)$$

where u is a dimensionless parameter in the interval $[0,1]$, $E(x, y, z)$ and $S(x, y, z)$ denote the coordinates of the curve's end and starting points, respectively. $C_E(x, y, z)$ and $C_S(x, y, z)$ depict the control points at the starting and endpoints of the curve, respectively (**Figure 37**).

The number of points for the curve construction is a predefined input parameter to the problem, where each point is allocated a distinct value within the interval $[0, 1]$. The support lines of C_E and C_S are represented by the tangents to the trajectory \vec{t}_E and \vec{t}_S at points E and S , respectively. **Figure 38a** shows different structures of third order Bézier curves with fixed C_S and variable C_E positions ($C_{E1}, C_{E2}, C_{E3}, C_{E4}$). **Figure 38b** shows different structures of third order Bézier curves with fixed C_E position and variable C_S positions ($C_{S1}, C_{S2}, C_{S3}, C_{S4}$). The control point's position highly affects the shape of the curve. The direction of the \vec{SC}_S vector is governed by the azimuth and

inclination of the trajectory at point S whereas the direction of the $\overrightarrow{EC_E}$ vector is governed by the azimuth and inclination of the trajectory at point E .

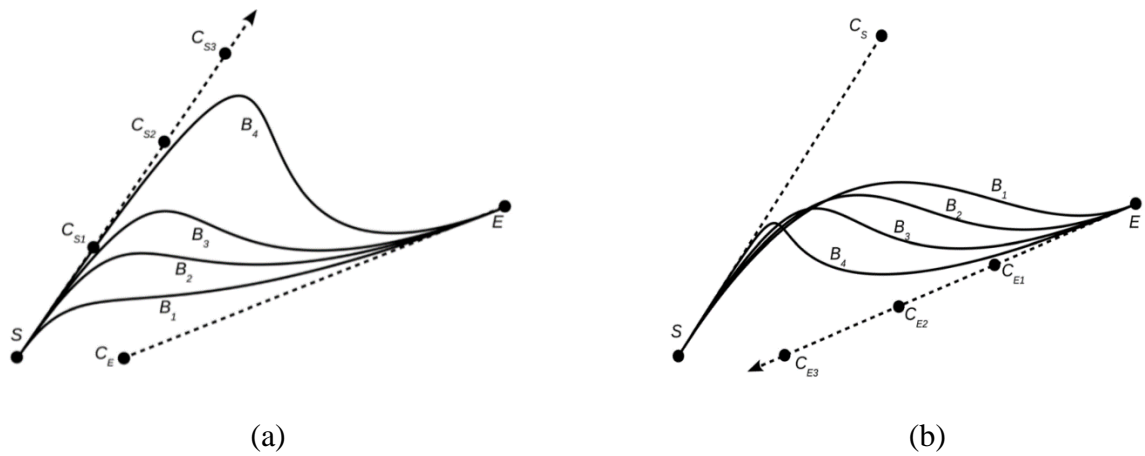


Figure 38 Third-order Bézier curves (a) with fixed C_E and varying C_S and (b) with varying C_E and fixed C_S . Extracted from [55].

Generally, the input data required to obtain a 3D well trajectory using Bézier curves [55] are the following:

- Location of the starting point (kickoff point) S of the trajectory
- Inclination (ϕ_S) and azimuth (β_S) of $\overrightarrow{SC_S}$
- Location of the end of the trajectory point of entry E
- The inclination (ϕ_E) and azimuth (β_E) of the reservoir target $\overrightarrow{EC_E}$

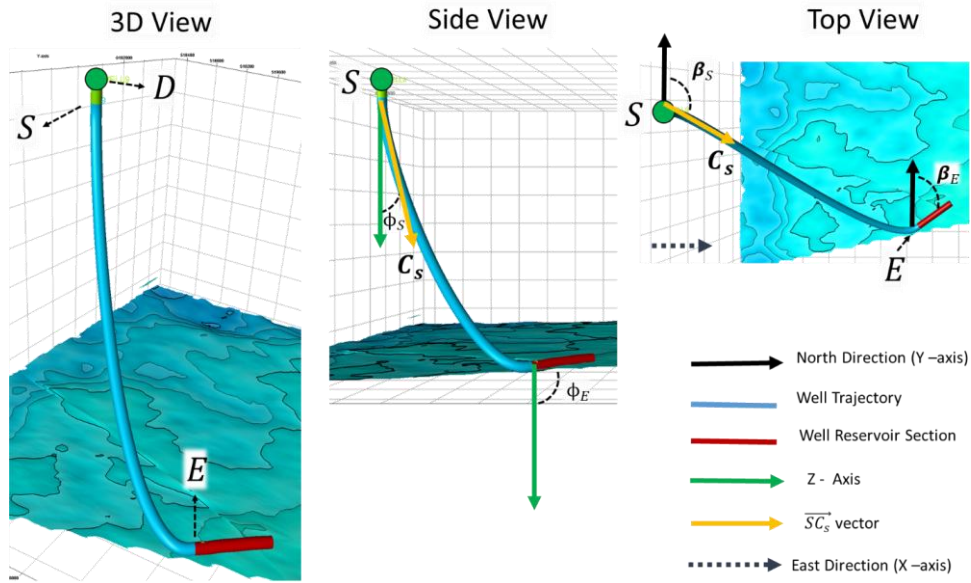


Figure 39 3D, side and top views of a typical 3D well trajectory associated with Bézier curve input parameters.

Figure 39 illustrates the different input parameters required for applying a general Bézier curve for well trajectory design. However, in well trajectory design, the inclination ϕ_s must always be defined at 0 degrees (C_s running along the Z-axis), since even a small inclination at this level can violate the DLS constraint. Furthermore, since C_s lies along the z-axis, the azimuth at β_s will have no affect on the position of C_s . This can be illustrated by computing the coordinates of unit vector \vec{t}_s (Equation 31), which result in (0, 0, 1). Furthermore, the inclination and azimuth at the target (ϕ_E, β_E) should be provided because the reservoir well sections are defined prior to the designing the well trajectory.

The azimuth and inclination at any point on the Bézier curve can be used to determine a unit tangent vector \vec{t} at the point by

$$\vec{t} = (\sin\phi\sin\beta, \sin\phi\cos\beta, \cos\phi) \quad (30)$$

Hence, the unit vector at the starting point \vec{t}_s is given by

$$\vec{t}_s = (\sin\phi_s\sin\beta_s, \sin\phi_s\cos\beta_s, \cos\phi_s) \quad (31)$$

In addition, the unit vector at the endpoint \vec{t}_E is given by

$$\vec{t}_E = (\sin\phi_E \sin\beta_E, \sin\phi_E \cos\beta_E, \cos\phi_E) \quad (32)$$

The positions of the control points are defined by multiplying its unit vector \vec{t} by a predefined scalar parameter d as follows:

$$C_S = S + d_S \vec{t}_S \quad (33)$$

$$C_E = E - d_E \vec{t}_E \quad (34)$$

After all the points on the curve are generated, the length of the curve can be approximated by aggregating the arc length between each two consecutive points on the curve as follows:

$$L(u_i) = \sum_i^1 \frac{\alpha_i}{\sin\alpha_i} DL_i \quad (35)$$

Where:

$$\alpha_i = \cos^{-1}(\vec{t}(u_i), \vec{t}(u_{i-1})) \quad (36)$$

and

$$DL_i = \sum_{i=0}^1 (Dx_i)^2 + (Dy_i)^2 + (Dz_i)^2 \quad (37)$$

The DLS constraint is magnitude of vector k expressed in degrees/30.48 m, which depicts the magnitude of curvature:

$$DLS = \|\vec{k}\| * 30.48 \quad (38)$$

Vector k can be obtained by a mathematical formulation combining the first derivative (\dot{B}) and second derivative (\ddot{B}) of the $B(u)$ function, as follows:

$$\vec{k} = \frac{\ddot{B}}{\dot{B} * \dot{B}} - \frac{\dot{B} * \ddot{B}}{(\dot{B} * \dot{B})^2} * \dot{B} \quad (39)$$

For more details on the mathematical formulation of the Bézier curves, refer to Sampaio [55].

5.1.2 Well Trajectory Optimization Using Non-Gradient Based Algorithms

All the employed optimizers follow the optimization workflow presented in **Figure 40**. However, each optimizer has a specific convergence criterion and an exclusive mathematical formulation for the update of decision variables. In the case of multiple well trajectories to be optimized, the presented workflow is applied sequentially; that is, each well is optimized on its own, independently of other wells. In the following, we describe the different steps of the optimization workflow.

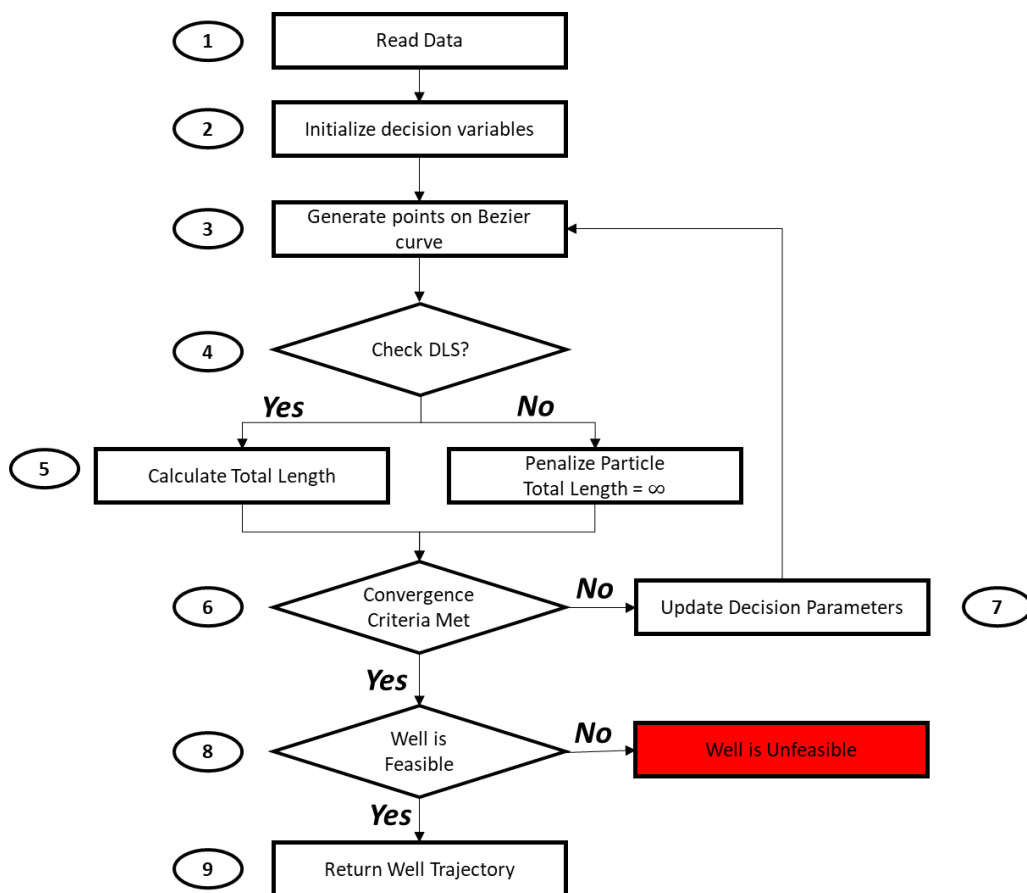


Figure 40 Well trajectory optimization using nongradient – based algorithms.

Step 1 – Input data

The algorithm reads the coordinates of the drilling center along with the reservoir well section provided as a set of connected segments, which is the drilling target specified by

reservoir engineers in a typical field development planning process. The algorithm automatically identifies the drilling center (starting point) then calculates the Euclidean distance between the starting point and the two endpoints of the drilling targets. Accordingly, it identifies the well heel based on the lower distance between the two calculations. The starting point and the well heel are stored to be used for the Bézier curve generation.

Step 2 – Initialize decision parameters

The decision parameters of the algorithm are the vertical distance between the drilling center point and the kickoff point S ; scalar parameter d_S , which controls the position of C_S ; and scalar parameter d_E , which controls the position of C_E . For the initial guess, these parameters are randomly defined for each searching agent or particle. However, as the algorithm progresses, the parameters change based on the optimizer's mathematical formulation.

Step 3 – Generate point on Bézier curve

After the decision variables are defined, the coordinates of the Bézier curve points are generated using Equation 29.

Step 4 – Check DLS

Using the points' coordinate as an input, the algorithm calculates DLS using Equation 38. If the DLS is greater than a predefined constraint, the solution is penalized by giving it a high total length. If not, the algorithm progresses to step 5.

Step 5 – Calculate total length

The algorithm calculates the total length of the curve using Equation 37.

Step 6 – Convergence

If the optimizer reached the convergence criteria, the algorithm stops and provides the optimal well trajectory solution. If not, the workflow reiterates until convergence is met. Each optimizer is associated with different convergence criteria, which will be discussed later in the result section.

Step 7 – Update decision parameters

At each iteration, the decision parameters are updated based on specific mathematical formulation relating the results of the previous iterations. the objective function used in our study is the length of the well trajectory obtained in step 4. note that each optimizer has its own mechanism in updating the decision parameters.

Steps 8 and 9 – Well feasibility

If the convergence criteria are attained and the optimizer reaches a feasible solution, the algorithm outputs the coordinates of the optimal well trajectory. otherwise, it flags the result as an unfeasible solution.

5.1.2.1 Tested Optimizers

Since the problem in hand is a non-smooth optimization problem where the gradient computation is difficult, only non-gradient-based optimization algorithms were considered. Such algorithms are computational intelligent models tailored for solving complex optimization problems. Four popular non-gradient-based algorithms were applied: particle swarm optimization (PSO), genetic algorithm (GA), differential evolution (DE), and brute force (BF). All these algorithms are available as built-in modules in Python. In the section below, we will briefly discuss the formulation of all these optimizers.

Genetic algorithm (GA)

Genetic algorithms (GA) are evolutionary, non-gradient-based algorithms that were inspired by the process of natural evolution of organisms. The concept was introduced by John Holland in the mid-1960s. A GA's functionality is based on three important components:

- Survival of the fittest (selection)
- Reproduction processes where genetic traits are propagated (crossover)
- Variation (mutation)

In GAs, an entire pool of solutions is iterated towards the optimal objective at the same time. Using a pool of solutions will increase the probability of the algorithm to moving out of local optima to target a global optimum, which is generally the ultimate objective of any optimization algorithm. For more details on the mathematical formulation of GA, please refer to Chambers [94].

Differential evolution (DE)

Similar to GA, differential evolution is an evolutionary, non-gradient-based algorithm that was introduced by Storn in 1997. It translates solutions as vectors and uses operations such as vector addition, scalar multiplication, and exchange of components (crossover) to construct new solutions from the existing ones. When a new solution, is created, it is compared to its parent. If the candidate is better than its parent, it replaces the parent in the population. Otherwise, the candidate is discarded. DE indirectly incorporates an elitism scheme in which no solution can be deleted from the population unless a better one is found. For more details on the mathematical formulation of DE, refer to Storn and Price [95].

Particle swarm optimization (PSO)

Please refer to section 3.1.1

Brute force (BF)

Brute-force search is a dummy algorithm that iterates over all possible candidates for the solution and checks whether each candidate satisfies the objective function and its constraints.

5.1.3 Use Of Trained Machine Learning Models for Well Trajectory Design

The ultimate objective of this work is to speed up the generation and optimization of well trajectories through ML models that are drastically fast at inference time. Optimizing well trajectories through traditional optimizers is relatively fast when considered on its own. However, when traditional optimizers are deployed within an integrated optimization workflow that includes facility, pipeline, and reservoir well section placement, they will be computationally exhaustive. To overcome this drawback, we propose an ML approach to predict the optimized location of the control points (d_S and d_E) that minimize the length of the curve while honouring the DLS. The dataset used for training the ML models was generated using one of the optimizers employed, namely the DE.

5.1.3.1 Machine learning algorithms

We apply three widely used ML regression methods, namely, artificial neural networks (ANNs), support vector regression (SVR), and random forests. These methods are trained on the dataset generated and numerically compared to identify which method provides the model that best approximates the values of the target output.

Artificial neural networks (ANNs)

Inspired by how the human brain processes information, artificial neural networks (ANNs) are composed of many connected computational units referred to as

neurons. The first layer of an ANN consists of several input neurons that is equal to the number of input features. This layer is followed by several hidden layers, and a final output layer with several neurons equal to the number of output features. The regular operation of a feed-forward connection in a neural network is given by:

$$z_i^{l+1} = w_i^{l+1} a^l + b_i^{l+1} \quad (40)$$

$$a_i^{l+1} = g(z_i^{l+1}) \quad (41)$$

where i is the neuron index, l is the layer index, z is an input vector to a layer, a is the output vector from a layer, w and b are the weight and bias parameters to be learned during training of the network, and $g(\cdot)$ is a fixed nonlinear activation function used to compute the output of a neuron that will be subsequently fed as an input to the neuron at the next layer. The network is trained using the backpropagation algorithm with the mean squared error (MSE) specified as the loss function, given by:

$$\mathcal{L}_{\text{ANN}}(X, Y) = \frac{1}{m} \sum_{i=0}^m (\hat{Y}_i - Y_i)^2 \quad (42)$$

where \hat{Y} is the predicted output value by the ANN model.

Support vector regression (SVR)

Inspired by its widely known classification counterpart, known as support vector machines, SVR aims at fitting a symmetrical tube of width $\varepsilon > 0$ around the estimated function so that absolute errors that fall below ε are ignored both above and below the estimate. To deal with nonlinear functions to be estimated, SVR is characterized by the use of nonlinear kernel functions that map the input into a higher dimensional space, referred to as the kernel space, where a linear relationship between the input and output features can be found. The SVR model is trained using the following loss function:

$$\mathcal{L}_{\text{SVR}} = \frac{1}{2} |w|^2 + C \sum_{i=1}^m |w \cdot \phi(X_i) + b - Y_i|_{\varepsilon} \quad (43)$$

where C is a regularization parameter, and $\phi(\cdot)$ is the nonlinear kernel function used, which we select to be the radial basis function.

Random forest

Random forest is an ensemble technique that averages the result of a collection of de-correlation decision trees. As trees are notoriously known to be noisy, averaging their results is beneficial in reducing variance. In random forest, multiple decision trees are fitted on randomly drawn bootstrap samples from the training dataset, where a subset of the initial input features are selected. The random forest model then averages the results of each decision tree to make a prediction. Given an input sample x , the prediction \hat{y} by the random forest model is then computed through:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^N T_i(x) \quad (44)$$

where $T(\cdot)$ denotes a decision tree and N is the total number of decision trees.

5.1.3.2 Problem formulation

Let X denote our input matrix of size $m \times n_x$ where m is the number of data samples and n_x is the number of input features. We use $n_x = 3$ input features, which are the angle at drilling center (Ψ), the Euclidean distance (ED) between the drilling center and the heel, and the angle between drilling direction and well direction (Ω). A diagram of the input and output features of our ML model is shown in **Figure 41**. 3D, side, and top views of the well trajectory are illustrated in **Figure 42** to better visualize the input features.

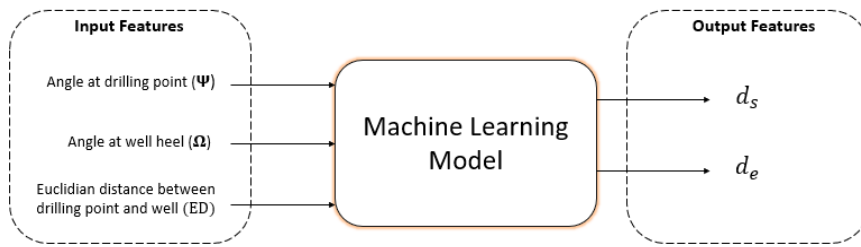


Figure 41 Input and output features of the ML models used for the characterization of well trajectory.

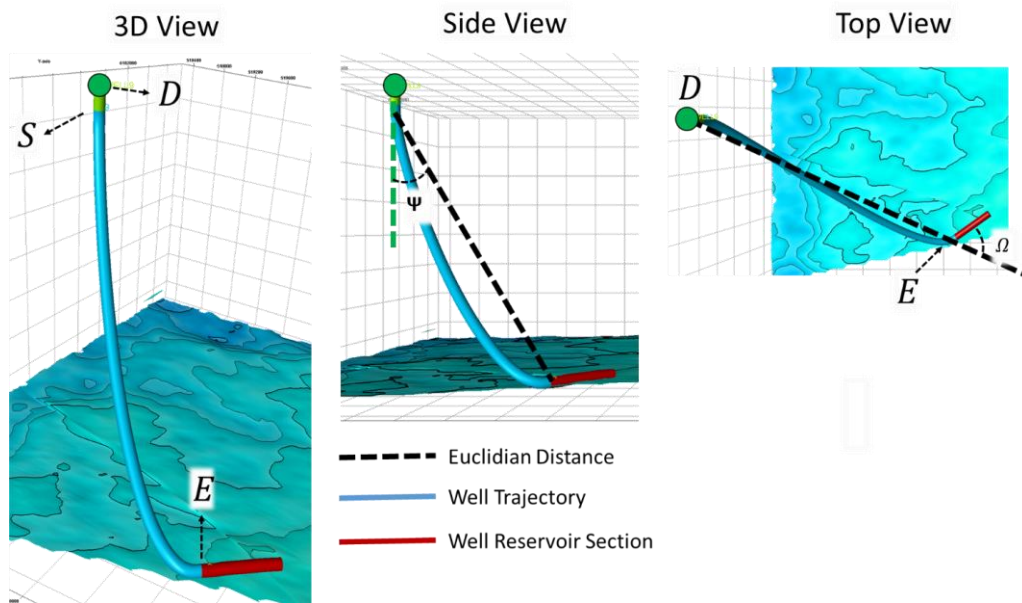


Figure 42 3D, side, and top view of a typical 3D well trajectory associated with the machine learning input parameters.

The rationale behind using the angle Ω is to reduce the number of samples used for training the ML model. This can be demonstrated in **Figure 43**. Assuming that all four wells lie on the same horizontal plane with equal Euclidean distance between the drilling center and the heel of each well, then the optimized lengths of all trajectories are equal since they all run through the same path. If we considered the drilling azimuth and the well azimuth as input features, it would produce the same output result with four different inputs values. This is avoided by using the angle Ω . The value of this

angle in all wells demonstrated in **Figure 43** will be the same, making them a single sample. This feature allows the dataset to be generated on a single drilling azimuth, which could be projected to any drilling azimuth in a 3D space. Thus, using the angle Ω as an input feature to the ML model allows it to produce predictions regardless of the drilling azimuth, and also reduces the need to generate data samples at various drilling azimuths.

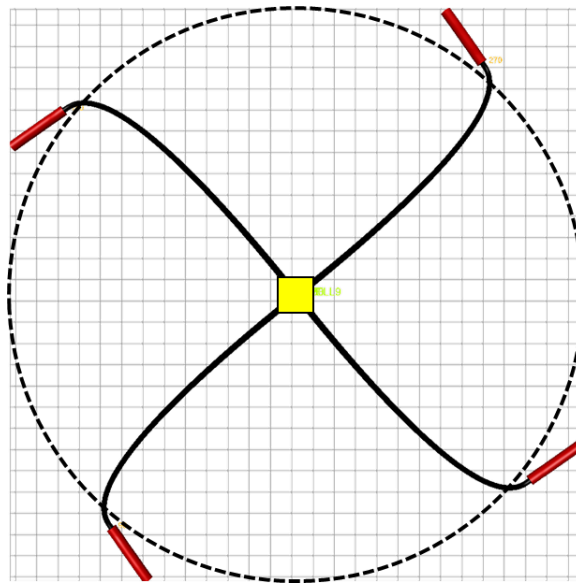


Figure 43 Four horizontal wells generated at an equal Euclidean distance from center of the circle (drilling center). Illustration of the efficiency of using the combined angle as an input feature to the ML models.

Let Y denote our output matrix of size $m \times n_y$ where n_y is the number of output features. The target output consists of $n_x = 2$ features: the scalar parameter at the start point d_s and the scalar parameter at the endpoint d_E . These output features can be computed using the differential evolution optimizer since it reaches a competitive result with lowest computational time. However, although optimization approaches result in accurate values of the desired output features to construct the Bézier curve, they suffer

from a large computational overhead, making them disadvantageous for adoption in parallelized approaches for integrated field development. To avoid the computational challenges associated with optimization strategies, we develop a regressive ML model that can learn a mapping from the input X to the target features vector Y . The prediction of d_S and d_E is treated as a regression problem since they take on continuous values. In this respect, after training a certain ML algorithm on data samples obtained through optimization approaches, the learned ML model can be used at inference mode to approximate the values of d_S and d_E in an accelerated manner and with low error.

5.2 Results And Discussion

In this section, we first describe the test case that was used in the optimization process, and then we discuss the characteristics and the results of the employed non-gradient optimizers. Furthermore, we elaborate on the training and the preparation of the ML models, the results achieved through its application on the test case, and finally we compare the results achieved by the optimizer and those achieved by the ML models.

5.2.1 Test case

For testing the performance of the optimizers along with the trained ML models, we defined a drilling center at 0 m depth and generated around it 500 horizontal reservoir well sections randomly in 3D cylindrical space with a radius of 5000 m, as shown in **Figure 44**. Depth of these wells ranges from 2000 m to 3000 m, a range that was also used for the ML training dataset. Furthermore, the horizontal length was fixed at 500 m since it has no impact on the optimization process of the Bézier curve.

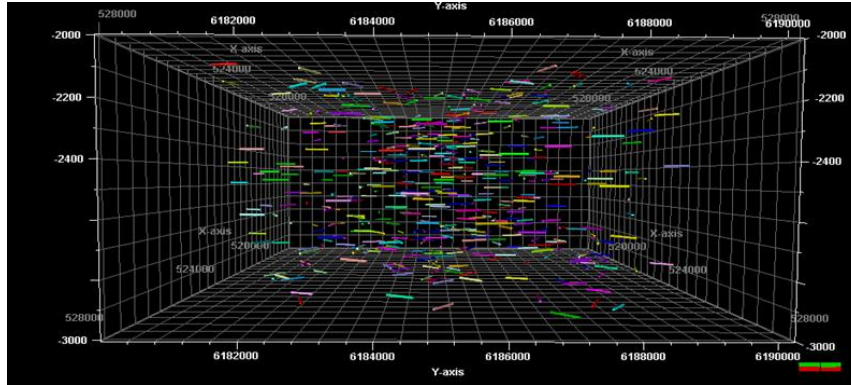


Figure 44 A side view of 500 horizontal wells generated randomly in 3D space.

5.2.2 Nongradient-Based Optimization Algorithms

5.2.2.1 Optimizer's configuration and characteristics

The optimization workflow was fully generated using Python. The source codes of GA and DE were taken from Solgi [96] and Nelson [97], respectively, and BF and PSO source codes were developed in house. The DLS constraint was fixed at 7° per 30.48 m throughout all optimization runs. Furthermore, based on several optimization trials, we noticed that the kick-off point was always at the minimum boundary, so it was excluded from the optimization process. All optimizers had the same boundary conditions for d_S and d_E . The value of d_S varies between 1 m and 1200 m while the value of d_E varies between 1 m and 100 m. Furthermore, the convergence criteria for PSO and DE were defined based on a recent publication by Dbouk et al. [98]; these criteria are the difference of the minimum particles' cost values and the average particles' cost value divided by the average particles' cost value. As for GA, it was left as specified by its author as the number of iterations with no change in cost value. Moreover, BF was run on all possible decision parameters combinations with a step size of 12 and 2 for d_S and d_E , respectively.

A sensitivity analysis was conducted to set a value for the convergence criteria of each optimizer. The maximum number of iterations and number of search agents were specified to be 200 and 15, respectively, for all optimizers. The values specified in Table 7 were found to be the most efficient in terms of time and cost value combined.

Table 7 Convergence criteria of the employed optimization algorithms

Algorithm	Convergence Criteria	Value
PSO	<u>average – minimum</u> average	0.05
GA	Number of iterations with no change of value	50
DE	<u>average – minimum</u> Average	0.001

5.2.2.2 Algorithms application on test case

We applied all the considered optimizers on the 500-well test case, and the results are presented in Table 2. All optimizers have concluded with 100% feasible wells honouring the defined DLS constraint. The BF algorithm achieved a total length of 2,186,879 m for all 500 wells in 5,025 seconds. As for the PSO algorithm, it achieved a total length of 2,184,251 m in 2,075 seconds. Furthermore, the GA algorithm recorded a total length of 2,186,000 m in 1,463 seconds and the DE algorithm recorded a total length of 2,184,248 m in 853 seconds. PSO and DE obtained a comparable total length and were the shortest length among all algorithms. They both achieved approximately 1,750 m and 2,660 m shorter total length than GA and BF, respectively. In terms of time, the DE algorithm was the fastest, recording 1.7 second per optimized well, which makes it the most efficient algorithm among all the algorithms. Accordingly, it was employed to optimize the ML dataset in the next section.

5.2.3 Machine Learning (ML) Models

5.2.3.1 Dataset generation

Introducing the combined angle allowed generating the data samples on a single drilling azimuth, as shown in **Figure 45**. A number of reservoir's well sections were generated from a number of center points that are 50m in distance along a 5000m line. These sections were produced from the center points based on an incremental angle α starting with 0° and ending at 180° . For clarity, α is illustrated at 45° in **Figure 45**. However, when the dataset was generated, α was incremented by 5° , which is small enough to generate samples that the ML algorithms can learn a good hypodissertation function from. The rationale behind having the boundaries between 0 and 180° is that if the wells are flipped relative to the drilling line ($\alpha > 180^\circ$) then we will have redundant data samples because the optimized track of these wells will have similar optimized values of d_S and d_E to wells with $\alpha < 180^\circ$. This configuration was repeated on 14 different depth horizons (75-m depth increment) from 2000 m to 3000 m. Each horizon included 3,564 data samples; hence, a total of 49,896 data samples was generated and optimized using the DE algorithms.

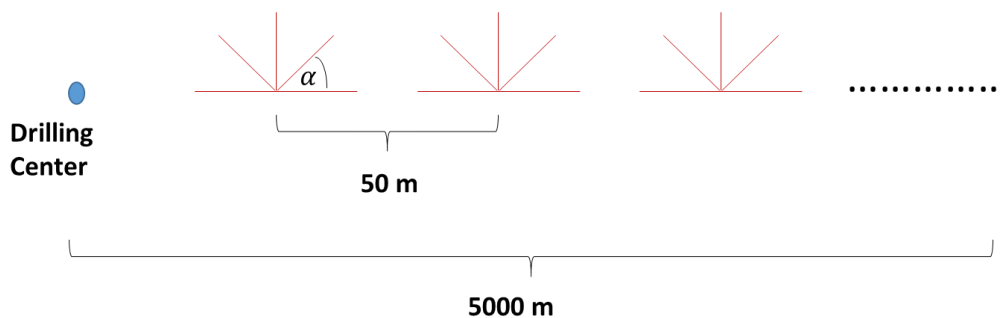


Figure 45 A schematic of the synthetic wells that were generated to train the ML models.

5.2.3.2 Model hyperparameters

The ANN model was developed using the Keras deep learning library and was trained using the adaptive moment estimation (ADAM) algorithm. The architecture of our ANN model developed for predicting d_s and d_e is illustrated in **Figure 46**. It consists of an input layer with 3 neurons, followed by four hidden layers with 50, 40, 30, and 20 neurons, respectively, and an output layer with 2 neurons. All neurons of the hidden layers use the rectified linear unit (ReLU) as an activation function, whereas the neurons of the output layer use a linear activation function. The choice for the number of layers, neurons, and activation function were obtained via a process of hyperparameter tuning for optimal training, validation, and test set performance.

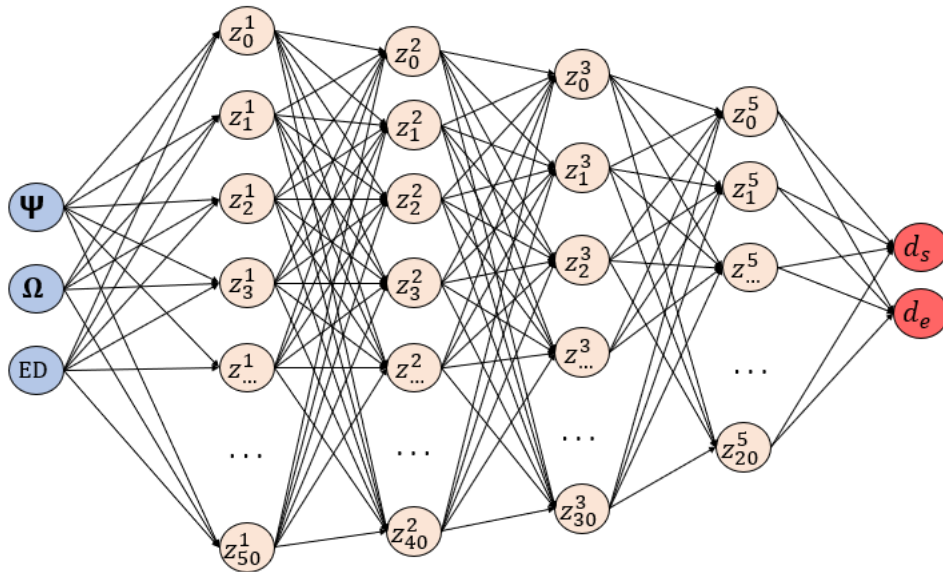


Figure 46 Architecture of the developed ANN model.

The SVR and random forest models were developed using the scikit-learn library. The hyperparameters of the SVR and random forest models were tuned using the grid search approach, which passes through all possible combinations of hyperparameters and selects the combination that yields the optimal test set performance. For the SVR model, a value of 0.1 for the tube width (ϵ) and a value of 30

for the regularization term (C) were found to give the best results. For the random forest model, 100 trees were used with no maximum depth specified.

5.2.3.3 Dataset splitting and pre-processing:

The dataset was randomly split into 90% for training and 10% for testing. This resulted in 44,895 samples for training, and 4,989 samples for testing. As a pre-processing step, the dataset was scaled using standard scaling as follows:

$$x' = \frac{x - \mu}{\sigma} \quad (45)$$

where x is a data sample, x' is the scaled version of the sample, and μ and σ are the mean and variance of the samples in the dataset, respectively. Scaling the features was an especially important pre-processing step to avoid problems such as exploding gradients.

5.2.3.4 Evaluation metrics

The ML models developed were evaluated using the mean squared error (MSE) and coefficient of determination (R^2 score) metrics that are commonly used in the evaluation of regression models. These metrics are computed as follows:

$$SE = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 \quad (46)$$

$$R^2 = 1 - \frac{\sum_{i=1}^m (\hat{y}_i - y_i)^2}{\sum_{i=1}^m (y_i - y_{\text{mean}})^2} \quad (47)$$

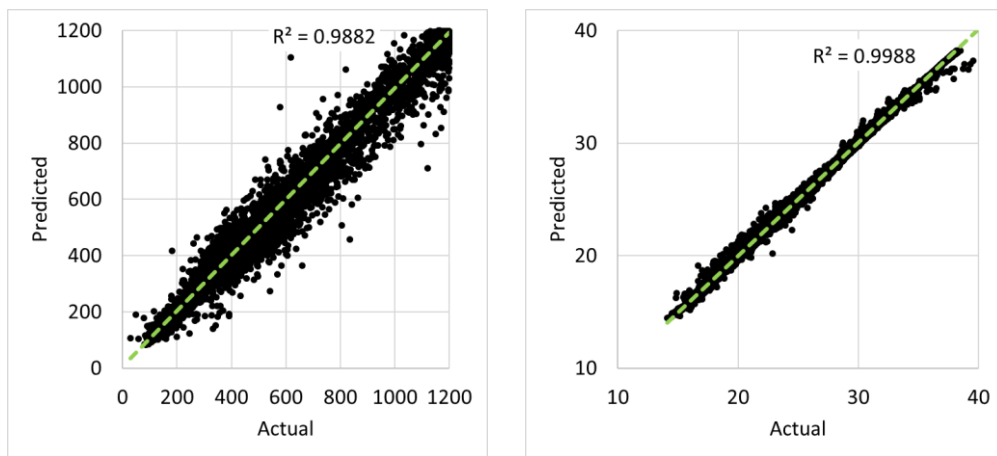
where \hat{y} is the predicted output value, y is the actual output value, and y_{mean} is the mean of all the actual output values.

5.2.3.5 Numerical evaluation

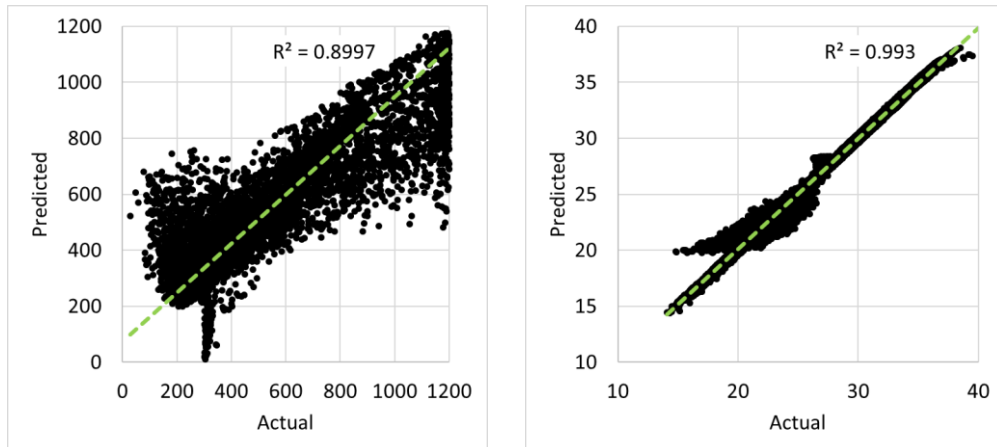
Cross-plots for each ML model on the training set for both d_S and d_E are provided in **Figure 47**. These plots show the predicted value by the model on the y-axis, and the actual value on the x-axis. A model with ideal performance would have all points situated on or very close to the identity line. We notice that the ANN and random forest models achieved remarkably high R^2 scores on both d_S and d_E whereas the SVR model delivered poorer performance, highlighted by a lower R^2 score and indicating that the model was not able to capture all the inherent nonlinearities between the input and output features in the kernel space, especially on the d_S feature where a significant drop in the R^2 is noticed. Although the ANN and random forest models achieved similar R^2 scores, we notice that the ANN model has many more outliers with relatively higher errors compared with the random forest model, where the points are situated closer to the identity line, hence providing better predictions. Despite the presence of multiple outliers in the case of the ANN model, their number is very small compared with the total number of samples in the training set. Therefore, these samples do not affect the overall performance of the ANN model, which still provides very accurate predictions.

Similar cross-plots are shown in **Figure 48** for the ML models on the test set. We notice a similar trend in the results observed on the training samples in **Figure 47**. This indicates that the models do not overfit on the training data and are able to generalize on unforeseen examples. Specifically, the ANN and random forest models provided very accurate test set predictions and achieved very high R^2 scores, and the SVR model still delivered inaccurate predictions. **Table 8** summarizes the results achieved by the different models in terms of MSE and R^2 scores. The ANN and random forest models achieved very low MSE scores and high R^2 scores, and the SVR

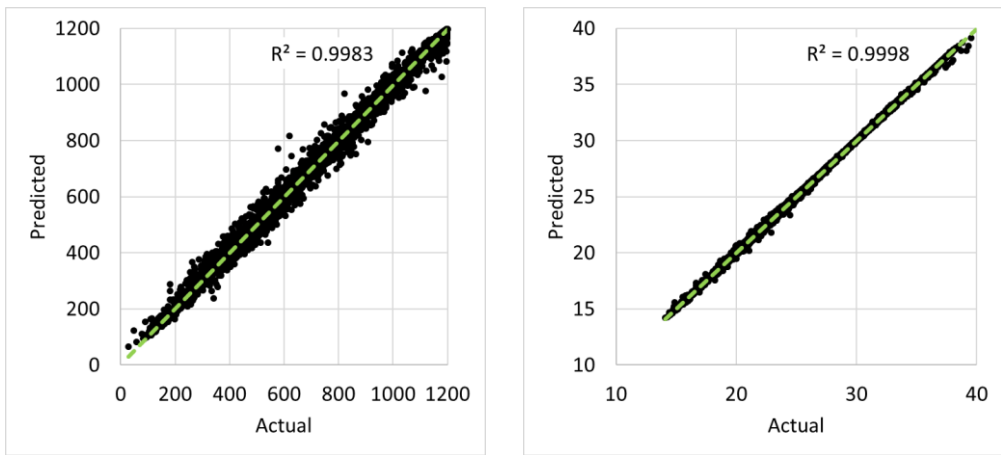
underperformed and showed a high MSE value on the d_S output feature. These results confirm the observations in the cross-plots of **Figure 47** and **Figure 48** that highlighted the strong performance attained by the ANN and random forest models, which have learned to accurately approximate the optimal values for d_S and d_E that would result by using conventional optimization techniques. These models can now be used in inference mode to replace an optimizer in predicting d_S and d_E , providing a boost in terms of computational time and cost.



A – Artificial Neural Networks

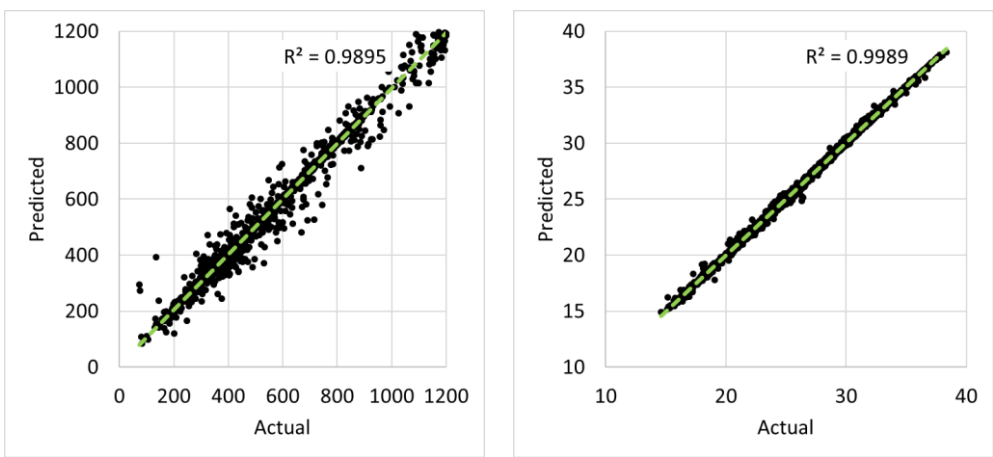


B – Support Vector Regression

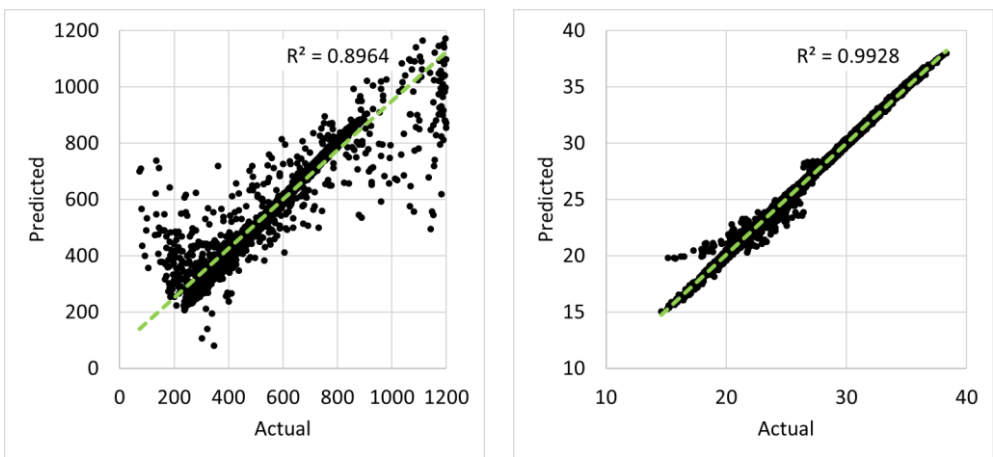


C – Random Forest.

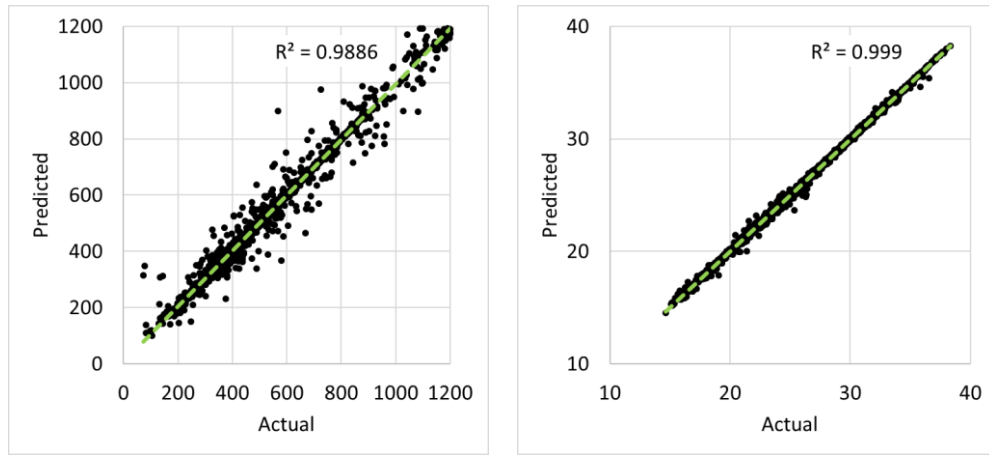
Figure 47 d_S (left) and d_E (right) cross-plots for the training dataset.



A – Artificial Neural Networks



B – Support Vector Regression



C – Random Forest

Figure 48 d_S (left) and d_E (right) cross-plots for the testing dataset.

Table 8 Numerical results achieved by the ML models for each output feature on the test set

ML Model	ANN		SVR		Random Forest	
Output Feature	d_S	d_E	d_S	d_E	d_S	d_E
MSE	0.01	0.001	0.102	0.007	0.01	0.001
R^2	0.989	0.999	0.891	0.993	0.989	0.999

5.2.3.6 ML models results on test case

The trained ML models were applied to the 500-wells test case, and the results are presented in Table 9. Although relatively high R^2 and low MSE values were achieved by model on the testing and training datasets, there is still some error in predicting d_S and d_E . This error triggered unfeasible well trajectory solutions. The random forest model was the best performer as it resulted in 97.6% feasibility, with only 12 unfeasible wells and with a relative low computation time of 0.015 seconds to generate a single trajectory. As for the ANN model, it had a lower performance, with 90% feasibility (50 wells were deemed unfeasible) and higher computation time of 0.0433 seconds for a well. Although the SVR model showed high speed, this model

concluded with 97 wells unfeasible. Combining both speed and feasibility, the random forest model was deemed the most efficient in terms of time and the number of feasible wells obtained.

Table 9 Computation time and percentage of feasibility wells achieved by the ML models. Application on a test dataset of 500 wells.

ML model	Percentage of feasible wells (%)	Computation time per well (sec)
Random Forest	97.6	0.015
ANN	90	0.043
SVR	80.4	0.002

5.2.4 Comparison Of the ML Approach and The Optimization Approach

To solve the issue of unfeasible wells, we employ the optimization algorithm when the solution fails with the ML models. In that sense, the trained ML model will work back-to-back with the optimizer to reach a competitive solution with lowest computation time possible. Since DE and random forest outperformed other competitors in terms of time and the length of the curve, we present them as a combined optimizer for well trajectory optimization. This combination was deemed 29.4 times faster than DE, which was the fastest optimizer among all. Note that the random forest model cost only 0.015 seconds per well, which was increased by 84% when the DE was employed for the 12 unfeasible wells. Furthermore, this combination obtained a comparable total length of the curve, with only 25-m difference from the DE algorithm among all 500 wells combined (Table 10).

Table 10 Cumulative/total well length and computation time for optimizers and the combined ML and optimizer approach. Application on a test dataset of 500 wells.

ML model	Total length (m)	Total time (seconds)	Total time per well (seconds)
Random forest - DE	2184263	28	0.056
BF	2186879	3000	6
GA	2186001	1463	2.9
PSO	2184251	2075	4.2
DE	2184248	853	1.7

5.3 Summary

Well trajectory design is a key element in field development planning and aims at determining the well path that minimizes the cost while honoring a set of surface, subsurface, and drilling constraints. Typically, optimizers are used to plan the well trajectory by calculating the values of a predefined set of parameters constituting the well's optimal trajectory. The optimization process is inherently fast due to the relatively simple problem to solve. However, it is not fast enough when the well trajectory design is part of a bigger planning tool where many wells are being planned in an iterative process that requires tens of thousands of well trajectory design operations.

A novel machine learning based optimization algorithm for well trajectory design achieves significant improvements in computational time compared to traditional optimization approaches. We used the Bézier curve to model the well trajectory and employed an optimization workflow to minimize the total well measured depth while honoring a dogleg severity constraint. We first compared several non-gradient optimizers and selected the most optimal among them which was shown to be the

differential evolution algorithm. We used the differential evolution optimizer to generate a large synthetic dataset that systematically, efficiently, and comprehensively cover the well trajectories. Three machine learning algorithms were used to train a model that predicts the well trajectory: artificial neural networks, support vector regression, and random forests. The random forests model produced the lowest achieved error and percentage of resulting feasible wells on both the training and testing datasets. Using a machine learning model to design a well trajectory was three orders of magnitude faster than the differential evolution algorithm which, in turn, was the fastest among the different optimization algorithms that we have tested.

To the best of our knowledge, our proposed method is the first attempt in the literature at using machine learning for optimizing well trajectory. The main contribution of the proposed method is the drastic reduction of computational time, which allows the method to be deployed in an integrated field development optimization, which includes several optimization modules.

CHAPTER 6

SUMMARY, CONCLUSIONS AND FUTURE WORK

In this work, we introduced and applied advanced techniques relevant to well placement optimization under uncertainty. We first developed a static well placement algorithm – the so called the black hole operator – that “optimizes” well placement guided by the $NHCT_{map}$ or any other potential reservoir quality map. Next, we merged the BHO with the traditional particle swarm optimization to form the BHPSO algorithm that automatically and efficiently optimizes well placement, accounting for the dynamic performance of the reservoir. Then, we developed an efficient uncertainty management workflow that enables BHPSO to be applied on a reduced number of geological realizations. Finally, we presented an efficient ML based well trajectory optimizer that drastically reduced the computational time needed to reach a feasible solution.

6.1 Conclusions

The key contributions of our work on well placement optimization are as follows:

- We introduced a new algorithm that combines an efficient static, map-based optimizer, the black hole operator, with the standard version of the popular particle swarm optimization. The proposed BHPSO algorithm is based on a novel idea that incorporates the $NHCT_{map}$ in the optimization process and ensures a guided (rather than random) well placement at each optimization iteration

- The combination of PSO with the black hole operator depicted efficiency, robustness and practicability, and was successful in jointly optimizing well count, location, and type.
- BHPSO has a key feature: the number of optimization parameters is independent of the well count as only the location of the first well is identified using PSO. This leads to an extensive reduction in computational time, avoids over-parameterization problems and, most importantly, allows the method to be employed in practical field development planning projects. Over-parameterization, especially, poses a real challenge in adopting optimization methods in real field development projects, where project time and CPU limitations practically eliminate most of the available optimization methods.
- Strengths of the proposed method were illustrated on the Olympus optimization challenge where the new method systematically outperformed PSO in terms of both CPU and optimal solution reached. BHPSO leads to a 40 to 50% NPV increase compared to the initial guess, with a relatively low number of iterations (less than 30 iterations) and particles (10 particles).

Our main findings regarding uncertainty management are as follows:

- We introduced the Statistical $NHCT_{map}$ ($SNHCT_{map}$) to BHPSO method for well placement optimization under uncertainty. $SNHCT_{map}$ guides the BHPSO algorithm in placing wells in statistically highly performing locations at each optimization iteration. This approach has also made it possible to run BHPSO using a “single” representative geological realization while obtaining a statistically competitive result that is robust to uncertainty.

- The performance of the proposed method is demonstrated by comparing it with two popular scenario reduction methods, through a scenario-based optimization workflow applied to the upper geological zone of the synthetic Olympus field. The optimal scenarios obtained using the proposed method were statistically dominant over the optimal scenarios obtained from the other two scenario reduction-based methods.

Our main findings regarding well trajectory modelling and optimization are as follows:

- We introduced a novel ML-based optimization algorithm for well trajectory design. We used the Bézier curve to model the well trajectory and employed an optimization workflow to minimize the length of the curve while honoring a DLS constraint.
- The differential evolution (DE) algorithm was deemed the most efficient among other non-gradient-based algorithms in terms of computation time and the cost value, which is the total measured depth (TMD). Hence, it was employed for generating that dataset used for training the ML model.
- Three ML models were tested and applied to a representative sample of wells that were generated randomly in the 3D space. Among all tested ML models, the random forest model was deemed the most efficient, achieving 97.6% feasibility. The random forest model was combined with the DE algorithm and applied to the test case. This combination drastically reduced the computation time while achieving comparable results with the DE, the most efficient optimizer among other non-gradient-based algorithms.

6.2 Future Work

Many challenges could be addressed in areas relevant to our work. Our suggestions for future research are as follows:

- The black hole operator is a portable operator that can be connected to any non-gradient-based algorithm. It will be useful to test the performance of connecting other efficient algorithms, i.e., differential evolution, genetic algorithms, etc., to the black hole operator.
- Black holing is carried out using a circular shape on the $NHCT_{map}$. It would be worth testing different shapes in the black holing technique, i.e., rectangular, ellipsoidal etc., especially in peripheral injection where the injection area is relatively thin compared to the production area.
- BHPSO was run through an efficient uncertainty management workflow which drastically reduced the number of realizations involved. This novel approach performed well on the Olympus field, which consists of 50 realizations. However, the method still needs further validation by applying it to real field cases, which could help better define the b variable for the $SNHCT_{map}$.
- In our third contribution, well trajectory was modeled using the Bézier curve. Among other methods found in literature is the minimum curvature method which was developed by Shokir [52]. Although the Bézier curve is more flexible and simpler to optimize, one might argue the Shokir model could lead to a more accurate trajectory. Hence, it is worth assessing this model within our proposed optimization methodology and comparing it with the Bézier curve model. However, it should be noted that optimizing such a complex

model would require high computational time which might be infeasible for training the machine learning model.

- Another issue that could be investigated is incorporating the dogleg severity constraint as one additional input parameter for the machine learning model; this requires an increased number of testing samples but can potentially achieve better results.
- Lastly, additional constraints to the well trajectory model can be considered: i.e., strain energy, drag.

BIBLIOGRAPHY

- [1] Meddaugh WS. Reservoir modeling for mature fields - impact of workflow and up-scaling on fluid flow response. Soc Pet Eng 68th Eur Assoc Geosci Eng Conf Exhib Inc SPE Eur 2006, EAGE 2006 Oppor Matur Areas 2006;2:699–716. <https://doi.org/10.2118/99833-ms>.
- [2] Siavashi M, Blunt MJ, Raisee M, Pourafshary P. Three-dimensional streamline-based simulation of non-isothermal two-phase flow in heterogeneous porous media. *Comput Fluids* 2014;103:116–31. <https://doi.org/10.1016/j.compfluid.2014.07.014>.
- [3] Samier P, Quettier L, Thiele M. Applications of Streamline Simulations to Reservoir Studies 2007;2000. <https://doi.org/10.2523/66362-ms>.
- [4] Li H, Durlofsky LJ. Upscaling for compositional reservoir simulation. *SPE J* 2016;21:873–87. <https://doi.org/10.2118/173212-pa>.
- [5] Ma E, Al-Houti R, Dashti L, Ali F, Ryzhov S, Ibrahim M, et al. Upscaling a 900 million-cell static model to dynamic model of the world’s largest clastic oil field - Greater Burgan field, Kuwait. Soc Pet Eng - Kuwait Oil Gas Show Conf KOGS 2013 2013;141–51. <https://doi.org/10.2118/167280-ms>.
- [6] Rezapour A, Ortega A, Sahimi M. Upscaling of Geological Models of Oil Reservoirs with Unstructured Grids Using Lifting-Based Graph Wavelet Transforms. *Transp Porous Media* 2019;127:661–84. <https://doi.org/10.1007/s11242-018-1219-7>.
- [7] Cheshire IM, Bowen G. Parallelization in reservoir simulation. *SPE Lat Am Caribb Pet Eng Conf Proc* 1992;1992-March:103–6. <https://doi.org/10.2523/23657-ms>.
- [8] Dogru AH, Sunaidi HA, Fung LS, Habiballah WA, Al-Zamel N, Li KG. A parallel reservoir simulator for large-scale reservoir simulation. *SPE Reserv Eval Eng* 2002;5:11–23.
- [9] Tanaka S, Wang Z, Dehghani K, He J, Velusamy B, Wen XH. Large scale field development optimization using high performance parallel simulation and cloud computing technology. *Proc - SPE Annu Tech Conf Exhib* 2018;2018-Sept. <https://doi.org/10.2118/191728-ms>.
- [10] Hassani H, Sarkheil H, Foroud T, Karimpooli S. A proxy modeling approach to optimization horizontal well placement. *45th US Rock Mech / Geomech Symp* 2011.
- [11] Mansouri V, Khosravianian R, Wood DA, Aadnoy BS. 3-D well path design using a multi objective genetic algorithm. *J Nat Gas Sci Eng* 2015;27:219–35. <https://doi.org/10.1016/j.jngse.2015.08.051>.
- [12] Meira LAA, Coelho GP, Santos AAS, Schiozer DJ. Selection of Representative Models for Decision Analysis Under Uncertainty. *Comput Geosci* 2016;88:67–82. <https://doi.org/10.1016/j.cageo.2015.11.012>.
- [13] Shirangi MG. Closed-loop field development with multipoint geostatistics and statistical performance assessment. *J Comput Phys* 2019;390:249–64. <https://doi.org/10.1016/j.jcp.2019.04.003>.
- [14] Shirangi MG, Durlofsky LJ. A general method to select representative models for decision making and optimization under uncertainty. *Comput Geosci* 2016;96:109–23. <https://doi.org/10.1016/j.cageo.2016.08.002>.

- [15] Onwunali JE, Durlofsky LJ. A new well-pattern-optimization procedure for large-scale field development. *SPE J* 2011;16:594–607. <https://doi.org/10.2118/124364-PA>.
- [16] Khademi G, Karimaghaee P. Hybrid FDG optimization method and kriging interpolator to optimize well locations. *J Pet Explor Prod Technol* 2016;6:191–200. <https://doi.org/10.1007/s13202-015-0175-9>.
- [17] Eberhart R, James Kennedy. A New Optimizer Using Particle Swarm Theory. *Int Symp Micro Mach Hum Sci* 1999:39–43.
- [18] Sen MK, Datta-Gupta A, Stoffa PL, Lake LW, Pope GA. Stochastic reservoir modeling using simulated annealing and genetic algorithms. *SPE Form Eval* 1995;10:49–55. <https://doi.org/10.2118/24754-pa>.
- [19] Goldberg, D. E., Holland JH. GUEST EDITORIAL Genetic Algorithms and Machine Learning. *Mach Learn* 1988:95–9.
- [20] Hazlett RD, Babu DK. Optimal well placement in heterogeneous reservoirs through semianalytic modeling. *SPE J* 2005;10:286–96. <https://doi.org/10.2118/84281-pa>.
- [21] Bangerth W, Klie H, Wheeler MF, Stoffa PL, Sen MK. On optimization algorithms for the reservoir oil well placement problem. *Comput Geosci* 2006;10:303–19. <https://doi.org/10.1007/s10596-006-9025-7>.
- [22] Li L, Jafarpour B, Mohammad-Khaninezhad MR. A simultaneous perturbation stochastic approximation algorithm for coupled well placement and control optimization under geologic uncertainty. *Comput Geosci* 2013;17:167–88.
- [23] Sarma P, Chen WH. Efficient well placement optimization with gradient-based algorithms and adjoint models. *Soc Pet Eng - Intell Energy Conf Exhib Intell Energy* 2008 2008;2:1097–114. <https://doi.org/10.2118/112257-ms>.
- [24] Harding TJ, Radcliffe NJ, King PR. Hydrocarbon Production Scheduling with Genetic Algorithms. *SPE J* 1998;3:99–107. <https://doi.org/10.2118/36379-PA>.
- [25] Anderson MG, Downing K, Jensen JP. Reservoir Production Optimization Using Genetic Algorithms and Artificial Neural Networks. *Nor Univ Sci Technol* 2009.
- [26] Ramezanzpour M, Siavashi M. Application of SiO₂ –water nanofluid to enhance oil recovery: A new hybrid optimization approach using pattern search and PSO algorithms. *J Therm Anal Calorim* 2019;135:565–80. <https://doi.org/10.1007/s10973-018-7156-4>.
- [27] Siavashi M, Yazdani M. A Comparative Study of Genetic and Particle Swarm Optimization Algorithms and Their Hybrid Method in Water Flooding Optimization. *J Sol Energy Eng Trans ASME* 2018;140:1–10. <https://doi.org/10.1115/1.4040059>.
- [28] Bittencourt AC, Horne RN. Reservoir development and design optimization. *Proc - SPE Annu Tech Conf Exhib* 1997;Sigma:545–58. <https://doi.org/10.2523/38895-ms>.
- [29] Güyagüler B, Ghorayeb K. Integrated optimization of field development, planning, and operation. *Proc. - SPE Annu. Tech. Conf. Exhib.*, vol. 4, 2006. <https://doi.org/10.2118/102557-ms>.
- [30] Yeten B, Durlofsky LJ, Aziz K. Optimization of nonconventional well type, location, and trajectory. *SPE J* 2003;8:200–10. <https://doi.org/10.2118/86880-PA>.
- [31] Emerick AA, Petrobras SA, Silva E, Messer B, Almeida LF, Szwarcman D, et al. Well placement optimization using a genetic algorithm with nonlinear constraints. *SPE Reserv Simul Symp Proc* 2009;1:98–117.

- <https://doi.org/10.2118/118808-ms>.
- [32] Onwunalu JE, Durlofsky LJ. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Comput Geosci* 2010;14:183–98. <https://doi.org/10.1007/s10596-009-9142-1>.
- [33] Nwankwor E, Nagar AK, Reid DC. Hybrid differential evolution and particle swarm optimization for optimal well placement. *Comput Geosci* 2013;17:249–68. <https://doi.org/10.1007/s10596-012-9328-9>.
- [34] Isebor OJ, Echeverri D, Durlofsky LJ. *Generalized Field-Development Optimization With Derivative-Free Procedures* 2014.
- [35] Ding S, Jiang H, Li J, Tang G. Optimization of well placement by combination of a modified particle swarm optimization algorithm and quality map method. *Comput Geosci* 2014;18:747–62.
- [36] Ding S, Jiang H, Li J, Liu G, Mi L. Optimization of Well Location, Type and Trajectory by a Modified Particle Swarm Optimization Algorithm for the PUNQ-S3 Model. *J Ind Intell Inf* 2015;4:27–33. <https://doi.org/10.12720/jiii.4.1.27-33>.
- [37] Hamida Z, Azizi F, Saad G. An efficient geometry-based optimization approach for well placement in oil fields. *J Pet Sci Eng* 2017;149:383–92. <https://doi.org/10.1016/j.petrol.2016.10.055>.
- [38] Alrashdi Z, Sayyafzadeh M. (M+ Λ) Evolution strategy algorithm in well placement, trajectory, control and joint optimisation. *J Pet Sci Eng* 2019;177:1042–58. <https://doi.org/10.1016/j.petrol.2019.02.047>.
- [39] Park H, Yang C, Al-aruri AD, Fjerstad PA. Improved decision making with new efficient work flows for well placement optimization 2017;152:81–90. <https://doi.org/10.1016/j.petrol.2017.02.011>.
- [40] da Cruz PS, Home RN, Deutsch C V. The quality map: A tool for reservoir uncertainty quantification and decision making. *SPE Reserv Eval Eng* 2004;7:6–14. <https://doi.org/10.2118/87642-PA>.
- [41] Molina A, Rincon A. Exploitation plan design based on opportunity index analysis in numerical simulation models. *SPE Lat Am Caribb Pet Eng Conf Proc* 2009;3:1295–302. <https://doi.org/10.2118/122915-ms>.
- [42] Abd Karim MG, Abd Raub MR. Optimizing development strategy and maximizing field economic recovery through simulation opportunity index. *Soc Pet Eng - SPE Reserv Characterisation Simul Conf Exhib 2011, RCSC 2011* 2011:554–9. <https://doi.org/10.2118/148103-ms>.
- [43] Abdy Y, Amari M, Sharifzadeh A, Al-Anzi E, Al-Ansari M. An integrated and effective method for well placement using a numerical simulation model: A North Kuwait field case study. *Soc Pet Eng - Kuwait Int Pet Conf Exhib 2012, KIPCE 2012 People Innov Technol to Unleash Challenging Hydrocarb Resour* 2012;2:960–77. <https://doi.org/10.2118/163378-ms>.
- [44] Al-Khazraji AK, Shuker MT. Development of heterogeneous immature brownfield with waterdrive using dynamic opportunity index: A case study from Iraqi oilfields. *Soc Pet Eng - SPE North Africa Tech Conf Exhib 2015, NATC 2015* 2015:112–24. <https://doi.org/10.2118/175708-ms>.
- [45] Chandra S. *Selected Topics on Improved Oil Recovery*. Springer Singapore; 2018. <https://doi.org/10.1007/978-981-10-8450-8>.
- [46] Short J. *Introduction to Directional and Horizontal Drilling*. 1993. <https://doi.org/10.1002/9781118998632.ch10>.
- [47] Hosseini S, Ghanbarzadeh A, Hasheni A. Optimization of Dogleg Severity in

- Directional Drilling Oil Wells Using Particle Swarm Algorithm 2014;48:139–51.
- [48] Inglis T. Petroleum Engineering and Refining. Using Eng Lit Second Ed 1987;2:469–95. <https://doi.org/10.1201/b11072-20>.
- [49] Devereux S. Practical Well Planning and Drilling. 1998.
- [50] Amorin R, Broni-Bediako E. Application of Minimum Curvature method to wellpath calculations. Res J Appl Sci Eng Technol 2010;2:679–86.
- [51] Sampaio Jr. JHB. Planning 3D Well Trajectories Using Cubic Functions. J Energy Resour Technol 2006;128:257–67. <https://doi.org/10.1115/1.2358140>.
- [52] Shokir EMEM, Emera MK, Eid SM, Wally AW. A new optimization model for 3D well design. Oil Gas Sci Technol 2004;59:255–66. <https://doi.org/10.2516/ogst:2004018>.
- [53] Scholes H. A Three-Dimensional Well Planning Method for HDR Geothermal Wells 1983. <https://doi.org/10.2118/12101-MS>.
- [54] Sampaio Jr. JHB. Planning 3D Well Trajectories Using Spline-in-Tension Functions. J Energy Resour Technol 2007;129:289–99. <https://doi.org/10.1115/1.2790980>.
- [55] Sampaio Jr. JHB. Designing Three-Dimensional Directional Well Trajectories Using B ezier Curves 2017;139:1–8. <https://doi.org/10.1115/1.4034810>.
- [56] Biswas K, Vasant PM, Vintaned JAG, Watada J. A Review of Metaheuristic Algorithms for Optimizing 3D Well-Path Designs. Arch Comput Methods Eng 2021;28:1775–93. <https://doi.org/10.1007/s11831-020-09441-1>.
- [57] Helmy MW, Khalaf F, Darwish TA. Well Design Using a Computer Model. SPE Drill Complet 1998;13:42–6. <https://doi.org/10.2118/37709-PA>.
- [58] Atashnezhad A, Wood DA, Fereidounpour A, Khosravianian R. Designing and optimizing deviated wellbore trajectories using novel particle swarm algorithms. J Nat Gas Sci Eng 2014;21:1184–204. <https://doi.org/10.1016/j.jngse.2014.05.029>.
- [59] Mansouri V, Khosravianian R, Wood DA, Aadnøy BS. Optimizing the separation factor along a directional well trajectory to minimize collision risk. J Pet Explor Prod Technol 2020;10:2113–25. <https://doi.org/10.1007/s13202-020-00876-7>.
- [60] Wood DA. Hybrid bat flight optimization algorithm applied to complex wellbore trajectories highlights the relative contributions of metaheuristic components. J Nat Gas Sci Eng 2016;32:211–21. <https://doi.org/10.1016/j.jngse.2016.04.024>.
- [61] Sha L, Pan Z. FSQGA based 3D complexity wellbore trajectory optimization. Oil Gas Sci Technol 2018;73. <https://doi.org/10.2516/ogst/2018008>.
- [62] Biswas K, Vasant PM, Gamez Vintaned JA, Watada J. Cellular automata-based multi-objective hybrid grey wolf optimization and particle swarm optimization algorithm for wellbore trajectory optimization. J Nat Gas Sci Eng 2021;85:103695. <https://doi.org/10.1016/j.jngse.2020.103695>.
- [63] Dada MA, Mellal M, Makhloufi A, Belhouchet H. A field development strategy for the joint optimization of flow allocations, well placements and well trajectories. Energy Explor Exploit 2021;39:502–27. <https://doi.org/10.1177/0144598720974425>.
- [64] Rahim S, Li Z, Trivedi J. Reservoir Geological Uncertainty Reduction: an Optimization-Based Method Using Multiple Static Measures. Math Geosci 2015;47:373–96. <https://doi.org/10.1007/s11004-014-9575-5>.
- [65] Ballin PR, Journal AG, Aziz K. Prediction Of Uncertainty In Reservoir Performance Forecast. J Can Pet Technol 1992;31. <https://doi.org/10.2118/92-04->

- 05.
- [66] McLennan J, Deutsch C V. Local Ranking of Geostatistical Realizations for Flow Simulation. *Cent Comput Geostatistics* 2005;2010.
 - [67] Deutsch C V. Fortran programs for calculating connectivity of three-dimensional numerical models and for ranking multiple realizations. *Comput Geosci* 1998;24:69–76. [https://doi.org/10.1016/S0098-3004\(97\)00085-X](https://doi.org/10.1016/S0098-3004(97)00085-X).
 - [68] Fenik DR, Nouri A, Deutsch C V. Criteria for ranking realizations in the investigation of SAGD reservoir performance. *Can Int Pet Conf 2009, CIPC 2009* 2009;1–10. <https://doi.org/10.2118/2009-191>.
 - [69] Li S, Deutsch C V, Si J. Ranking Geostatistical Reservoir Models with Modified Connected Hydrocarbon Volume. *Proc Geostats 2012* 2012;1–11.
 - [70] Steagall DE, Schiozer DJ. *Uncertainty Analysis in Reservoir Production Forecasts During Appraisal and Pilot Production Phases* 2007. <https://doi.org/10.2523/66399-ms>.
 - [71] Rodrigues HWL, Prata BA, Bonates TO. Integrated optimization model for location and sizing of offshore platforms and location of oil wells. *J Pet Sci Eng* 2016;145:734–41. <https://doi.org/10.1016/j.petrol.2016.07.002>.
 - [72] Rahim S, Li Z. Well placement optimization with geological uncertainty reduction. *IFAC-PapersOnLine* 2015;28:57–62. <https://doi.org/10.1016/j.ifacol.2015.08.157>.
 - [73] Scheidt C, Caers J. Representing spatial uncertainty using distances and kernels. *Math Geosci* 2009;41:397–419. <https://doi.org/10.1007/s11004-008-9186-0>.
 - [74] Singh AP, Maučec M, Carvajal GA, Mirzadeh S, Knabe SP, Al-Jasmi AK, et al. Uncertainty quantification of forecasted oil recovery using dynamic model ranking with application to a ME carbonate reservoir. *Soc Pet Eng - Int Pet Technol Conf 2014, IPTC 2014 Unlocking Energy Through Innov Technol Capab* 2014;3:2284–95. <https://doi.org/10.3997/2214-4609-pdb.395.iptc-17476-ms>.
 - [75] Wang H, Ciaurri DE, Durlofsky LJ, Cominelli A. Optimal well placement under uncertainty using a retrospective optimization framework. *SPE J* 2012;17:112–21. <https://doi.org/10.2118/141950-PA>.
 - [76] Shirangi MG, Mukerji T. Retrospective Optimization of Well Controls Under Uncertainty Using Kernel Clustering. *25th Annu SCRF Meet* 2012;1–36. <https://doi.org/10.13140/RG.2.1.3784.2003>.
 - [77] Liu Z, Forouzanfar F. Ensemble clustering for efficient robust optimization of naturally fractured reservoirs. *Comput Geosci* 2018;22:283–96. <https://doi.org/10.1007/s10596-017-9689-1>.
 - [78] Barros EGD, Maciel S, Moraes RJ, Fonseca RM. Automated clustering based scenario reduction to accelerate robust life-cycle optimization. *16th Eur Conf Math Oil Recover ECMOR 2018* 2018. <https://doi.org/10.3997/2214-4609.201802179>.
 - [79] Sarma P, Chen WH, Xie J. Selecting representative models from a large set of models. *Soc Pet Eng - SPE Reserv Simul Symp* 2013 2013;2:1313–25. <https://doi.org/10.2118/163671-ms>.
 - [80] Meira LA, Coelho GP, Silva CG, Schiozer DJ, Santos AS. RMFinder 2.0: An improved interactive multi-criteria scenario reduction methodology. *SPE Lat Am Caribb Pet Eng Conf Proc* 2017;0. <https://doi.org/10.2118/185502-ms>.
 - [81] Harb A, Kassem H, Ghorayeb K. Black hole particle swarm optimization for well

- placement optimization. *Comput Geosci* 2019. <https://doi.org/10.1007/s10596-019-09887-8>.
- [82] Ghorayeb K, Harb A, Kassem H. BLACK HOLE PARTICLE SWARM OPTIMIZATION FOR OPTIMAL WELL PLACEMENT IN FIELD DEVELOPMENT PLANNING AND METHODS OF USE. US 20200080406A1, 2020.
- [83] Harb A, Moallem A, Ghorayeb K. BHPSO combined with statistical net hydrocarbon thickness map for well placement optimization under uncertainty. *Comput Geosci* 2021. <https://doi.org/10.1007/s10596-021-10040-7>.
- [84] Schulze-Riegert R, Nwakile M, Skripkin S, Whymark M, Baffoe J, Geissenhoener D, et al. Olympus challenge—standardized workflow design for field development plan optimization under uncertainty. *Comput Geosci* 2020;24:2059–77. <https://doi.org/10.1007/s10596-019-09905-9>.
- [85] Fonseca RM, Rossa E Della, Emerick AA, Hanea RG, Jansen JD. Introduction to the special issue: Overview of OLYMPUS Optimization Benchmark Challenge. *Comput Geosci* 2020;24:1933–41. <https://doi.org/10.1007/s10596-020-10003-4>.
- [86] Okwu MO, Tartibu LK. Particle Swarm Optimisation. *Stud Comput Intell* 2021;927:5–13. https://doi.org/10.1007/978-3-030-61111-8_2.
- [87] Zhang W, Ma D, Wei JJ, Liang HF. A parameter selection strategy for particle swarm optimization based on particle positions. *Expert Syst Appl* 2014;41:3576–84. <https://doi.org/10.1016/j.eswa.2013.10.061>.
- [88] Jiang M, Luo YP, Yang SY. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Inf Process Lett* 2007;102:8–16.
- [89] Trelea IC. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inf Process Lett* 2003;85:317–25.
- [90] Chao O, Weixing L. Comparison between PSO and GA for parameters optimization of PID controller. 2006 IEEE Int Conf Mechatronics Autom ICMA 2006 2006;2006:2471–5. <https://doi.org/10.1109/ICMA.2006.257739>.
- [91] Sun L, Song X, Chen T. An improved convergence particle swarm optimization algorithm with random sampling of control parameters. *J Control Sci Eng* 2019;2019. <https://doi.org/10.1155/2019/7478498>.
- [92] Hegstad BK, Sætrum J. Using multiple realizations from an integrated uncertainty analysis to make more robust decisions in field development. *Soc Pet Eng - 30th Abu Dhabi Int Pet Exhib Conf ADIPEC 2014 Challenges Oppor Next 30 Years* 2014;3:1653–9. <https://doi.org/10.2118/171831-ms>.
- [93] Meira LAA, Coelho GP, da Silva CG, Abreu JLA, Santos AAS, Schiozer DJ. Improving representativeness in a scenario reduction process to aid decision making in petroleum fields. *J Pet Sci Eng* 2020;184:106398. <https://doi.org/10.1016/j.petrol.2019.106398>.
- [94] Chambers LD. *The Practical Handbook of GENETIC ALGORITHMS: Applications*. 2000.
- [95] Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Australas Plant Pathol* 2009;38:284–7. <https://doi.org/10.1071/AP09004>.
- [96] Solgi R. *Genetic Algorithms* 2020. <https://pypi.org/project/geneticalgorithm/>.
- [97] Nelson A. *Differential Evolution* 2014. https://github.com/scipy/scipy/blob/master/scipy/optimize/_differentialevolution.

- py.
- [98] Dbouk HM, Ghorayeb K, Kassem H, Hayek H, Torrens R, Wells O. Facility Placement Layout Optimization. J Pet Sci Eng 2021.