



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIAS DE LA TIERRA
INSTITUTO DE GEOFÍSICA

MÉTODOS METAHEURÍSTICOS DE OPTIMIZACIÓN GLOBAL
PARA LA SIMULACIÓN ESTOCÁSTICA BASADA EN CÓPULAS
DE PROPIEDADES PETROFÍSICAS A PARTIR DE ATRIBUTOS
SÍSMICOS ELÁSTICOS

T E S I S

QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS DE LA TIERRA

PRESENTA:

YARILIS GÓMEZ MARTÍNEZ

DIRECTOR DE TESIS:

DR. MARTÍN A. DÍAZ VIERA
(INSTITUTO MEXICANO DEL PETRÓLEO)

MIEMBROS DEL COMITÉ TUTOR:

DRA. ELSA LETICIA FLORES MÁRQUEZ
(INSTITUTO DE GEOFÍSICA, UNAM)
DR. RAÚL DEL VALLE GARCÍA
(INSTITUTO MEXICANO DEL PETRÓLEO)

CIUDAD DE MÉXICO

1 DE DICIEMBRE, 2020



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**Métodos metaheurísticos de optimización global para la
simulación estocástica basada en cópulas de propiedades
petrofísicas a partir de atributos sísmicos elásticos**

por

Yarilis Gómez Martínez

Ingeniera Geofísica

Tesis presentada para obtener el grado de

Maestro en Ciencias de la Tierra

en el

POSGRADO EN CIENCIAS DE LA TIERRA

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Ciudad de México, 1 de diciembre, 2020

*Todo lo que enaltece y honra
implica sacrificio*
ERNESTO GUEVARA DE LA SERNA

Agradecimientos

Gracias a cada una de las personas que me apoyaron e hicieron que este sueño se cristalizara en una hermosa realidad.

CRÉDITOS DE LA INFORMACIÓN PETROLERA PROPORCIONADA POR EL CNIH A LA UNAM.

“Esta tesis utilizó información petrolera proporcionada por la Comisión Nacional de Hidrocarburos, de acuerdo al anexo C de la licencia de uso de la información a favor de la UNAM, de fecha 11 de diciembre de 2017, bajo la nomenclatura CNIH-C-00417. Dicha información es propiedad de la Nación y su acopio, resguardo, uso, administración y actualización, así como de publicación de la misma corresponde a la Comisión Nacional de Hidrocarburos.”

CÓDIGO DE ÉTICA

Declaro conocer el Código de Ética de la Universidad Nacional Autónoma de México, plasmado en la Legislación Universitaria. Con base en las definiciones de integridad y honestidad ahí especificadas, aseguro mediante mi firma al calce que el presente trabajo es original y enteramente de mi autoría. Todas las citas de, o referencias a, las obras de otros autores aparecen debida y adecuadamente señaladas, así como acreditadas mediante los recursos editoriales convencionales.



Yarilis Gómez Martínez

Abstract

The simulation of petrophysical properties is of interest for the study of hydrocarbon reservoirs. Due to the intrinsic uncertainty of the measurements of each property, it is convenient to use stochastic models of dependence between petrophysical and elastic properties. Making possible to model the variable of interest through its relationship with others, usually known as secondary variables or support variables. For this reason, it is convenient to use dependency models based on copulas, since it does not require the choice of an specific dependency between the main and secondary variables.

Due to the shortcomings of classical simulation methods, particularly for dependencies between variables, joint simulation algorithms have been proposed based on the optimization of a multi-objective function. This function makes possible to link the joint distribution structures between the variables (through copulas), and the spatial dependence of the primary variable (through variogram). The multi-objective function approach is independent of the global optimization method used. Therefore it is necessary to establish which of them are compatible with this methodology to be able to choose the best in each case. To do this, a study of various metaheuristic methods of global optimization was carried out, choosing the two that best suited the nature of the problem: generalized simulated annealing and differential evolution.

The implementations made so far of this methodology are characterized by doing the co-simulation by copulas and the spatial simulation separately using a modification of the GSLIB library. That is why it was of special interest to implement the different methods so that both stages are carried out in a single library programmed in Python. Parametric copulas were used to estimate the joint probability distribution. Although non-parametric distributions allow a better fit to the empirical distributions of the variables, they are more expensive computationally.

The simulation was executed for three study cases, consisting of wells located in different areas. For this, the total porosity of the rock was simulated using the acoustic impedance given the strong dependence between these two variables. In general, both methods were effective, thus validating the proposed methodology. For a fourth case study where the porosity values were unknown, the methodology was applied obtaining satisfactory simulations of said property. Although the present work only considered a one-dimensional case, it is possible to extend it to more dimensions.

Resumen

La simulación de propiedades petrofísicas es de interés para el estudio de yacimientos de hidrocarburos. Debido a la incertidumbre intrínseca de las mediciones de cada propiedad, es conveniente utilizar modelos estocásticos de dependencia entre propiedades petrofísicas y elásticas. Haciendo posible modelar la variable de interés a través de su relación con otras a las que se les conoce como variables secundarias o variables de apoyo. Por este motivo se proponen modelos de dependencia basados en cópulas, ya que no requiere la elección de una dependencia específica entre las variables principales y secundarias.

Debido a las deficiencias de los métodos de simulación clásicos, en particular para las dependencias entre variables, se han propuesto algoritmos de simulación conjunta basados en la optimización de una función multiobjetivo. Esta función permite vincular las estructuras de dependencia entre las variables (mediante cópulas), y de dependencia espacial de la variable primaria (mediante variograma). El enfoque de función multiobjetivo es independiente del método de optimización global a emplear. Se hace necesario establecer entonces, cuales de ellos son compatibles con esta metodología para poder elegir los mejores en cada caso. Para ello se realizó un estudio de varios métodos metaheurísticos de optimización global, escogiendo los dos que más se ajustaban a la naturaleza del problema: recocido simulado generalizado y evolución diferencial.

Las implementaciones hasta ahora realizadas de esta metodología se caracterizan por hacer por separado la cosimulación mediante cópulas y la simulación espacial usando una modificación de la biblioteca GSLIB. Es por esto que fue de especial interés implementar los diferentes métodos de forma que ambas etapas se realicen en una biblioteca única programada en Python. Para la estimación de la distribución de probabilidad conjunta se usaron cópulas paramétricas. Si bien las distribuciones no paramétricas permiten un mejor ajuste a las distribuciones empíricas de las variables, son más costosas computacionalmente.

Se realizó la simulación para tres casos de estudio, consistente en pozos ubicados en diferentes áreas. Para esto se simuló la porosidad total de la roca a partir de la impedancia acústica debido a la fuerte dependencia entre estas dos variables. En general ambos métodos resultaron efectivos validándose así la metodología propuesta. Para un cuarto caso de estudio donde se desconocían los valores de porosidad, se aplicó la metodología obteniendo simulaciones satisfactorias de dicha propiedad. Aunque el presente trabajo solo consideró un caso unidimensional, es posible extenderlo a más dimensiones.

Índice general

Preámbulo	I
Abstract	IV
Resumen	v
Índice	VIII
1. Introducción	1
1.1. Antecedentes y motivación	2
1.2. Planteamiento del problema	3
1.3. Objetivos	3
1.4. Novedad científica	4
1.5. Contenido del documento	4
2. Metodología de cosimulación estocástica espacial	5
2.1. Análisis exploratorio univariado	6
2.1.1. Estimación no paramétrica de funciones de distribución	8
2.2. Análisis de dependencia espacial	9
2.3. Análisis bivariado de dependencia	11
2.3.1. Cópulas	12
2.3.2. Distribución condicional	15
2.4. Algoritmo de simulación con cópula condicional	16
2.5. Observaciones finales	17
3. Métodos metaheurísticos de optimización global estocástica	19
3.1. Recocido Simulado	20
3.1.1. Recocido Simulado Generalizado	23
3.2. Recocido Cuántico	25
3.3. Enjambres de Partículas	27
3.4. Colonias de Hormigas	29
3.5. Algoritmos Genéticos	32
3.5.1. Algoritmo de Evolución Diferencial	35
3.6. Comparación entre metaheurísticas	37

4. Optimización multiobjetivo	39
4.1. Problema de optimización	39
4.2. Implementación computacional	40
4.3. Recursos computacionales	43
5. Validación de los métodos de optimización	44
5.1. Validación con funciones de prueba	45
5.1.1. Análisis comparativo	48
5.2. Validación a partir de casos de estudios	51
5.2.1. Caso de estudio I	52
5.2.2. Caso de estudio II	78
5.2.3. Caso de estudio III	81
5.2.4. Discusión de casos	83
6. Aplicación a casos de estudio	85
6.1. Caso de estudio IV	85
6.1.1. Análisis estadístico y variográfico de los datos	85
6.1.2. Cosimulación con cópulas paramétricas	86
6.2. Discusión del caso de aplicación	91
7. Conclusiones y trabajos futuros	92
7.1. Conclusiones	92
7.2. Trabajos futuros	93
Apéndices	96
A. Simulación estocástica	96
A.1. Fundamentos teóricos	96
A.1.1. Funciones aleatorias	97
A.2. Métodos de simulación estocástica espacial	98
A.2.1. Método Secuencial Gaussiano	99
A.2.2. Método Secuencial Indicador	100
A.2.3. Método Gaussiano Truncado	101
B. Modelos de dependencia bivariada	104
B.1. Modelos de dependencia usando cópulas	104
B.2. Otros modelos dependencia	105
B.2.1. Medidas de asociación monótona	106
B.2.2. Dependencia de colas	109
C. Métodos de Monte Carlos con Cadenas de Markov	110
C.1. Cadenas de Markov	110
C.2. Método de Monte Carlo con cadenas de Markov	113
C.2.1. Algoritmo de Metropolis-Hastings	114
C.2.2. Algoritmo de Metropolis	115

C.2.3. Muestreo de Gibbs	115
D. Código básico de la implementación computacional	117
D.1. Etapas inicial y final	117
D.2. Implementación para DE	120
D.3. Implementación para GSA	121
E. Figuras del Capítulo 5	123
E.1. Figuras del caso de estudio I	123
E.2. Figuras del caso de estudio II	123
E.2.1. Análisis estadístico y variográfico de los datos	123
E.2.2. Cosimulación con cópulas paramétricas	123
E.3. Figuras del caso de estudio III	132
E.3.1. Análisis estadístico y variográfico de los datos	132
E.3.2. Cosimulación con cópulas paramétricas	132
Bibliografía	147

Índice de tablas

3-1. Comparación entre métodos metaheurísticos.	38
5-1. Estadística descriptiva para la función Ackley.	49
5-2. Estadística descriptiva para la función Rastrigin.	50
5-3. Estadística descriptiva para la función Rosenbrock.	51
5-4. Resumen de la comparación entre las dos variantes de cosimulación.	68
5-5. Resumen de la comparación entre las dos variantes de implementación.	71
5-6. Resumen de la comparación considerando o no puntos de condicionamiento.	78
5-7. Resumen de la comparación de los tres casos de estudio.	83

Índice de figuras

2-1. Esquema que muestra la metodología de cosimulación.	6
2-2. Diferentes tipos de kernel. Figura de Brian Amberg, con licencia CC-BY-SA 3.0.	9
2-3. Variograma experimental y teórico.	10
2-4. Ejemplo de distribución condicional a partir de una muestra. (a) Diagrama de dispersión (Y, X) de una muestra. (b) Función de distribución condicional de X condicionados a un valor de $Y = 8000$. (c) Función de distribución condicional de toda la muestra.	16
3-1. SA para altas y bajas temperaturas. Tomado de (Takyamamoto, 2017).	21
3-2. Esquema del método GSA.	25
3-3. Diferencia en la exploración del espacio por CSA, FSA y GSA. Las líneas representan el paso de la posición actual a la siguiente, marcando el espacio de búsqueda. Tomado de (Xiang y Gong, 2000).	26
3-4. Diferencia entre SA y QA. Tomado de (Das y Chakrabarti, 2005).	27
3-5. Representación esquemática de PSO. En líneas discontinua se representa la trayectoria, la flecha negra señala la mejor posición individual y la roja la mejor posición del grupo. Tomado de (Pérez López, 2005).	28
3-6. Ejemplo de funcionamiento del algoritmo ACO. N representa el nido y F lugar donde se encuentra la comida, a la trayectoria de ida y b de regreso. En línea naranja el rastro de feromonas, blancas las trayectorias posibles y en rojo se marca la trayectoria óptima. 1 representa el rastro inicial de feromonas, 2 la evaporación y 3 la trayectoria óptima. Figura de Johann Dréo, con licencia CC-BY-SA 3.0.	30
3-7. Ejemplo de funcionamiento del algoritmo GA. Tomado de (Campos Soberanis, 2019).	33
3-8. Esquema del método DE.	36
3-9. Ejemplo de la propiedad de coalecencia para DE con la función <i>Peacks</i> . A la izquierda se encuentra la población y a la derecha los vectores diferencia para diferentes generaciones. Tomado de (Price <i>et al.</i> , 2006).	37
5-1. Función Ackley. Tomado de (Gavana, 2016).	46
5-2. Función Rastrigin. Tomado de (Gavana, 2016).	47
5-3. Función Rosenbrock. Figura de Morn the Gorn, con licencia de dominio público.	48
5-4. Gráficos de error y tiempo de ejecución para la función Ackley.	50
5-5. Gráficos de error y tiempo de ejecución para la función Rastrigin.	50
5-6. Gráficos de error y tiempo de ejecución para la función Rosenbrock.	51
5-7. Registro del pozo <i>Lakach-1</i>	52

5-8. Estadística descriptiva para el pozo <i>Lakach-1</i>	53
5-9. Variograma de Phit para el pozo <i>Lakach-1</i>	54
5-10. Gráfico de dispersión de Ip contra Phit y estadística descriptiva bivariada de los datos.	54
5-11. Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de Ip y Phit respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.	55
5-12. Registros de los datos y simulados de la Phit y el error de estimación, para el pozo <i>Lakach-1</i> (DE a la izquierda, GSA a la derecha).	56
5-13. Estadística descriptiva del error en la simulación para el pozo <i>Lakach-1</i>	57
5-14. Variogramas y función de distribución condicional de la simulación para el pozo <i>Lakach-1</i> (DE a la izquierda, GSA a la derecha).	58
5-15. Estadística descriptiva de la simulación para el pozo <i>Lakach-1</i>	59
5-16. Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).	59
5-17. Gráfico de dispersión de Ip contra Phits (DE a la izquierda, GSA a la derecha) y estadística descriptiva bivariada de los datos simulados.	60
5-18. Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de Ip y Phits respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.	61
5-19. Registros de los datos y simulados de la Phit y el error de estimación, para el pozo <i>Lakach-1</i> (DE a la izquierda, GSA a la derecha).	62
5-20. Estadística descriptiva del error en la simulación para el pozo <i>Lakach-1</i>	63
5-21. Variogramas y función de distribución condicional de la simulación (DE a la izquierda, GSA a la derecha).	64
5-22. Estadística descriptiva de la simulación para el pozo <i>Lakach-1</i>	65
5-23. Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).	65
5-24. Gráfico de dispersión de Ip contra Phits (DE a la izquierda, GSA a la derecha) y estadística descriptiva bivariada de los datos simulados.	66
5-25. Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de Ip y Phits respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.	67
5-26. Registros de los datos y simulados de la Phit y el error de estimación, para DE, GSA, SA (de izquierda a derecha), en el pozo <i>Lakach-1</i>	68
5-27. Estadística descriptiva del error en el pozo <i>Lakach-1</i>	69

5-28. Variogramas y función de distribución condicional de la simulación para el pozo <i>Lakach-1</i> (de izquierda a derecha: DE, GSA y SA).	70
5-29. Estadística descriptiva de la simulación para el pozo <i>Lakach-1</i> . Histograma y gráfico de caja para los simulados (de izquierda a derecha: DE, GSA y SA).	70
5-30. Función de distribución empírica de probabilidad para las variables simuladas (de izquierda a derecha: DE, GSA y SA).	71
5-31. Gráfico de dispersión de I_p contra Phits (de izquierda a derecha: DE, GSA y SA) y estadística descriptiva bivariada de los datos simulados.	71
5-32. Registros de los datos, de la simulación condicional de la Phit y del error de estimación, para el pozo <i>Lakach-1</i> (DE a la izquierda, GSA a la derecha).	72
5-33. Estadística descriptiva del error en la simulación condicional para el pozo <i>Lakach-1</i>	73
5-34. Variogramas y función de distribución condicional de la simulación condicional para el pozo <i>Lakach-1</i> (DE a la izquierda, GSA a la derecha).	74
5-35. Estadística descriptiva de la simulación para el pozo <i>Lakach-1</i>	75
5-36. Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).	75
5-37. Gráfico de dispersión de I_p contra Phits (DE a la izquierda, GSA a la derecha) y estadística descriptiva bivariada de los datos simulados.	76
5-38. Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de I_p y Phits respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.	77
5-39. Registro del pozo <i>F03-FB-102</i>	79
5-40. Registros de los datos y simulados de la Phit y el error de estimación, para el pozo <i>F03-FB-102</i> (DE a la izquierda, GSA a la derecha).	80
5-41. Registro del <i>Pozo 2</i>	81
5-42. Registros de los datos y simulados de la Phit y el error de estimación, para el <i>Pozo 2</i> (DE a la izquierda, GSA a la derecha).	83
6-1. Registro del <i>Pozo 3</i>	86
6-2. Estadística descriptiva para el <i>Pozo 3</i>	87
6-3. Registros de los datos de V_p , R_{hob} e I_p así como de Phit simulada para el <i>Pozo 3</i> (DE a la izquierda, GSA a la derecha).	87
6-4. Variogramas y función de distribución condicional de la simulación para el <i>Pozo 3</i> (DE a la izquierda, GSA a la derecha).	88
6-5. Estadística descriptiva de la simulación para el <i>Pozo 3</i>	89
6-6. Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).	90
6-7. Gráfico de dispersión de I_p contra Phits y estadística descriptiva bivariada de los datos simulados (DE a la izquierda, GSA a la derecha).	90

6-8. Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de Ip y Phits respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.	91
A-1. Inversión de la función de distribución y selección de la categoría. Tomado de (Deutsch, 2002).	101
A-2. Simulación gaussiana truncada: la variable de gaussiana continua se trunca en una serie de umbrales para crear una realización de la variable categórica. Tomado de (Deutsch, 2002).	102
A-3. Ejemplo de distribución normal con dos umbrales (y_1^t y y_2^t), tres códigos de facies (1, 2 y 3) y tres valores de y continuos para cada facies (y_1 , y_2 , y y_3). La proporción de cada facies es el área bajo la curva en cada intervalo. Tomado de (Deutsch, 2002).	102
C-1. Grafo orientado correspondiente al ejemplo 1. Tomado de (Nicholls y Tan, 2006).	111
C-2. Grafo que representa una cadena irreducible. Tomado de (Nicholls y Tan, 2006).	112
C-3. Ejemplo de un algoritmo de Gibbs para dos parámetros. Tomado de (Ausín, 2012).	116
E-1. Gráfico de dispersión (Phit,Phits) para DE a la izquierda y GSA a la derecha para el pozo <i>Lakach 1</i> . Parte superior: cosimulación paramétrica, parte central: variante híbrida y parte inferior: considerando datos de condicionamientos. . . .	124
E-2. Gráfico de dispersión (Phit,Phits) para DE, GSA y SA de izquierda a derecha en el pozo <i>Lakach 1</i>	125
E-3. Estadística descriptiva para el pozo <i>F03-FB-102</i>	125
E-4. Variograma de Phit para el pozo <i>F03-FB-102</i>	126
E-5. Gráfico de dispersión de Ip contra Phit y estadística descriptiva bivariada de los datos.	126
E-6. Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de Ip y Phit respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.	127
E-7. Estadística descriptiva del error en la simulación para el pozo <i>F03-FB-102</i>	127
E-8. Variogramas y función de distribución condicional de la simulación para el pozo <i>F03-FB-102</i> (DE a la izquierda, GSA a la derecha).	128
E-9. Estadística descriptiva de la simulación para el pozo <i>F03-FB-102</i>	129
E-10. Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).	129
E-11. Gráfico de dispersión de Ip contra Phits (DE a la izquierda, GSA a la derecha) y estadística descriptiva bivariada de los datos simulados.	130

E-12. Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de Ip y Phits respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.	131
E-13. Estadística descriptiva para el <i>Pozo 2</i>	132
E-14. Variograma de Phit para el <i>Pozo 2</i>	133
E-15. Gráfico de dispersión de Ip contra Phit y estadística descriptiva bivariada de los datos.	133
E-16. Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de Ip y Phit respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.	134
E-17. Estadística descriptiva del error en la simulación para el <i>Pozo 2</i>	134
E-18. Variogramas y función de distribución condicional de la simulación para el <i>Pozo 2</i> (DE a la izquierda, GSA a la derecha).	135
E-19. Estadística descriptiva de la simulación para el <i>Pozo 2</i>	136
E-20. Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).	136
E-21. Gráfico de dispersión de Ip contra Phits (DE a la izquierda, GSA a la derecha) y estadística descriptiva bivariada de los datos simulados.	137
E-22. Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de Ip y Phits respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.	138

Capítulo 1

Introducción

La modelación de yacimientos proporciona los elementos necesarios para mejorar la producción de hidrocarburos mediante la predicción de patrones de recuperación. En particular la simulación de propiedades petrofísicas es de interés en diferentes áreas. Este proceso depende de un Modelo Geológico Petrofísico de Yacimiento (MGPY), el cual consiste en un modelo de distribución espacial de propiedades petrofísicas, dentro de un marco geológico en un yacimiento. Para obtener dicho modelo es necesario la integración de datos (provenientes de núcleos, de registros de pozo, de la sísmica, etc.) con un modelo geológico conceptual (Casar González *et al.*, 2012).

Los MGPY se basan en técnicas que permiten obtener modelos numéricos confiables que representan el modelo geológico y el petrofísico. Las técnicas geoestadísticas facilitan combinar información de manera integral, analizar la influencia de los diversos parámetros que intervienen en el modelo y realizar un análisis del grado de incertidumbre asociado. Estos modelos pueden ser estáticos o dinámicos, dependiendo de si incluyen datos relacionados con el flujo (Cosentino, 2001).

Las rocas que forman parte del yacimiento son usualmente interpretadas como un medio poroso. Las características y propiedades de esta red porosa, son las propiedades petrofísicas, y están relacionadas con los procesos de litificación y diagénesis de la roca. La información petrofísica se obtiene de las diferentes curvas de los registros geofísicos de pozo y por lo general se realiza un proceso de calibración con la información petrofísica obtenida de los núcleos mediante pruebas de laboratorio.

Debido a la incertidumbre intrínseca de las mediciones o que no se dispone de una cantidad representativa de las mismas, es conveniente utilizar modelos estocásticos de dependencia entre propiedades petrofísicas y elásticas para la modelación de yacimientos. De manera que sea posible modelar la variable de interés a través de su relación con otras, a estas se les conoce como variables secundarias o variables de apoyo. Por ejemplo, la impedancia acústica usualmente se encuentra correlacionada con la porosidad, la permeabilidad con la porosidad, la saturación de agua con el módulo de rigidez y otras.

Los modelos de dependencia entre variables primarias y secundarias se basan en leyes físicas, relaciones funcionales empíricas, o principalmente en técnicas de regresión. Estas últimas no permiten representar adecuadamente la variabilidad de los datos, que en la mayoría de los casos constituye una parte fundamental de la información. Más aún, el enfoque de regresión muchas veces está restringido a suponer dependencias lineales entre las variables, que en la

práctica pueden ser poco frecuentes o limitadas a ciertos casos particulares (Díaz-Viera y Casar-González, 2005; Maldonado, 2014; Mendoza-Torres *et al.*, 2017).

1.1. Antecedentes y motivación

Existe una gran variedad de métodos de simulación espacial estocásticos ampliamente reconocidos en el estado del arte, como pueden ser la simulación secuencial Gaussiana o indicador, la simulación Gaussiana truncada y la simulación basada en objetos (Deutsch, 2002). En algunos de ellos la dependencia entre las variables no se tiene en cuenta o se parte de hipótesis demasiado restrictivas sobre el tipo de dependencias subyacentes.

Una metodología sistemática para la cosimulación, es decir, simular variables espaciales preservando estructuras de dependencia bivariada, fue propuesta inicialmente por Deutsch y Cockerham (1994). La dependencia espacial y la distribución conjunta de las variables se vinculan en una función multiobjetivo (FMO). Esta metodología está implementada utilizando un método de recocido simulado clásico en la biblioteca GSLIB (Clayton V. Deutsch, 1997), en la que se utilizan los histogramas, la función de distribución condicional y el coeficiente de correlación lineal para aproximar la distribución conjunta.

De manera general, para establecer estas dependencias se requiere del conocimiento de la función de distribución de probabilidad conjunta de las variables. No es posible acceder a esta información de manera directa, es por ello que se ajustan modelos de distribución de probabilidad conocidos a los datos. Una manera eficiente de obtener estas distribuciones es mediante el uso de unas funciones denominadas cópulas (Nelsen, 2006; Salvadori *et al.*, 2007; Joe, 2014).

Para saldar las deficiencias de los métodos de simulación clásicos respecto a las dependencias entre variables tanto Díaz-Viera y Casar-González (2005) como Díaz-Viera *et al.* (2006) utilizaron algoritmos de cosimulación general basado en t -cópulas como parámetros de entrada para recocido simulado clásico consiguiendo una simulación espacial que respeta la dependencia entre las variables. Esta nueva metodología fue extendida mediante distribuciones no paramétricas basadas en cópulas de Bernstein en trabajos posteriores (Erdely y Diaz-Viera, 2010; Díaz Viera *et al.*, 2016; Ramírez, 2018).

Esta aplicación de las cópulas ha mostrado ser efectiva para modelación de fracturas utilizando simulación basada en objetos (Mendoza-Torres *et al.*, 2017), y para la simulación de propiedades petrofísicas en general (Maldonado, 2014; Díaz Viera *et al.*, 2016). En particular, comparaciones entre los métodos de cosimulación secuencial Gaussiana y cosimulación espacial basada en cópulas de Bernstein han demostrado que este último representa una alternativa más sistemática y flexible, brindando mejores resultados (Ramírez, 2018; Le *et al.*, 2020).

El enfoque de FMO es independiente del método de optimización global a emplear. Se hace necesario establecer entonces, cuales de ellos son compatibles con esta metodología para poder elegir los mejores en cada caso. En este sentido, los métodos metaheurísticos de optimización global permiten buscar de manera eficiente el óptimo de dicho problema.

El presente trabajo fue desarrollado dentro de el Grupo de Modelación de Ciencias de la Tierra, que ha desarrollado parte de las metodologías antes expuestas para la cosimulación espacial estocásticas de propiedades petrofísicas. Partiendo de un modelo geológico conceptual y de que existe una fuerte relación de dependencia entre las propiedades petrofísicas y elásticas en un pozo, en este trabajo se simulará la primera a partir de la segunda. Para ello se estudiarán

varios métodos de optimización global estocástico para minimizar la FMO, y en particular nos centraremos en el conjunto de los conocidos como metaheurísticos, que serán analizados con más detalle posteriormente.

Las implementaciones hasta ahora realizadas de esta metodología se caracterizan en hacer por separado la cosimulación mediante cópulas, de la cual se obtienen realizaciones, y la simulación espacial usando una modificación de **GSLIB**, donde las realizaciones anteriores son datos de entrada. Fue de especial interés implementar los diferentes métodos en una biblioteca única programada en Python.

Para la estimación de la distribución de probabilidad conjunta se usarán cópulas paramétricas. Si bien las distribuciones no paramétricas permiten un mejor ajuste a las distribuciones empíricas de las variables, son más costosas computacionalmente. La metodología propuesta en este trabajo está enfocada a escala de registro de pozos, aunque se puede extender a escala sísmica de manera similar a los trabajos de Maldonado (2014).

1.2. Planteamiento del problema

Debido al deficiente manejo de dependencias de los métodos de simulación espacial estocásticos tradicionales se han ido adoptando métodos alternativos de simulación como son la utilización de cópulas y la optimización de FMO. La metodología establecida por Díaz-Viera y Casar-González (2005) ha mostrado ser efectiva para la cosimulación espacial, sin embargo quedan aún muchas interrogantes por resolver en cada una de sus etapas. En particular el método de optimización empleado hasta ahora no se ha comparado con otros similares presentes en la literatura.

En este trabajo se propone una metodología de cosimulación estocástica espacial que consiste de dos etapas. En la primera se modelan las estructuras: distribución conjunta entre todas las variables mediante cópulas, y de dependencia espacial de la variable a simular a través del variograma. En la segunda se simula espacialmente la variable primaria mediante métodos de optimización con una FMO que vincule ambas estructuras y se establece una comparación entre dichos métodos. En particular se tomó como variable primaria a la porosidad total y como secundaria a la impedancia acústica.

Por lo tanto, el *problema* de esta investigación consiste en aplicar varios métodos metaheurísticos de optimización global para la cosimulación espacial estocástica de la porosidad total a partir de la impedancia acústica en un pozo, de manera tal que dicha simulación respete la dependencia, y los métodos de optimización elegidos aprovechen al máximo la información disponible. Para resolver dicho problema partiremos de la siguiente *hipótesis*: a partir de la impedancia acústica obtenida en registros de un pozo es posible obtener simulaciones estocásticas de la porosidad total aplicando métodos metaheurísticos de optimización global mediante el uso de cópulas.

1.3. Objetivos

Para llevar a cabo esta investigación se trazó como *objetivo general*: Desarrollar una metodología de cosimulación espacial usando cópulas paramétricas para la simulación de propiedades petrofísicas a partir de atributos sísmicos elásticos en yacimientos petroleros.

A partir del mismo se plantearon los siguientes *objetivos específicos*:

- Analizar varios métodos metaheurísticos de optimización global y seleccionar los que mejor se ajusten para simular la porosidad total en un pozo a partir de la impedancia acústica.
- Implementar computacionalmente los métodos de optimización global seleccionados e incorporarlos a la metodología desarrollada.
- Validar los métodos de optimización escogidos y la metodología propuesta a partir de tres casos de estudio donde se conoce la porosidad total.
- Aplicar dicha metodología a un cuarto caso de estudio para simular la porosidad total que se desconoce.

1.4. Novedad científica

La metodología presentada se encuentra bien establecida y tiene múltiples aplicaciones. En implementaciones anteriores se utiliza solo un recocido simulado discreto y el ajuste de distribuciones se realiza mediante histogramas, por lo que en la implementación de los métodos escogidos evitamos especialmente estas limitaciones. Podemos identificar como puntos más novedosos de este trabajo los siguientes:

- Se utilizaron varios métodos de optimización global (recocido simulado generalizado y evolución diferencial).
- Se escogieron familias de cópulas paramétricas para las dependencias entre variables.
- Se realizó una implementación en una biblioteca única de Python, utilizando paquetes estándar.

1.5. Contenido del documento

En el siguiente capítulo se detallan los fundamentos teóricos de la metodología de cosimulación a aplicar, haciendo énfasis en la teoría de cópulas. En el capítulo 3 se estudian diferentes métodos de optimización global y se comparan de acuerdo a sus características. De estos métodos se escogen los dos mejores para su uso posterior. La implementación de estos métodos y la metodología de cosimulación se encuentra expuesta en el capítulo 4. El capítulo 5 se dedica especialmente a las validaciones correspondientes. Por último se discute la aplicación de la metodología a un caso de estudio en el capítulo 6.

Capítulo 2

Metodología de cosimulación estocástica espacial

La manera más utilizada para generar realizaciones de una función aleatoria es a través de la simulación estocástica espacial, entre los métodos más comunes se encuentran por ejemplo los métodos Secuencial Gaussiano y Secuencial Indicador (véase el apéndice A). Estos son métodos ampliamente reconocidos en el estado del arte, sin embargo están limitados a casos donde la correlación espacial está caracterizada por una estacionariedad débil y la dependencia entre las variables se tiene en cuenta de forma limitada (Deutsch, 2002; Maldonado, 2014). El método de cosimulación Secuencial Gaussiano, por ejemplo, resulta en una disminución de la longitud de correlación de los valores en las colas, siendo incapaz de simular grandes regiones conectadas o con valores extremos (Deutsch, 2002; Mendoza-Torres *et al.*, 2017).

La metodología que utilizaremos ha sido propuesta por Deutsch y Cockerham (1994); Díaz-Viera y Casar-González (2005), ya que permite preservar la estructura de dependencia entre las variables a la vez que genera dichas realizaciones. Este enfoque consiste en obtener realizaciones de la variable de interés empleando una técnica de optimización multiobjetivo donde tanto la distribución espacial como la estructura de dependencia se aproximen simultáneamente a sus equivalentes teóricas. Para ello se definirá una función multiobjetivo (FMO), donde la distribución espacial estará representada a partir de un modelo de correlación espacial (variograma teórico), mientras que la estructura de dependencia se modela utilizando cópulas bivariadas, las cuales pueden ser paramétricas o no paramétricas (Díaz-Viera *et al.*, 2006; Maldonado, 2014).

Adoptaremos la metodología anterior teniendo en cuenta algunas modificaciones. Se utilizarán cópulas paramétricas para el análisis de dependencia. Su ventaja radica en que son más eficientes computacionalmente, aunque las no paramétricas representen mejor la dependencia de los datos. Dentro de las paramétricas se escogieron las arquimedianas ya que son las más usadas para problemas estocástico de este tipo. Otra diferencia importante es el análisis de varios métodos de optimización y la implementación computacional que se verá más adelante en los capítulos 4 y 3 respectivamente.

La metodología consta de dos etapas (figura 2-1). En una primera etapa se establece un modelo de dependencia, formado por una cópula y las distribuciones marginales de las dos variables. En esta etapa se obtiene también el variograma teórico a partir de información a priori de la variable a simular. En la segunda etapa se realiza una cosimulación espacial estocástica utilizando un método de optimización global y la FMO definida.

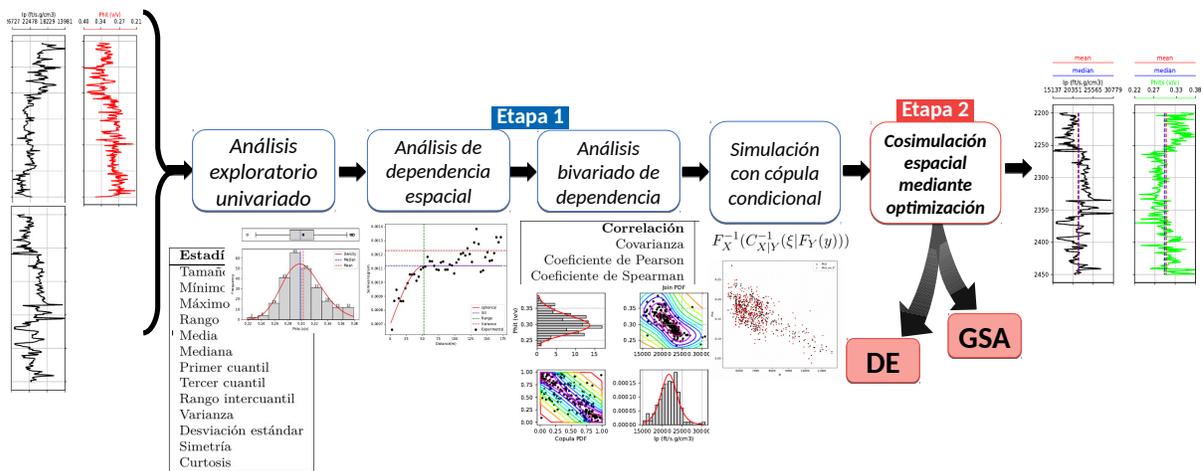


Figura 2-1: Esquema que muestra la metodología de cosimulación.

Analizaremos en este capítulo la primera etapa para un caso unidimensional (un pozo), aunque esta metodología es extensible a un contexto más general (Maldonado, 2014). Los métodos de optimización necesarios para la segunda etapa serán estudiados con más detalles en el capítulo 3, y la FMO e implementación de dichos métodos para este problema serán vistos en el capítulo 4.

Para la aplicación de esta metodología es necesario tener en cuenta los siguientes supuestos:

- Dependencia espacial: que las variables tengan correlación espacial consigo mismas.
- Dependencia bivariada: que exista una fuerte relación entre las variables, no necesariamente lineal.

Como punto de partida se conoce la información de la variable secundaria en la región a estimar. La variable de interés o variable primaria solo se conoce parcialmente y se utilizará información externa, de otro pozo o zona por ejemplo, para estimar los parámetros correspondientes a la región donde se realizará la simulación. Primeramente se realiza un análisis univariado del cual se obtiene la información fundamental de cada una de las variables y en particular se obtienen ajustes de sus funciones de distribución univariada. Luego se realiza un análisis variográfico de la variable primaria con el cual se estimará un variograma teórico. Realizaremos un análisis bivariado para estudiar el patrón de dependencia entre ellas y extraer en última instancia su distribución conjunta, nos auxiliaremos para ello de la teoría de cópulas. La metodología descrita se verá con más detalle en el resto del capítulo. En el apéndice A se encuentran algunas definiciones formales de esta metodología.

2.1. Análisis exploratorio univariado

El análisis univariado explora un conjunto de variables a partir de muestras de cada una por separado. Estudia el patrón de respuesta de cada variable por sí misma: para ello se examina el rango de valores, como se agrupan, si hay valores extremos, etc. Se basa en el concepto de

variable aleatoria, que no es más que una variable cuyos valores solo pueden conocerse con cierto grado de certeza. Para su estudio se necesita conocer la probabilidad de ocurrencia de los diferentes valores posibles, por lo que se recurre al conjunto de sus valores conocidos, denominado *muestra o realizaciones*. En este sentido el análisis univariado no es solo una forma de describir y resumir datos sino que nos brinda la capacidad de analizar las variables aleatorias asociadas a dichos datos (Kachigan, 1986).

Se considera que la función de distribución de una variable aleatoria contiene toda la información de su comportamiento de manera individual, es por esto que obtenerla es una tarea fundamental. Se define como *función de distribución (CDF)* $F_X : \mathbb{R} \rightarrow [0, 1]$ de una variable aleatoria X a

$$F_{\mathbf{X}}(x) = \mathbb{P}(X \leq x). \quad (2-1)$$

Si esta función es diferenciable se dice que la variable aleatoria es *absolutamente continua* y se le llama *densidad de probabilidad (PDF)* a la función f_X tal que

$$F_{\mathbf{X}}(x) = \int_{-\infty}^x f_X(t) dt. \quad (2-2)$$

Estos conceptos pueden ser extendido para más de una variable siendo objeto de estudio del análisis bivariado y multivariado en general.

Como solo se conocen de cada variable valores de una realización solo es posible estimar sus propiedades. Para ello se estiman los estadígrafos de cada variable (media, mediana, varianza, desviación estándar, entre otros), se grafican los histogramas y gráficos de caja, y se ajustan funciones de distribución de acuerdo a los datos disponibles.

Existen dos formas de estimar la función de distribución a partir de una muestra. La primera mediante una función de distribución cuya forma está dada por un grupo de parámetros (enfoque paramétrico), pudiendo estimar los parámetros de ajuste a partir de datos muestrales. La otra variante se basa en estimar directamente la distribución a partir de una muestra (enfoque no paramétrico). El enfoque principal de nuestro trabajo es paramétrico, aunque también se exploró una variante híbrida (sección 5.2.1.3) para un caso de estudio.

De los diferentes parámetros que se pueden estimar a partir de una muestra, los más comunes son los siguientes:

- Parámetros básicos: tamaño de la población, mínimo y máximo
- Medidas de tendencia central:
 - media aritmética
 - mediana o segundo cuartil (separa los datos mayores y menores en dos partes iguales)
 - primer cuartil (valor por encima del 25 % y por debajo del 75 % de los datos)
 - tercer cuartil (valor por encima del 75 % y por debajo del 25 % de los datos)
- Medidas de dispersión:
 - rango (diferencia entre el valor máximo y mínimo)
 - rango intercuartil (diferencia entre el tercero y primer cuartil)
 - varianza (mide la desviación de la variable respecto a la media)

- desviación estándar (raíz cuadrada de la varianza)
- Medidas de forma:
 - simetría (indica si los valores se concentran por encima o por debajo de la media)
 - curtosis o apuntamiento (mide la concentración de valores muy cerca de la media).

2.1.1. Estimación no paramétrica de funciones de distribución

La estimación paramétrica utiliza familias de distribuciones que dependen de parámetros para estimar la función de distribución de los datos, sin embargo la no paramétrica estima esta función de distribución de manera más flexibles sin necesidad de calcular parámetros para su ajuste. Una manera elemental de estimar dicha distribución es a través de la *función de distribución empírica* (Dudley, 2014), que se define como

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{[x_{mi}, \infty)\}(x) = \frac{\text{número de elementos en la muestra } \leq x}{n}. \quad (2-3)$$

Aunque también se puede definir a partir del operador *rango* (rank) definido en el apéndice B (Le *et al.*, 2020).

La principal propiedad de la función de distribución empírica de la muestra es su aproximación a la función de distribución poblacional cuando aumenta el tamaño muestral. Este resultado queda establecido a través del teorema de Glivenko-Cantelli (Dudley, 2014).

Las principales carencias de la función de distribución empírica son que se necesita una muestra grande para obtener una buena aproximación y no permite incorporar información a priori sobre la distribución de la variable aleatoria. En particular \hat{F}_n será una función escalonada, aún cuando la variable aleatoria considerada sea continua, es por esto que se encuentran emplean diferentes técnicas de suavizado disponibles en la literatura. Los principales métodos de estimación no paramétrica (con suavizado) son: mediante *splines* tipo B (Brinks, 2008; Shen *et al.*, 2008), con polinomios de Bernstein (Erdely y Diaz-Viera, 2010) y estimación por kernel (para variables aleatorias absolutamente continuas) (Wand y Jones, 1994; Chacón y Duong, 2018).

Estimación por kernel: la estimación de la densidad por núcleo (KDE por sus siglas en inglés) o ventana de Parzen-Rosenblatt es un método no paramétrico que, como su nombre lo indica, permite estimar la función de densidad de probabilidad de una variable aleatoria. Es ampliamente empleado cuando las inferencias acerca de la población se realizan sobre la base de un conjunto finito de muestras (Wand y Jones, 1994; Chacón y Duong, 2018).

Para una variable aleatoria el estimador KDE se define como

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_{mi}) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right), \quad (2-4)$$

donde K es una función conocida como *núcleo* o *kernel* y x_{mi} son los valores de la muestra. Al parámetro de suavidad $h > 0$ se le conoce como *ancho de banda* y a $K_h(x) = K(x/h)/h$ se le conoce como núcleo escalado.

El núcleo es una función con soporte finito e integral unitaria (Wand y Jones, 1994), es decir, que debe satisfacer $\int_S K(x)dx = 1$, ($K(y) = 0, \forall y \notin S$). Este núcleo puede interpretarse como una función de pesos que permiten la *entrada* de determinados valores y su importancia, es por eso que también se le suele denominar *función ventana* en algunas aplicaciones.

Existen varias funciones del núcleo utilizadas comúnmente: Gaussiano o normal, uniforme, triangular, cuadrático o doblemente pesado (*biweight*), triplemente pesado (*triweight*), de Epanechnikov, entre otros; como se muestra en la figura 2-2. El núcleo Epanechnikov es óptimo en el sentido del error cuadrático medio, aunque el kernel normal se utiliza con mayor frecuencia.

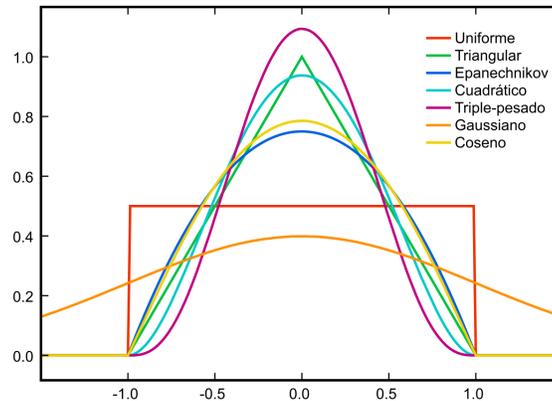


Figura 2-2: Diferentes tipos de kernel. Figura de Brian Amberg, con licencia CC-BY-SA 3.0.

2.2. Análisis de dependencia espacial

Para el análisis de dependencia espacial utilizaremos modelos basados en funciones aleatorias. Para esto se usan diferentes medidas como las funciones de covarianzas y de semivarianzas o comúnmente llamado *variograma*. El variograma es la medida de correlación espacial más utilizada en la práctica y se define como la varianza de la diferencia entre los valores de la función aleatoria en dos puntos con una separación fija (Díaz Viera, 2002):

$$\gamma(h) = \text{var}[Z(\mathbf{x}) - Z(\mathbf{x} + h)], \quad (2-5)$$

sin embargo en la práctica la manera de estimarlo es a partir de

$$\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [Z(\mathbf{x}_i) - Z(\mathbf{x}_i + h)]^2, \quad (2-6)$$

donde Z es la función aleatoria y $N(h)$ es el número de parejas de observaciones que se encuentran separadas por un vector de desplazamiento de longitud h . En la práctica se aceptan vectores con una cierta tolerancia.

A partir de la ecuación 2-6 se estima un *variograma experimental*; sin embargo como este no es continuo para poder utilizar la información de correlación espacial, se le ajusta un modelo matemático conocido como *variograma teórico* que lo aproxima mediante una función continua.

Propiedades del variograma: (Lantuéjoul, 2002; Wackernagel, 2003)

- Es no negativo $\gamma(h) \geq 0$ y par $\gamma(h) = \gamma(-h)$.
- Para $h = 0$, $\gamma(0) = 0$.
- Una función es un semivariograma si y sólo si es una función condicionalmente definida negativa, esto es:

$$\sum_{i=1}^N \sum_{j=1}^N w_i \gamma(\mathbf{x}_i - \mathbf{x}_j) w_j \leq 0 \quad \text{si} \quad \sum_{i=1}^N w_i = 0. \quad (2-7)$$

- Como consecuencia de las dos anteriores, el variograma podría ser discontinuo en el origen. Al salto en el origen se le conoce como pepita o *efecto de pepita*. En la práctica esta discontinuidad se debe a la resolución del muestreo de los datos.
- Si Z no tiene ninguna dependencia espacial, el variograma es constante de $\gamma(h) = \text{var}(Z)$ en todas partes excepto en el origen, donde es cero.
- Si Z es estacionaria y ergódica (véase teorema.C.1.2, del apéndice C), entonces existe el límite $\lim_{h \rightarrow \infty} \gamma(h) = \text{var}(Z)$, que corresponde a la varianza. Este límite también se llama *meseta* y generalmente coincide con el valor de la varianza de los datos.

Se denomina *alcance* a la distancia en la que la diferencia entre el variograma y la meseta se vuelve no significativo, es decir la distancia a partir de la cual los datos ya no se encuentran correlacionados. En los modelos teóricos usuales los parámetros principales de ajuste son el efecto pepita, el alcance y la meseta (figura 2-3).

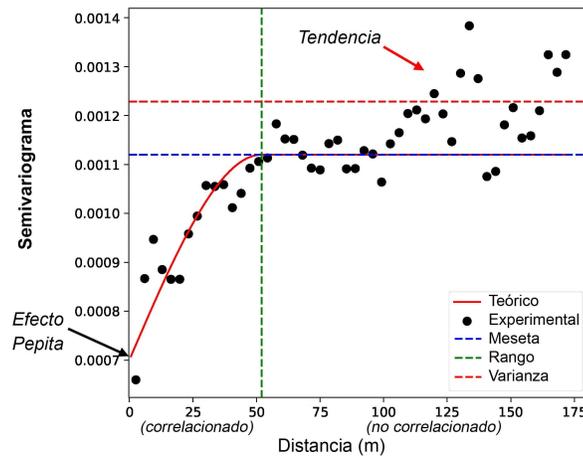


Figura 2-3: Variograma experimental y teórico.

Para que el variograma sea una medida representativa de la variabilidad espacial es necesario que la variable cumpla ciertas condiciones relacionadas con el concepto de estacionaridad (Díaz Viera, 2002; Cuador Gil, 2004). En particular, basta que cumpla con la conocida como hipótesis intrínseca (para más detalles véase el apéndice A).

Si $Z(\mathbf{x})$ es no estacionaria, su valor esperado depende de de la ubicación \mathbf{x} y se le conoce como deriva o *tendencia*. Para evitar este fenómeno es posible utilizar una función $m(\mathbf{x}) = E[Z(\mathbf{x})]$, generalmente polinómica, que recoja la parte determinista de la variable, es decir

$$Z(\mathbf{x}) = m(\mathbf{x}) + R(\mathbf{x}); \quad (2-8)$$

a $R(\mathbf{x})$ se le conoce como residuo. La forma más común de manejar este fenómeno es mediante el método de kriging residual (Díaz Viera, 2002), que se basa en utilizar una estimación de la tendencia por mínimos cuadrados (o mínimos cuadrados generalizados), realizar el análisis con los residuos y recuperar los valores de interés sumando la tendencia estimada.

Un ejemplo particular son las tendencias lineales, es decir, $m(\mathbf{x}) = a + b \cdot \mathbf{x}$, puede ser estimada mediante las ecuaciones $b = \frac{Cov(Z,x)}{Var(x)}$, $a = E[Z] - b \cdot E[x]$. Una importante propiedad de este tipo de tendencias es que, sustituyendo la ecuación (2-8) en el variograma, este puede escribirse de la forma

$$\begin{aligned} \gamma(h) &= \frac{1}{2} E \left[(R(x) - R(x+h) - bh)^2 \right] \\ &= \frac{1}{2} E[(bh)^2] + \frac{1}{2} E \left[(R(x) - R(x+h))^2 \right] \\ &= \frac{1}{2} (bh)^2 + \gamma_R(h), \end{aligned}$$

donde $\gamma_R(h)$ es el variograma del residuo. Es decir, se tiene una ecuación explícita para el cálculo del variograma sin tendencia a partir del variograma de los datos:

$$\gamma_R(h) = \gamma(h) - \frac{1}{2} (bh)^2. \quad (2-9)$$

En general el algoritmo a seguir es:

1. En caso de que exista tendencia: eliminarla.
2. Seleccionar las distancias h y ángulos en el caso de que se trabaje con variogramas direccionales.
3. Calcular el variograma experimental basado en la muestra seleccionada.
4. Seleccionar un variograma teórico que se ajuste al experimental.

2.3. Análisis bivariado de dependencia

Un *vector aleatorio* reúne de forma ordenada a varias variables aleatorias relacionadas, por lo que se puede considerar una generalización de dicho concepto. Estos son el objeto de estudio del análisis multivariado y es de especial interés la estructura de dependencia que puede existir entre las variables que componen a los vectores aleatorios.

Una de las mayores ventajas de incorporar la estructura de dependencia es que es posible atrapar y reproducir las características de cada variable por separado y reproducir simultáneamente su comportamiento conjunto. Por lo tanto, necesitamos definir la distribución conjunta de las variables y extraer de ella el complejo patrón de dependencia que frecuentemente aparece

entre propiedades petrofísicas. El concepto de cópulas, que se verá más adelante, nos permitirá capturar esta información.

Llamaremos *función de distribución (CDF)* $F_{\mathbf{X}} : \mathbb{R}^d \rightarrow [0, 1]$ de un vector aleatorio $X = (X_1, \dots, X_d)$ a

$$F_{\mathbf{X}}(x_1, x_2, \dots, x_d) = \mathbb{P} \left(\bigcap_{i=1}^d \{X_i \leq x_i\} \right). \quad (2-10)$$

Para $d \geq 2$ se le conoce como *función de distribución conjunta*.

Una idea fundamental será la de distribución marginal la cual proporciona la probabilidad de un subconjunto de las variables aleatorias sin necesidad de conocer los valores de las otras. En particular nos interesan las marginales 1-dimensionales F_j . Si $F_{\mathbf{X}}$ es la función de distribución de X entonces la marginal F_j es la función de distribución de su componente X_j (Jaworski *et al.*, 2010; Joe, 2014).

Si bien de una función de distribución conjunta nos es posible obtener las distribuciones marginales, el procedimiento inverso no es evidente. En particular si las variables aleatorias X_1, X_2, \dots, X_d son *independientes* se puede demostrar que

$$F_{\mathbf{X}}(x_1, x_2, \dots, x_d) = \prod_{i=1}^d F_{X_i}(x_i). \quad (2-11)$$

Esto no ocurre cuando existe dependencia entre las variables, ya que durante la obtención de distribuciones marginales se pierde información, se hace necesario entonces encontrar una forma de obtener una relación entre las funciones de distribución (Erdely Ruiz y Gutiérrez Peña, 2012).

Para la estimación de las distribuciones marginales se pueden utilizar métodos paramétricos y no paramétricos (como los descritos en la sección 2.1.1), estos pueden ser extendidos permitiéndonos estimar las distribuciones multivariadas. En la sección 2.3.1 utilizaremos el concepto de cópula para estudiar las relaciones entre las variables, lo que nos permitirá además obtener la distribución conjunta a partir de las distribuciones marginales. En la sección 2.3.2 se verá el concepto de distribución condicional que resultará fundamental en nuestro estudio.

2.3.1. Cópulas

La manera más general de abordar la dependencia entre variables aleatorias es a través del estudio de las cópulas. Una de las ventajas que brinda la metodología propuesta es que el número de funciones a incorporar en la FMO se reduce considerablemente, ya que la cópula al reproducir la estructura de dependencia entre variables aleatorias es capaz de reproducir la distribución individual de cada una de ellas y los coeficientes de correlación lineal (Maldonado, 2014). Para más detalles sobre el análisis de dependencia véase el apéndice B.

Una *cópula* bidimensional (o 2-cópula o simplemente cópula) es una función C con las siguientes propiedades

1. $\text{Dom } C = [0, 1]^2$,
2. $C(x, 0) = C(0, y) = 0 \quad \forall x, y$,

3. para todo $B = [x_1, x_2] \times [y_1, y_2]$:

$$V_C(B) = C(x_2, y_2) - C(x_2, y_1) - C(x_1, y_2) + C(x_1, y_1) \geq 0, \quad (2-12)$$

4. para todo $(u, v) \in \text{Dom } C$: $C(u, 1) = u$ y $C(1, v) = v$.

A la propiedad 2 se le conoce como conexión a tierra y se dice que C está conectada a tierra o *aterrizada*. La propiedad 3 se conoce como *2-creciente* y puede ser satisfecha por otras funciones $H : \mathbb{R}^2 \rightarrow \mathbb{R}$, y a V_H se le conoce como H -volumen del rectángulo B . La propiedad 4 implica que la imagen de C es un subconjunto de $[0, 1]$, es decir $0 \leq C(x, y) \leq 1 \quad \forall x, y$ (Erdely Ruiz y Gutiérrez Peña, 2012; Díaz Viera *et al.*, 2016).

Una función H con $\text{Dom } H = S_1 \times S_2$ con $S_1, S_2 \subset [0, 1]$, que cumpla el resto de propiedades (2-4), se denomina *subcópula*, este concepto será usado exclusivamente para el teorema de Sklar que enunciaremos a continuación. En el apéndice B se pueden encontrar algunas de las propiedades más importantes de las cópulas y subcópulas (Erdely Ruiz y Gutiérrez Peña, 2012).

Teorema 2.3.1 (Teorema de Sklar). *Sea F una función de distribución conjunta con marginales F_X y F_Y . Entonces existe una subcópula $\bar{C}_{X,Y}$ tal que para toda $x, y \in \mathbb{R} \cup \{-\infty, +\infty\}$:*

$$F(x, y) = \bar{C}_{X,Y}(F_X(x), F_Y(y)), \quad (2-13)$$

y está determinada de forma única en $\text{Im } F_X \times \text{Im } F_Y$.

A $\bar{C}_{X,Y}$ es posible extenderla de forma continua definiendo así una cópula $C_{X,Y}$, llamada *función cópula asociada* a (X, Y) , que contiene información acerca de la relación de dependencia entre ambas variables. Si F_X y F_Y son absolutamente continuas entonces la cópula así definida es única, obteniéndose un método para construir las cópulas a partir de la función de distribución conjunta conociendo las distribuciones marginales (Nelsen, 2006).

Teorema 2.3.2 (Corolario). *Sea F una función de distribución conjunta con marginales F_X y F_Y continuas, y sea $C_{X,Y}$ la única cópula que satisface el Teorema 2.3.1. Entonces, para cualquier $(u, v) \in [0, 1]^2$:*

$$C_{X,Y}(u, v) = F(F_X^{-1}(u), F_Y^{-1}(v)). \quad (2-14)$$

En la ecuación 2-14, F_X^{-1}, F_Y^{-1} son las cuasi-inversas de F_X y F_Y o *funciones cuantil* de X e Y , respectivamente, y se definen como

$$F^{-1}(p) = \inf\{x \in \mathbb{R} : p \leq F(x)\}. \quad (2-15)$$

El teorema de Sklar es central en la teoría de cópulas y es la base de la mayoría de las aplicaciones de esa teoría a la estadística. De esta forma es posible construir modelos de probabilidad conjunta de una manera más flexible, ya que podemos elegir por separado los modelos univariados para las variables aleatorias de interés y la función cópula que representa mejor la dependencia entre ellos (Díaz Viera *et al.*, 2016). Dicha función contiene información valiosa acerca del tipo de dependencia que existe entre variables aleatorias, en Mikusiński *et al.* (1991) se las interpreta como un representante canónico de todas las distribuciones conjuntas cuyas variables aleatorias X, Y poseen cierto tipo particular de dependencia entre ellas.

Hay muchas formas de desarrollar modelos multivariados paramétricos a partir de un conjunto de variables aleatorias. Las estructuras de dependencia adecuadas dependen del contexto de las aplicaciones (Joe, 2014), es por esto que las cópulas se suelen agrupar en familias o clases.

Clase de Fréchet: esta clase agrupa tres cópulas que representan dependencias extremas: la cópula de independencia (C^\perp) y las cotas superior e inferior de Fréchet-Hoeffding (C^+ y C^-).

Supongamos que X y Y son *independientes*, con distribuciones $F_X(x), F_Y(y)$ continuas. Entonces, $F(x, y) = F_X(x) * F_Y(y)$, por lo que la única cópula a la que se refiere el teorema de Sklar es en este caso

$$C^\perp(u_1, u_2) := u_1 u_2. \quad (2-16)$$

$$C^\perp = C(F_1(x_1), F_2(x_2)) = F_1(x_1) \cdot F_2(x_2) = F(x_1, x_2), \quad (2-17)$$

conocida como cópula de independencia.

La cota superior de Fréchet-Hoeffding es

$$C^+(u_1, u_2) := \min\{u_1, u_2\}. \quad (2-18)$$

Ninguna otra cópula puede tomar un valor superior a esta. Cuando las variables aleatorias están relacionadas de acuerdo con esta, se dice que poseen *dependencia positiva perfecta* o comonotonicidad.

La cota inferior de Fréchet-Hoeffding es

$$C^-(u_1, u_2) := \max\{u_1 + u_2 - 1, 0\}. \quad (2-19)$$

Ninguna otra cópula puede tomar un valor inferior a esta. Cuando las variables aleatorias están relacionadas de acuerdo con esta, se dice que poseen *dependencia negativa perfecta* o contramonotonicidad. Esta solo es una cópula unicamente para dimensión 2, aunque sigue siendo una cota inferior para más dimensiones.

Las cópulas Fréchet C^+ y C^- son cópulas simples que unen variables aleatorias con correlación de Spearman +1 y -1 respectivamente, y es posible combinarlas linealmente para obtener cualquier otro valor (McClarren, 2018).

Dado que muchas familias incluyen a estas cópulas, es necesario conocer las propiedades de dependencia para distinguir familias paramétricas y comprender cuándo se pueden usar. Las familias que no incluyen a C^+ tienen un rango limitado de dependencia y en general no son útiles para el modelado estadístico (Joe, 2014).

Cópulas de Frank: se encuentra dentro de la familia de cópulas arquimedianas. Para $-\infty < \delta < \infty$ la función de distribución de la cópula es

$$C(u, v; \delta) = -\delta^{-1} \log \left(\frac{1 - e^{-\delta} - (1 - e^{-\delta u})(1 - e^{-\delta v})}{1 - e^{-\delta}} \right), \quad 0 \leq u, v \leq 1 \quad (2-20)$$

Algunas propiedades son las siguientes (Joe, 2014):

- Cuando $\delta \rightarrow -\infty$ esta cópula es la cota inferior (C^-), cuando $\delta \rightarrow 0^+$ se convierte en la cópula de independencia (C^\perp) y cuando $\delta \rightarrow \infty$ coincide con la cota superior (C^+).
- Simetría de reflexión de la densidad de probabilidad $c(1-u, 1-v; \delta) = c(u, v; \delta)$. Frank (1979) caracterizó esto como la única simetría de reflexión para cópulas arquimedianas.
- Si (U, V) está en esta familia con el parámetro δ , entonces $(U, 1-V)$ tiene una cópula dentro de la misma familia, ya que $u - C(u, 1-v; \delta) = C(u, v; -\delta)$.
- Construido a partir de la arquimediana con $\varphi(t; \delta) = -\log \left[\frac{(1 - e^{-\delta t})}{(1 - e^{-\delta})} \right]$; para $0 \leq t \leq 1$. El inverso es $\varphi^{-1}(s; \delta) = -\delta^{-1} \log \left[1 - (1 - e^{-\delta}) e^{-s} \right]$.
- El coeficiente de Spearman es: $\rho_S = 1 + 12\delta^{-1} [D_2(\delta) - D_1(\delta)]$; y el coeficiente de Kendall es: $\tau = 1 + 4\delta^{-1} [D_1(\delta) - 1]$, donde $D_k(x) = kx^{-k} \int_0^x t^k (e^t - 1)^{-1} dt$ para $k = 1, 2$.
- Las esquinas de la función de densidad de cópula son $c(0, 0; \delta) = c(1, 1; \delta) = \delta(1 - e^{-\delta})^{-1}$.
- La función de distribución condicional (véase la ecuación 2-21) se expresa como:

$$C_{2|1}(v|u; \delta) = e^{-\delta u} \left[(1 - e^{-\delta}) (1 - e^{-\delta v})^{-1} - (1 - e^{-\delta u}) \right]^{-1}.$$

El enfoque de cópulas ha sido ampliamente utilizado para la modelación de patrones de dependencia en varias áreas de la industria del petróleo, más allá de la cosimulación estocástica. Por ejemplo, en la toma de decisiones (de Melo e Silva Accioly y Chiyoshi, 2004), para análisis económicos (Lorenzo Valdés *et al.*, 2016) y de riesgo (Cediel, 2016).

2.3.2. Distribución condicional

Partiendo del concepto de probabilidad condicional se define la *función de distribución condicional*, de un vector aleatorio bidimensional de la siguiente manera:

$$F_{X|Y}(x|y) = \mathbb{P}(X \leq x | Y = y). \quad (2-21)$$

Si ambas variables aleatorias son continuas se puede calcular también mediante límites como:

$$F_{X|Y}(x|y) = \frac{\partial F / \partial y(x, y)}{f_Y(y)}. \quad (2-22)$$

En el apéndice A se encuentran las definiciones formales y generalizaciones a más dimensiones de estos conceptos.

De la ecuación 2-22 se deriva el concepto de *función de densidad de probabilidad condicional* de X dada la aparición del valor y de Y , la cual se puede escribir como:

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x, y)}{\int_{-\infty}^{\infty} f_{X,Y}(x, y) dx} = \frac{f_{X,Y}(x, y)}{f_Y(y)}, \quad (2-23)$$

donde $f_{X,Y}(x, y)$ es la densidad conjunta de X e Y , mientras que $f_Y(y)$ es la densidad marginal para Y . En este caso es estrictamente necesario que $f_Y(y) > 0$ (McClarren, 2018).

Los conceptos anteriores permiten, entre otras cosas, simular y estimar una variable aleatoria, conocida como *variable primaria* a partir de la información conocida de otra u otras (*variables secundarias*). En el caso bivariado para cualquier valor de la variable secundaria Y la distribución condicional está definida en el espacio de la variable primaria X . Si se conoce una muestra de ambas variables, asociado a cada valor fijo y se tiene un conjunto de valores muestrales $\{(x_1; y), \dots, (x_k; y)\}$ (figura 2-4,a); y para cada uno de estos se conoce a su vez la probabilidad asociada a ese valor condicionada a $Y = y$ (figura 2-4,b). De manera que, si quisiéramos construir la distribución condicional de toda muestra bivariada, entonces para cada valor de la variable secundaria se deben considerar los valores de la probabilidad asociados a todas las muestras que tienen a dicho valor en la componente de la variable secundaria (figura 2-4,c).

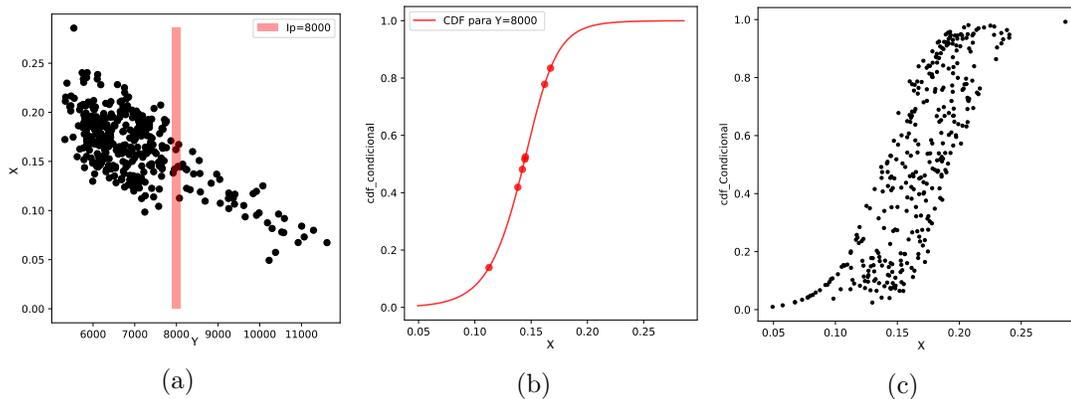


Figura 2-4: Ejemplo de distribución condicional a partir de una muestra. (a) Diagrama de dispersión (Y, X) de una muestra. (b) Función de distribución condicional de X condicionados a un valor de $Y = 8000$. (c) Función de distribución condicional de toda la muestra.

2.4. Algoritmo de simulación con cópula condicional

Existen varias técnicas clásicas para obtener una realización aleatoria de una distribución cuya función de distribución es conocida. La más común es llamada *inversión de la función de distribución*: se basa en la propiedad de que, si U tiene una distribución uniforme sobre $[0, 1]$, para cualquier función de distribución F , entonces $V = F^{-1}(U)$ se distribuye de acuerdo a F :

$$U \sim U(0, 1) \Rightarrow F_X^{-1}(U) \sim X, \quad (2-24)$$

donde F_X^{-1} es la función cuantil de la variable aleatoria X . Si F_X^{-1} tiene una expresión analítica simple, entonces esta es una manera eficiente de generar realizaciones de X evaluando realizaciones de una variable aleatoria con distribución uniforme sobre $[0, 1]$. En general existen varios métodos para calcular la función cuantil de una variable aleatoria.

Hay una manera directa de producir realizaciones a partir de la distribución conjunta de un vector aleatorio producida por cópulas (McClarren, 2018). Considerando las funciones de distribución marginales para las variables aleatorias X y Y , $F_X(x)$ y $F_Y(y)$ respectivamente,

y una cópula $C(u, v)$. El procedimiento para una distribución conjunta dada por $F_{X,Y}(x, y) = C(F_X(x), F_Y(y))$ es el siguiente:

1. Generar realizaciones ξ_1^* y ξ_2^* de $\xi_1, \xi_2 \sim U(0, 1)$, respectivamente.
2. Hacer $\omega = C_{X|Y}^{-1}(\xi_1^*|\xi_2^*)$.
3. Entonces las muestras (x, y) son $x = F_X^{-1}(\xi_1^*)$ y $y = F_Y^{-1}(\omega)$.

es decir,

$$(F_X^{-1}(\xi_1), F_Y^{-1}(C_{X|Y}^{-1}(\xi_1|\xi_2))) \sim (X, Y). \quad (2-25)$$

A este método se le conoce como *inversión condicional de cópula general* (o de distribuciones multivariadas generales). Se ha usado la fórmula de función de densidad condicional (2-22) para definir $C_{X|Y}(u|v) = \partial C / \partial v$.

De lo anterior se tiene que, en general, para cualquier distribución conjunta dada por $F_{X,Y}(x, y) = C(F_X(x), F_Y(y))$, si F_Y es absolutamente continua, la ecuación (2-22) es equivalente a

$$F_{X|Y}(x|y) = C_{X|Y}(F_X(x)|F_Y(y)). \quad (2-26)$$

Este procedimiento puede ser adaptado para obtener muestras de la distribución condicional, teniendo en cuenta que

$$F_X^{-1}(C_{X|Y}^{-1}(\xi|F_Y(y))) \sim (X|Y = y). \quad (2-27)$$

para $\xi \sim U(0, 1)$. Quedando de esta manera el siguiente algoritmo:

1. Generar realizaciones ξ^* de $\xi \sim U(0, 1)$ y calcular $\eta = F_Y(y)$.
2. Hacer $\omega = C_{X|Y}^{-1}(\xi^*|\eta)$.
3. Entonces las muestras de $(X|Y = y)$ son $x = F_X^{-1}(\omega)$.

Este procedimiento de muestreo es simple de realizar con la posible excepción de no conocer $C_{X|Y}^{-1}$. En esos casos, se suele usar un solucionador no lineal para realizar la inversión.

Es importante no confundir los algoritmos anteriores con el concepto de *simulación espacial condicional*. Diremos que una simulación espacial es condicional si la realización resultante $Z_C^*(\mathbf{x})$ reproduce los valores duros de la muestra de datos. Esto significa que solo se toman aquellas realizaciones $Z^*(\mathbf{x})$ en que para los puntos muestrales $\{\mathbf{x}_i, i = 1, \dots, n\}$ coinciden los valores simulados $Z^*(\mathbf{x}_i)$ con los valores reales o experimentales $Z(\mathbf{x}_i) = x_{mi}$ (Díaz Viera, 2002). Este tipo de condicionamiento es válido para cualesquiera de los métodos de simulación estocástica espacial descritos. En particular la decisión de aplicarla a nuestra metodología dependerá de la disponibilidad de información de la variable primaria para cada caso de estudio.

2.5. Observaciones finales

La metodología aplicada para esta investigación se basa en: primeramente un análisis univariado de las variables primaria y secundaria, aquí se estiman los estadígrafos y se realiza un ajuste de las marginales. Luego se hace un análisis de dependencia espacial basado en modelos

de la variable primaria como función aleatoria, específicamente se hace un ajuste del variograma teórico. También se realiza un análisis bivariado estimando la función de distribución conjunta a través de una cópula y las marginales. Tanto el variograma teórico como la distribución conjunta forman parte de los parámetros de entrada para la siguiente etapa, que es la cosimulación espacial donde se define una FMO que se minimiza mediante los métodos de optimización global que se seleccionarán en el capítulo 3.

Capítulo 3

Métodos metaheurísticos de optimización global estocástica

En este capítulo realizaremos una revisión bibliográfica y análisis de una serie de métodos metaheurísticos de optimización global que pudieran ser utilizados en simulación estocástica. Además se detallarán algunos conceptos básicos necesarios para la comprensión de estos métodos. Nuestro objetivo es seleccionar los más adecuados en términos de eficiencia y desempeño para su implementación.

El muestreo y la optimización son dos problemas íntimamente relacionados. El muestreo es un problema relativamente más fácil y que se puede considerar como una primera fase de la segunda (Liang *et al.*, 2010). Sin embargo también puede afirmarse que un problema de optimización es posible reducirlo, por lo menos conceptualmente, a un problema de generación de realizaciones con una adecuada distribución de probabilidad.

Los métodos de optimización global pueden ser clasificados como deterministas o estocásticos de acuerdo al manejo de la incertidumbre. Debido a la naturaleza de los datos que se utilizan en este trabajo es más conveniente emplear el enfoque probabilístico. En este contexto es necesario comprender los Métodos de Monte Carlo con cadena de Markov ya que se utilizarán en el desarrollo y justificación de los métodos empleados, es por ello que nos referiremos a esta teoría en el apéndice C.

Los métodos estocásticos de optimización global se basan en la teoría de probabilidad para el análisis de las variables, estos realizan una búsqueda exhaustiva por el espacio. Los métodos de búsqueda aleatoria son métodos de optimización estocástica capaces de generar soluciones razonablemente buenas de forma rápida y sencilla, por lo que son frecuentemente utilizados en la práctica (Serrano Cádiz, 2011).

Una heurística es una técnica diseñada para resolver un problema específico con un esfuerzo computacional razonable y con un alto grado de confianza sobre la calidad de la solución, aunque no se pueda garantizar su optimalidad. Se usa el calificativo heurístico en contraposición a exacto, en el que se presupone que el algoritmo siempre converge a una solución óptima bajo ciertos supuestos (Riojas y Álvarez, 2005). Los métodos metaheurísticos de optimización global son heurísticas de nivel superior proporcionando un marco general para diseñar o mejorar las heurísticas. Estos permiten la exploración de espacios de búsqueda mediante el uso de diferentes estrategias, especialmente cuando la información sobre el problema o la capacidad de cálculo son limitadas (Bianchi *et al.*, 2009). Para realizar esta búsqueda de manera eficiente

podemos auxiliarnos en principios físicos y biológicos de la naturaleza, aunque no existe una clara diferenciación entre estocásticos y metaheurísticos.

Una gran ventaja es que suelen hacer pocas suposiciones sobre el problema de optimización que se está resolviendo, por lo que pueden ser útiles para gran variedad de problemas. Si bien no garantizan que se pueda encontrar una solución globalmente óptima en un tiempo finito, a menudo puede encontrar buenas soluciones con menos esfuerzo computacional que los algoritmos de optimización clásicos y otros métodos iterativos (Blum y Roli, 2003).

Varias taxonomías han sido desarrolladas pero la más común distingue entre *métodos basados en trayectoria* y *métodos basados en población*. La primera incluye todos los que utilizan un mecanismo de exploración caracterizable por una trayectoria en el espacio de búsqueda, mientras que la segunda se refiere a los que utilizan un conjunto de soluciones potenciales en lugar de explotar una única solución (Soria *et al.*, 2016).

3.1. Recocido Simulado

En metalurgia la técnica de recocido consiste en calentar un metal a altas temperaturas, para darle una gran cantidad de energía a su estructura interna. Luego se deja que se enfríe lentamente, manteniéndolo en cada etapa del enfriamiento con una temperatura constante, si la disminución de la temperatura es muy rápida, puede causar defectos en su red cristalina (sólido amorfo). Esta estrategia de disminución controlada de la temperatura conduce a un estado sólido cristalizado homogéneo (estado estable) y que corresponde a un mínimo absoluto de energía (Avner, 1988). El método de recocido simulado (SA por sus siglas en inglés) traslada el proceso físico a la solución de un problema de optimización, donde la función objetivo del problema (que denotaremos por E en analogía con el concepto físico de energía) se minimiza con la ayuda de la introducción de un parámetro control que llamaremos temperatura (T).

En cada iteración, el método de recocido simulado evalúa algunos vecinos del estado actual i y probabilísticamente decide entre efectuar una transición a un nuevo estado j o quedarse en el estado i . Típicamente la comparación entre estados vecinos se repite hasta que se encuentre un estado óptimo que minimice la energía del sistema o hasta que se cumpla cierto tiempo computacional u otras condiciones.

La probabilidad de hacer la transición al nuevo estado j es una función $P(\Delta E, T)$ de la diferencia de valor de la función objetivo $\Delta E = E(j) - E(i)$ entre los dos estados y de la temperatura T . Si ΔE es negativo, es decir, con la transición disminuye E , el movimiento es aceptado con probabilidad $P = 1$. Cuando ΔE es positivo la probabilidad de transición es siempre distinta de cero, es decir, el sistema puede pasar a una peor solución que el estado actual. Esta propiedad impide que el sistema se quede atrapado en un óptimo local.

Este proceso se repite de manera sistemática hasta que hayan ocurrido un número suficiente de modificaciones, manteniendo la temperatura constante (Maldonado, 2014). Después, se baja la temperatura, y el proceso anterior se repite. Cuando T llega a cero, el algoritmo solo aceptará cambios a estados que mejoren la función objetivo (figura 3-1).

El funcionamiento de este método se basa en lo siguiente: A la función objetivo $E(x)$ se le puede asociar a una densidad de probabilidad de la forma

$$q_T(x) \propto H \circ E(x)/T = H(E(x)/T),$$

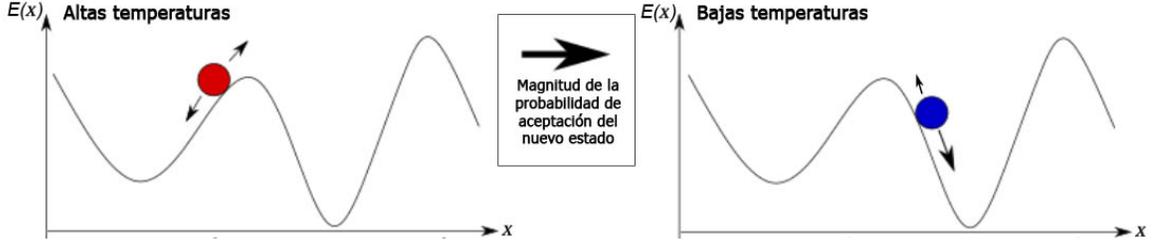


Figura 3-1: SA para altas y bajas temperaturas. Tomado de (Takyamamoto, 2017).

conocida como función de generación. Donde $H(\vartheta) : \mathbb{R} \rightarrow \mathbb{R}$ es una función positiva decreciente y T es la temperatura, que actúa como un parámetro de escala. Por ejemplo, para $H(\vartheta) = \exp(-\vartheta)$, se obtiene $q_T(x) = \exp(-E(x)/T)$, conocida como distribución de Boltzmann (Serrano Cádiz, 2011).

T está relacionado con la varianza. Cuanto más pequeño es el valor de T , más estrecha se hará la función $q_T(x)$ en torno a su máximo global (mínimo global para E). De este modo, cuando T es pequeño la convergencia del método de optimización es más rápida. Para el caso límite $q_T(x)$ se convierte en una delta de Dirac.

De esta forma se garantiza que los valores \hat{x} que minimizan la función objetivo son los mismos que maximizan la función $q_T(x)$, es decir, sus modas coinciden:

$$\hat{x} = \arg \min_x E(x) = \arg \max_x q_T(x)$$

Además, muestrear desde una densidad $q_T(x)$ significa generar números aleatorios de acuerdo a su función de distribución asociada.

En la práctica, esta técnica de muestreo utiliza el algoritmo de Metrópolis, interpretando al sistema termodinámico en equilibrio como una distribución estacionaria de la cadena de Markov (véase apéndice C), esto se hace para cada temperatura.

$$\alpha(j|i) = \min\left(1, \frac{h(j)}{h(i)}\right) \quad (3-1)$$

donde $h(i) = \exp\{-E/T\}$ es, precisamente, la distribución de Boltzman (Sen y Stoffa, 2015).

La temperatura juega un papel muy importante en el control de la evolución del sistema. A temperaturas altas, el sistema hará saltos grandes entre los estados, mientras que a temperaturas más bajas, los cambios serán menores.

Para el algoritmo de recocido simulado se debe especificar los siguientes valores de los parámetros de control de la temperatura:

- La temperatura inicial.

La elección de la temperatura inicial T_0 con la cual empezar el proceso iterativo es a veces crítica en el SA. Si es demasiado alta, al sistema le tomará muchas iteraciones para alcanzar un estado óptimo. Contrariamente, si es demasiado baja, el sistema *crystalizará* demasiado pronto, quedando en un óptimo local. Esto no sucede con variantes donde la disminución de la función objetivo es grande en cada paso (ejemplo recocido simulado muy rápido, VFSA, por sus siglas en inglés), donde T_0 puede ser relativamente arbitraria.

Un criterio estándar es tomar T_0 de forma tal que el valor de la función objetivo a dicha temperatura esté próxima a su valor medio. Esto asegura que al principio del proceso casi todos los nuevos estados propuestos sean aceptados (Maldonado, 2014).

- La longitud de las cadenas homogéneas de Markov, es decir, el criterio para el cambio de la etapa de temperatura.

No existe consenso sobre este punto. Para cada temperatura suele existir diferentes comportamientos según el problema en cuestión. Volviendo a la analogía del proceso físico, si el material es enfriado con la lentitud suficiente, la estructura cristalina resultante presenta una configuración de mínima energía. Por el contrario, si la temperatura disminuye demasiado rápido, se empezarán a observar distorsiones en la estructura cristalina. Por lo tanto, para alcanzar el estado de mínima energía se debe conseguir un estado de equilibrio térmico antes de disminuir la temperatura. Cerca del final del proceso, es de esperar que el sistema esté en la cuenca del mínimo global, la temperatura es prácticamente cero, y el proceso asintóticamente se convierte en un método de descenso de gradiente.

- La ley de disminución de la temperatura o esquema de enfriado.

Este es quizá el más crítico de los parámetros a ajustar y juega un rol clave en el proceso de optimización llevado a cabo por el SA. De los múltiples esquemas existentes, el *esquema geométrico* es el más aceptado, debido a su simplicidad:

$$T_{k+1} = \alpha T_k.$$

Geman y Geman (1984) demostraron que una condición necesaria y suficiente para que el mínimo global pueda ser alcanzado es que la temperatura no disminuya más rápido que

$$T_k = \frac{T_0}{\ln(1+k)},$$

para una función de generación uniforme. Cuando el estado siguiente se busca de acuerdo a una distribución Gaussiana en la vecindad de su estado real (es decir, una distribución de búsqueda local) (Kirkpatrick *et al.*, 1983), este método a veces se denomina recocido simulado clásico (CSA, por sus siglas en inglés) o máquina de Boltzmann.

Este esquema, junto con una modificación en la etapa de generación propuesto por Szu y Hartley (1987) (buscando de acuerdo a una distribución Cauchy-Lorentz en lugar de Gaussiana), permite obtener un esquema inversamente proporcional al número de iteraciones:

$$T_k = \frac{T_0}{k},$$

conocido como recocido simulado rápido (FSA, por sus siglas en inglés) o máquina de Cauchy.

Para un espacio de parámetros $N \times M$ -dimensional, Ingber (1989) propuso la técnica de VFSA. Este propone un esquema de enfriado más rápido, modificando nuevamente la distribución utilizada en la etapa de generación, donde se le permite a cada parámetro tener diferentes grados de perturbaciones de su posición actual. El esquema de enfriado

resultante garantiza estadísticamente la convergencia al mínimo global y está dado por:

$$T_l(k) = T_{0l} \exp\left(-c_l k^{\frac{1}{NM}}\right)$$

(T_{0l} es la temperatura inicial y c_l el parámetro de control de la variable l -ésima).

- El criterio de terminación del programa.

El algoritmo termina cuando la solución obtenida permanece igual en un número determinado de transiciones. Es importante notar que no es necesario dejar que la temperatura alcance el cero.

En ausencia de resultados teóricos generales, el usuario tiene que recurrir a un ajuste empírico de estos parámetros. Para ciertos problemas, la tarea es aún más complicada por la gran sensibilidad de los resultados y el tiempo de cálculo al ajuste. Por lo tanto, la velocidad de enfriamiento ideal no puede determinarse de antemano y debe ajustarse empíricamente para cada problema.

Algunas mejoras en los tiempos de cálculo y costo computacional pueden ser obtenidos en tres direcciones específicas: el ajuste interactivo de parámetros, la paralelización del algoritmo y la incorporación de otros enfoques basados en la física. Los algoritmos de recocido simulado adaptativo abordan este problema conectando el enfriamiento al progreso de la búsqueda. Otro enfoque adaptativo como el recocido termodinámico simulado, ajusta automáticamente la temperatura en cada paso en función de la diferencia de energía entre los dos estados, de acuerdo con las leyes de la termodinámica.

Ventajas: Se observa que la técnica de recocido simulado, a diferencia de otros métodos de búsqueda local, alcanza una solución de buena calidad (es decir, mínimo absoluto o mínimo relativamente *bueno* para la función objetivo). Más aún, es un método general aplicable a múltiples problemas, fácil de implementar y polinómico. Ofrece una gran flexibilidad, ya que se pueden añadir fácilmente nuevas restricciones al algoritmo.

Desventajas: Es necesario un compromiso entre la calidad de las soluciones y el tiempo requerido para procesarlas, por lo que el tiempo de cálculo que conlleva puede ser excesivo en ciertas aplicaciones. El protocolo de enfriamiento no es fácil de diseñar debido especialmente al gran número de parámetros (como los vistos anteriormente). Aunque los valores estándar publicados para estos parámetros de manera general permiten un buen funcionamiento del método, la naturaleza esencialmente empírica de ellos no puede garantizar la idoneidad para una gran variedad de problemas.

3.1.1. Recocido Simulado Generalizado

Como fue mencionado anteriormente los métodos de recocido simulado evalúan vecinos del estado actual y decide entre efectuar una transición a un nuevo estado. La forma de generar los vecinos, para el caso continuo es a partir de una distribución de probabilidad que suele depender de la distancia a recorrer, la dimensión del espacio de búsqueda, la temperatura u otros parámetros.

El recocido simulado generalizado (GSA por sus siglas en inglés), fue obtenido por Tsallis y Stariolo (1996) y se refiere a la generalización de los recocidos simulados clásico y rápido (CSA y FSA). Tanto CSA como FSA se generaban a partir de una distribución de Boltzman, esto estaba justificado teóricamente a partir de una maximización de la entropía de información de un sistema físico. GSA se obtiene al maximizar una entropía generalizada propuesta en Tsallis (1988) dependiente de un parámetro (la cual tiende a la entropía de información cuando el parámetro tiende a 1). De lo anterior se deriva la forma Tsallis-Stariolo de la distribución de visitas Cauchy-Lorentz, esta converge puntualmente a la distribución de Boltzmann cuando el parámetro tiende a 1, y tiene la forma:

$$g_{q_v}(\Delta x(t)) \propto \frac{[T_{q_v}(t)]^{-\frac{D}{3-q_v}}}{\left[1 + (q_v - 1) \frac{\Delta x(t)^2}{[T_{q_v}(t)]^{\frac{2}{3-q_v}}}\right]^{\frac{1}{q_v-1} + \frac{D-1}{2}}}, \quad (3-2)$$

donde t es el tiempo artificial y $x(t)$ es el estado actual. El parámetro q_v controla la forma de esta distribución y se conoce como *coeficiente de visita*. Esta distribución de visitas se utiliza para generar una distancia de salto Δx bajo la temperatura de visita del estado actual $T_{q_v}(t)$.

El nuevo estado se acepta si la función objetivo mejora, y en caso contrario podría aceptarse de acuerdo con una probabilidad de aceptación. Se utiliza un algoritmo generalizado de Metropolis-Hastings para la probabilidad de aceptación:

$$p_{q_a} = \text{máx} \left\{ 0, \text{mín} \left\{ 1, [1 - (1 - q_a) \beta \Delta E]^{-\frac{1}{1-q_a}} \right\} \right\}, \quad (3-3)$$

donde q_a es un parámetro conocido como *coeficiente de aceptación* y $\beta = (KT_{q_a})^{-1}$, que está controlada por una temperatura artificial de aceptación T_{q_a} .

El esquema de enfriado de GSA está controlado por el coeficiente de visita de la manera siguiente:

$$T_{q_v}(t) = \frac{2^{q_v-1} - 1}{(1+t)^{q_v-1} - 1} T_{q_v}(1). \quad (3-4)$$

No existe una manera específica de seleccionar la temperatura artificial de aceptación, aunque una técnica simple para acelerar la convergencia general del método es tomarla de la forma

$$T_{q_a}(t) = \frac{T_{q_v}(t)}{t}. \quad (3-5)$$

El rendimiento de GSA puede mejorarse aún más combinando GSA con un método de búsqueda local, realizando una búsqueda local al final de cada cadena de Markov (Xiang *et al.*, 2013). En la figura 3-2 se muestra un esquema que resumen el método de GSA.

Propiedades de q_v y q_a : Cuando $q_v = 1$ y $q_a = 1$, GSA recupera CSA, es decir

$$g_1(\Delta x) \propto \exp\left(-\frac{(\Delta x)^2}{T}\right); \quad (3-6)$$

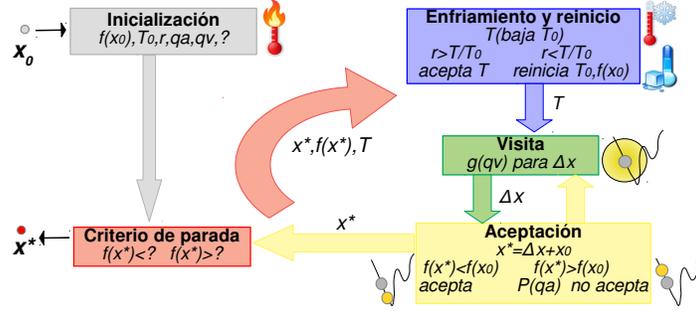


Figura 3-2: Esquema del método GSA.

y cuando $q_v = 2$ y $q_a = 1$, GSA recupera FSA:

$$g_2(\Delta x) \propto \frac{T}{[T^2 + (\Delta x)^2]^{\frac{D+1}{2}}} \quad (3-7)$$

Cuando $T \rightarrow 0$, GSA se comporta como el algoritmo de descenso de gradiente. Para valores mayores de $q_v = 2$ el enfriamiento es más rápido que el de CSA y FSA. GSA no solo converge más rápido que FSA y CSA, sino que también tiene la capacidad de escapar de un mínimo local más fácilmente. Dado que GSA tiene una gran probabilidad de tener un salto largo, la probabilidad de encontrar el mínimo global con GSA es mayor que con FSA y CSA, ya que explora un área mucho mayor como se muestra en la figura 3-3. Un q_v más grande (bajo la restricción $q_v < 3$) hace que el enfriamiento sea más rápido y salte aún más. Los valores negativos de q_a con mayor valor absoluto hace que GSA acepte menos las escaladas. Se ha demostrado en algunos casos que GSA es superior a FSA y CSA en la optimización del problema de estructura cristalina.

Los valores predeterminados de q_v y q_a se establecen en 2.62 y -5 respectivamente de acuerdo con Tsallis y Stariolo (1996); Xiang *et al.* (2013).

Ventajas: Se trata de una generalización de CSA y FSA por lo que es posible utilizarlos con una misma implementación, aprovechando sus ventajas. Modificando los parámetros es posible obtener un método que converja más rápido que estos.

Desventajas: Posee las mismas desventajas que otros SA.

3.2. Recocido Cuántico

Esté método ha mostrado poder resolver problemas *NP-complejo* que se consideran irresolubles con las técnicas existentes. Es un enfoque inspirado en la física cuántica y SA, sin embargo la función objetivo es cambiada por el Hamiltoniano de los estados del sistema. El hamiltoniano cuántico H es el observable que representa la energía total del sistema, es decir, es una propiedad que puede ser determinada físicamente. Los posibles valores de la energía (función

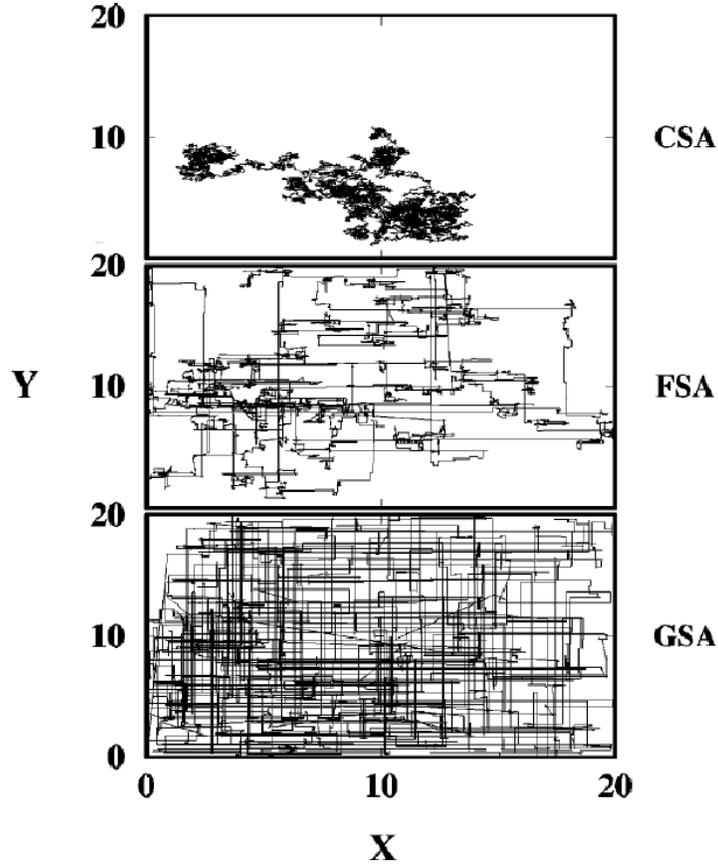


Figura 3-3: Diferencia en la exploración del espacio por CSA, FSA y GSA. Las líneas representan el paso de la posición actual a la siguiente, marcando el espacio de búsqueda. Tomado de (Xiang y Gong, 2000).

objetivo) vienen dados por los valores propios del operador hamiltoniano (Crosson y Harrow, 2016).

El recocido cuántico comienza a partir de una superposición mecánica cuántica de todos los estados posibles (estados candidatos) con pesos iguales. Luego, el sistema evoluciona siguiendo la ecuación de Schrödinger dependiente del tiempo. Las amplitudes de todos los estados candidatos continúan cambiando, realizando un paralelismo cuántico, de acuerdo con la intensidad del campo transversal (dependiente del tiempo), el cual funciona como parámetro de control (equivalente a la temperatura en SA). Esto causa la tunelización cuántica entre estados como se observa en la figura 3-4.

Un hamiltoniano diseñado para el recocido cuántico puede escribirse como

$$\mathcal{H} = \mathcal{H}_F + \Gamma(t)\mathcal{H}_D \quad (3-8)$$

donde H_F es un hamiltoniano que codifica la función a optimizar, H_D es otro hamiltoniano que introduce un campo transversal externo y $\Gamma(t)$ es un coeficiente de campo transversal que se

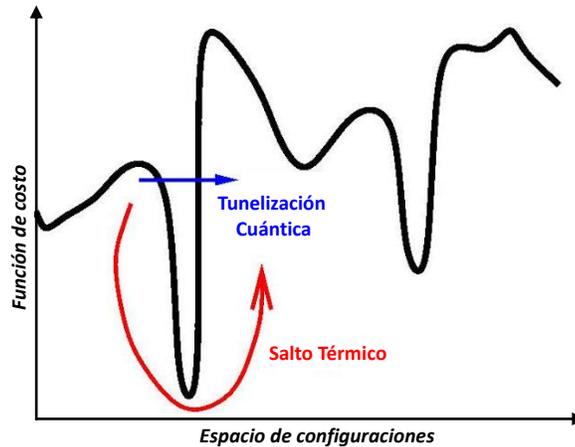


Figura 3-4: Diferencia entre SA y QA. Tomado de (Das y Chakrabarti, 2005).

utiliza para controlar la intensidad del campo externo.

Si el sistema evoluciona muy lentamente (es decir, adiabáticamente), eventualmente se asentará en un estado fundamental final que, con cierta probabilidad, corresponderá al valor óptimo de la función codificada en H_F . El campo transversal finalmente se apaga, y se espera que el sistema haya alcanzado el estado fundamental que corresponde a la solución del problema de optimización original (Sharma y Maharjan, 2018).

La simulación exacta de un proceso de recocido cuántico en una computadora digital es costosa. Los métodos cuánticos de Monte Carlo son procedimientos estocásticos que se utilizan para resolver la ecuación de Schrödinger. Consisten en utilizar los métodos clásicos de MCMC (apéndice C) para estimar los estados de baja energía de los hamiltonianos (Monte Carlo hamiltoniano). El recocido cuántico simulado (SQA) es un algoritmo clásico que utiliza métodos cuánticos de Monte Carlo para simular recocido cuántico hamiltonianos. Todavía se desconoce la complejidad computacional de SQA.

Ventajas: Conserva algunas de las características del SA, pues se trata de un métodos de búsqueda global, es aplicable a múltiples problemas, y posee flexibilidad. Puede resolver un mayor número de problemas, incluyendo la categoría NP-complejo. La convergencia es mucho más rápida que la caminata aleatoria debido al efecto de tunelización cuántica.

Desventajas: La simulación exacta es muy costosa y la implementación del algoritmo es muy difícil. Todavía se desconoce la complejidad computacional del método.

3.3. Enjambres de Partículas

La optimización por enjambre de partículas (PSO por sus siglas en inglés) es un algoritmo de búsqueda inspirado en el comportamiento colectivo de ciertos grupos de animales voladores capaces de comunicarse, en particular las abejas y algunas aves migratorias, en su búsqueda de alimento o mejores lugares (Poli, 2008).

Un algoritmo PSO trabaja con una población (llamada nube o enjambre) de soluciones candidatas (llamadas partículas). Dichas partículas se desplazan a lo largo del espacio de búsqueda conforme unas simples reglas matemáticas. El movimiento de cada partícula se ve afectado por su mejor posición local hallada hasta el momento, así como por las mejores posiciones globales encontradas por otras partículas a medida que recorren el espacio de búsqueda. Cuando se descubren nuevas y mejores posiciones, estas pasan a orientar los movimientos de las partículas. El proceso se repite con el objetivo, no garantizado, de hallar en algún momento una solución lo suficientemente satisfactoria (Tellez Cortes *et al.*, 2015).

Cada partícula se mueve de modo errático por el espacio, recordando en todo momento cual es la región donde ha encontrado un óptimo local. A su vez, el enjambre sabe colectivamente cual es la región del espacio, de entre todas las exploradas, donde se han encontrado el mínimo hasta ese momento. Cada partícula variará individualmente su movimiento con arreglo a estas dos direcciones, moviéndose hacia algún lugar intermedio. Es posible que una partícula durante ese movimiento encuentre una región con un mínimo menor que la conocida hasta entonces por ella (óptimo local), o incluso que la conocida por el enjambre (óptimo global); en este último caso, todo el enjambre orientará la búsqueda hacia esa nueva dirección (figura 3-5).



Figura 3-5: Representación esquemática de PSO. En líneas discontinua se representa la trayectoria, la flecha negra señala la mejor posición individual y la roja la mejor posición del grupo. Tomado de (Pérez López, 2005).

Cada partícula ajusta su vuelo actualizando su posición y velocidad de acuerdo a ecuaciones de actualización (Shi y Eberhart, 1998). Para la velocidad, la ecuación está definida por

$$\vec{V}_i = (w * \vec{V}_i) + c_p * \text{rand}_p * (\vec{P}_i - \vec{X}_i) + c_g * \text{rand}_g * (\vec{G} - \vec{X}_i) \quad (3-9)$$

donde w es conocido como peso de inercia, que controla el desplazamiento de las partículas. c_p y c_g son parámetros de control para darle mayor importancia a la búsqueda local o global del enjambre, respectivamente. \vec{P}_i es la mejor posición individual (óptimo local), y \vec{G} es la mejor posición de grupo (óptimo global). Los números rand_p y rand_g se eligen de forma aleatoria entre 0 y 1 para modificar los parámetros de control local o global, respectivamente.

La ecuación de actualización de posición para cada partícula está definida por:

$$\vec{X}_i = \vec{X}_i + \vec{V}_i * dt \quad (3-10)$$

donde dt es el radio de avance, entendido como el paso que utilizan las partícula para avanzar dentro del espacio de búsqueda. \vec{V}_i es el vector velocidad actual de la i -ésima partícula y \vec{X}_i el vector posición actual de la i -ésima partícula. Es necesario gestionar el radio de avance que experimenta cada partícula de forma tal que la nueva posición esté siempre dentro del espacio.

El algoritmo queda de la manera siguiente (Tellez Cortes *et al.*, 2015):

1. Inicialización: A cada partícula se le asigna una posición y velocidad inicial; y se eligen los parámetros de control c_p y c_g . También se guarda la mejor posición de todas ellas \vec{G} e individualmente sus posiciones actuales. En la práctica, para generar las posiciones iniciales se suelen generar puntos de Poisson del espacio de búsqueda.

Para cada instante de tiempo ($t = t + dt$) se realizan las siguientes operaciones sobre cada partícula:

2. Actualizar la posición y velocidad de las partículas de acuerdo a las ecuaciones (3-9)-(3-10).
3. Se compara la posición actual de cada partícula con el óptimo local conocido por ellas y con el óptimo global; actualizándolos si es menor.

El algoritmo termina cuando se alcanza un límite máximo de tiempo u otros criterios de parada especiales, como puede ser que todas las partículas se encuentren en una misma posición. Esto último puede suceder en un óptimo local cuando todas las partículas exploran muy cerca de él y se pierde el carácter aleatorio de su movimiento. A este fenómeno se le conoce como *convergencia del enjambre*. En la literatura pueden encontrarse análisis de este fenómeno así como de *divergencia* u *oscilación* del enjambre, estableciendo pautas de selección de los parámetros y nuevas variantes en la PSO.

A medida que el algoritmo de PSO avanza, si se actualizan los vectores dimensión por dimensión, el punto solución es más fácil de encontrar si se halla en un eje del espacio de búsqueda, en una diagonal o si está justo en el centro. Una primera forma de evitar este sesgo, es por ejemplo tomar como referencia problemas no sesgados, y luego rotarlos o desplazarlos. Otra opción es modificar el propio algoritmo para hacerlo menos sensible al sistema de coordenadas.

Ventajas: Facilidad de interpretación, comprensión e implementación del algoritmo. Permite el ajuste directo en la característica de la búsqueda, por medio de la asignación de valores de los parámetros de control c_p y c_g . Permite la visualización del proceso de búsqueda. Favorece la búsqueda global pues maneja un gran número de partículas comunicadas entre sí (Shi y Eberhart, 2001).

Desventajas: Es sensible a cambios en los valores de los parámetros, el mal manejo de estos valores puede ocasionar una convergencia prematura (Hu *et al.*, 2004). El sistema es también susceptible a la inicialización de las posiciones y velocidad, pudiendo existir posiciones donde se conduce irremediabilmente hacia óptimos locales. Existen sesgos relacionados con el sistema de coordenadas.

3.4. Colonias de Hormigas

El algoritmo de colonias de hormigas (ACO, por sus siglas en inglés) como su nombre lo indica se basa en el comportamiento de estos insectos. Las hormigas (inicialmente) vagan de

manera aleatoria y una vez encontrada comida regresan a su colonia dejando un rastro de feromonas (figura 3-6-1). Si otras hormigas encuentran dicho rastro, es probable que estas no sigan caminando aleatoriamente sino que sigan el rastro de feromonas, regresando y reforzándolo si estas encuentran comida finalmente (Dorigo, 1992).

Un detalle importante es que al paso del tiempo el rastro de feromonas comienza a evaporarse, reduciéndose así su fuerza de atracción. Cuanto más tiempo le tome a una hormiga viajar por el camino y regresar de vuelta otra vez, más tiempo tienen las feromonas para evaporarse. Un camino corto, en comparación, es marchado más frecuentemente, y por lo tanto la densidad de feromonas se hace más grande en caminos cortos que en los largos (figura 3-6-2). La evaporación de feromonas también tiene la ventaja de evitar convergencias a óptimos locales. Si no hubiese evaporación en absoluto, los caminos elegidos por la primera hormiga tenderían a ser excesivamente atractivos para las siguientes hormigas. En este caso, el espacio de búsqueda de soluciones se vería más limitado.

Por tanto, cuando una hormiga encuentra un buen camino entre la colonia y la fuente de comida, hay más posibilidades de que otras hormigas sigan este camino y con una *retroalimentación positiva* se conduce finalmente a todas las hormigas a un solo camino (figura 3-6-3). La idea del ACO es imitar este comportamiento con hormigas simuladas, conocidas como *agentes*, caminando a través de un grafo que representa el problema en cuestión (Pedemonte, 2007).

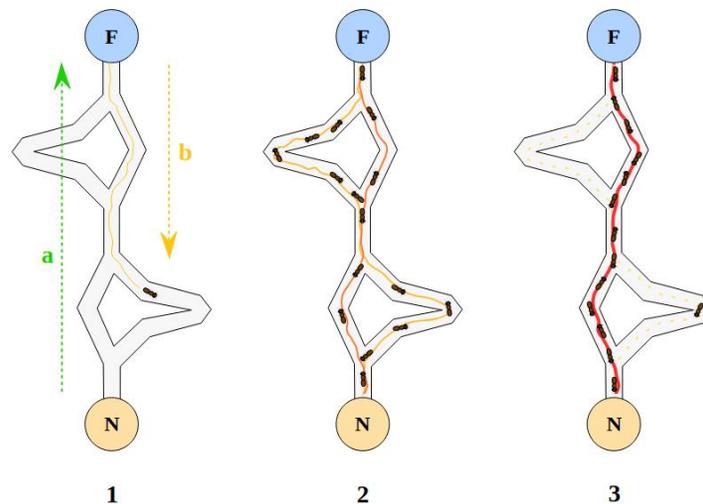


Figura 3-6: Ejemplo de funcionamiento del algoritmo ACO. N representa el nido y F lugar donde se encuentra la comida, a la trayectoria de ida y b de regreso. En línea naranja el rastro de feromonas, blancas las trayectorias posibles y en rojo se marca la trayectoria óptima. 1 representa el rastro inicial de feromonas, 2 la evaporación y 3 la trayectoria óptima. Figura de Johann Dréo, con licencia CC-BY-SA 3.0.

Formalmente, cada agente cumple con dos funciones principales: la construcción de un camino mediante la *incorporación de componentes* y dejar un rastro mediante la *actualización de feromonas* en una matriz de feromona, a modo de memoria; ambas funciones permite incorporar la experiencia adquirida por la colonia en el comportamiento individual (Arito, 2010).

En cada iteración del algoritmo, cada hormiga h se mueve entre estados construyendo itera-

tivamente una solución. La probabilidad P_{ij}^h de moverse de un estado i a un estado j depende de la combinación de dos valores:

$$P_{ij}^h = \frac{\tau_{ij} \cdot \eta_{ij}}{\sum_{k \in N_i^h} \tau_{ik} \cdot \eta_{ik}} \quad (3-11)$$

donde τ_{ij} es el nivel de rastro (matriz de feromona), indicando que tan competente ha sido en el pasado este particular movimiento y η_{ij} es la conveniencia del movimiento, calculado por alguna información heurística que indica a priori su factibilidad (típicamente se utiliza $\text{costo}(i, j)^{-1}$). Hemos denotado por N_i^h a la vecindad factible del nodo i , es decir, sólo aquellas ubicaciones admisibles.

A esta formulación se le suelen añadir parámetros para controlar la influencia de cada uno, de forma tal que si se elimina la influencia de τ_{ij} el algoritmo es controlado solo por la heurística a priori (algoritmo avaricioso) y si se elimina la influencia de η_{ij} se observa el fenómeno conocido como *estancamiento*, en el cual todas las hormigas siguen exactamente el mismo camino y construyen exactamente la misma solución.

Una vez que todas las hormigas han construido sus soluciones, se realiza la actualización de la matriz de feromona mediante la aplicación de la regla de *actualización en línea con retraso* (en inglés, *online delayed pheromone update rule*):

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_h \Delta_{ij}^h, \quad (3-12)$$

donde ρ es el coeficiente de evaporación de feromonas y Δ_{ij}^h es la cantidad de feromonas depositadas por la h -ésima hormiga. El objetivo de la regla es aumentar la cantidad de feromonas en los componentes de la solución que han sido utilizados por las soluciones de mayor calidad.

Variantes de la fórmula (3-12) definen diferentes ACO (EAS, AS-LBT, ANT-Q, entre otras), otros ACO se obtienen modificando la regla de transición de estados (ACS, MMAS y ANTS) (Pedemonte, 2007).

Los algoritmos ACO se consideran de construcción adaptativa, en el sentido de que una colonia de hormigas modifica las soluciones asignando un mayor rastro de feromonas a las conexiones, que están en las mejores soluciones. Durante la construcción de soluciones las hormigas seleccionan con mayor preferencia las que tienen mayor cantidad de feromonas.

Ventajas: Es especialmente útil en la solución de problemas combinatorios. La retroalimentación positiva permite llegar a buenas soluciones rápidamente en problemas NP-duros (Asif, 2019). Es fácilmente adaptable para sistemas distribuidos favoreciendo la paralelización. Se puede combinar con algoritmos de búsqueda locales puesto que generan buenas soluciones iniciales, así los algoritmos de búsqueda locales necesita un número mucho menor de iteraciones para alcanzar un óptimo local (ejecutando más búsquedas locales que con soluciones generadas aleatoriamente).

Desventajas: Posee una definición muy general que puede dificultar su análisis teórico. Los algoritmos básicos de ACO presentan un peor rendimiento con respecto a los tiempos de carga medios y requisitos de computación en comparación con otras metaheurísticas (Ambrosino *et al.*, 2010), por lo cual se le ha introducido diversos cambios y extensiones.

3.5. Algoritmos Genéticos

Los algoritmos genéticos (GA) están basados en los principios darwinistas de la evolución biológica. Son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Haciendo evolucionar una población de individuos sometida a acciones aleatorias semejantes a las que actúan en la evolución biológica, así como a una selección natural en función de la cual se decide cuáles son los individuos más adaptados, que sobreviven, y los menos aptos, que son descartados.

En una población, se codifica la información de cada solución (*individuo*) en una cadena, llamada *cromosoma*; los símbolos que forman la cadena son llamados *genes*. Los cromosomas evolucionan a través de iteraciones, llamadas *generaciones*. En cada generación un proceso análogo al de selección natural elegirá que cromosomas son aptos. Las siguientes generaciones (nuevas cromosomas), son generadas aplicando los operadores genéticos repetidamente: selección, cruzamiento, mutación y reemplazo (Whitley, 1994).

El proceso de codificación es independiente del algoritmo, en general se utiliza una codificación binaria y se introducen restricciones para que los nuevos cromosomas coincidan con soluciones factibles del problema.

La función objetivo se define sobre la representación genética y mide la calidad de la solución representada. En algunos problemas, es difícil o incluso imposible definir una expresión analítica. En estos casos, se puede utilizar una simulación para determinar el valor de la función objetivo de un cromosoma o incluso algoritmos genéticos interactivos.

Aunque existen variaciones, en términos generales, la estructura de un algoritmo genético es la siguiente (figura 3-7):

1. Crear una población inicial aleatoria formada por Np cromosomas diferentes.
2. Evaluar la aptitud (a través de la función objetivo) de cada cromosoma de la población.
3. Crear una nueva población con la anterior y repetir los siguientes pasos hasta que se hayan creado Np nuevos cromosomas:
 - a) Seleccionar parejas de cromosomas de la población existente, la probabilidad de selección es proporcional a la aptitud de los cromosomas.
 - b) Cruzar las parejas seleccionadas para generar un nuevo descendiente (cruzamiento).
 - c) Aplicar un proceso de mutación aleatorio sobre el nuevo cromosoma.
 - d) Añadir el nuevo cromosoma a la población sustituyendo cromosomas menos aptos (menor valor de la función objetivo).
4. Reemplazar la antigua población por la nueva y si no se cumple un criterio de parada volver al paso 2.

La forma en que se seleccionan los cromosomas que participan en cada cruce difiere en las distintas implementaciones de los algoritmos genéticos. Estas tienden a favorecer la selección de aquellos cromosomas mejor adaptados. Algunas de las estrategias más comunes (Gestal *et al.*, 2010) son:

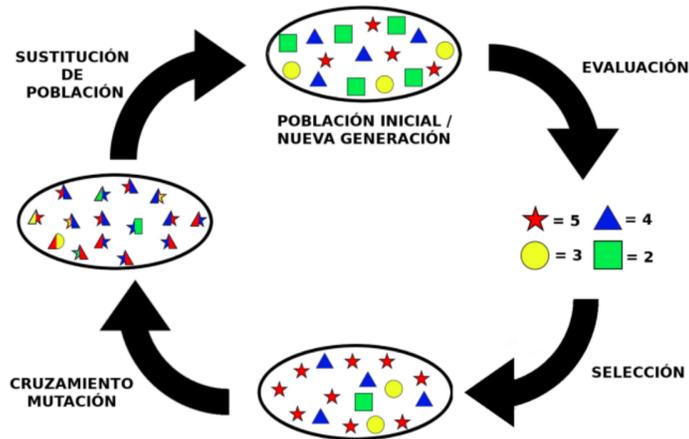


Figura 3-7: Ejemplo de funcionamiento del algoritmo GA. Tomado de (Campos Soberanis, 2019).

- Método de ruleta: la probabilidad de que un cromosoma sea seleccionado es proporcional a su valor relativo de la función objetivo, es decir, la función objetivo dividida por la suma de los valores de todos los cromosomas de la población. De esta forma si la aptitud de un cromosoma es el doble que el de otro, también lo será la probabilidad de que sea seleccionado. Este método presenta problemas si la aptitud de unos pocos cromosomas es superior en varios órdenes de magnitud al resto, ya que estos serán seleccionados de forma repetida y casi todos los cromosomas de la siguiente generación serán *descendientes* de los mismos *padres* (endogamia).
- Método Rank: la probabilidad de selección de un cromosoma es inversamente proporcional a la posición que ocupa tras ordenar todos los cromosomas de mayor a menor según su función objetivo. Este método es menos agresivo que el método de ruleta evitando la endogamia.
- Selección competitiva (o por torneo): se seleccionan aleatoriamente dos parejas de cromosomas de la población (usualmente todos con la misma probabilidad). De cada pareja se selecciona el mejor adaptado (*ganador*). Finalmente, se comparan los dos finalistas y se selecciona el de mayor aptitud. Este método tiende a generar una distribución de la probabilidad de selección más equilibrada que las dos anteriores.
- Selección truncada: se realizan selecciones aleatorias de cromosomas, habiendo descartado primero un grupo de cromosomas con menor aptitud de la población.

Las etapas donde se genera la nueva población resultan decisivas en el desarrollo del algoritmo. Los nuevos cromosomas se generan mediante los operadores de cruzamiento y mutación. El objetivo del cruzamiento es generar, a partir de cromosomas ya existentes (parentales), nuevos cromosomas (descendencia) que combinen las características de los anteriores. Para los casos de codificación discreta las estrategias más empleadas son:

- Cruzamiento a partir de un punto: se selecciona aleatoriamente una posición que actúa como punto de corte. Cada cromosoma parental se divide en dos partes y se intercambian

las partes anterior y posterior. Como resultado de este proceso, por cada cruce, se generan dos nuevos cromosomas.

- Cruzamiento a partir múltiples puntos: se seleccionan aleatoriamente varias posiciones que actúan como puntos de corte. Cada cromosoma parental se divide por los puntos de corte y se intercambian las partes. Como resultado de este proceso, por cada cruce, se generan dos nuevos cromosomas.
- Cruzamiento uniforme: el valor que toma cada posición del nuevo cromosoma se obtiene de uno de los dos parentales. Por lo general, la probabilidad de que el valor proceda de cada parental es la misma, aunque podría, por ejemplo, estar condicionada a la aptitud de cada uno. A diferencia de las anteriores estrategias, con esta, de cada cruce se genera un único descendiente.

Tras generar cada nuevo cromosoma de la descendencia, este se somete a un proceso de mutación en el que, cada una de sus posiciones, puede verse modificada con cierta probabilidad. Este paso es importante para añadir diversidad al proceso y evitar que el algoritmo caiga en mínimos locales por que todos los cromosomas sean demasiado parecidos de una generación a otra. Existen dos estrategias clásicas para controlar la magnitud del cambio que puede provocar una mutación:

- La primera consiste en perturbar el valor de la posición que se desea mutar sumándole un valor obtenido de una distribución de probabilidad (usualmente uniforme $U(-d, d)$ o normal $N(0, \sigma)$). Cuanto mayor la desviación estándar, mayores serán los cambios introducidos.
- La segunda se consigue reemplazándolo por un nuevo valor aleatorio dentro del rango permitido para dicha variable. Esta estrategia conlleva cambios más abruptos.

Ventajas: Es útil en la solución de problemas combinatorios. Realizan cambios aleatorios en sus soluciones candidatas y luego utilizan la función de aptitud para determinar si esos cambios producen una mejora, esto permite la exploración simultánea de soluciones. Son intrínsecamente paralelos, pues operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales. Resultan menos afectados por los mínimos locales que las técnicas tradicionales. Permite manipular muchas variables simultáneamente, esto es importante en caso de tener varios objetivos a resolver. Usan operadores probabilísticos, en vez de los típicos operadores deterministas de otros métodos (Eberhart y Shi, 1998).

Desventajas: Para problemas de alta complejidad la función de evaluación puede tornarse demasiado costosa en términos de tiempo y recursos. Puede haber casos en los cuales dependiendo de los parámetros podría no llegar a converger a una solución óptima o bien terminar en una convergencia prematura con resultados no satisfactorios. Es sensible a parámetros como el número de generaciones y el tamaño de la población. El diseño, la creación de la función de aptitud y la selección de los criterios de mutación entre otros, necesitan de conocimiento del problema para obtener buenos resultados. La necesidad de codificación puede considerarse como una desventaja para algunos problemas reales.

3.5.1. Algoritmo de Evolución Diferencial

El algoritmo de evolución diferencial (DE por sus siglas en inglés) es un método perteneciente a la computación evolutiva desarrollada en 1995 para la optimización en espacios continuos (Storn y Price, 1997). Lo que caracteriza a la DE es el uso de vectores de prueba que compiten con los individuos de la población actual a fin de sobrevivir. La población inicial se genera de forma aleatoria, los individuos se recombinan y mutan para producir una nueva generación. Para esto se seleccionan varios individuos como padres, uno de los padres es el padre principal, el cual se perturba con una combinación de los demás vectores padres. Si el vector de prueba resultante es mejor que el padre principal, entonces se reemplaza, de lo contrario, se mantiene al padre principal.

El algoritmo asume que las variables están codificadas como un vector de números reales cuya dimensión es igual al número de variables del problema y la población está compuesta de Np vectores. El dominio de las variables del problema está restringido entre valores mínimos y máximos (x_{min}, x_{max}). DE se compone básicamente de 4 pasos:

- Inicialización: La población se puede inicializar utilizando diferentes técnicas de muestreo aleatorio. Puede ser un muestreo aleatorio simple, el conocido como el hipercubo latino o se puede especificar una población inicial aumentando la velocidad de convergencia.
- Mutación: El vector de prueba se genera mediante

$$v_p = x_c + F \cdot (x_a - x_b) \quad (3-13)$$

donde x_c es el padre principal y $F \in (0, 2)$ es un parámetro de control conocido como *constante de mutación*. Para la selección de los padres $x_{a,b,c}$ existen diferentes estrategias (Quintero y Coello, 2004).

- Recombinación y selección: El vector anterior es aceptado o rechazado de forma aleatoria de acuerdo a una *probabilidad de cruce* especificada Cr . Luego se comparan los vectores de prueba con los originales, de manera que pasa a la generación siguiente aquel que tenga el mejor valor de función objetivo.

La inicialización se realiza al principio y los otros pasos se realizan de forma iterativa. Las condiciones de parada pueden estar relacionadas con número de generaciones, tiempo transcurrido, calidad de solución alcanzada, entre otras. En la figura 3-8 se muestra un esquema que resume el método.

Estrategias de mutación Existen varias estrategias diferentes que se pueden generalizar a través de la convención ($s/n/t$).

s representa la forma de seleccionar el vector a perturbar; puede ser el mejor vector de la generación anterior ($s = best$) o cualquier vector elegido aleatoriamente ($s = rand$), aunque se pueden establecer otro tipo de elección de s .

n es el número de vectores diferencia considerados para la perturbación del padre principal. Para la perturbación de un vector simple ($n = 1$), se eligen aleatoriamente 3 diferentes vectores, y la diferencia de dos de ellos se suma al tercero; para la perturbación de dos vectores diferentes ($n = 2$), se eligen 5 vectores distintos; se calcula la diferencia para cada par de cualesquiera de los cuatro primeros vectores y se hace una suma ponderada con el quinto.

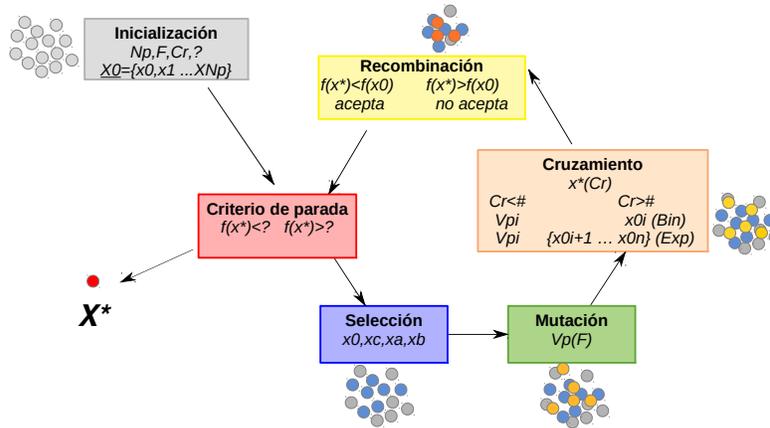


Figura 3-8: Esquema del método DE.

t se entiende como el tipo de cruzamiento a ser utilizada (*exp*: exponencial; *bin*: binomial). El cruzamiento en DE se realiza en para cada componente del vector dentro de un ciclo limitado por el valor de Cr . El número de componentes coincide con la dimensión d de la función objetivo. En el cruzamiento exponencial, la primera vez en la que un número aleatorio entre $(0, 1)$ supera el valor de Cr se suspende este ciclo y las componentes que queden por alterar quedan intactas. En el caso binomial, los cruzamientos se realiza en cada una de las d componentes siempre y cuando al elegir un número aleatorio entre $(0, 1)$ este sea menor que el valor de Cr . De lo anterior podemos deducir que para valores altos de Cr , los cruzamientos exponencial y binomial se comportan de forma similar.

Ventajas Una de las mayores ventajas que ofrece la perturbación basada en diferencia de vectores es que tanto el tamaño de un paso como su orientación se adaptan automáticamente a la forma de la función objetivo. A esto se le conoce como propiedad de coalescencia.

La figura 3-9 muestran un ejemplo de esta propiedad para la función *Peacks* (Price *et al.*, 2006). En la primera generación se calcula la posición inicial y los vectores diferencias. A medida que avanza el algoritmo la población se une alrededor de mínimos locales y aparecen nubes en los vectores de diferencia. Durante las generaciones iniciales (generaciones 2 – 16 en el ejemplo) la distribución de los vectores es multimodal y en el espacio de los vectores de diferencia contiene pasos adaptados para buscar dentro de cada cuenca y pasos capaces de transportar vectores de unas cuencas a las otras; hasta que poco a poco la población se agrupa alrededor del mínimos global y las diferencias favorecen más la búsqueda local. Una vez que la población se instala en la cuenca óptima (generaciones 17 – 33), la distribución del vector de diferencia se vuelve unimodal, los pasos exhiben una escala y una orientación apropiada para una búsqueda local haciéndose más pequeños. El proceso finaliza cuando se encuentra el mínimo y los vectores diferencia se anulan (generación 34).

DE codifica todos los parámetros como números de punto flotante, independientemente de su tipo. Incluso las variables enteras y discretas se codifican como valores reales para agregar diversidad a sus distribuciones de diferencias. El punto que se destaca aquí es que codificar parámetros continuos como números de punto flotante y manipularlos con operadores aritméticos

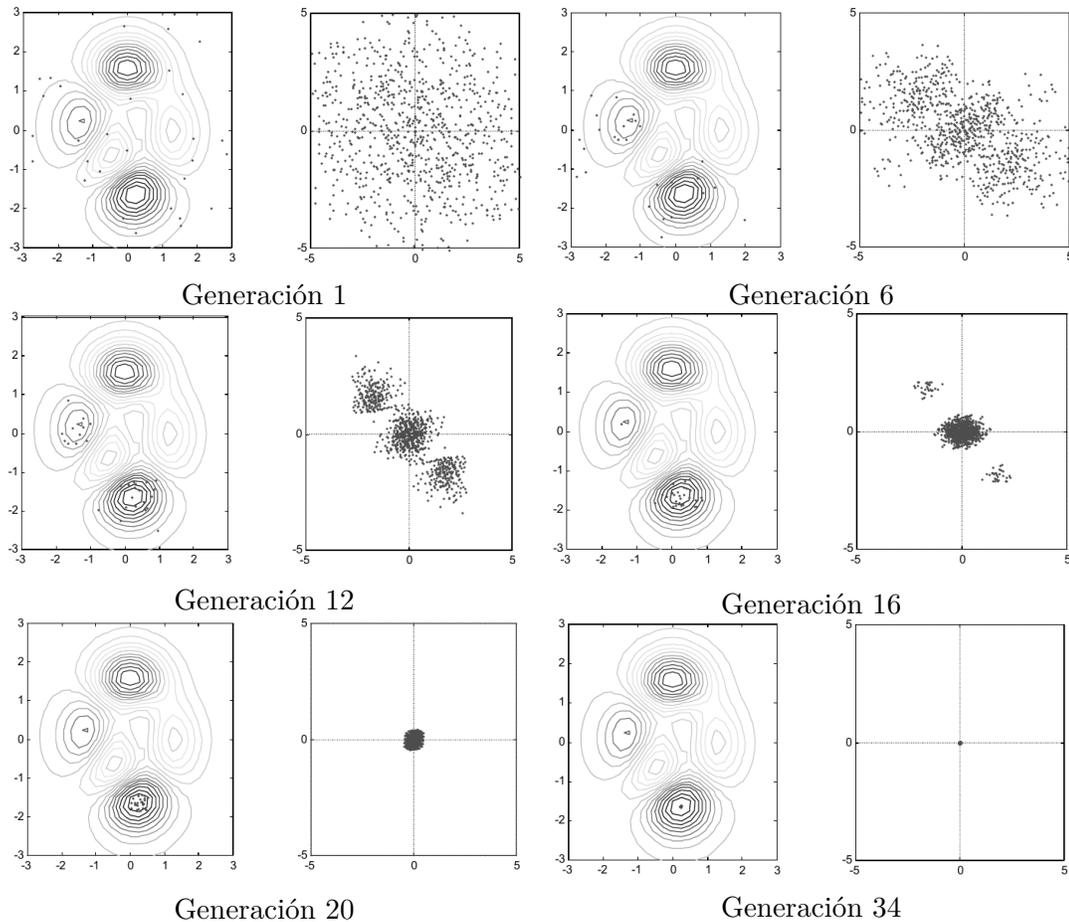


Figura 3-9: Ejemplo de la propiedad de coalecencia para DE con la función *Peacks*. A la izquierda se encuentra la población y a la derecha los vectores diferencia para diferentes generaciones. Tomado de (Price *et al.*, 2006).

ofrecen varias ventajas significativas sobre el enfoque tradicional de *cambio de bit* de otros algoritmos genéticos para la optimización continua de parámetros (Price *et al.*, 2006). Las ventajas incluyen:

- Facilidad de uso.
- Utilización eficiente de la memoria.
- Menor complejidad computacional: puede ser escalado a problemas grandes.
- Menor esfuerzo computacional: convergencia más rápida.
- Mayor libertad en el diseño de una distribución de mutaciones.

3.6. Comparación entre metaheurísticas

Después de detallar varios métodos metaheurísticos es necesario realizar una comparación entre ellos para saber cuales se pueden implementar en la cosimulación espacial (Guerrero *et al.*,

2016). En la tabla 3-1 se muestra un resumen de las ventajas y desventajas de las diferentes familias presentadas. De acuerdo a la naturaleza del problema quedan descartado: el recocido cuántico simulado (SQA) por su difícil implementación, el de colonia de hormigas (ACO) ya que se utiliza para problemas generalmente discretos y el de enjambre de partículas (PSO) que aunque se ha usado en problemas estocásticos similares requiere de complejas adaptaciones para su uso. Por tanto los métodos elegidos para esta investigación serán tomados de recocido simulado (SA) y algoritmos genéticos (GA). Específicamente se escogieron los métodos de recocido simulado generalizado (GSA) y evolución diferencial (DE).

Tabla 3-1: Comparación entre métodos metaheurísticos.

Métodos	Ventajas	Desventajas
SA	Alcanza una solución de buena calidad para la función objetivo. Aplicable a múltiples problemas. Fácil de implementar y polinómico. Ofrece una gran flexibilidad.	Balance entre la calidad de las soluciones y el tiempo de ejecución. El protocolo de enfriamiento no es fácil de diseñar (muchos parámetros a tener en cuenta).
SQA	Conserva algunas de las características del SA. Aplicable a problemas NP-complejos. Posee flexibilidad. La convergencia es mucho más rápida.	La simulación exacta es muy costosa. Difícil implementación. Se desconoce la complejidad computacional.
PSO	Facilidad de interpretación, comprensión e implementación. Permite el ajuste directo de la búsqueda. Fácil visualización. Favorece la búsqueda global.	Sensible a cambios en los valores de los parámetros. Muy dependiente de la posición y velocidad inicial. Existen sesgos relacionados con el sistema de coordenadas.
ACO	Útil en la solución de problemas combinatorios. Llega a buenas soluciones rápidamente. Adaptable a sistemas distribuidos y paralelización. Se puede combinar con algoritmos de búsqueda locales.	Los resultados del algoritmo básico son inferiores a otras metaheurísticas.
GA	Útil en la solución de problemas combinatorios. Permite la exploración simultánea de soluciones. Son intrínsecamente paralelos. Menos afectados por los mínimos locales. Manipula muchos parámetros simultáneamente. Usan operadores probabilísticos.	La función de evaluación puede ser demasiado costosa en términos de tiempo y recursos. Dependiendo de los parámetros podría no llegar al óptimo global. Es sensible al número de generaciones y tamaño de la población. Requiere conocimiento del problema. Necesita de codificación.

Capítulo 4

Optimización multiobjetivo

Para simular propiedades petrofísicas a partir de atributos sísmicos elásticos se propuso en el capítulo 2 una metodología basada en la utilización de cópulas y la optimización. Dentro de ella se identificaron dos etapas fundamentales: la primera permitía obtener, para un caso de estudio, un variograma y una función de distribución conjunta que teóricamente debe respetar la simulación resultante, la cual sería obtenida en la segunda etapa utilizando un método de optimización global. Se hace necesario entonces definir el problema de optimización que, utilizando esta información, permita obtener dichas simulaciones. Una vez planteado este problema es posible resolverlo mediante los métodos de optimización escogidos en el capítulo 3, por lo que otro aspecto fundamental que se tratará en este capítulo es la implementación computacional de la metodología.

4.1. Problema de optimización

Un problema de optimización está compuesto por un espacio de soluciones, una función objetivo y un método de optimización que será utilizado para su solución.

Nos proponemos simular una variable primaria en una malla conociendo explícitamente su dependencia con otra variable que llamaremos secundaria. Por otra parte la dimensión del problema de optimización coincide con el número de observaciones y nodos de la malla, ya que el valor de la variable a simular en cada nodo será considerado como una *variable del problema*. El espacio de soluciones estará limitado por las características del problema.

En particular se tomará como variable primaria a la porosidad total (Phit) y como variable secundaria a la impedancia acústica (Ip). La elección se basa en la dependencia entre ellas, la cual ha sido demostrada por Le *et al.* (2020) y Ramírez (2018). En el presente trabajo no se realizaron análisis de dependencia adicionales entre otras propiedades petrofísicas y elásticas. La impedancia acústica se estimó a partir de los registros de pozo de velocidad de la onda longitudinal (Vp) y la densidad de la roca (Rhob) mediante la ecuación siguiente:

$$Ip = Vp * Rhob. \tag{4-1}$$

El rango de valores posibles para la porosidad total (variable del problema) estará limitado por las características petrofísicas del medio, por lo que se encuentra entre 0 y 0.4.

Seguiremos un enfoque de optimización para una función multiobjetivo (FMO), que denotaremos por \mathcal{O} , la cual es una suma ponderada de las siguientes funciones:

$$\mathcal{O}_1 = \sum_h \frac{[\gamma_x^*(h) - \gamma_x(h)]^2}{\gamma_x(h)^2}, \quad (4-2)$$

$$\mathcal{O}_2 = \sum_{x \in \eta_p} \sum_{y \in \eta_s} [F^*(x|y) - F(x|y)]^2, \quad (4-3)$$

donde $\gamma_x(h)$ son los semivariogramas de la variable primaria; $F(x|y)$ las distribuciones condicionales de la variable primaria respecto a la secundaria; η_s y η_p son las clases secundaria y primaria respectivamente; y el asterisco denota que se trata de la realización simulada.

Lo anterior quiere decir que

$$\mathcal{O} = \omega_1 \mathcal{O}_1 + \omega_2 \mathcal{O}_2, \quad (4-4)$$

donde ω_1 y ω_2 representan los pesos o importancia que se le dará a cada función. Esta función tiene una naturaleza estocástica que hace muy difícil estudiarla de forma analítica.

Los métodos de optimización escogidos fueron recocido simulado generalizado (GSA) y evolución diferencial (DE), de acuerdo al análisis realizado en el capítulo 3.

4.2. Implementación computacional

Las implementaciones anteriores (Clayton V. Deutsch, 1997) realizaban por separado la simulación mediante cópulas y la simulación espacial. Esta última utiliza como datos de entrada realizaciones de la cosimulación y para la optimización utiliza un algoritmo de optimización global (recocido simulado SA), usando una modificación de la biblioteca GSLIB. Este algoritmo es discreto, ya que el paso de SA asociado a la perturbación se realiza en dos etapas: una primera donde se intercambian valores conocidos, y cuando no es suficiente se pasa a una segunda donde se genera un nuevo valor escogiéndolo de forma aleatoria de los valores conocidos. Es decir, en cada paso existe un número limitado de posibles nuevos valores de la variable. En la segunda etapa Deutsch y Cockerham (1994) añade una restricción adicional sobre el grupo de valores donde se pueden hacer los intercambios y sobre los valores nuevos de la variable primaria, condicionando a la variable secundaria mediante una tabla. Esta restricción, si bien se considera que podría ser favorable, no fue tenida en cuenta en los métodos implementados para este trabajo, no obstante el condicionamiento de la variable primaria respecto a la secundaria se obtiene de la segunda función objetivo.

La implementación de esta metodología se realizó usando el lenguaje de programación **Python 3.6.9**, el cual es de código abierto y multiplataforma. Esto entre otras ventajas permite la reutilización de códigos en programas como *Petrel*[®] (Schlumberger, 2010) y *OpendTect* (Huck, 2012). A diferencia de trabajos precedentes se implementaron los dos algoritmos: GSA y DE en dicho lenguaje definidos en la biblioteca de optimización de SciPy (Virtanen *et al.*, 2020): `scipy.optimize.dual_annealing` y `scipy.optimize.differential_evolution` respectivamente, modificándose el primero para que fuera compatible con el recocido simulado clásico. Estos algoritmos implementados son continuos, en el sentido de que en principio no existe un número finito de valores posibles para la variable, y además, en cada paso la cantidad de valores estará solo restringida por la aritmética de la máquina.

De manera adicional se consideró una simulación condicional utilizando puntos de condicionamiento, que será utilizada en la sección 5.2.1.5.

Para la primera etapa de la metodología:

1. Se calcularon los estadígrafos univariados y bivariados con la librería NumPy (Oliphant, 2006).
2. Se adaptaron los módulos de la biblioteca `geostatmodels` (Johnson, 2019), que nos permite el cálculo de variogramas experimentales $1D$ y $2D$, y además combinar modelos teóricos clásicos. Estas extensiones podrían ser de interés, además, en etapas posteriores de esta investigación. Pues nos permitiría aplicar esta metodología en más dimensiones, de manera similar a los trabajos de Maldonado (2014) a escala sísmica.
3. Se ajustaron las distribuciones marginales y las cópulas utilizando el *criterio de información bayesiano* y los módulos de la biblioteca `OpenTURNS` (Baudin *et al.*, 2015).

En la segunda etapa, para la FMO se tomaron pesos iguales ($\omega_1 = \omega_2 = 1$), asignando igual importancia a ambas funciones. Los detalles de la implementación se explicarán a continuación.

Función Objetivo \mathcal{O}_1 : para el cálculo del variograma experimental de la variable primaria en cada paso del proceso de optimización es necesario conocer las coordenadas de los puntos de interés y el variograma teórico. Se adaptaron los módulos de la biblioteca `geostatmodels` (Johnson, 2019) para el cálculo del variograma experimental y el modelo teórico.

Función Objetivo \mathcal{O}_2 : para el cálculo de la función de distribución condicional es necesario conocer los valores de la variable secundaria en cada punto y la distribución conjunta de ambas variables. En cada paso del proceso de optimización se realiza el cálculo de las distribuciones condicionales teórica y simulada. Esto se realiza condicionando la variable simulada en dicho paso a partir de los valores conocidos de la variable secundaria. En el primer caso se utiliza la distribución conjunta teórica y para el segundo caso se obtiene una distribución conjunta de los pares primario simulado y secundario conocido mediante un ajuste de sus distribuciones marginales y una cópula paramétrica asociada.

La distribución condicional teórica se tomará a partir de una estimación de la distribución bivariada de los datos y para el cálculo de la distribución condicional simulada se utilizarán varias cópulas paramétricas, escogiendo la que mejor se ajuste. Si bien se pierde así la posibilidad de un ajuste más exacto de las distribuciones empíricas, se logra disminuir el costo computacional.

Realización de valores iniciales: para acelerar la convergencia de los métodos de optimización se decidió generar valores iniciales de manera condicional a partir de la distribución bivariada de los datos según la ecuación (2-27). Esta distribución bivariada se estimó tanto con una cópula paramétrica de Frank (véase la sección 5.2.1.2) como utilizando KDE (sección 5.2.1.3).

El manejo de las distribuciones marginales, conjunta y condicional de las variables involucradas se realizó utilizando la biblioteca `OpenTURNS` (Baudin *et al.*, 2015).

Los dos métodos de optimización escogidos se validarán en el capítulo 5. A continuación se describen los parámetros esenciales para la implementación de los algoritmos de optimización. Los valores que se mencionarán fueron utilizados para los casos de estudio que se verán en los capítulos siguientes. Aunque estos parámetros se escogieron de la literatura no se puede afirmar que la elección realizada sea óptima para problema.

Los parámetros a tener en cuenta para GSA son:

1. Temperatura inicial T_0 : se estimó utilizando el método sugerido en Dréo *et al.* (2006); Maldonado (2014), para esto se fijó un valor inicial x_0 y se evaluó la FMO en 100 perturbaciones aleatorias de este, se calculó la media de la variación de la FMO ($m_{\Delta\mathcal{O}} = \frac{1}{100} \sum[\mathcal{O}(x_0) - \mathcal{O}_{pert}]$) y se eligió una tasa inicial de aceptación (τ_0) para obtener

$$T_0 = \left(-\frac{m_{\Delta\mathcal{O}}}{\log(\tau_0)} \right).$$

2. Razón de reinicio r : a medida que baja la temperatura es necesario establecer mecanismos de control, la técnica de reinicio eleva la temperatura cuando

$$\frac{T}{T_0} < r$$

y continúa la búsqueda, evitando que el algoritmo quede atascado en un mínimo local (Ingber, 2000). El valor recomendado por Virtanen *et al.* (2020) es de $r = 2e - 5$.

3. Coeficientes de aceptación q_a y de visita q_v : se utilizaron los valores de $q_a = -5$ y $q_v = 2.7$; los cuales son óptimos para GSA como se demuestra en Tsallis y Stariolo (1996).

Los parámetros fundamentales para DE son:

1. Tamaño de la población Np : el tamaño mínimo de cualquier población es de $Np = 5$ para poder utilizar todas las estrategias posibles, sin embargo Price *et al.* (2006) utiliza valores de hasta $Np = 25$ para optimizar funciones de prueba con dimensión $d \leq 100$ y hasta $Np = 40$ para problemas estocásticos. Como nuestro problema es estocástico y su dimensión será $d > 100$ se escogió $Np = 35$.
2. Estrategia: como estrategia se escogió *best1bin*, descrita en la subsección 3.5.1. Al tomar $s = best$, de toda la población se escoge como padre principal aquel de menor función objetivo (4-4), es decir, que mejor respete la distribución condicional y el variograma. De esta forma se garantiza que ese padre es una realización de las simulaciones con cópula condicional y además respeta la dependencia espacial; el vector de prueba en la ecuación (3-13) es una perturbación de dicha realización.
3. Probabilidad de cruce Cr : este parámetro determina la probabilidad (aproximada) de que los valores de un vector se hereden a la siguiente generación. Los resultados de Storn y Price (1997) mostraron que todas las funciones de prueba podían resolverse con $0 \leq Cr \leq 0.2$ (evitando saltos bruscos) o $0.9 \leq Cr \leq 1$ (favoreciendo la exploración), concluyendo que las funciones optimizables con Cr pequeño eran separables y las otras altamente

dependientes. Luego de probar con valores altos y bajos se encontró que nuestro problema se satisfacía con $Cr = 0.2$.

4. Constante de mutación F : en Price *et al.* (2006) se recomienda tomar en general $F < 1$ ya que para valores mayores lo considera menos confiable y consume mayor tiempo. Para evitar la convergencia prematura, F tiene que tener suficiente magnitud para contrarrestar los efectos del cruzamiento, pudiéndose establecer un valor crítico teórico de

$$F_{\text{crit}} = \sqrt{\frac{(1 - Cr/2)}{Np}}$$

por encima del cual aumenta la varianza de la población, sin embargo Zaharie (2002) encuentra que el valor crítico empírico es mucho mayor y recomienda utilizar $F > 0.3$. En nuestro caso $F_{\text{crit}} = 0.16$ y se escogió $F = 0.5$.

Interrupción: debido a que se conoce que la FMO alcanza su óptimo con valor cero, se decidió añadir a ambos métodos un parámetro de tolerancia para disminuir el tiempo de ejecución. De este modo si para alguna realización x , $\mathcal{O}(x) < tol = 1e - 3$ el algoritmo se detiene y x es considerado óptimo global.

4.3. Recursos computacionales

Todos los cálculos presentados en este trabajo se hicieron con una laptop que cuenta con las siguientes características

- SO: Linux 5.4.0 (Kubuntu 18.04) de 64bits
- CPU: 4× Intel® Core™ i5-254M (2.6GHz)
- RAM: 12GB
- GPU: Intel HD Graphics 3000 (integrada)

En el apéndice D se encuentra detallado partes sustanciales del código básico de *Python* con los puntos fundamentales de la implementación.

Capítulo 5

Validación de los métodos de optimización

El objetivo de este capítulo es establecer los criterios bajo los cuales la metodología desarrollada cumple con el objetivo para el que fue diseñada. En primera instancia necesitamos evaluar los métodos de optimización escogidos en situaciones diversas y difíciles, ya que son claves en nuestro estudio. El otro requerimiento es que sean compatibles con la metodología de cosimulación, para ello necesitamos verificar que su comportamiento ante problemas reales sea el esperado, reproduciendo la estructura espacial y la dependencia entre las variables de las soluciones conocidas.

Primeramente validaremos los métodos de optimización y su implementación para funciones arbitrarias utilizando funciones de prueba. También validaremos la implementación realizada de toda la metodología utilizando la porosidad total e impedancia acústica como variable primaria y secundaria respectivamente en varios casos de estudio.

Para estudiar la validez de un algoritmo es necesario medir diferentes características que sean independientes de los datos de entrada. En particular el término eficiencia computacional o desempeño es usado para describir aquellas propiedades de los algoritmos que están relacionadas con la cantidad de recursos utilizados por los mismos. Las dos medidas más comunes suelen ser cuanta memoria operativa requiere y el tiempo de ejecución necesarios para cumplir su objetivo (Hernández Torres *et al.*, 2014; Nesmachnow y Salsano, 2007). En nuestro estudio utilizaremos solo esta última para medir la eficiencia.

No todos los algoritmos de optimización se comportan de igual manera para diferentes funciones objetivos, y aún los que teóricamente tienen garantías de alcanzar el óptimo global requieren un número de iteraciones que puede llegar a ser demasiado grande (Wetter y Wright, 2004; Greenhalgh y Marshall, 2000). La capacidad de encontrar el óptimo global en un número de pasos o tiempo acotado se conoce como efectividad (Si *et al.*, 2019; Akhmedova, 2012; Shi *et al.*, 2016). Es por esto que todos los algoritmos incorporan en su implementación criterios de parada como un número máximo de iteraciones, tiempo máximo o número máximo de evaluaciones de la función objetivo, u otros relacionados con la optimalidad. Utilizaremos la calidad del óptimo relativo al algoritmo como medida de su efectividad, en el sentido de que cuando falla en encontrar el óptimo global se considerará como mejor a aquel que tenga menor valor de la función objetivo (Hernández Torres *et al.*, 2014).

Para los dos casos de validación mencionados se utilizarán la eficiencia computacional (tiem-

po de ejecución) y la efectividad (calidad del óptimo relativo) para comparar los métodos de optimización seleccionados.

5.1. Validación con funciones de prueba

La evaluación y validación de algoritmos de optimización se realiza con frecuencia mediante el uso de un conjunto elegido de funciones de prueba descritos en la literatura. Estas funciones de prueba están diseñadas para medir diferentes características de estos algoritmos, tales como: tasa de convergencia, precisión, robustez y rendimiento general. Utilizaremos un conjunto de funciones de prueba que son completamente artificiales y simples, es decir, se expresan en una forma analítica cerrada y no tienen antecedentes directos de ninguna aplicación práctica. En cambio, permiten un análisis detallado de ciertas características especiales de la topología: modalidad, continuidad, separabilidad y otros. Una característica común a todas las funciones escogidas para nuestra validación es que presentan un único mínimo global (Jamil y Yang, 2013).

Características de las funciones de prueba:

- **Modalidad:** Una función con más de un óptimo local se llama multimodal. El número de picos corresponde a la modalidad de función (para un óptimo local se dice unimodal, dos bimodal, etc.). Si los algoritmos encuentran estos picos durante un proceso de búsqueda, existe una tendencia a que el algoritmo pueda quedar atrapado en uno de dichos picos. Esto tendrá un impacto negativo en el proceso de búsqueda, ya que puede desviar la búsqueda de las soluciones óptimas globales.
- **Cuencas:** una disminución relativamente pronunciada que rodea un área grande se llama cuenca. Los algoritmos de optimización pueden ser fácilmente atraídos por tales regiones. Una vez en estas regiones, el proceso de búsqueda del algoritmo se ve severamente obstaculizado. Esto se debe a la falta de información para dirigir el proceso de búsqueda hacia el mínimo. Un problema puede tener múltiples cuencas.
- **Valles:** Un valle ocurre cuando un área estrecha de poco cambio está rodeada por regiones de fuerte descenso. Al igual que con las cuencas, los algoritmos se sienten inicialmente atraídos por esta región. El avance del proceso de búsqueda de un algoritmo puede ralentizarse considerablemente en el fondo del valle.
- **Separabilidad:** La separabilidad es una medida de la independencia entre las variables que intervienen en las funciones a optimizar. Si todas las variables son independientes, se puede realizar una secuencia de procesos de optimización también independientes. Como resultado, cada variable se puede optimizar por separado para encontrar el óptimo de la función. En general, las funciones separables son relativamente fáciles de resolver, en contraposición con las no separables.
- **Dimensionalidad:** La dificultad de un problema generalmente aumenta con su dimensión. Para problemas altamente no lineales, esta propiedad puede ser una barrera importante para casi todos los algoritmos de optimización.

En particular para nuestra validación solo se tomó dimensión 2, sin embargo, para los casos de estudio pudiera resultar de gran importancia dado el gran número de parámetros que se manejan.

- Escala: Otra dificultad que los algoritmos pueden encontrar es que existan diferencias de magnitud de varios órdenes entre el dominio y la función.

A continuación listamos las funciones de prueba utilizadas para esta validación y algunas de sus propiedades.

Función Ackley: fue propuesta por David Ackley en su tesis doctoral de 1987 (Ackley, 1987). Es una función de prueba continua, diferenciable, no separable, escalable y multimodal. Es obtenida modulando una función exponencial con una onda cosenoidal de amplitud moderada (figura 5-1). Su topología se caracteriza por una región externa casi plana (debido a la exponencial dominante) y un agujero central o pico donde las modulaciones por la onda cosenoidal se vuelven cada vez más influyentes.

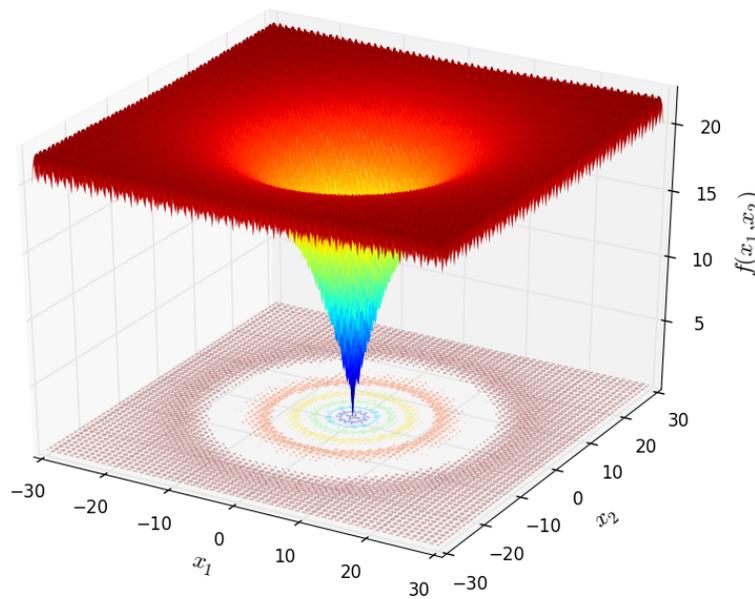


Figura 5-1: Función Ackley. Tomado de (Gavana, 2016).

Como señala Ackley, esta función causa complicaciones moderadas a la búsqueda, ya que si bien un algoritmo de optimización estrictamente local que realice escaladas seguramente quedará atrapado en un óptimo local, una estrategia de búsqueda que explore un vecindario un poco más grande podría cruzar valles intermedios hacia óptimos cada vez mejores (Bäck, 1996).

En un dominio bidimensional se define por:

$$f(x, y) = -20 \exp[-0.2\sqrt{0.5(x^2 + y^2)}] - \exp[0.5(\cos 2\pi x + \cos 2\pi y)] + e + 20 \quad (5-1)$$

donde $-5 \leq x, y \leq 5$

Su punto óptimo global es: $f(0,0) = 0$.

Función Rastrigin: es un ejemplo típico de función continua, diferenciable y multimodal no lineal. Fue propuesto por primera vez por Rastrigin (Rastrigin, 1974) como una función bidimensional (figura 5-2) y ha sido generalizado en Rudolph (1990). Encontrar el mínimo de esta función es un problema bastante difícil debido a su gran espacio de búsqueda y su gran número de mínimos locales.

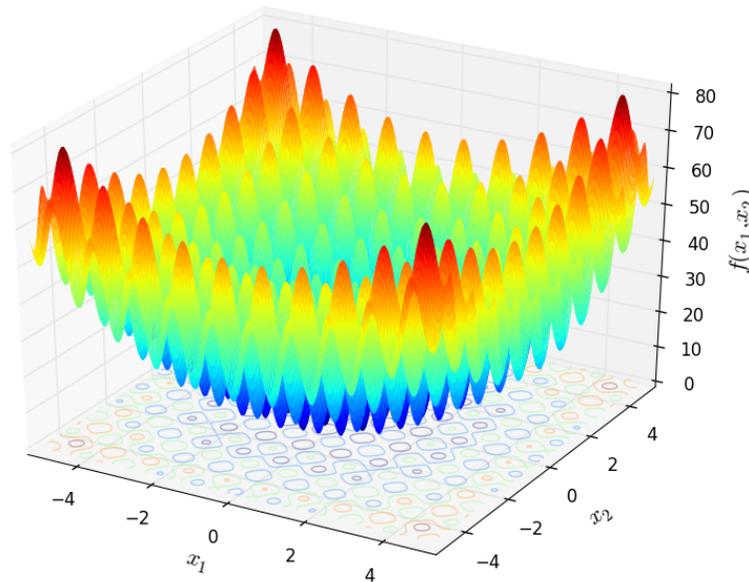


Figura 5-2: Función Rastrigin. Tomado de (Gavana, 2016).

En un dominio n-dimensional se define por:

$$f(\bar{x}) = An + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)] \quad (5-2)$$

dónde $A = 10$ y $x_i \in [-5.12, 5.12]$.

Su punto óptimo global es: $f(\bar{0}) = 0$

Función Rosenbrock: fue introducida por Howard H. Rosenbrock en 1960 (Rosenbrock, 1960). También se conoce como el valle de Rosenbrock o la función de *plátano de Rosenbrock*. El mínimo global está dentro de un valle plano largo, estrecho y de forma parabólica (figura 5-3). Encontrar el valle es trivial. Sin embargo, converger al mínimo global es difícil. Esta función es continua, diferenciable, no separable, escalable y unimodal, forma parte del conjunto de funciones de pruebas de Schwefel y De Jong (Bäck, 1996).

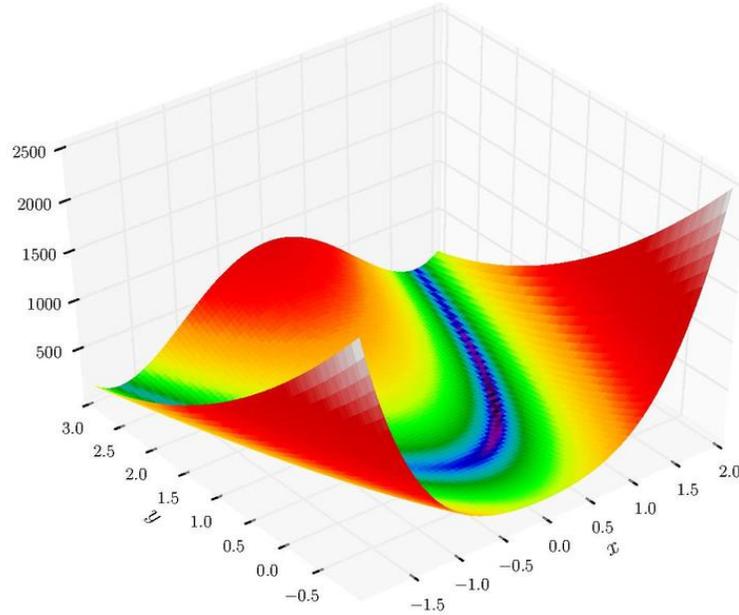


Figura 5-3: Función Rosenbrock. Figura de Morn the Gorn, con licencia de dominio público.

La función está definida por:

$$f(x, y) = (a - x)^2 + b(y - x^2)^2 \quad (5-3)$$

donde $-\infty \leq x, y \leq \infty$. Por lo general, estos parámetros se configuran de manera que $a = 1$ y $b = 100$.

Su punto óptimo global es: $f(a, a^2) = 0$.

5.1.1. Análisis comparativo

La validación se realizó con los algoritmos elegidos en el capítulo 3. Se evaluó la tasa de convergencia y el rendimiento general (tiempo de ejecución). El error debe disminuir en cada iteración para asegurar una correcta optimización de los métodos. Si el error es constante durante varias iteraciones puede que el algoritmo este estancado en mínimos locales. Para establecer la comparación se fijaron los siguientes parámetros para cada método (todos los parámetros que se presentan a continuación son adimensionales).

Parámetros para el algoritmo de evolución diferencial (DE)

- Máximo número de generación = 300
- Tamaño de la población (número de individuos) = 30
- Constante de mutación = 1
- Probabilidad de cruce = 0.5

Parámetros para el algoritmo de recocido simulado generalizado (GSA)

- Máximo número de iteraciones = 2000
- Temperatura inicial
 - para las funciones Ackley y Rastrigin = 25
 - para la función Rosenbrok = 1000
- Razón de reinicio de la temperatura = $2e - 07$
- Coeficiente de visita = 2.7
- Coeficiente de aceptación = -5

Los parámetros de DE se escogieron empíricamente de manera tal que se asegurara la optimización para las funciones de prueba escogidas. En GSA los tres últimos parámetros se escogieron de la literatura (Tsallis y Stariolo, 1996; Xiang *et al.*, 2013), por el contrario el número de iteraciones y la temperatura inicial se escogieron empíricamente. Para la función Rosenbrock se observó que aumentando la temperatura aumentaba la precisión, sin embargo no se encontró una temperatura inicial que satisfaga la precisión deseada.

En los gráficos siguientes se muestra el desempeño de los algoritmos teniendo en cuenta el error, entendido como el cuadrado de la distancia entre el óptimo encontrado y el óptimo real, el número de iteraciones y el tiempo de ejecución. A la izquierda se encuentran los del algoritmo de evolución diferencial y a la derecha los de recocido simulado generalizado. Ambos lograron resolver problemas bidimensionales con un error de precisión de $1e - 16$, se observa como disminuye el error al aumentar las generaciones o iteraciones, según sea el caso.

En las tablas 5-1-5-3 se muestra la estadística descriptiva de el error, el valor de la función objetivo al evaluar en el óptimo encontrado, el número de iteraciones y tiempo necesarios para encontrarlo. En la izquierda con el algoritmo de evolución diferencial y en la derecha con recocido simulado generalizado, para estos cálculos se realizaron 50 corridas con los mismos parámetros para estimar la consistencia del método respecto a la generación y estado inicial. Estos resultados pudieran variar al aumentar las dimensiones de estos problemas de acuerdo con los resultados obtenidos en Xiang *et al.* (2017).

Para la función Ackley ambos algoritmos se detuvieron porque el valor de la función era menor que la precisión fijada (figura 5-4). Para ambos métodos se obtuvo el óptimo global exacto (resultando en un error de 0.0) en la mayoría de los casos salvo en seis ocasiones donde GSA se obtuvo con errores menores del orden de $1e - 29$. Si bien debido a la diferencia entre los dos métodos GSA realiza más iteraciones es 1.8 veces más rápido que DE (ver tabla 5-1).

Tabla 5-1: Estadística descriptiva para la función Ackley.

Parámetros	Método	DE				GSA			
		Error	Valor F.	Iter.	Tiempo(s)	Error	Valor F.	Iter.	Tiempo(s)
Mínimo		0.0	0.0	158	1.152	0.0	0.0	1259.0	0.492
Máximo		0.0	0.0	184	1.550	1.26e-29	1.07e-14	2000.0	0.844
Media		0.0	0.0	168.66	1.298	6.31e-31	6.39e-16	1777.12	0.701
Mediana		0.0	0.0	168.5	1.255	0.0	0.0	1844.0	0.717
Rango intercuantil		0.0	0.0	5.75	0.104	0.0	0.0	192.25	0.092
Desviación estándar		0.0	0.0	4.998	0.079	2.09e-30	1.97e-15	201.771	0.080

Para la función Rastrigin ambos algoritmos se detuvieron porque el valor de la función era menor que la precisión fijada (figura 5-5). Con ninguno de los métodos se obtuvo el óptimo

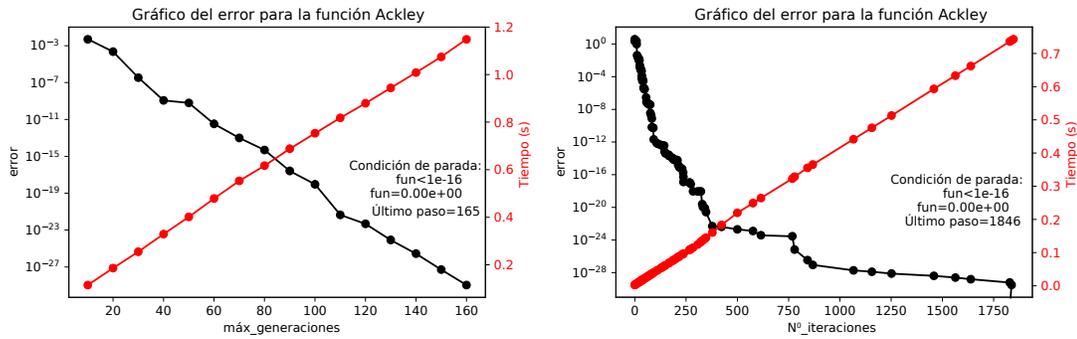


Figura 5-4: Gráficos de error y tiempo de ejecución para la función Ackley.

global exacto, aunque DE lo obtuvo con errores menores del orden de $1e - 17$. Para el caso de GSA el óptimo global se obtuvo con errores de hasta 0.99, si bien en el 75% de las corridas lo obtuvo con errores menores del orden de $1e - 17$. DE necesitó menos generaciones en esta ocasión que para la función Ackley, lo cual repercutió positivamente también en el tiempo de ejecución. Por otra parte GSA fue 3.7 veces más rápido que DE (ver tabla 5-2).

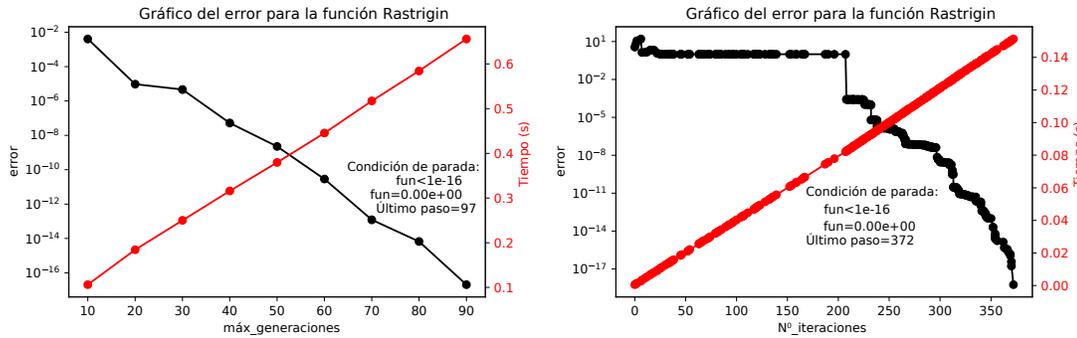


Figura 5-5: Gráficos de error y tiempo de ejecución para la función Rastrigin.

Tabla 5-2: Estadística descriptiva para la función Rastrigin.

Método	DE				GSA			
	Error	Valor F.	Iter.	Tiempo(s)	Error	Valor F.	Iter.	Tiempo(s)
Mínimo	4.26e-20	0.0	83.0	0.644	4.47e-19	0.0	175.0	0.070
Máximo	1.38e-17	0.0	101.0	0.958	0.990	0.995	2000.0	0.830
Media	6.73e-18	0.0	93.66	0.777	5.94e-2	0.060	497.72	0.208
Mediana	6.93e-18	0.0	94.5	0.745	1.02e-17	0.0	264.0	0.116
Rango intercuantil	7.21e-18	0.0	6.0	0.098	7.10e-18	0.0	198.0	0.079
Desviación estándar	4.34e-18	0.0	4.255	0.061	0.235	0.236	509.757	0.209

Para la función Rosenbrock DE se detuvo porque el valor de la función era menor que la precisión fijada salvo en un caso (figura 5-6), GSA realizó las 2000 iteraciones en todas las ocasiones sin obtener una buena precisión. Con ninguno de los métodos se obtuvo el óptimo global exacto, aunque en el 75% de las corridas DE lo obtuvo con errores menores del orden $1e - 16$. Para el caso de GSA el óptimo global se obtuvo con errores mayores que el orden de

$1e-7$. DE necesitó más generaciones en esta ocasión que para la función Ackley con un tiempo de ejecución similar (ver tabla 5-3).

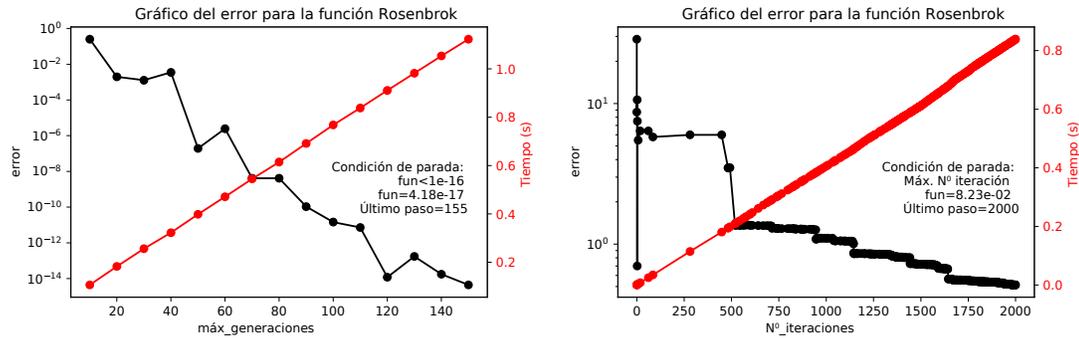


Figura 5-6: Gráficos de error y tiempo de ejecución para la función Rosenbrock.

Tabla 5-3: Estadística descriptiva para la función Rosenbrock.

Método Parámetros	DE				GSA			
	Error	Valor F.	Iter.	Tiempo(s)	Error	Valor F.	Iter.	Tiempo(s)
Mínimo	5.1e-18	1.72e-19	133.0	1.056	1.49e-7	2.97e-8	2000.0	0.815
Máximo	4.78e-5	9.1e-6	300.0	2.278	2.958	0.377	2000.0	0.90
Media	9.61e-7	2.01e-7	179.94	1.390	7.58e-2	1.10e-2	2000.0	0.838
Mediana	7.04e-17	4.11e-17	171.5	1.347	4.62e-3	9.47e-4	2000.0	0.835
Rango intercuantil	1.84e-16	5.46e-17	13.75	0.107	2.9e-02	6.07e-3	0.0	0.015
Desviación estándar	6.68e-06	1.4e-6	36.610	0.268	0.413	5.27e-2	0.0	0.019

La validación presentada en esta sección se consideró de manera preliminar para evaluar la implementación de los métodos de optimización. Es importante tener en cuenta que las funciones analizadas están definidas en espacios bidimensionales, para dimensiones grandes estos resultados pueden cambiar (Xiang *et al.*, 2017). Los casos de interés para nuestra metodología tienen una dimensión mucho mayor, igual al número de puntos donde se realizará la simulación.

Los errores alcanzados en las 50 corridas por DE, en general; son menores para las tres funciones de prueba. Los tiempos de ejecución para las funciones Ackley y Rastrigin fueron menores utilizando GSA, donde además ambos algoritmos fueron efectivos pues alcanzaron la precisión deseada. La función más difícil fue la Rosenbrock, la cual no pudo ser optimizada por GSA, aunque sí con DE. Por lo tanto, para esta validación preliminar, se puede afirmar que DE fue más efectivo y GSA más eficiente computacionalmente.

5.2. Validación a partir de casos de estudios

Para validar la metodología se tomarán tres casos de estudio con diferentes características formado por pozos provenientes de diversas fuentes. Para cada uno de ellos se estudiará un intervalo productor de petróleo del que se extraerá la información necesaria para construir la función objetivo. Los datos de registro de pozo se filtraron a cada metro de profundidad mediante promedios locales. Este filtro se ha empleado con anterioridad (Ramírez, 2018; Maldonado, 2014) y nos permite reducir, además, el número de variables de la optimización y en consecuencia los recursos de computo necesarios.

En cada pozo se realizó un análisis estadístico univariado, variográfico y bivariado de los datos de las variables de interés (más detalles en el capítulo 2). Denotaremos por $Phit$ a la porosidad total e Ip a la impedancia acústica calculada a partir de los registros de densidad de la roca ($Rhob$) y velocidad de la onda longitudinal (Vp) mediante la ecuación 4-1.

Finalmente se realizó una cosimulación estocástica espacial mediante los métodos de optimización descritos, con el fin de obtener una porosidad simulada ($Phits$). Nuestro objetivo es evaluar de manera cualitativa dicha simulación, de esta forma si se obtienen buenos resultados entonces podemos afirmar que la metodología es consistente. Para el Caso de estudio I se evaluarán diferentes implementaciones de dicha cosimulación. De las variantes que se analizarán se escogerá la que mejor resultado brinde y se continuará la validación con los otros dos casos de estudio. En todos los casos de estudio se compararán los resultados obtenidos utilizando DE y GSA. Para más detalles de la implementación de esos métodos, véase la sección 4.2.

5.2.1. Caso de estudio I

Para el primer caso de estudio era de interés buscar un pozo ubicado en algún área perteneciente a los Estados Unidos Mexicanos. Se seleccionó el pozo *Lakach-1*, ubicado en la zona de aguas profundas del Golfo de México, con un tirantes de agua de $988m$. Este pozo pertenece a la provincia petrolera Cordilleras Mexicanas y resultó productor en areniscas turbidíticas del Mioceno inferior en la parte sur de la provincia (Schlumberger Oilfield Services, 2019; Ramírez Da Silveira, 2014). Estos datos fueron proporcionados por el CNIH. Se seleccionó el intervalo productor ($3035 - 3380m$). En la figura 5-7 se muestra el registro del pozo con las curvas de impedancia acústica y porosidad.

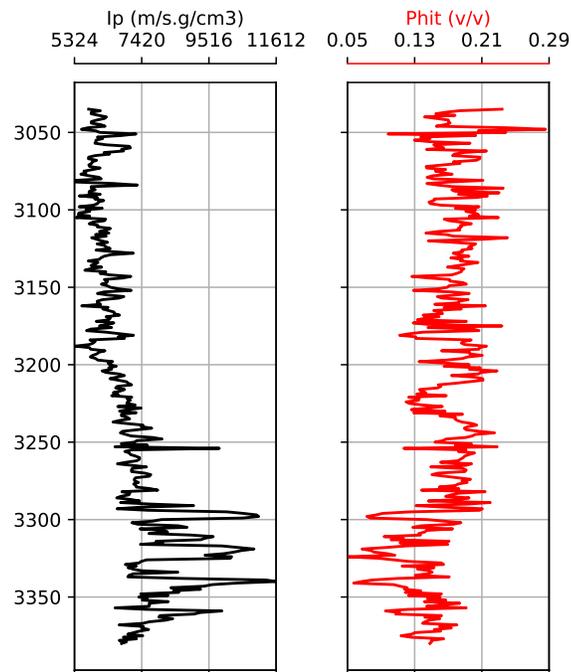
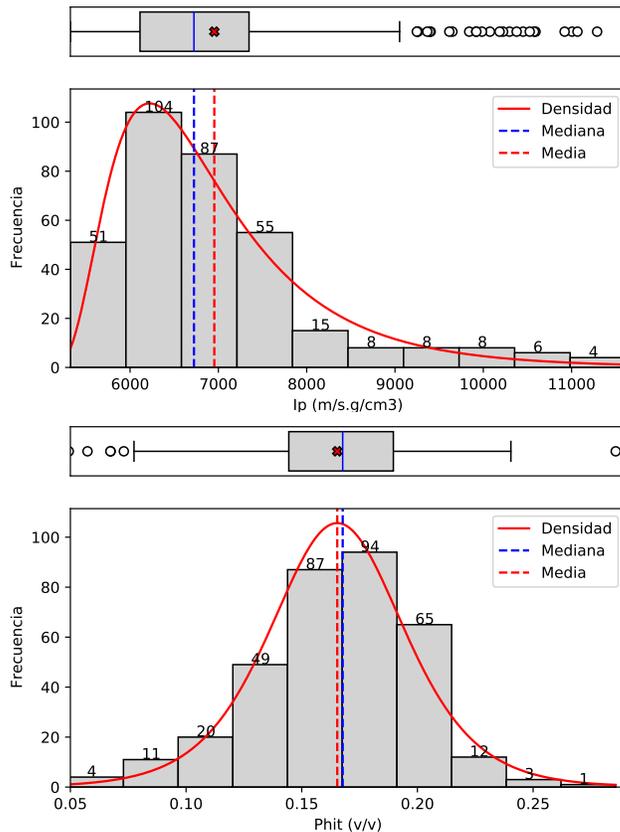


Figura 5-7: Registro del pozo *Lakach-1*.

Aunque las cópulas paramétricas son más eficiente computacionalmente, razón por las que fueron elegidas, utilizar las no paramétricas pudiera mejorar la representación de la dependencia. Por este motivo se analizará una variación utilizando una cópula no paramétrica en la sección 5.2.1.3. En la sección 5.2.1.4 se verá con más detalles una comparación entre la implementación actual y la presentada por Ramírez (2018); Le *et al.* (2020).

5.2.1.1. Análisis estadístico y variográfico de los datos

Análisis estadístico univariado: en la figura 5-8 se muestra la estadística univariada del pozo. Se realizó un ajuste de las distribuciones marginales de tipo $LogNormal(\mu_l = 7.43459, \sigma_l = 0.555439, \gamma = 4977.04)$ para I_p ; y $Logística(\mu = 0.165352, \beta = 0.0193547)$ para $Phit$.



(a) Histograma y gráfico de caja para la impedancia acústica (arriba) y porosidad total (abajo).

(b) Tabla de la estadística descriptiva.

Estadística	$I_p(m/s.g/cm^3)$	$Phit(v/v)$
Tamaño	346.0	346.0
Mínimo	5324.432	0.0493
Máximo	11612.42	0.2857
Rango	6287.992	0.2364
Media	6953.937	0.165352
Mediana	6723.458	0.1677
Primer cuantil	6112.285	0.1444
Tercer cuantil	7347.029	0.1896
Rango intercuantil	1234.744	0.0452
Varianza	1391475	0.001229
Desviación estándar	1179.608	0.035055
Simetría	1.587036	-0.43234
Curtosis	2.632152	0.542704

Figura 5-8: Estadística descriptiva para el pozo *Lakach-1*

Variografía: Se supondrá que la variable es estacionaria ya que la media local y la varianza no parecen depender de las coordenadas. El modelo teórico de variograma ajustado fue esférico con una meseta de 0.00112 y un alcance de 52m. Se registró, además, un efecto pepita de 0.0007 (figura 5-9).

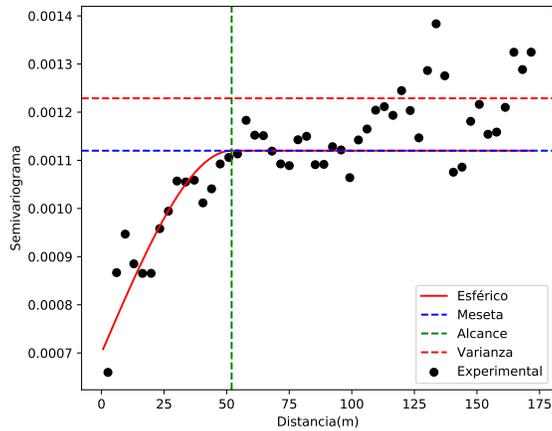


Figura 5-9: Variograma de Phit para el pozo *Lakach-1*.

Análisis bivariado utilizando cópulas: la figura 5-10 ilustra la dependencia entre ambas variables. Para relacionar la distribución de ambas variables de manera conjunta se utilizó una

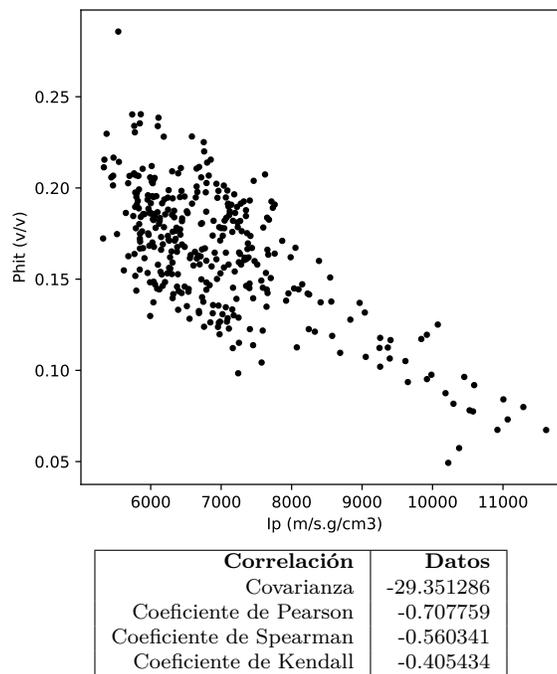


Figura 5-10: Gráfico de dispersión de I_p contra $Phit$ y estadística descriptiva bivariada de los datos.

cópula de *Frank* con parámetro $\delta = -4.2364$. Las funciones de distribución y densidad descritas se muestran en la figura 5-11.

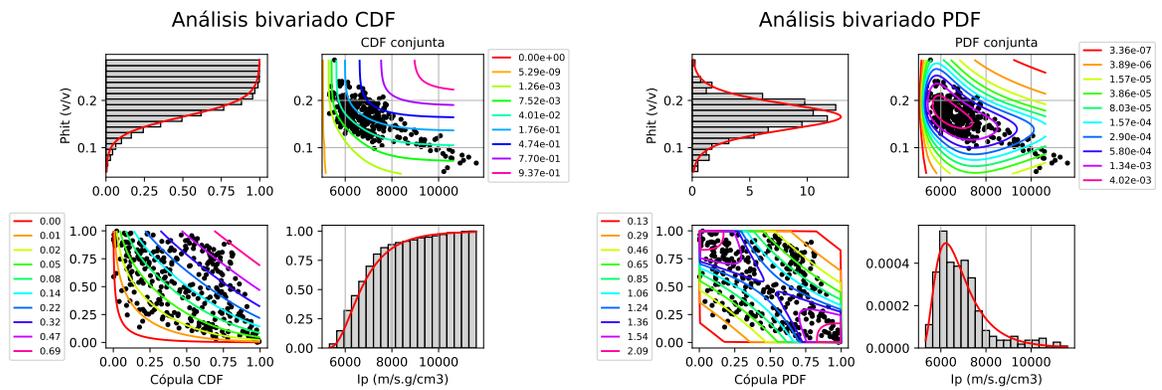


Figura 5-11: Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de I_p y $Phit$ respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.

5.2.1.2. Cosimulación con cópulas paramétricas

A continuación se verán los resultados de cosimulación utilizando cópulas paramétricas. Para ello se realizó un ajuste paramétrico de las distribuciones marginales y la cópula, y la distribución bivariada resultante se utilizó para generar los valores iniciales y para el cálculo de la distribución condicional durante el proceso de optimización.

Para este problema las simulaciones obtenidas poseen valores de 0.0006817 y 0.0009858 al ser evaluadas en la función objetivo para DE y GSA, respectivamente. Ambos algoritmos se detuvieron cuando el valor de la función objetivo fue menor que la precisión fijada, considerándose como optimizaciones satisfactorias en ambos casos. El tiempo de ejecución de DE fue de 1557.04s (aproximadamente 26min) tras 3169 generaciones, mientras que GSA demoró 673.66s (aproximadamente 11min) con 40 iteraciones. Para este problema tanto DE como GSA mostraron ser efectivos, aunque DE se mostró menos eficiente computacionalmente.

En las figuras 5-12-5-18 se muestran los resultados de las simulaciones obtenidas mediante ambos métodos del pozo escogido. La figura 5-12 muestra el registro de la simulación. En general ambas porosidades simuladas siguen el comportamiento espacial de los datos.

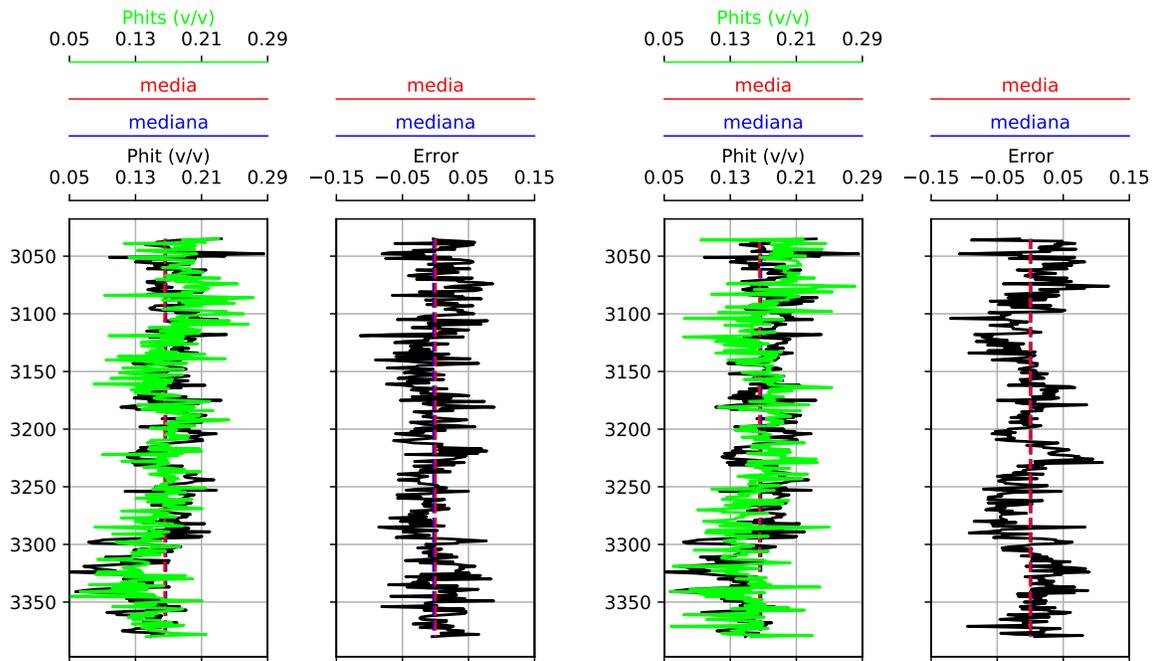
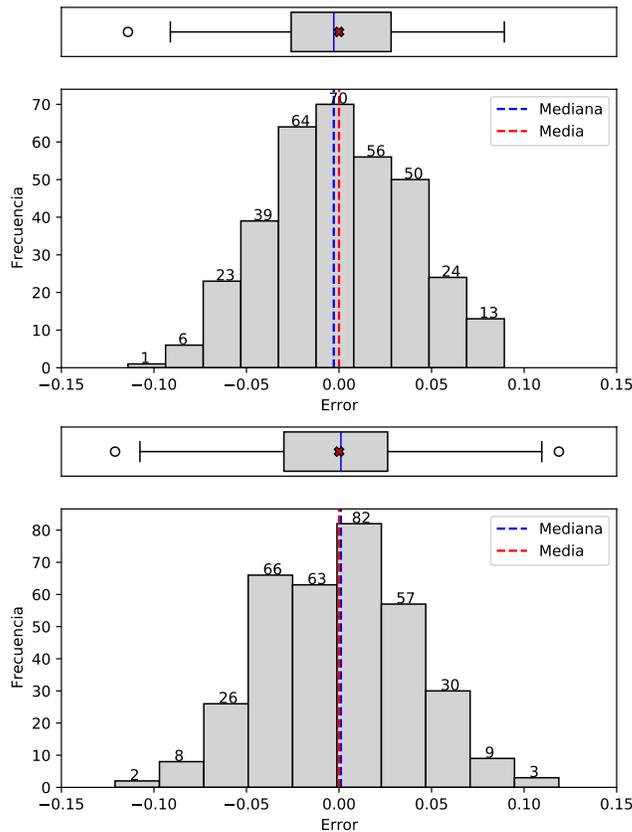


Figura 5-12: Registros de los datos y simulados de la Phit y el error de estimación, para el pozo *Lakach-1* (DE a la izquierda, GSA a la derecha).

Adicionalmente, se calculó la diferencia entre la variable simulada y los valores conocidos, obteniéndose un *error de estimación*. Dado que la estimación no es el objetivo de la simulación estos valores se consideran como una evaluación cualitativa de la misma. En la figura 5-13 se muestra la estadística descriptiva de dicho error. Se puede observar una mayor dispersión para la simulación obtenida con GSA y el error medio de DE es menor. GSA presentó apenas 2 valores extremos mientras DE solo tuvo uno, y aunque en ambos casos hubo puntos con errores mayores a 0.1, el 75% de ellos se mantuvo por debajo de 0.06. Para más detalles sobre el

comportamiento de los errores se puede ver el apéndice E



(a) Histograma y gráfico de caja del error para DE (arriba) y para GSA (abajo).

(b) Tabla de la estadística descriptiva de los errores.

Estadística	Error	(DE)	(GSA)
Tamaño		346.0	346.0
Mínimo		-0.114028	-0.120986
Máximo		0.089304	0.118784
Rango		0.203332	0.239770
Media		-0.000005	0.000016
Mediana		-0.002823	0.001027
Primer cuantil		-0.025853	-0.029789
Tercer cuantil		0.028150	0.026285
Rango intercuantil		0.054002	0.056074
Varianza		0.001439	0.001647
Desviación estándar		0.037932	0.040578
Simetría		0.016937	0.024793
Curtosis		-0.406918	-0.141979

Figura 5-13: Estadística descriptiva del error en la simulación para el pozo *Lakach-1*.

La figura 5-14 muestra la cercanía de las componentes teóricas y resultado de la simulación de la función objetivo. No existen diferencias apreciables entre ambos métodos en el ajuste del variograma aunque el de GSA fue ligeramente mejor para la distribución condicional.

Las simulaciones obtenidas respetan la distribución univariada de los datos (figura 5-15). Las distribuciones marginales estimadas para Phits fueron *Logística*($\mu = 0.165347, \beta = 0.0193355$) para DE y *Logística*($\mu = 0.165367, \beta = 0.0193717$) para GSA. Se calcularon también las distribuciones empíricas de la simulación y los datos obteniéndose un buen ajuste entre ambas (figura 5-16).

Las simulaciones respetan la dependencia bivariada de los datos (figura 5-17). La figura 5-18 muestra las funciones de distribución y densidad para las simulaciones. Las cópulas ajustadas fueron de tipo *Frank* con parámetros ($\delta = -4.23422$) para DE y ($\delta = -4.22488$) para GSA.

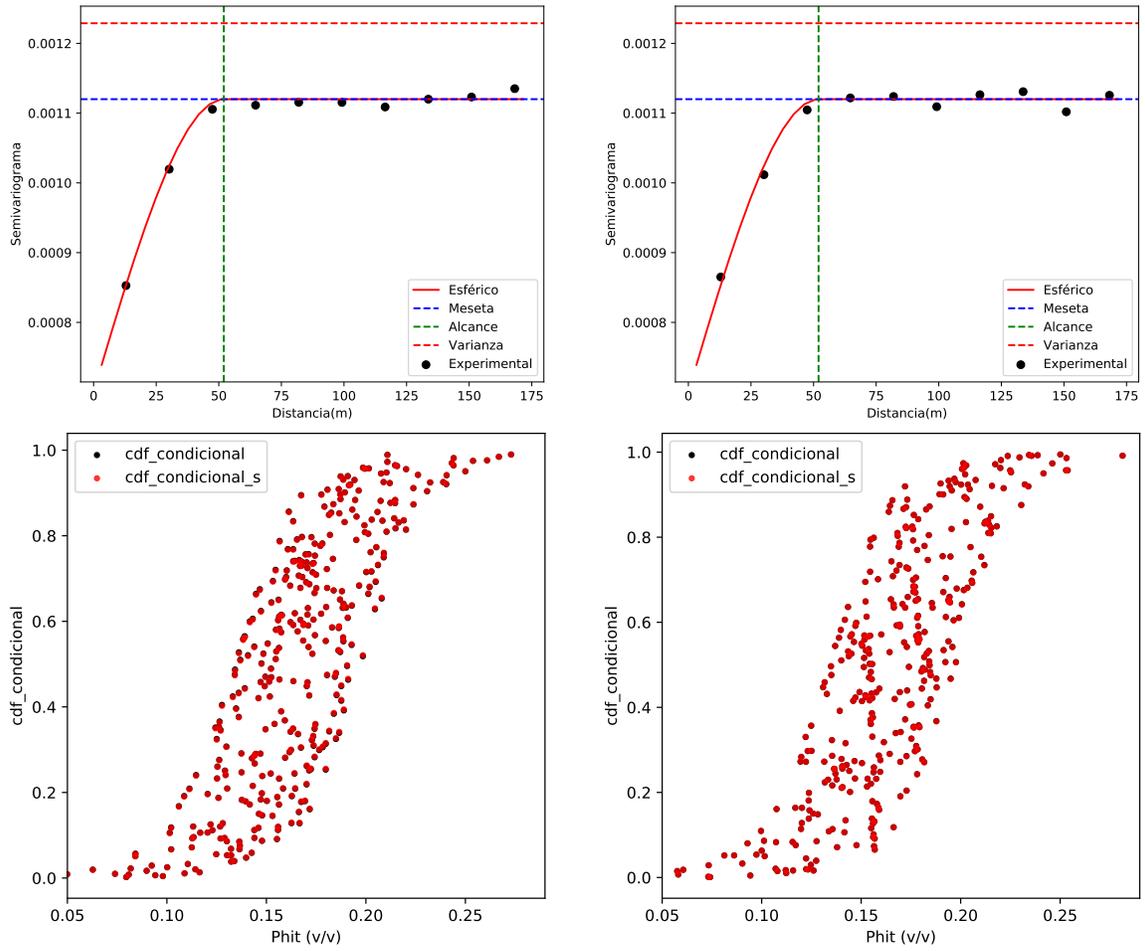
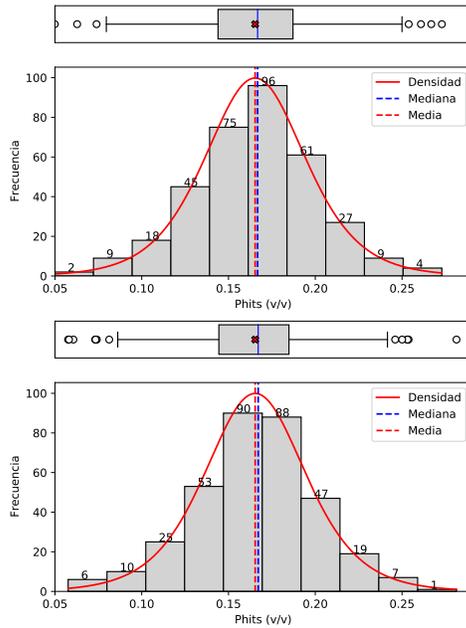


Figura 5-14: Variogramas y función de distribución condicional de la simulación para el pozo *Lakach-1* (DE a la izquierda, GSA a la derecha).



(a) Histograma y gráfico de caja para los simulados por DE (arriba) y GSA (abajo).

(b) Tabla de la estadística descriptiva.

Estadística	Phit(v/v)	Datos	(DE)	(GSA)
Tamaño		346.0	346.0	346.0
Mínimo		0.0493	0.049880	0.057571
Máximo		0.2857	0.272950	0.281384
Rango		0.2364	0.223070	0.223812
Media		0.165352	0.165347	0.165367
Mediana		0.1444	0.166771	0.167114
Primer cuantil		0.27516	0.143953	0.144444
Tercer cuantil		0.1896	0.187159	0.184844
Rango intercuantil		0.0452	0.043206	0.040401
Varianza		0.001229	0.001226	0.001231
Desviación estándar		0.035055	0.035020	0.035086
Simetría		-0.43234	-0.047183	-0.117585
Curtosis		0.542704	0.552923	0.620527

Figura 5-15: Estadística descriptiva de la simulación para el pozo *Lakach-1*.

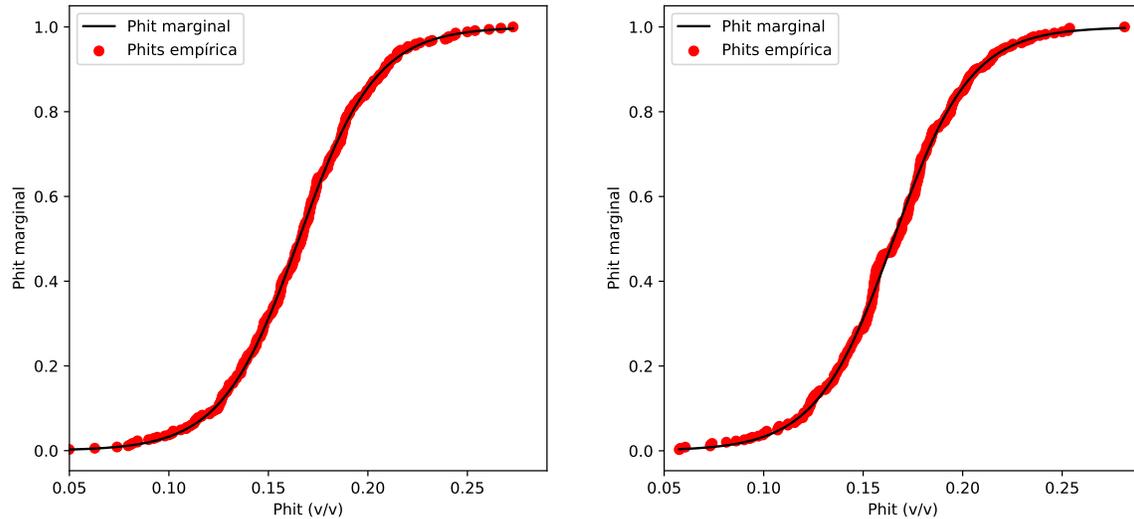


Figura 5-16: Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).

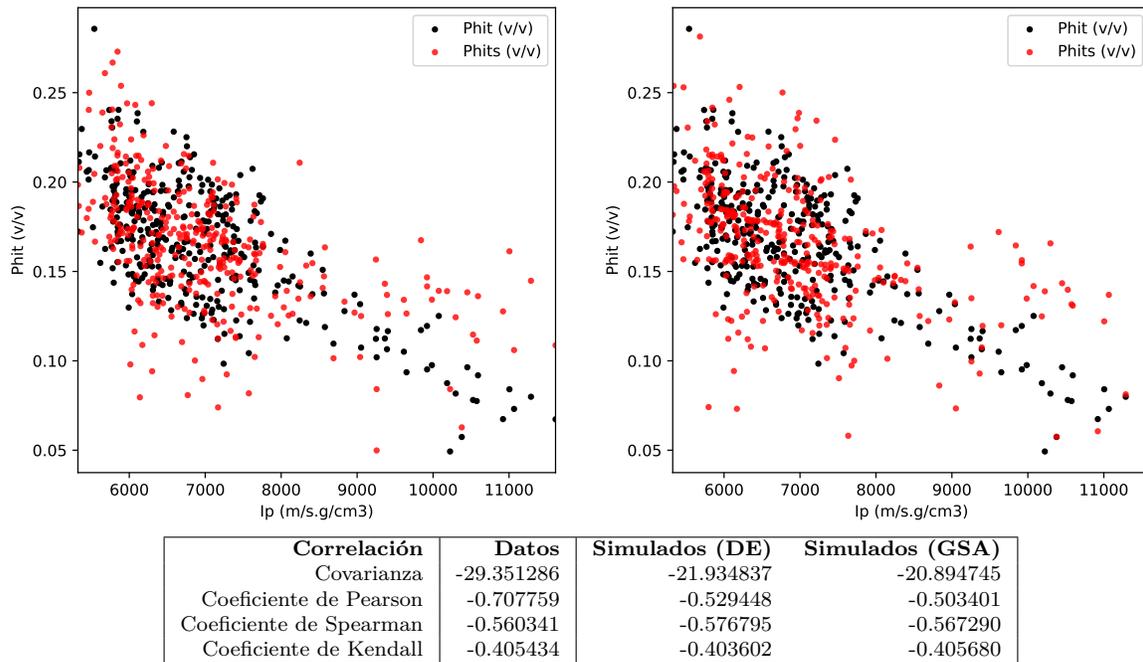


Figura 5-17: Gráfico de dispersión de I_p contra $Phits$ (DE a la izquierda, GSA a la derecha) y estadística descriptiva bivariada de los datos simulados.

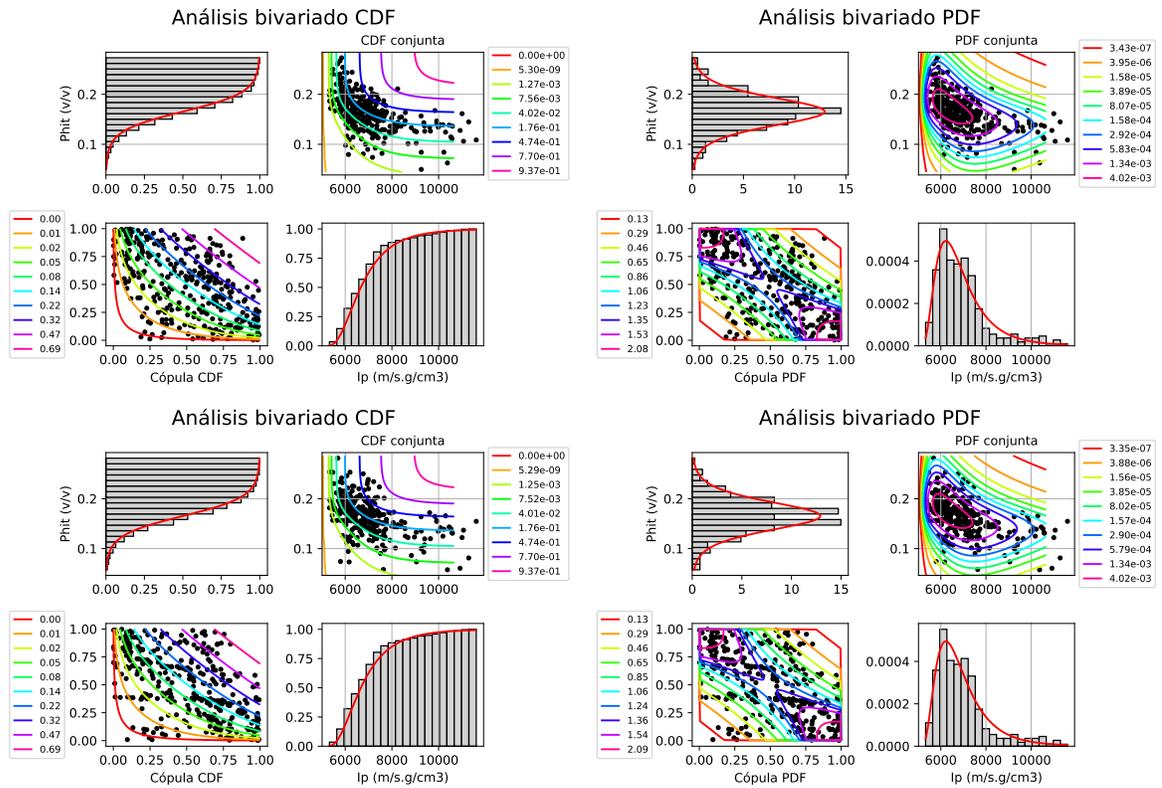


Figura 5-18: Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de I_p y $Phit$ respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.

5.2.1.3. Variante híbrida de cosimulación

En un primer momento se intentó utilizar estimación no paramétrica de la cópula tanto para generar los valores iniciales como durante el proceso de optimización. Esta variante fue descartada al presentarse dificultades con los recursos de cómputo disponibles, ya que realizar una estimación no paramétrica en cada paso es más costoso en términos de memoria, procesamiento y tiempo.

A continuación se muestra una variante de la cosimulación paramétrica donde se consideró una simulación de valores iniciales de entrada de los algoritmos de optimización utilizando estimación por núcleos (véase la sección 2.1.1). Es decir, se utilizó una distribución no paramétrica para generar la realización de valores iniciales, mientras que las distribuciones condicionales teórica y simulada utilizadas en la función objetivo \mathcal{O}_2 continuaron siendo paramétricas (véase la sección 4.2). De esta forma se restringe el espacio de búsqueda pues se parte de valores iniciales con una función objetivo menor, que deben estar más cerca del óptimo global, y se mantiene la misma eficiencia durante el proceso de optimización.

Para este problema las simulaciones obtenidas poseen valores de 0.0008946 y 0.0009867 al ser evaluadas en la función objetivo para DE y GSA, respectivamente. El tiempo de ejecución de DE fue de 965.89s (aproximadamente 16min) tras 1955 generaciones, mientras que GSA demoró 579.45s (aproximadamente 10min) necesitando apenas 34 iteraciones.

En las figuras 5-19-5-25 se muestran los resultados de las simulaciones obtenidas mediante ambos métodos. La figura 5-19 muestra el registro de la simulación. En general las porosidades simuladas siguen el comportamiento espacial de los datos.

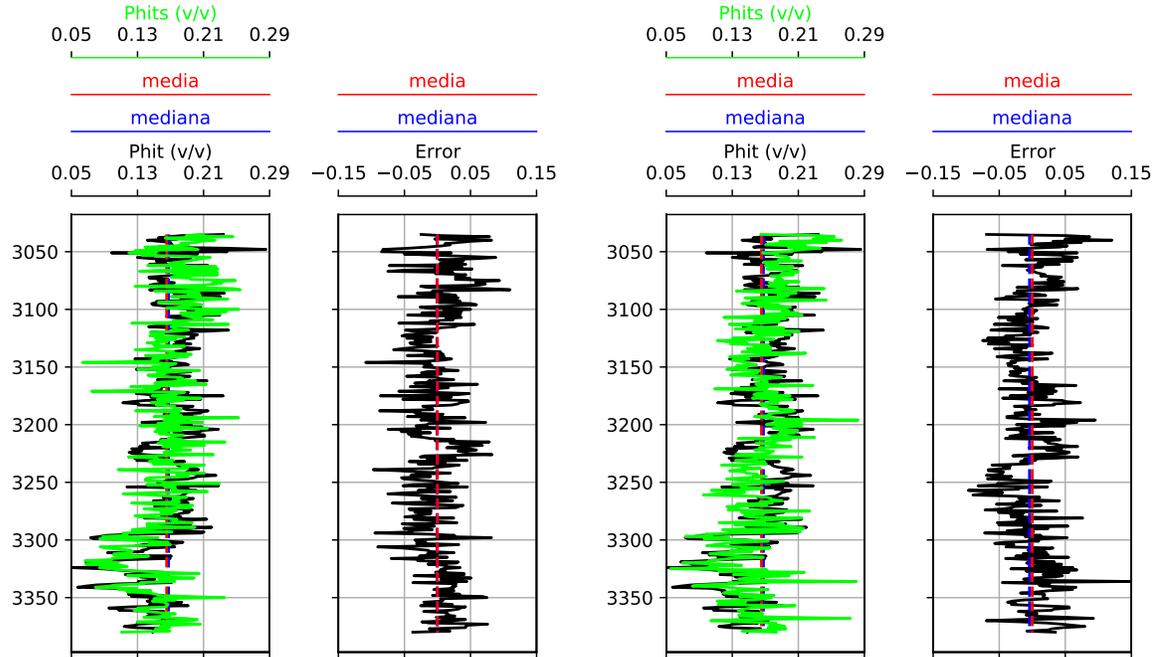
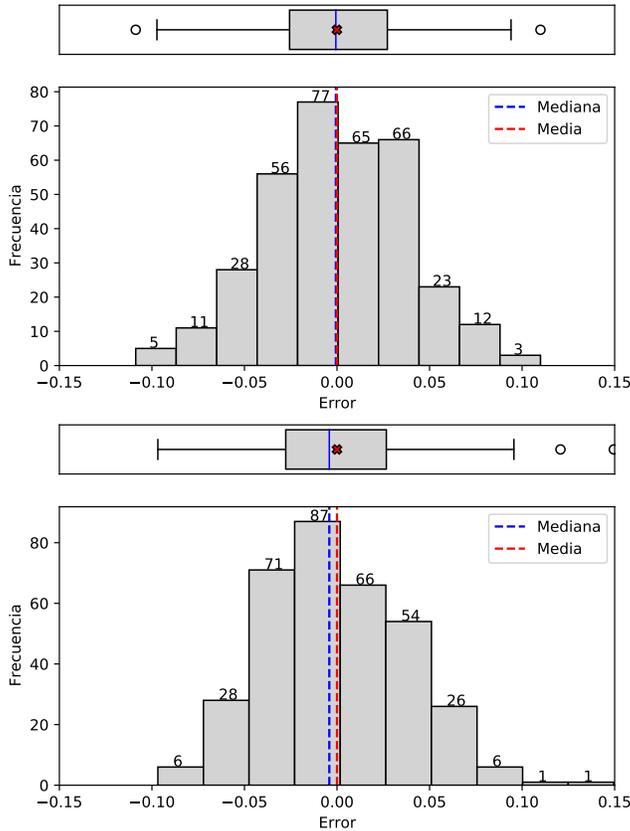


Figura 5-19: Registros de los datos y simulados de la Phit y el error de estimación, para el pozo *Lakach-1* (DE a la izquierda, GSA a la derecha).

En la figura 5-20 se muestra la estadística descriptiva del error. El error medio fue menor

para GSA y las dispersiones de dicho error se comportaron de manera similar. Ambos presentan 2 valores extremos y aunque hubo puntos con errores mayores a 0.1, el 75 % de ellos se mantuvo por debajo de 0.06.



(a) Histograma y gráfico de caja del error para DE (arriba) y para GSA (abajo).

(b) Tabla de la estadística descriptiva de los errores.

Estadística	Error	
	(DE)	(GSA)
Tamaño	346.0	346.0
Mínimo	-0.108747	-0.096790
Máximo	0.109928	0.149550
Rango	0.218675	0.246340
Media	-0.000065	-0.000005
Mediana	-0.000633	-0.004161
Primer cuantil	-0.025638	-0.027703
Tercer cuantil	0.027202	0.026659
Rango intercuantil	0.052840	0.054362
Varianza	0.001425	0.001497
Desviación estándar	0.037753	0.038690
Simetría	-0.050146	0.316156
Curtosis	-0.080388	0.077340

Figura 5-20: Estadística descriptiva del error en la simulación para el pozo *Lakach-1*.

La figura 5-21 muestra la cercanía de las componentes teóricas y resultado de la simulación de la función objetivo.

Las simulaciones obtenidas respetan la distribución univariada de los datos (figura 5-22). Las distribuciones ajustadas fueron de tipo *Logística* con parámetros ($\mu = 0.165, \beta = 0.0193$) en los dos casos. Se calcularon también las distribuciones empíricas de la simulación y los datos obteniéndose un buen ajuste entre ambas (figura 5-23).

Las simulaciones obtenidas respetan la dependencia bivariada de los datos (figura 5-24). La figura 5-25 muestra las funciones de distribución y densidad para las simulaciones. Las cópulas de ajuste fueron de tipo *Frank*, con parámetro $\delta = -4.23$ para ambos métodos.

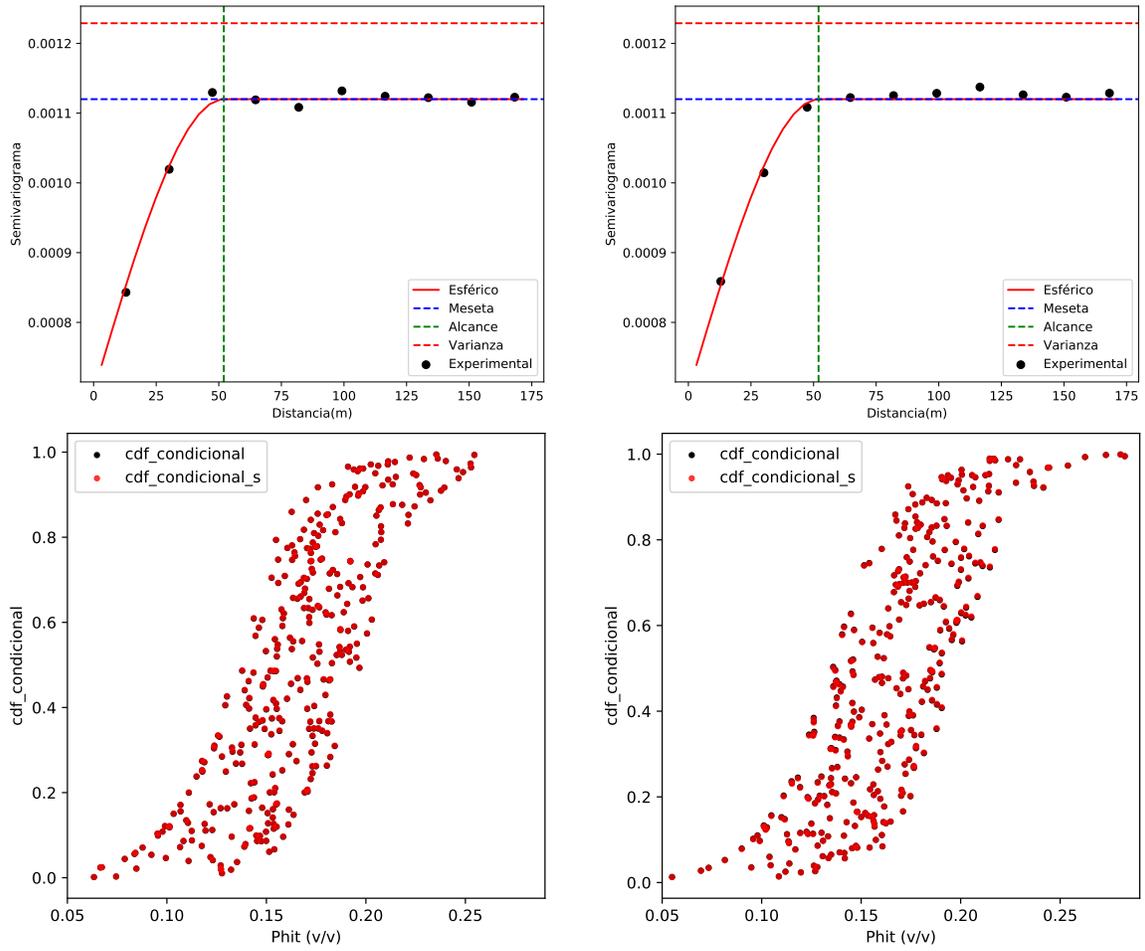
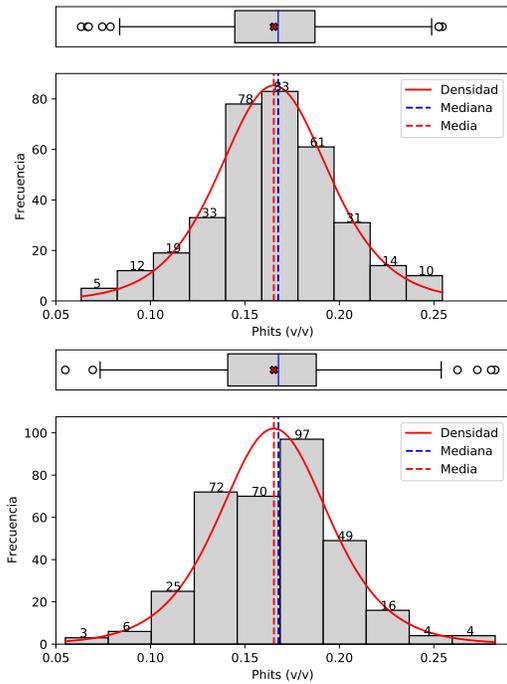


Figura 5-21: Variogramas y función de distribución condicional de la simulación (DE a la izquierda, GSA a la derecha).



(a) Histograma y gráfico de caja para los simulados por DE (arriba) y GSA (abajo).

(b) Tabla de la estadística descriptiva.

Estadística	Phit(v/v)	Datos	(DE)	(GSA)
Tamaño		346.0	346.0	346.0
Mínimo		0.0493	0.063353	0.054959
Máximo		0.2857	0.254528	0.282474
Rango		0.2364	0.191175	0.227515
Media		0.165352	0.165286	0.165346
Mediana		0.1444	0.167612	0.167761
Primer cuantil		0.27516	0.144661	0.140927
Tercer cuantil		0.1896	0.187081	0.187763
Rango intercuantil		0.0452	0.042420	0.046835
Varianza		0.001229	0.001232	0.001220
Desviación estándar		0.035055	0.035106	0.034925
Simetría		-0.43234	-0.145121	-0.131870
Curtosis		0.542704	0.253314	0.561330

Figura 5-22: Estadística descriptiva de la simulación para el pozo *Lakach-1*.

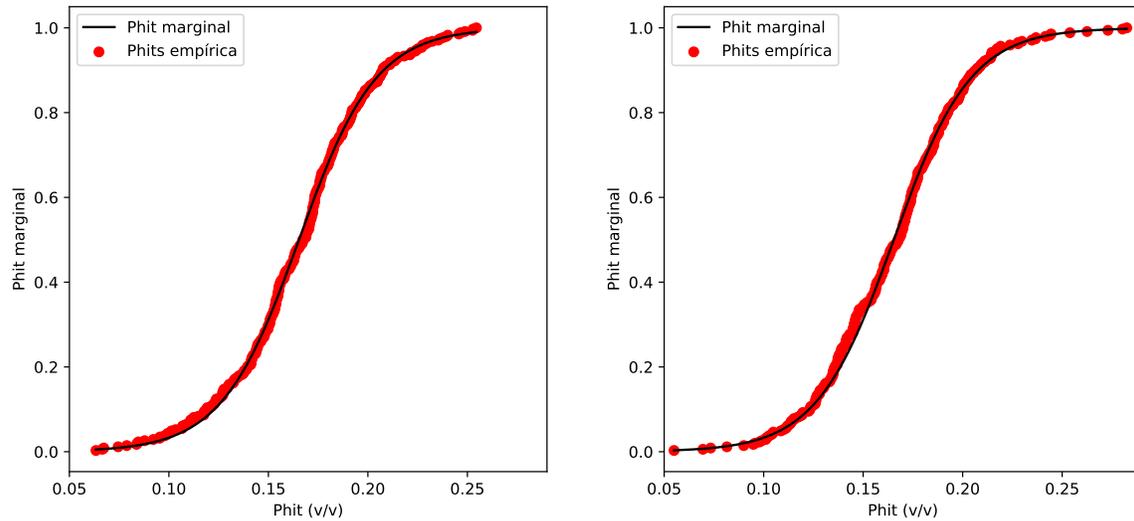


Figura 5-23: Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).

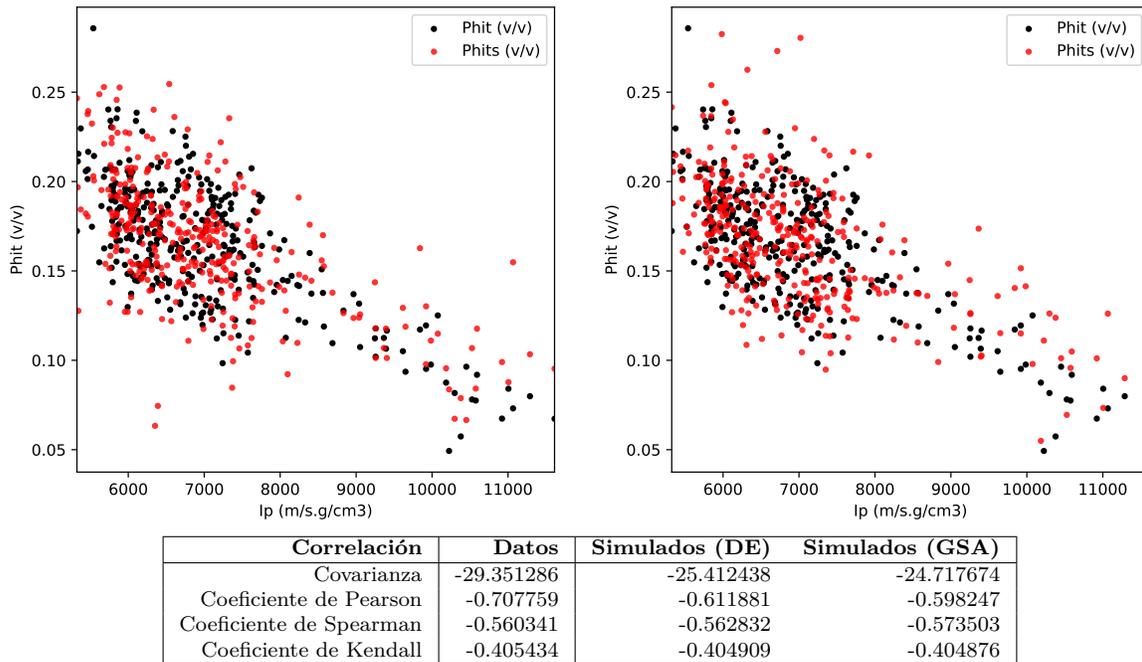


Figura 5-24: Gráfico de dispersión de I_p contra $Phits$ (DE a la izquierda, GSA a la derecha) y estadística descriptiva bivariada de los datos simulados.

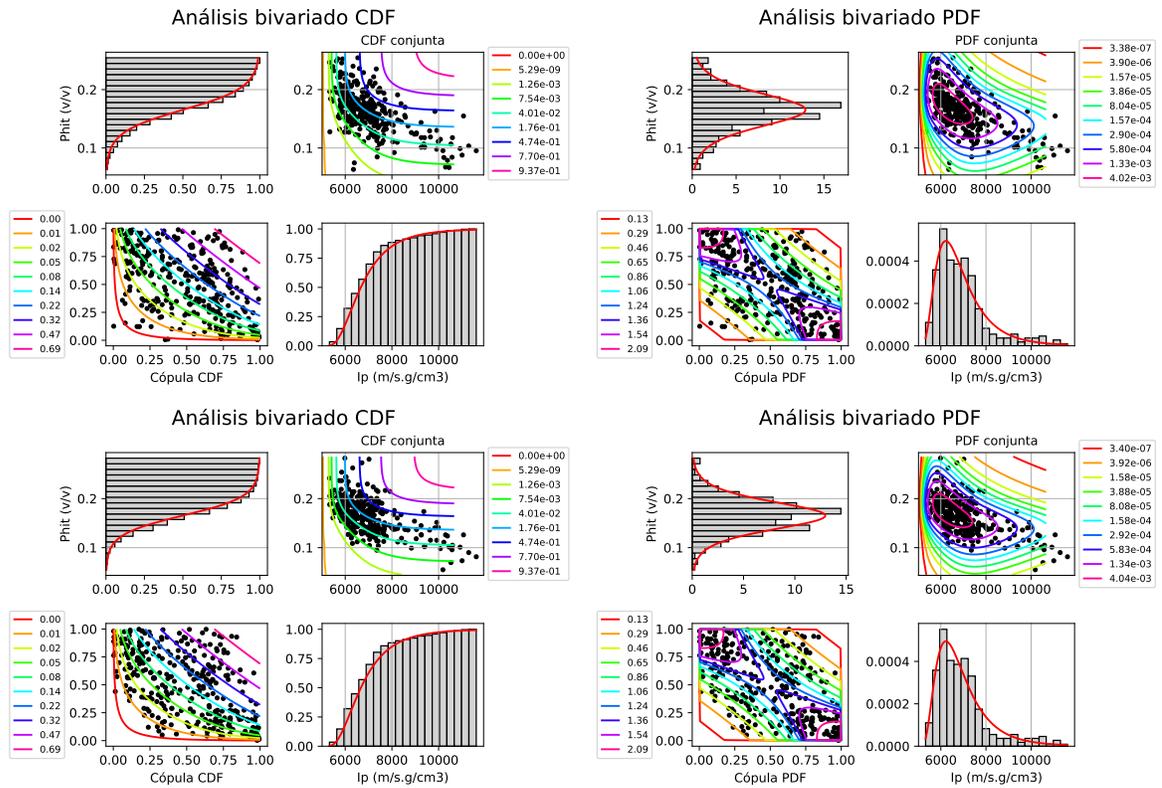


Figura 5-25: Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de I_p y $Phit$ respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.

En la tabla 5-4 se muestra una comparación entre la variante paramétrica e híbrida. La variante híbrida afectó de manera positiva a la simulación en términos de eficiencia y permitió un mejor ajuste de las distribuciones univariadas y bivariadas. Sin embargo, dicha mejora no es suficiente para elegirla por sobre la variante paramétrica. Por lo cuál se continuará usando esta variante.

Tabla 5-4: Resumen de la comparación entre las dos variantes de cosimulación.

Parámetros		Paramétrico		Híbrida	
		DE	GSA	DE	GSA
FMO	\mathcal{O}_1	0.0009563	0.0004837	0.000603	0.0005951
	\mathcal{O}_2	0.0000296	0.0001981	0.0002916	0.0003917
	\mathcal{O}	0.0006817	0.0009858	0.0008946	0.0009867
Desempeño	Niter	3169	40	1955	34
	Tiempo (s)	1557.04	673.66	965.89	579.45
Error	Media	-0.000005	0.000016	-0.000065	-0.000005
	Desviación estándar	0.037932	0.040578	0.037753	0.038690
	Rango	0.203332	0.239770	0.218675	0.246340

5.2.1.4. Comparación de la implementación en la metodología de cosimulación

Para dar una valoración más completa de nuestra metodología resulta de interés comparar nuestra implementación con la presentada por Ramírez (2018); Le *et al.* (2020), en los cuales se utilizó recocido simulado clásico (SA) para la optimización. En esos trabajos se implementó la metodología utilizando la biblioteca BCSASIM (Díaz Viera *et al.*, 2016), las principales diferencias entre dicha implementación y la nuestra fueron discutidas en la sección 4.2. Para el SA se utilizó una temperatura inicial de 100 con un factor de enfriamiento de 0.9. Al igual que con nuestra implementación se fijó una tolerancia de $1e - 3$.

En las figuras 5-26-5-31 se muestran los resultados de las simulaciones obtenidas mediante los métodos DE, GSA y SA, en ese orden. La figura 5-26 muestra el registro de la simulación. En general las porosidades simuladas tratan de seguir el comportamiento espacial de la variable de referencia.

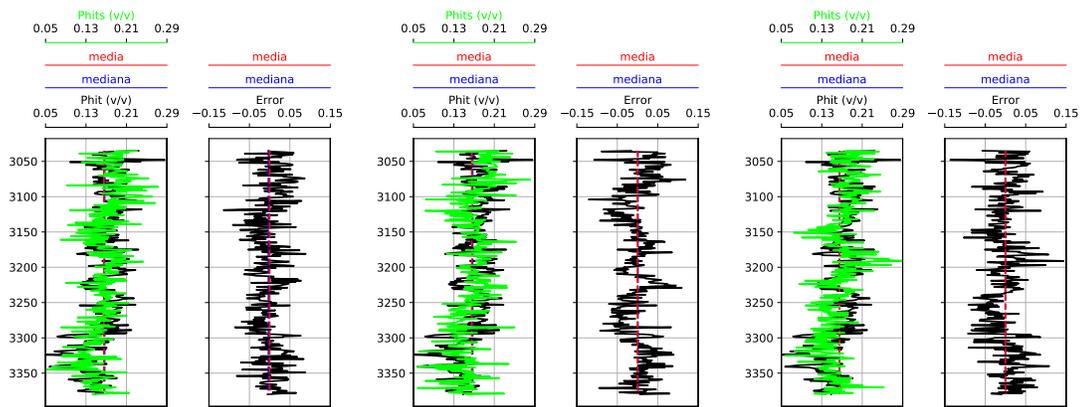
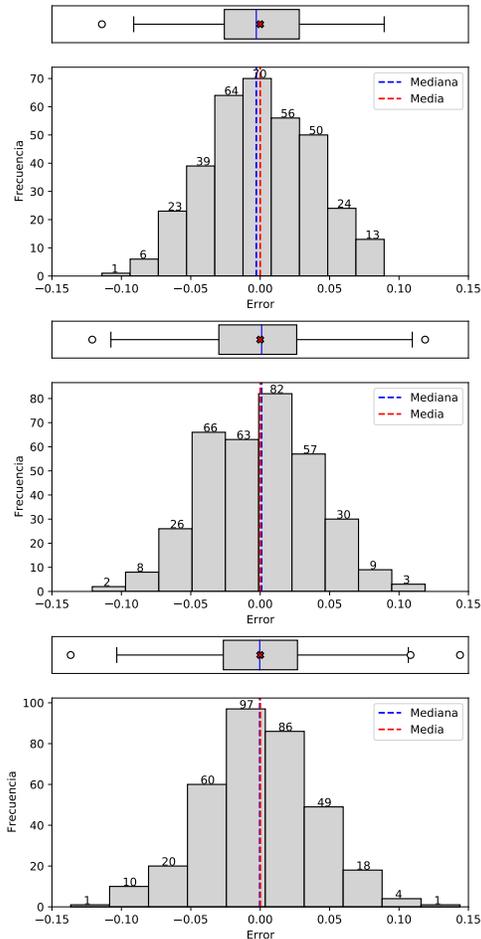


Figura 5-26: Registros de los datos y simulados de la Phit y el error de estimación, para DE, GSA, SA (de izquierda a derecha), en el pozo *Lakach-1*.

En la figura 5-27 se muestra la estadística descriptiva del error. El error medio fue mayor para SA, presentando además una mayor dispersión. Aunque los errores alcanzaran un máximo de 0.14 para SA y -0.12 para DE y GSA, en general con tres métodos el 75% de los puntos presentan errores menores de 0.06.



(a) Histograma y gráfico de caja del error para DE (arriba), GSA (medio) y SA (abajo).

(b) Tabla de la estadística descriptiva.

Estadística	Error	(DE)	(GSA)	(SA)
Tamaño		346.0	346.0	346.0
Mínimo		-0.114028	-0.120986	-0.136530
Máximo		0.089304	0.118784	0.143920
Rango		0.203332	0.239770	0.280450
Media		-0.000005	0.000016	0.000033
Mediana		-0.002823	0.001027	-0.000340
Primer cuantil		-0.025853	-0.029789	-0.026545
Tercer cuantil		0.028150	0.026285	0.026853
Rango intercuantil		0.054002	0.056074	0.053397
Varianza		0.001439	0.001647	0.001692
Desviación estándar		0.037932	0.040578	0.041131
Simetría		0.016937	0.024793	0.014720
Curtosis		-0.406918	-0.141979	0.294169

Figura 5-27: Estadística descriptiva del error en el pozo *Lakach-1*.

La figura 5-28 muestra la cercanía de las componentes teóricas y resultado de la simulación de la función objetivo. El variograma de la simulación con SA aproxima de forma casi exacta el teórico para distancias menores de 100, sin embargo la distribución condicional es peor que con los otros métodos (ver tabla 5-5).

Las simulaciones obtenidas respetan la distribución univariada de los datos (figura 5-29). Se calcularon también las distribuciones empíricas de la simulación y los datos obteniéndose un buen ajuste entre ambas (figura 5-30).

Las simulaciones obtenidas respetan la dependencia bivariada de los datos (figura 5-31), aunque en general se presenta una mayor dispersión para valores altos de I_p .

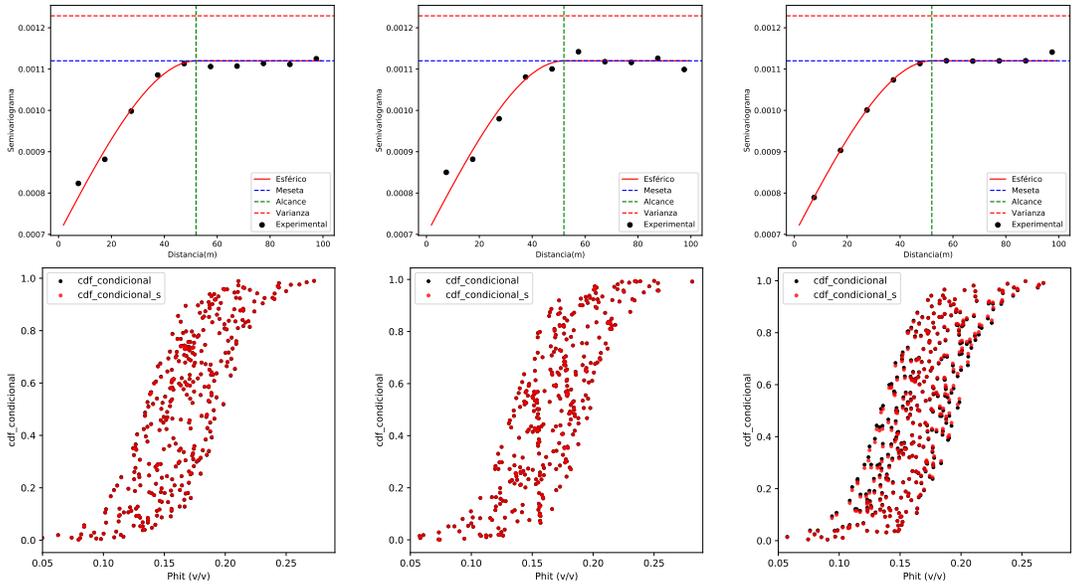


Figura 5-28: Variogramas y función de distribución condicional de la simulación para el pozo *Lakach-1* (de izquierda a derecha: DE, GSA y SA).

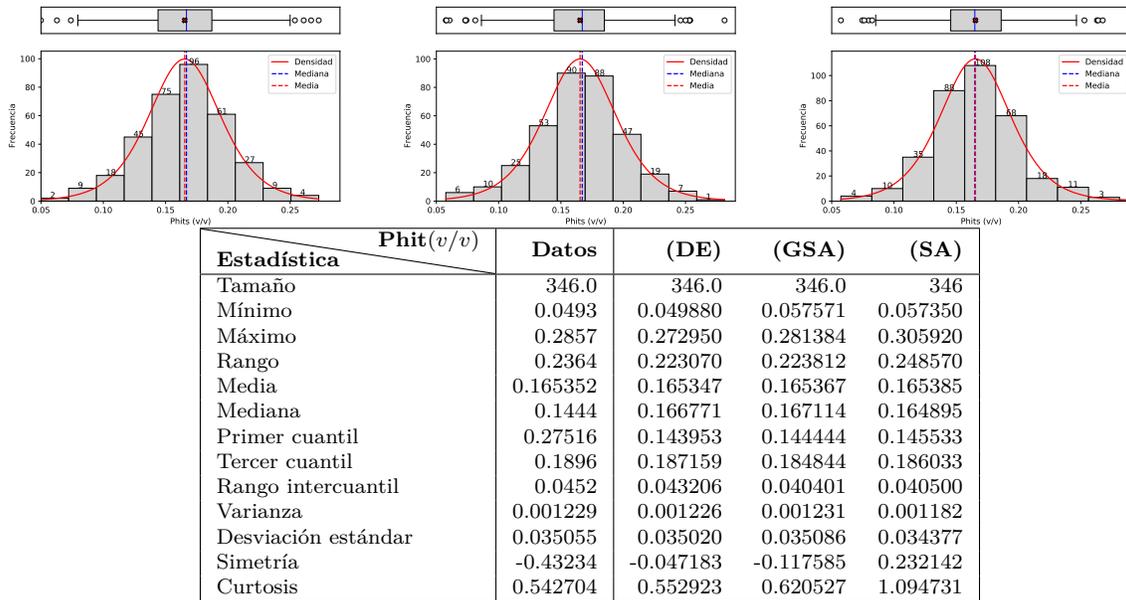


Figura 5-29: Estadística descriptiva de la simulación para el pozo *Lakach-1*. Histograma y gráfico de caja para los simulados (de izquierda a derecha: DE, GSA y SA).

En la tabla 5-5 se muestra un resumen entre ambas variante de implementación. En general nuestra implementación dio mejores resultados que SA implementada mediante BCSASIM (Díaz Viera *et al.*, 2016).

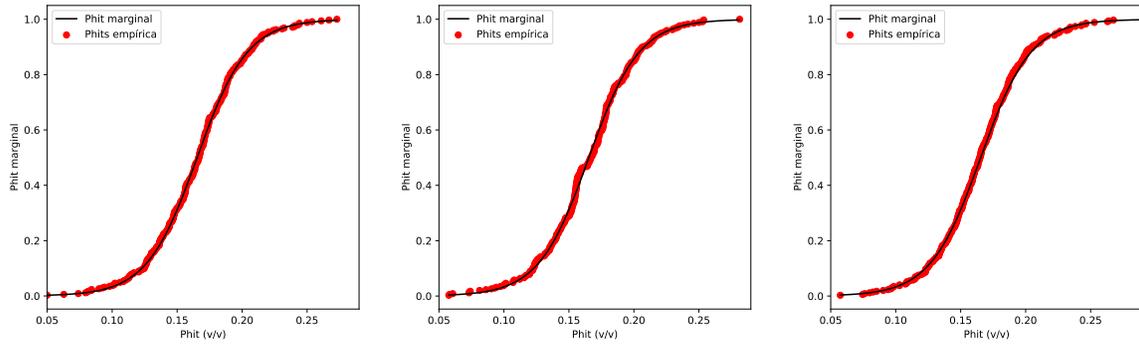


Figura 5-30: Función de distribución empírica de probabilidad para las variables simuladas (de izquierda a derecha: DE, GSA y SA).

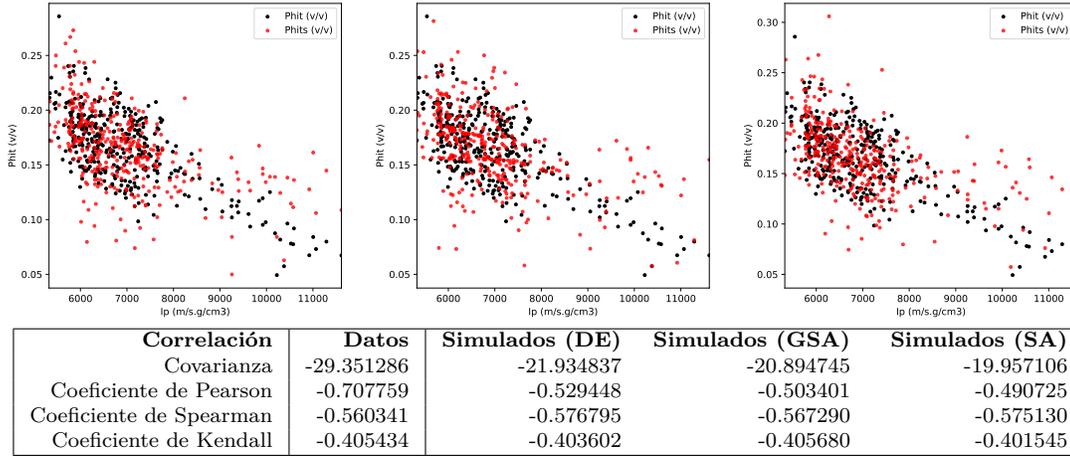


Figura 5-31: Gráfico de dispersión de Ip contra Phits (de izquierda a derecha: DE, GSA y SA) y estadística descriptiva bivariada de los datos simulados.

Tabla 5-5: Resumen de la comparación entre las dos variantes de implementación.

Parámetros		DE	GSA	SA
FMO	\mathcal{O}_1	0.0009563	0.0004837	0.000001
	\mathcal{O}_2	0.0000296	0.0001981	0.123774
	\mathcal{O}	0.0006817	0.0009858	0.123775
Condición de parada		precisión	precisión	max. número de evaluaciones
Desempeño	Niter	3169	40	—
	Tiempo (s)	1557.04	673.66	270.43
Error	Media	-0.000005	0.000016	0.000033
	Desviación estándar	0.037932	0.040578	0.041131
	Rango	0.203332	0.239770	0.280450

5.2.1.5. Simulación condicional

Se decidió realizar una *simulación espacial condicional* en el sentido clásico descrito en la sección 2.4. Para ello se tomaron 20 puntos de condicionamiento (datos duros). En las figuras 5-32-5-37 se muestran los resultados de las simulaciones obtenidas con esta restricción. La figura 5-32 muestra el registro de la simulación.

Las simulaciones obtenidas poseen valores de 0.0008017 y 0.000999 en la función objetivo para DE y GSA, respectivamente. Ambos algoritmos se detuvieron cuando el valor de la función objetivo fue menor que la precisión fijada, considerándose como optimizaciones satisfactorias en ambos casos. El tiempo de ejecución de DE fue de 1684.99s (aproximadamente 28min) tras 2964 generaciones, mientras que GSA demoró 1132.44s (aproximadamente 19min) necesitando 73 iteraciones. Para este problema tanto DE como GSA mostraron ser efectivos aunque GSA fue más eficiente computacionalmente.

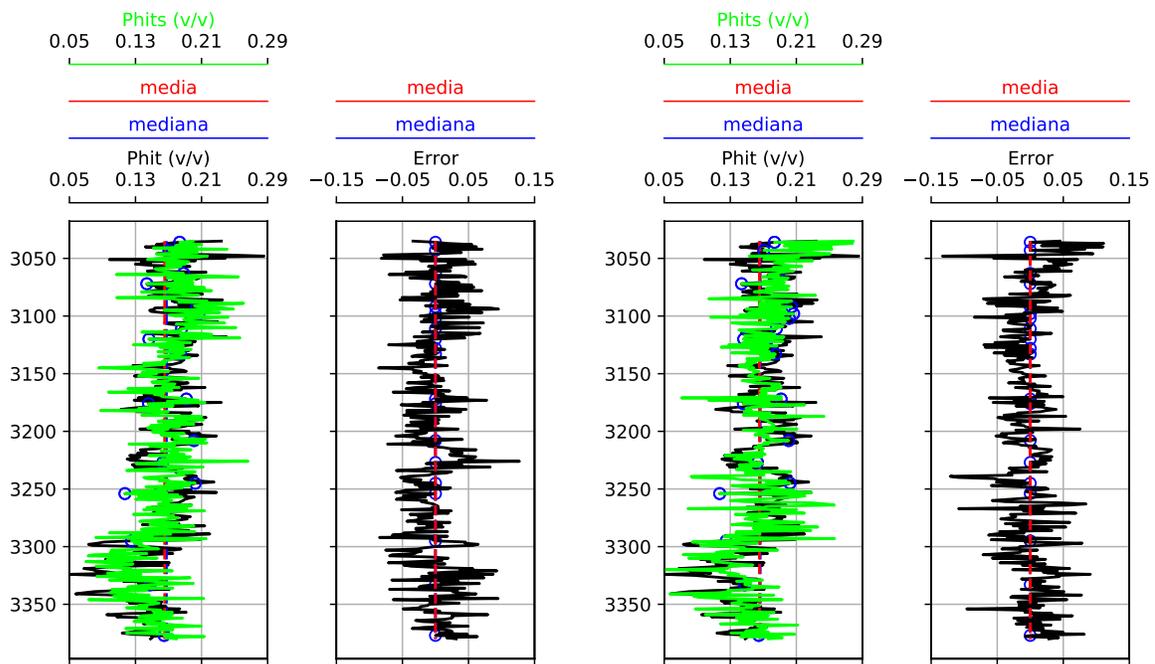
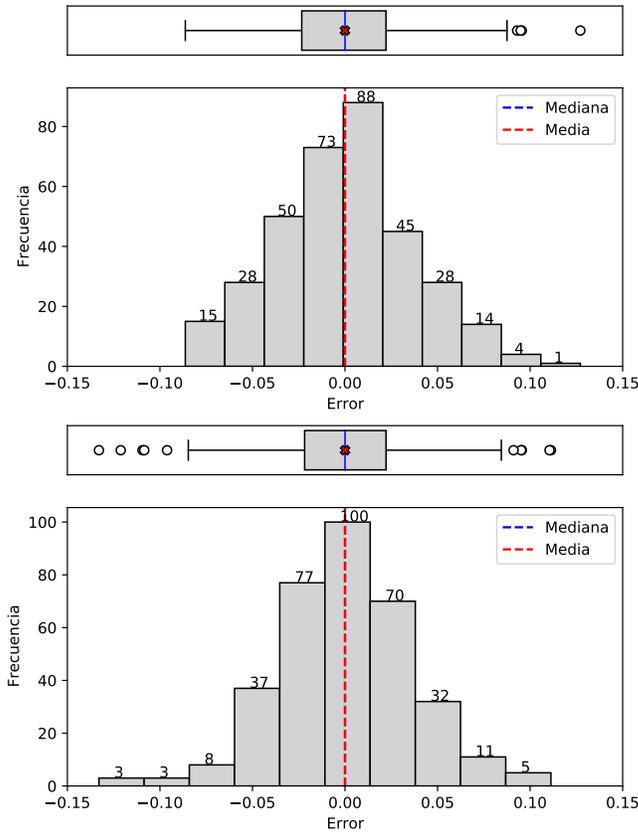


Figura 5-32: Registros de los datos, de la simulación condicional de la Phit y del error de estimación, para el pozo *Lakach-1* (DE a la izquierda, GSA a la derecha).

Adicionalmente, se calculó la diferencia entre la variable simulada y los valores conocidos. En la figura 5-33 se muestra la estadística descriptiva de dicho error. El error medio de DE es un poco mayor, debido a que sobrestimó un poco los valores de Phit, sin embargo GSA presentó 11 valores extremos mientras DE tuvo 5. No obstante la dispersión es similar en ambos métodos y aunque en ambos casos hubo puntos con errores mayores a 0.1, el 75% de todos los errores se mantuvo por debajo de 0.05.

La figura 5-34 muestra la cercanía de las componentes teóricas y resultado de la simulación de la función objetivo.

Las simulaciones obtenidas respetan la distribución univariada de los datos (figura 5-35). Las distribuciones ajustadas fueron de tipo *Logística* con parámetros ($\mu = 0.1653, \beta = 0.0193$)



(a) Histograma y gráfico de caja para los simulados por DE (arriba) y GSA (abajo).

(b) Tabla de la estadística descriptiva de los errores.

Estadística	Error	
	(DE)	(GSA)
Tamaño	346.0	346.0
Mínimo	-0.086286	-0.132948
Máximo	0.127134	0.111246
Rango	0.213420	0.244194
Media	-0.000033	0.000009
Mediana	0.000000	0.000000
Primer cuantil	-0.023383	-0.021829
Tercer cuantil	0.022142	0.022174
Rango intercuantil	0.045525	0.044003
Varianza	0.001358	0.001350
Desviación estándar	0.036849	0.036743
Simetría	0.267031	-0.042717
Curtosis	0.065946	0.903701

Figura 5-33: Estadística descriptiva del error en la simulación condicional para el pozo *Lakach-1*.

para ambos métodos. Se calcularon también las distribuciones empíricas de la simulación y los datos obteniéndose un buen ajuste entre ambas (figura 5-36).

Las simulaciones condicional obtenidas respetan la dependencia bivariada de los datos (figura 5-37). La figura 5-38 muestra las funciones de distribución y densidad para las simulaciones. Las cópulas de ajuste fueron de tipo *Frank*, con parámetro $\delta = -4.25$ para DE y $\delta = -4.22$ para GSA.

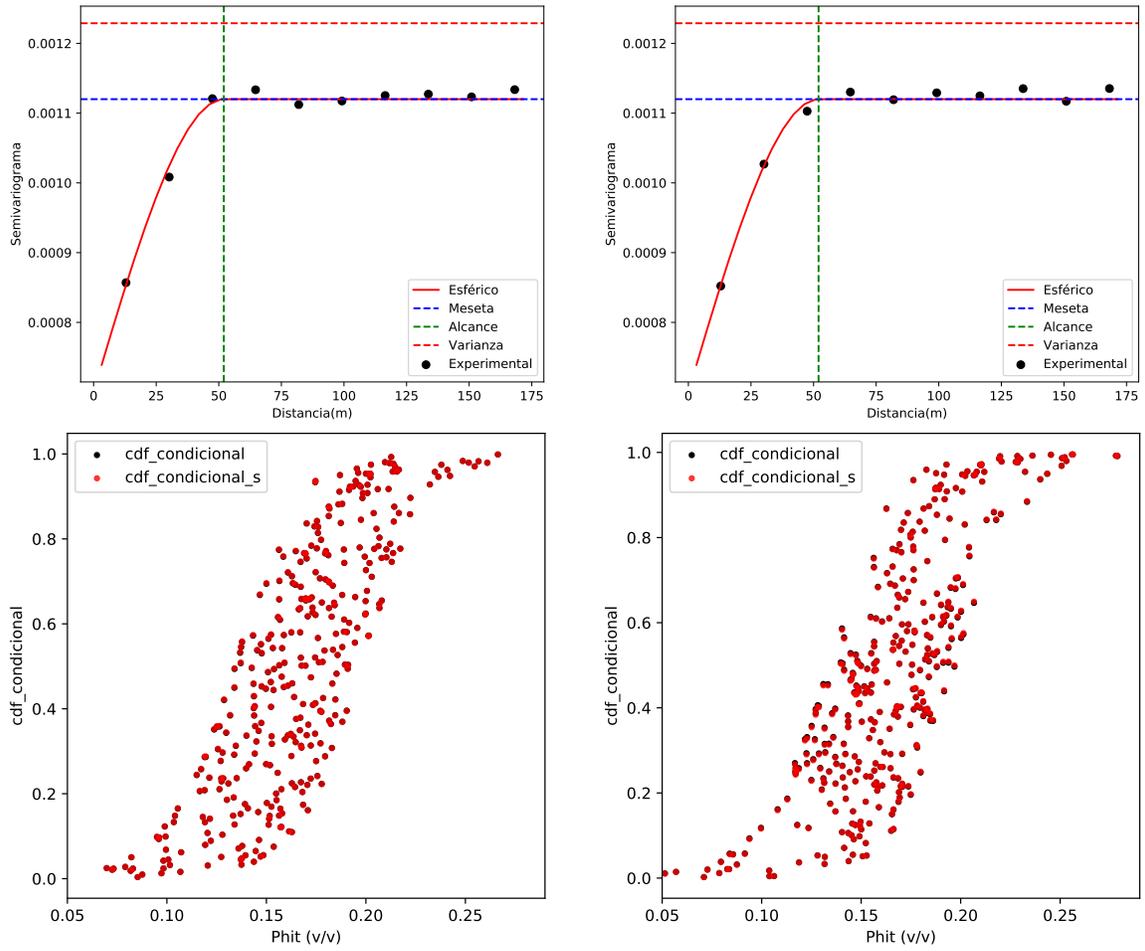
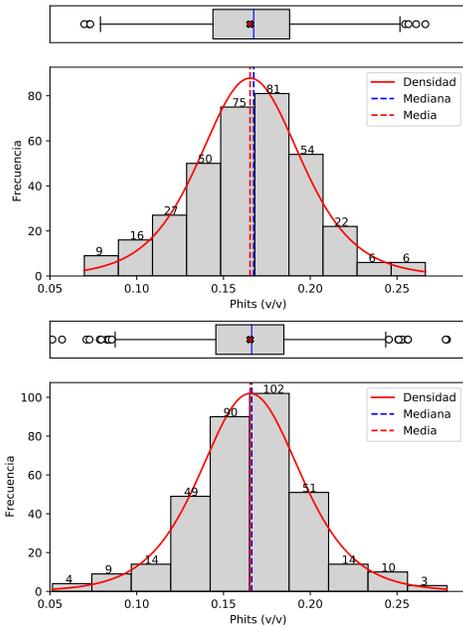


Figura 5-34: Variogramas y función de distribución condicional de la simulación condicional para el pozo *Lakach-1* (DE a la izquierda, GSA a la derecha).



(a) Histograma y gráfico de caja para los simulados por DE (arriba) y GSA (abajo).

(b) Tabla de la estadística descriptiva.

Estadística	Phit(v/v)	Datos	(DE)	(GSA)
Tamaño		346.0	346.0	346.0
Mínimo		0.0493	0.069773	0.051371
Máximo		0.2857	0.266334	0.278792
Rango		0.2364	0.196561	0.227421
Media		0.165352	0.165319	0.165360
Mediana		0.1444	0.167403	0.166233
Primer cuantil		0.27516	0.143918	0.145581
Tercer cuantil		0.1896	0.188110	0.184780
Rango intercuantil		0.0452	0.044192	0.039199
Varianza		0.001229	0.001230	0.001220
Desviación estándar		0.035055	0.035067	0.034928
Simetría		-0.43234	-0.117859	0.021226
Curtosis		0.542704	0.244534	1.055098

Figura 5-35: Estadística descriptiva de la simulación para el pozo *Lakach-1*.

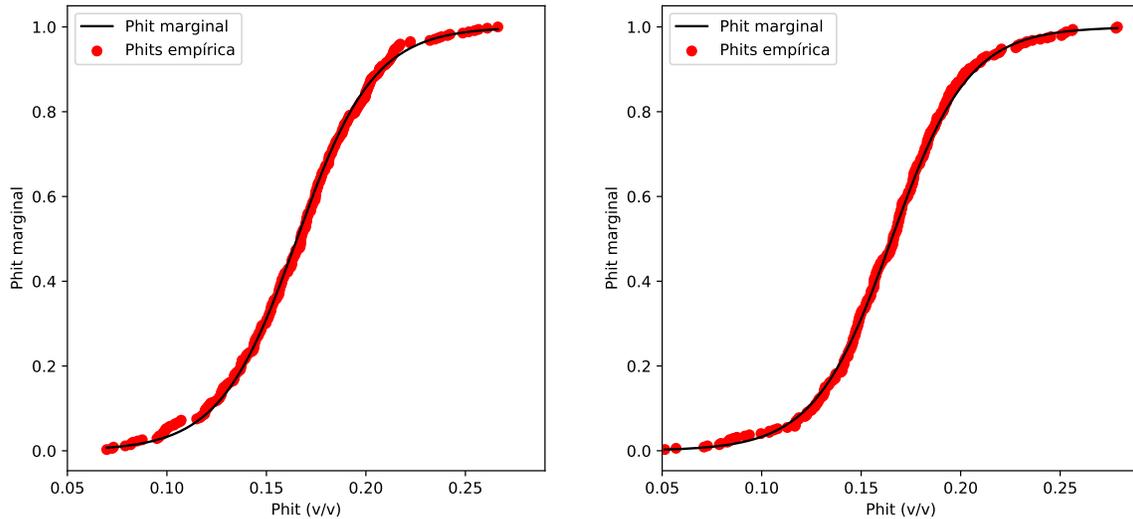


Figura 5-36: Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).

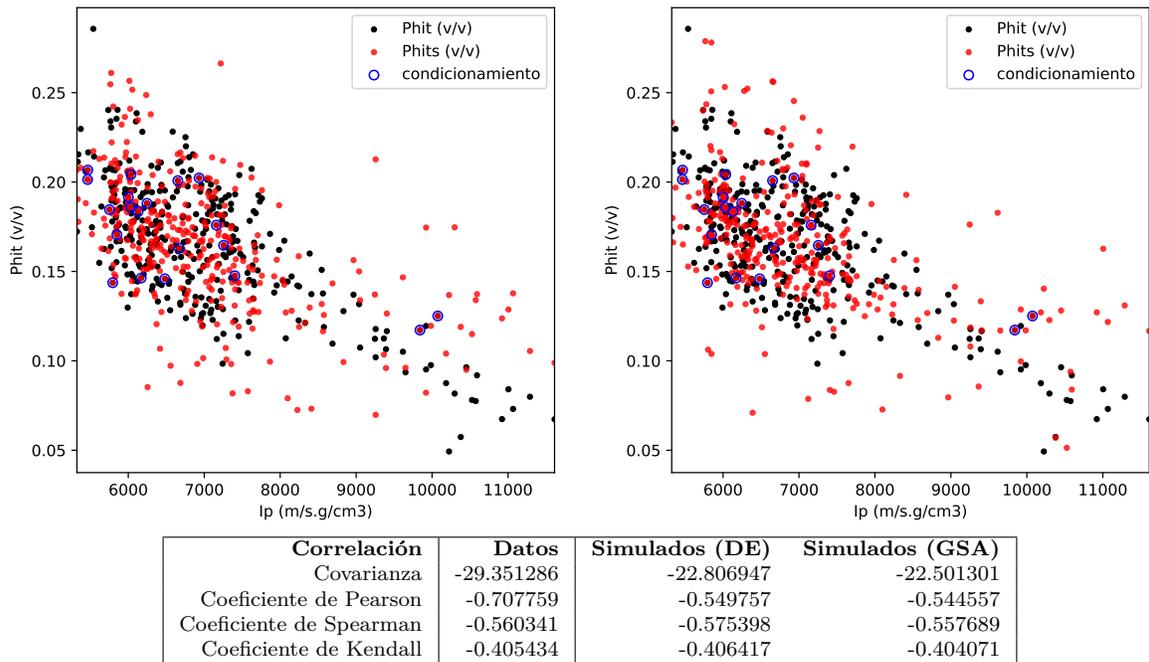


Figura 5-37: Gráfico de dispersión de I_p contra $Phits$ (DE a la izquierda, GSA a la derecha) y estadística descriptiva bivariada de los datos simulados.

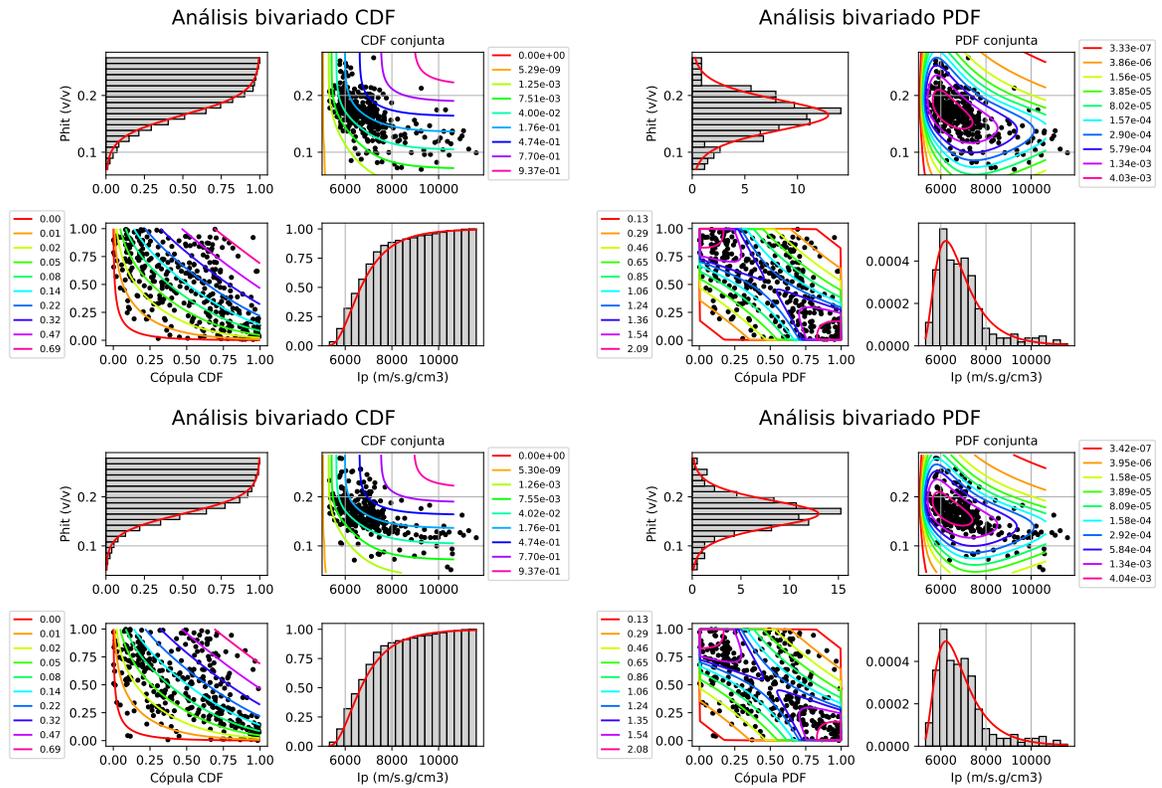


Figura 5-38: Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de I_p y $Phit$ respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.

De acuerdo a los resultados obtenidos (ver tabla 5-6) no se observó ninguna mejora con datos de condicionamiento en este caso de estudio. También se muestra una pérdida de eficiencia al considerar condicionamiento.

Tabla 5-6: Resumen de la comparación considerando o no puntos de condicionamiento.

Parámetros		No condicional		Condicional	
		DE	GSA	DE	GSA
FMO	\mathcal{O}_1	0.0009563	0.0004837	0.0007126	0.000676
	\mathcal{O}_2	0.0000296	0.0001981	0.0000889	0.0003229
	\mathcal{O}	0.0006817	0.0009858	0.0008017	0.000999
Desempeño	Niter	3169	40	2964	73
	Tiempo (s)	1557.04	673.66	1684.99	1132.44
Error	Media	-0.000005	0.000016	-0.000033	0.000009
	Desviación estándar	0.037932	0.040578	0.036849	0.036743
	Rango	0.203332	0.239770	0.213420	0.244194

5.2.1.6. Conclusión parcial

Para este primer caso de estudio se implementaron diferentes variantes del método de cosimulación. Al comparar con los resultados de SA implementado en la literatura, de forma general, se puede afirmar que nuestra implementación dio mejores resultados, en especial en lo que respecta a la distribución de Phit condicionada respecto a Ip. La variante híbrida implementada mejora la eficiencia y permitió un mejor ajuste de las distribuciones univariadas y bivariadas. Sin embargo, esta mejora no fue muy notable por lo que se descarta su utilización posterior. No se observó ninguna mejora cuando se consideran datos de condicionamiento, observándose una pérdida de eficiencia en dicha variante.

Todas las variantes propuestas permitieron simular de forma satisfactoria la porosidad total a partir de la impedancia acústica, respetándose la dependencia espacial y la dependencia entre las variables. Para este caso de estudio se puede afirmar la consistencia entre las variantes propuestas, lo cual demuestra flexibilidad en la metodología e implementación. Para los demás casos de estudio solo se utilizará la variante de cosimulación paramétrica sin considerar condicionamiento.

5.2.2. Caso de estudio II

Como segundo caso de estudio se tomó un pozo ubicado en el Mar del Norte (bloque Alemán) y fue obtenido de un repositorio público del TNO - Geological Survey of the Netherlands (<https://www.nlog.nl/data>). La producción de hidrocarburo está relacionada con un depósito de arenas finas correspondiente a las formaciones Graben superior del Grupo Scruff y Graben medio e inferior del Grupo Graben Central (Guerra Guerra, 2013). Se eligió el pozo *F03-FB-102*, puesto que presentaba toda la información necesaria para nuestro propósito. Este, junto a otros pozos del Bloque F03 pertenecen a un grupo de pozos de prueba del software OpendTect. Se escogió el intervalo (3190 – 3340m) perteneciente a la formación Graben inferior. En la figura 5-39 se muestra el registro del pozo escogido.

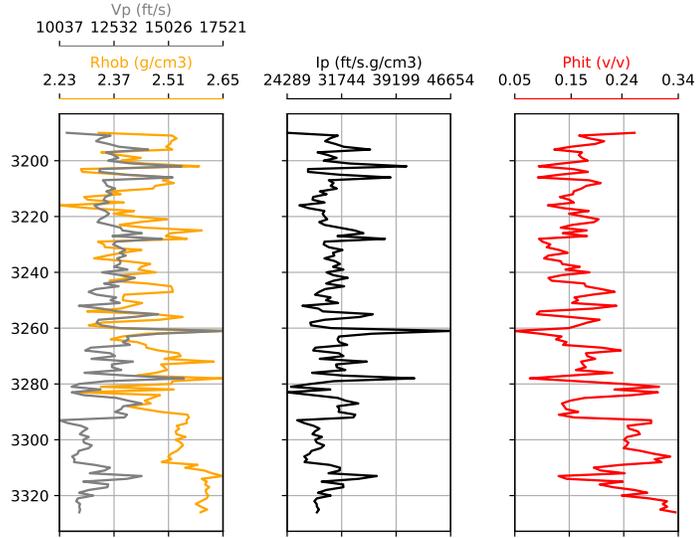


Figura 5-39: Registro del pozo *F03-FB-102*.

Para este caso de estudio se realizará un análisis y una simulación análogos al realizado con el Caso de estudio I, lo cual se presenta a continuación.

5.2.2.1. Análisis estadístico y variográfico de los datos

Análisis estadístico univariado: se realizó un ajuste de las distribuciones marginales de tipo *Gumbel* con parámetros ($\beta = 2458.48; \gamma = 28953.5$) para I_p ; y ($\beta = 0.0489963; \gamma = 0.156505$) para $Phit$.

Variografía: se detectó que los datos estaban afectados por una tendencia con respecto a la profundidad. En particular, para la variable primaria dicha tendencia pudo ser aproximada mediante la ecuación lineal:

$$\Phi^* = 0.0007312z - 2.793.$$

Luego de eliminar dicha tendencia se obtuvo un modelo teórico de variograma de tipo gaussiano con una meseta de 0.0035 y un alcance de 55m. Se registró, además, un efecto pepita de 0.002. Si bien este ajuste se consideró pobre, se considera suficiente para emplear la metodología de cosimulación estocástica espacial.

Análisis bivariado utilizando cópulas: para relacionar la distribución de ambas variables de manera conjunta se utilizó una cópula de *Frank* con parámetro $\delta = -5.21511$.

En el apéndice E.2 se muestran otros detalles del presente análisis.

5.2.2.2. Cosimulación con cópulas paramétricas

Las simulaciones obtenidas mediante DE y GSA poseen un valor de función objetivo de 0.0008754 y 0.0009942, respectivamente, deteniéndose ambos algoritmos al obtener un valor

menor que la precisión fijada. Ambas optimizaciones se consideraron satisfactorias. Para obtener este resultado DE realizó el cálculo de 1719 generaciones para un tiempo de ejecución de 363.72s (aproximadamente 6min). El tiempo de ejecución de GSA fue un poco menor, tomándole 229.04s (aproximadamente 4min) realizando 75 iteraciones. Para este problema tanto DE como GSA mostraron ser efectivos y GSA fue un poco más eficiente computacionalmente.

En el apéndice E.2 se muestran los resultados de las simulaciones obtenidas a partir del pozo escogido. La figura 5-40 muestra el registro de la simulación. Se puede observar que ambas porosidades simuladas siguen el comportamiento espacial de los datos.

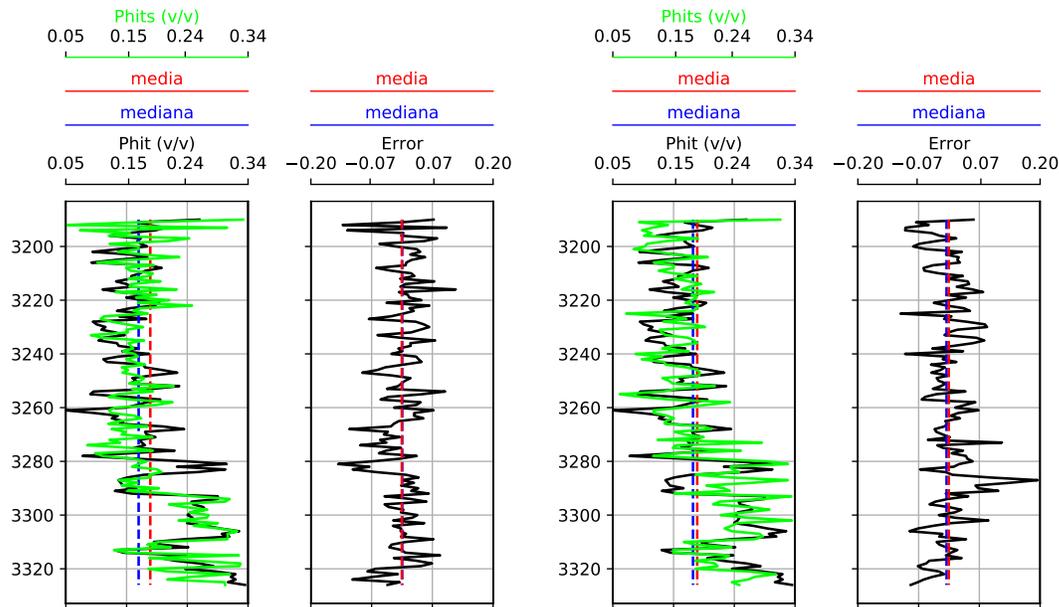


Figura 5-40: Registros de los datos y simulados de la Phit y el error de estimación, para el pozo *F03-FB-102* (DE a la izquierda, GSA a la derecha).

Al analizar el error de estimación podemos afirmar que ambos métodos presentan una dispersión similar aunque la media del error de GSA es un poco mayor. DE subestimó ligeramente los valores de Phit mientras que GSA lo sobrestimó, y aunque en ambos casos hubo puntos con errores mayores a 0.1, el 75% de ellos se mantuvo por debajo de 0.06. Para más detalles sobre el comportamiento de los errores se puede ver el apéndice E.

Se presenta un buen ajuste entre las componentes teóricas y experimentales de las funciones objetivos. En este caso el variograma experimental sin tendencia se calculó de acuerdo a lo expuesto en la sección 2.2. Las simulaciones obtenidas respetan la distribución univariada de los datos. Las distribuciones marginales estimadas para Phits fueron $Gumbel(\beta = 0.0485146, \gamma = 0.156411)$ para DE y $Gumbel(\beta = 0.0483109, \gamma = 0.156405)$ para GSA. Se calcularon también las distribuciones empíricas de la simulación y los datos obteniéndose un buen ajuste entre ambas. Las simulaciones obtenidas respetan la dependencia bivariada de los datos. Las cópulas de ajuste fueron de tipo *Frank*, con parámetros $\delta = -5.14096$ para DE y $\delta = -5.10947$ para GSA.

5.2.2.3. Conclusión parcial

Para este problema fue posible obtener simulaciones estocásticas de Phit a partir de Ip utilizando tanto DE como GSA. Este último fue más eficiente computacionalmente, aunque las diferencias no son tan notorias. El comportamiento de ambos algoritmos fue similar. Los variogramas ajustan de manera casi exacta, sobre todo para los valores menores de h , respetándose la dependencia espacial esperada ($\mathcal{O}_1 = 0.0004041$ para DE, $\mathcal{O}_1 = 0.0001672$ para GSA). La diferencia entre las distribuciones condicionales también fue muy pequeña, lo que demuestra que se mantuvo la dependencia entre las variables ($\mathcal{O}_2 = 0.0004712$ para DE, $\mathcal{O}_2 = 0.000827$ para GSA), esta dependencia también está avalada por la distribución bivariada y la proximidad de los coeficientes de correlación.

5.2.3. Caso de estudio III

Por último estamos interesados en tomar como caso de estudio a un grupo de pozos ubicado en el Mar del Norte (bloque Noruego) que fueron obtenidos de un repositorio que acompaña a Avseth *et al.* (2010). La producción de hidrocarburo se encuentra relacionada con un depósito arenoso de la formación Heimdal. Estos datos fueron ofrecidos por el autor de forma pública en <http://publishing.cambridge.org/resources/0521816017>. Se conforma de varios pozos de prueba de los cuales se escogió el *Pozo 2* para realizar la validación y obtener información a priori del área. Se escogió el intervalo (2154 – 2450m) recomendado en Avseth *et al.* (2010). En el capítulo siguiente se verá la utilización de esta información aplicando nuestra metodología. En la figura 5-41 se muestra el registro del pozo escogido.

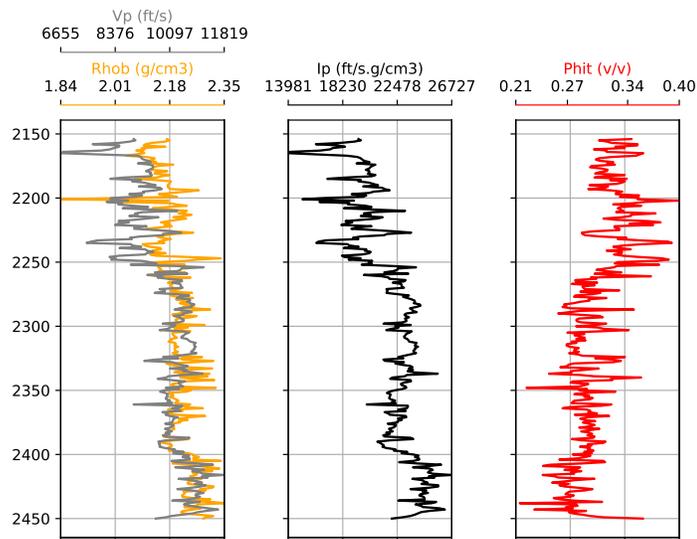


Figura 5-41: Registro del *Pozo 2*.

Para este caso de estudio se realizará un análisis y una simulación análogos al realizado con el Caso de estudio I, lo cual se presenta a continuación.

5.2.3.1. Análisis estadístico y variográfico de los datos

Análisis estadístico univariado: se realizó un ajuste de las distribuciones marginales de tipo $Normal(\mu = 21929.5; \sigma = 2251.15)$ para Ip; y $LogNormal(\mu_l = -1.49583; \sigma_l = 0.134007; \gamma = 0.0770454)$ para Phit.

Variografía: se supondrá que la variable es estacionaria ya que la media local y la varianza no parecen depender de las coordenadas. El modelo teórico de variograma fue esférico con una meseta de 0.00093 y un alcance de 110m. Se registró, además, un efecto pepita de 0.0004.

Análisis bivariada utilizando cópulas: para relacionar la distribución de ambas variables de manera conjunta se utilizó una cópula de *Frank* con parámetro $\delta = -7.83752$.

En el apéndice E.3 se muestran otros detalles del presente análisis.

5.2.3.2. Cosimulación con cópulas paramétricas

Las simulaciones obtenidas poseen un valor de 0.0031678 y 0.0009973 al ser evaluada en la función objetivo para DE y GSA, respectivamente. Deteniéndose el segundo algoritmo cuando este valor resultó menor que la precisión fijada, aunque en ambos casos se considera una optimización satisfactoria. El tiempo de ejecución de DE fue de 1677.64s (aproximadamente 28min) tras 4000 generaciones, mientras que GSA demoró 441.75s (aproximadamente 7min) realizando 40 iteraciones. Para este problema GSA fue más efectivo y eficiente que DE.

En el apéndice E.3 se muestran los resultados de las simulaciones obtenidas a partir del pozo escogido. La figura 5-42 muestra el registro de la simulación. En general se puede observar que las porosidades simuladas siguen el comportamiento espacial de la variable de referencia.

Al analizar el error de la estimación se puede decir que tanto los errores medios como las dispersiones son similares en ambos métodos. En ambos casos hubo puntos con errores mayores a 0.1, el 75 % de ellos se mantuvo por debajo de 0.04. Para más detalles sobre el comportamiento de los errores se puede ver el apéndice E.

Se presenta un buen ajuste entre las componentes teóricas experimentales de las funciones objetivos. Las simulaciones obtenidas respetan la distribución univariada de los datos. Las distribuciones marginales estimadas para Phits fueron $LogNormal(\mu_l = -1.52246, \sigma_l = 0.137184, \gamma = 0.0829109)$ para DE y $LogNormal(\mu_l = -1.49397, \sigma_l = 0.133678, \gamma = 0.0766207)$ para GSA. Se calcularon también las distribuciones empíricas de la simulación y los datos obteniéndose un buen ajuste entre ambas. Las simulaciones obtenidas respetan la distribución bivariada de los datos. Las cópulas de ajuste fueron de tipo *Frank*, con parámetros $\delta = -7.80898$ para DE y $\delta = -7.83679$ para GSA.

5.2.3.3. Conclusión parcial

Para este problema fue posible simular Phit de forma efectiva a partir de los datos de Ip tanto con DE como con GSA. Este último se mostró más eficiente computacionalmente, aunque para ambos algoritmos se obtuvo un comportamiento similar de las simulaciones, respetándose la dependencia espacial de Phit ($\mathcal{O}_1 = 0.0013139$ para DE, $\mathcal{O}_1 = 0.0000277$ para GSA) y la dependencia entre las variables ($\mathcal{O}_2 = 0.0018539$ para DE, $\mathcal{O}_2 = 0.0009696$ para GSA).

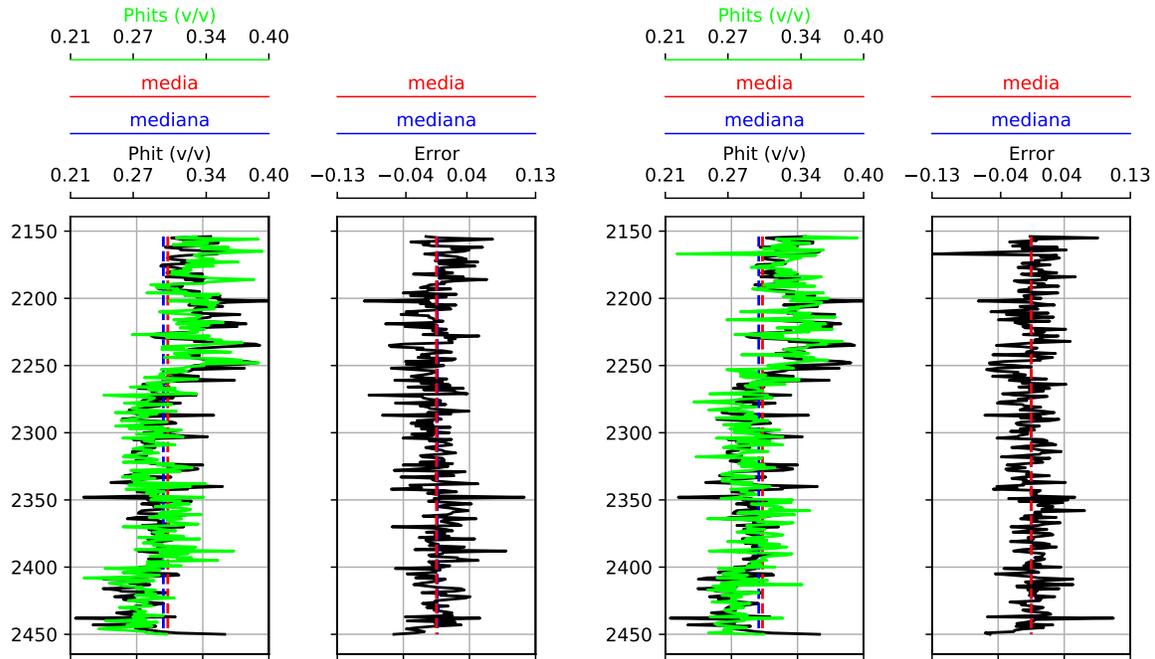


Figura 5-42: Registros de los datos y simulados de la Phit y el error de estimación, para el *Pozo 2* (DE a la izquierda, GSA a la derecha).

5.2.4. Discusión de casos

En esta sección hemos podido constatar la efectividad de la metodología de cosimulación propuesta para los tres casos de estudio. Nuestra implementación mostró ser más flexible que otras propuestas en la literatura permitiéndonos abordar casos con diferentes características geológicas y petrofísicas. En la tabla 5-7 se muestra una comparación de los resultados obtenidos.

Tabla 5-7: Resumen de la comparación de los tres casos de estudio.

Parámetros		Casos		I		II		III	
		DE	GSA	DE	GSA	DE	GSA		
Dimensión del problema		346		137		295			
FMO	\mathcal{O}_1	0.0009563	0.0004837	0.0004041	0.0001672	0.0018539	0.0009696		
	\mathcal{O}_2	0.0000296	0.0001981	0.0004712	0.000827	0.0013139	0.0000277		
	\mathcal{O}	0.0006817	0.0009858	0.0008754	0.0009942	0.0031678	0.0009973		
Condición de parada		precisión	precisión	precisión	precisión	máx. generaciones	precisión		
Desempeño	Niter	3169	40	1719	75	4000	40		
	Tiempo (s)	1557.04	673.66	363.72	229.04	1677.65	441.75		
Error	Media	-0.000005	0.000016	-0.000372	-0.000496	0.000015	-0.000019		
	Desviación estándar	0.037932	0.040578	0.047296	0.048520	0.028074	0.026496		
	Rango	0.203332	0.239770	0.258151	0.301163	0.210052	0.236623		

Los procesos de optimización se detuvieron cuando el valor de la función objetivo fue menor que la precisión fijada ($\mathcal{O} < 0.001$), salvo en la simulación mediante DE en el Caso de estudio III. Para las simulaciones resultantes los valores de la función objetivo \mathcal{O}_1 fueron mayores que los de \mathcal{O}_2 salvo una excepción. Esto quiere decir que la metodología permite de manera efectiva mantener la distribución condicional de los datos con un ligero sacrificio en la aproximación

del variograma. Si bien este comportamiento es deseable (ya que el variograma teórico es de forma natural mucho más suave que un variograma experimental), puede modificarse fácilmente cambiando el valor de los pesos de las funciones objetivo, dándole más importancia a una u otra.

No se puede establecer una relación directa entre el número de pasos y operaciones para ambos métodos, sin embargo, en todos los casos de estudio analizados GSA resultó más eficiente desde el punto de vista computacional que DE. Lo anterior quiere decir que a GSA le toma menos tiempo alcanzar iguales precisiones que DE, lo cual es consistente con los resultados de la sección 5.1 para funciones de prueba. Es importante destacar que el tiempo de cómputo es proporcional a la dimensión de los problemas para igual número de iteraciones, y la diferencia entre el tiempo de cómputo de ambos métodos se hace más evidente a medida que aumenta la dimensión del problema.

En general las porosidades simuladas siguieron el comportamiento espacial de los datos y, si bien no era objetivo de la simulación, los errores de estimación se mantuvieron en niveles bajos para los tres casos de estudio, siendo los de DE menores que los de GSA. Para ambos métodos las simulaciones obtenidas respetan la estadística univariada y bivariada, consiguiendo que los ajustes de las distribuciones de las variables simuladas fueran próximos a los de los datos.

Todo lo anterior nos permite valorar de manera positiva las simulaciones obtenidas. Los análisis realizados nos permiten concluir que tanto la metodología como la implementación realizada son consistentes y están listas para aplicarlas en cualquier caso de estudio. En el capítulo siguiente veremos los resultados de su aplicación.

Capítulo 6

Aplicación a un caso de estudio

Dada la importancia que entrañan las aplicaciones y por ser el objetivo final de la metodología se decidió dedicarle un capítulo independiente para hacer un análisis lo más detallado posible. Nos interesa específicamente obtener valores de Phit en puntos donde se desconoce el valor de dicha variable. Para ello necesitamos establecer un modelo de dependencia con respecto a Ip, basado en cópulas paramétricas, y un variograma teórico a partir de información a priori de la variable. Con estos datos se realiza una cosimulación espacial estocástica utilizando los métodos DE y GSA propuestos.

Se seleccionó un pozo como caso de estudio para esta aplicación. Para establecer la dependencia y el variograma se usará información externa, en este caso de un pozo cercano, aunque podría provenir de otras fuentes.

Otras posibles extensiones de dicha aplicación es utilizando la información de una traza sísmica (2D) o un cubo sísmico (3D), y combinado la información disponible de pozos y núcleos. Estas aplicaciones no pudieron hacerse porque no se contaba con toda la información necesaria.

6.1. Caso de estudio IV

Dentro del grupo de pozos del Mar del Norte (bloque Noruego), analizado en el capítulo anterior, se eligió el *Pozo 3* para aplicar nuestra metodología. Esto se hace posible pues posee registros de densidad (Rhob) y velocidad de la onda P (Vp) con los cuales es posible calcular la impedancia acústica (Ip). Para establecer las relaciones de dependencia bivariada y espacial de Phit se supuso que estas serían equivalentes a las obtenidas para el *Pozo 2*, analizado con anterioridad. Esta suposición es razonable, ya que se trata de los dos pozos más cercanos del grupo de datos (2.9km de distancia) y se encuentran en el mismo subcubo sísmico propuesto por Avseth *et al.* (2010). En la figura 6-1 se muestra el registro del pozo.

6.1.1. Análisis estadístico y variográfico de los datos

Análisis estadístico univariado: en la figura 6-2 se muestra la estadística univariada de Ip del pozo. Se realizó un ajuste de la distribución marginal de tipo *Logística*($\mu = 21644.3; \beta = 1373.27$) para Ip. Para Phit se tomó la distribución del *Pozo 2* (sección 5.2.3), es decir, una *LogNormal*($\mu_l = -1.49583; \sigma_l = 0.134007; \gamma = 0.0770454$).

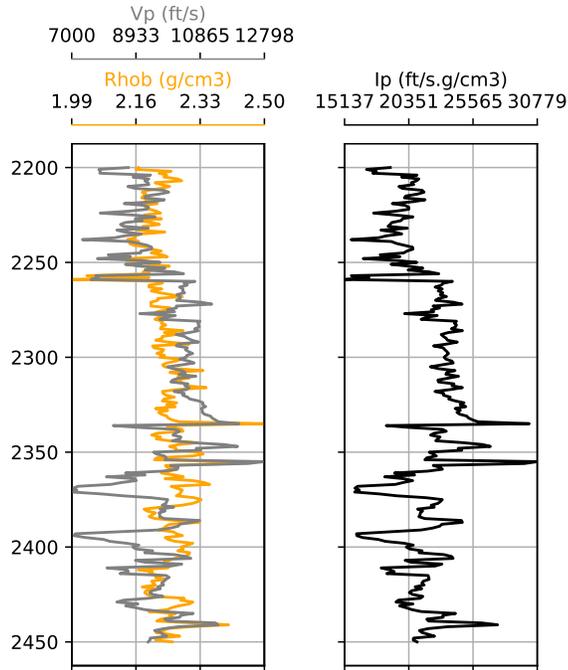


Figura 6-1: Registro del *Pozo 3*.

Variografía: el modelo teórico de variograma se extrajo del *Pozo 2* (véase sección 5.2.3), tomando así un modelo esférico con una meseta de 0.00093, alcance de 110m y efecto pepita de 0.0004. Se asumió que los datos estaban afectados por una tendencia equivalente a la tendencia detectada en I_p , escalando de forma proporcional en el rango de valores de Φ_{hit} . Esta tendencia se eliminó del modelo seleccionado mediante la ecuación lineal:

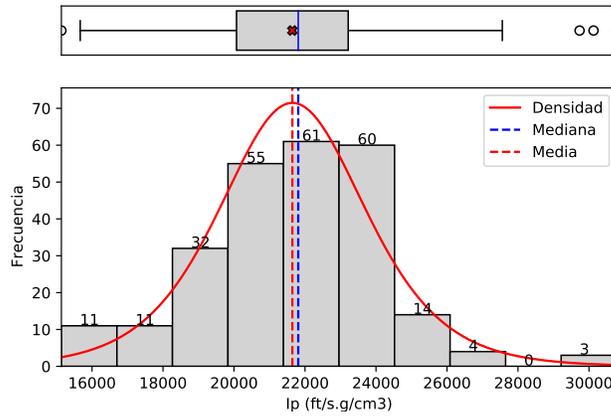
$$\Phi^* = 0.0002309z - 0.2337622.$$

Análisis bivariado utilizando cópulas: para establecer la dependencia entre las variables se utilizaron los datos del *Pozo 2* (véase sección 5.2.3). De esta forma elegimos una cópula de *Frank* con parámetro $\delta = -7.83752$.

6.1.2. Cosimulación con cópulas paramétricas

Las simulaciones obtenidas mediante DE y GSA poseen un valor de función objetivo de 0.0009882 y 0.0038747, respectivamente, deteniéndose DE al obtener un valor menor que la precisión fijada. Para obtener este resultado DE realizó el cálculo de 2484 generaciones demorando 865.58s (aproximadamente 14min). El tiempo de ejecución de GSA fue de 4047.07s (aproximadamente 67min) realizando las 500 iteraciones. Para este problema DE mostró ser mucho más efectivo que GSA.

En las figuras 6-3-6-8 se muestran los resultados de las simulaciones obtenidas mediante ambos métodos. La figura 6-3 muestra el registro de la simulación. Se puede observar que las simulaciones satisfacen una relación inversa respecto a la impedancia acústica. El comporta-



(a) Histograma y gráfico de caja para la impedancia acústica.

(b) Tabla de la estadística descriptiva.

Estadística	Ip(ft/s.g/cm ³)
Tamaño	251.0
Mínimo	15137.34
Máximo	30779.09
Rango	15641.75
Media	21644.27
Mediana	21809.34
Primer cuantil	20070.36
Tercer cuantil	23216.48
Rango intercuantil	3146.12
Varianza	6179503
Desviación estándar	2485.86
Simetría	0.0582
Curtosis	0.9716

Figura 6-2: Estadística descriptiva para el *Pozo 3*

miento de ambas porosidades es similar, aunque GSA muestra un poco más de dispersión.

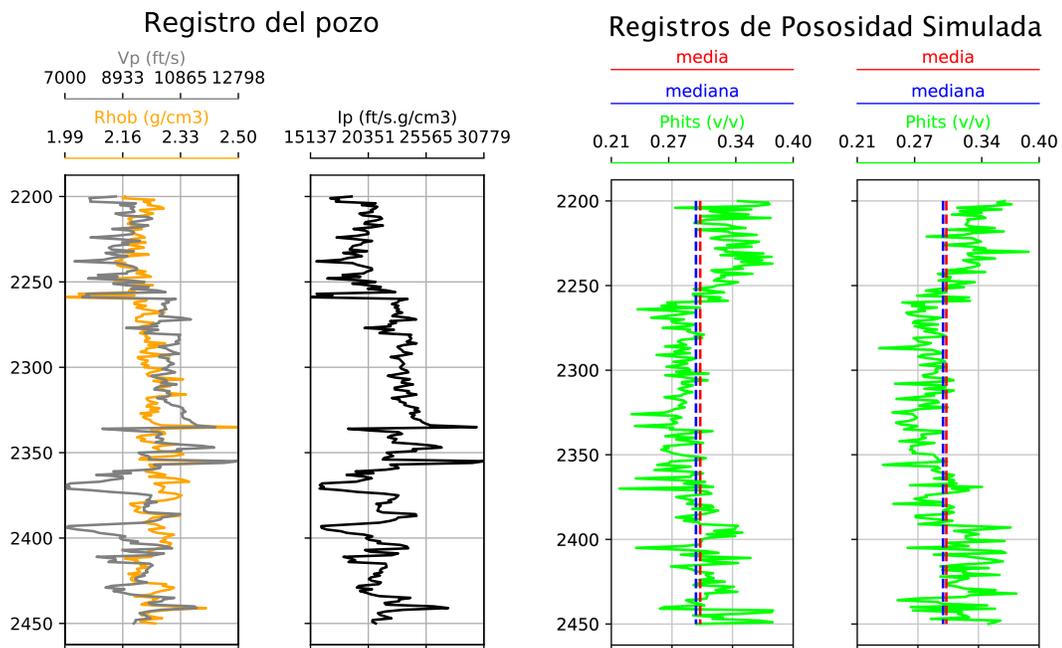


Figura 6-3: Registros de los datos de Vp, Rhob e Ip así como de Phit simulada para el *Pozo 3* (DE a la izquierda, GSA a la derecha).

La figura 6-4 muestra la cercanía de las componentes teóricas y resultado de la simulación de la función objetivo. El variograma experimental sin tendencia se calculó de acuerdo a lo expuesto en la sección 2.2.

Las distribuciones marginales estimadas para Phits fueron $LogNormal(\mu_l = -1.40151, \sigma_l = 0.120722, \gamma = 0.0548078)$ y $LogNormal(\mu_l = -1.57414, \sigma_l = 0.141858, \gamma = 0.0938357)$ para

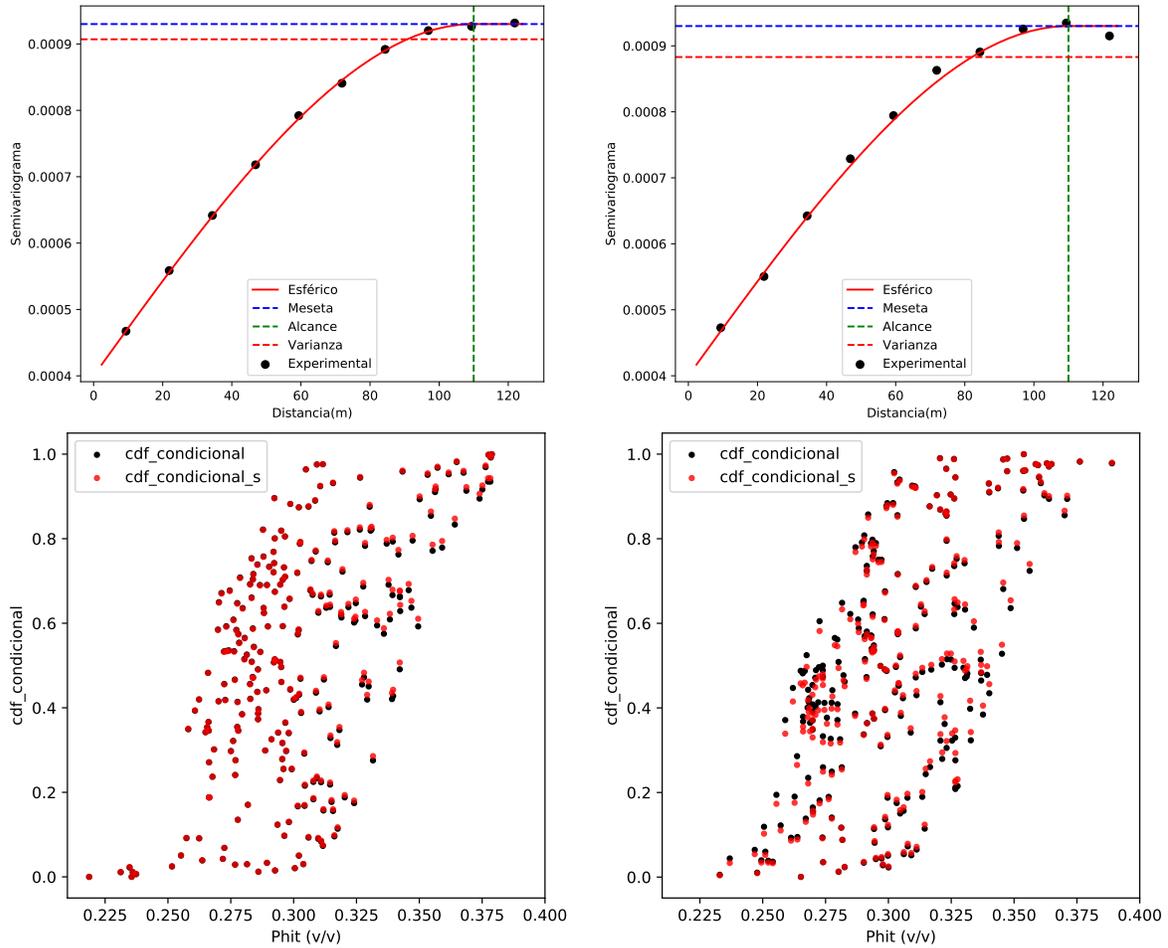
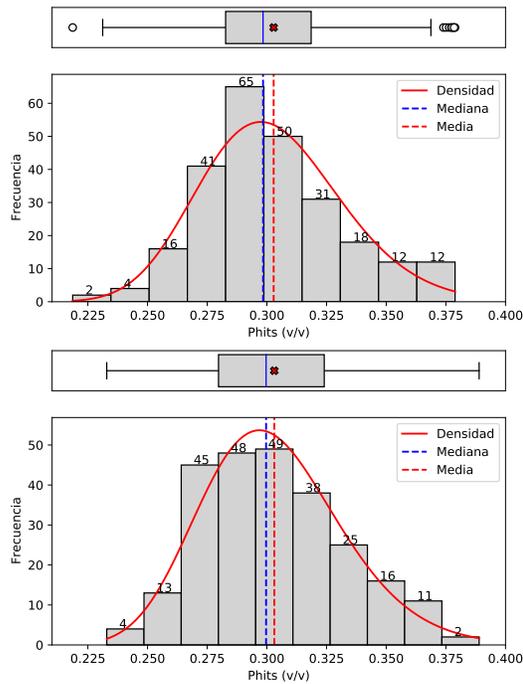


Figura 6-4: Variogramas y función de distribución condicional de la simulación para el *Pozo 3* (DE a la izquierda, GSA a la derecha).

las simulaciones obtenidas mediante DE y GSA, respectivamente. Se calcularon también las distribuciones empíricas de la simulación (figura 6-6).

Las simulaciones obtenidas respetan la dependencia bivariada que se tomó como modelo (figura 6-7). La figura 6-8 muestra las funciones de distribución y densidad para las simulaciones. Las cópulas de ajuste fueron de tipo *Frank*, con parámetros $\delta = -7.82$ para DE y $\delta = -7.646$ para GSA.



(a) Histograma y gráfico de caja para los simulados por DE (arriba) y GSA (abajo).

(b) Tabla de la estadística descriptiva.

Estadística	Phit(v/v)	(DE)	(GSA)
Tamaño		251.0	251.0
Mínimo		0.210022	0.232912
Máximo		0.218643	0.388934
Rango		0.378855	0.156022
Media		0.160212	0.303113
Mediana		0.302835	0.299670
Primer cuantil		0.298397	0.279744
Tercer cuantil		0.282721	0.324036
Rango intercuantil		0.318553	0.044292
Varianza		0.035832	0.000883
Desviación estándar		0.000907	0.029716
Simetría		0.030118	0.306184
Curtosis		0.416158	-0.383466

Figura 6-5: Estadística descriptiva de la simulación para el *Pozo 3*.

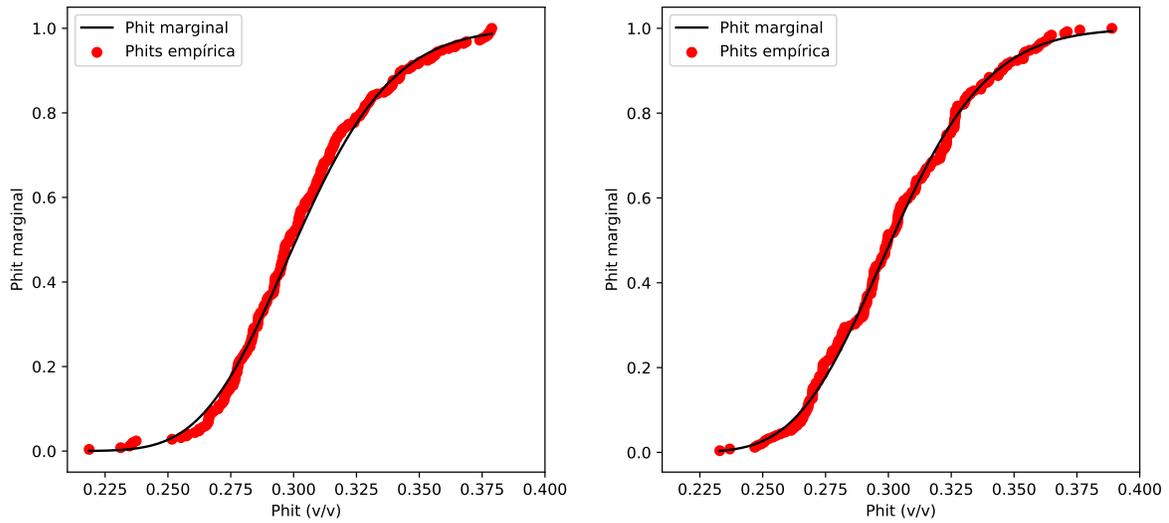
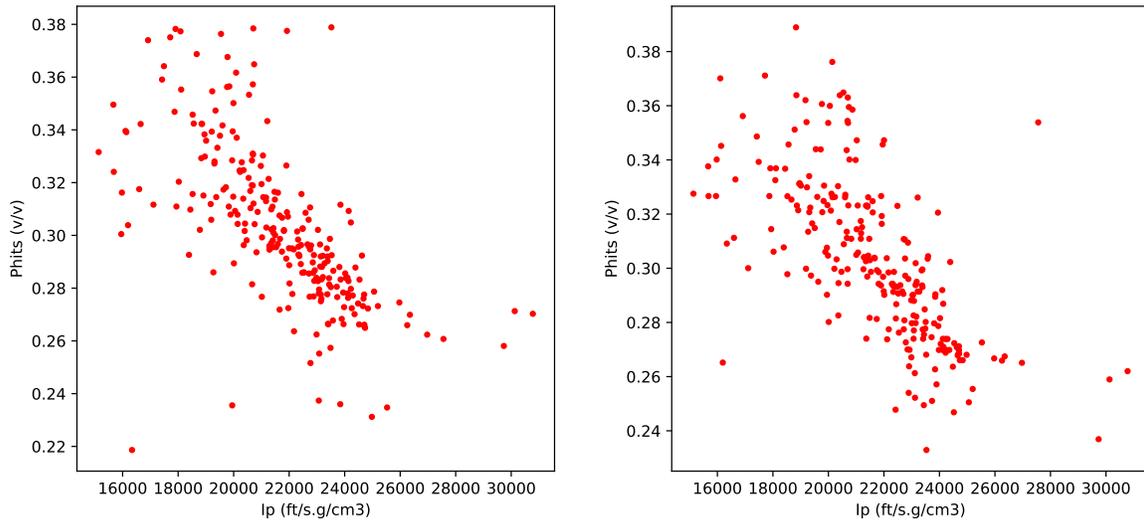


Figura 6-6: Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).



Correlación	Datos de referencia	Simulados (DE)	Simulados (GSA)
Covarianza	-52.561727	-48.995413	-50.262924
Coefficiente de Pearson	-0.763350	-0.651816	-0.67771
Coefficiente de Spearman	-0.792594	-0.776543	-0.771318
Coefficiente de Kendall	-0.596522	-0.595857	-0.5891

Figura 6-7: Gráfico de dispersión de I_p contra $Phits$ y estadística descriptiva bivariada de los datos simulados (DE a la izquierda, GSA a la derecha).

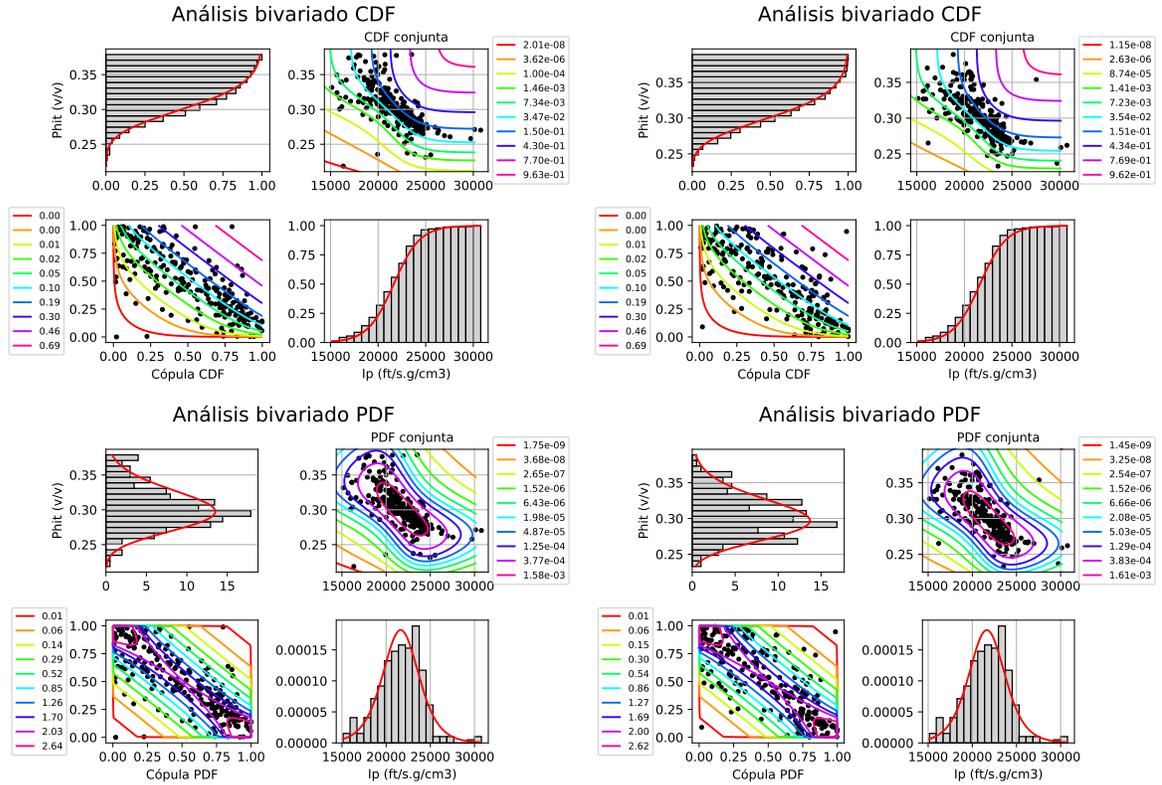


Figura 6-8: Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de Ip y Phits respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.

6.2. Discusión del caso de aplicación

Las simulaciones estocásticas obtenidas satisfacen una relación inversa respecto a la impedancia acústica, lo que da una medida de que respetan relaciones físicas empíricas con un comportamiento razonable. Los variogramas de las simulaciones ajustan de manera casi exacta con el variograma teórico propuesto, respetándose la dependencia espacial. La diferencia entre las distribuciones condicionales simuladas y propuesta también fueron muy pequeña, lo que demuestra que se mantuvo la dependencia entre las variables. Resulta interesante comprobar esta dependencia en las distribuciones bivariadas resultantes y los coeficientes de correlación.

Estos resultados son muy alentadores, pues demuestran que las simulaciones obtenidas con nuestra metodología cumplen todos los parámetros propuestos. Gracias a la flexibilidad en la implementación, es posible esperar que se obtengan buenos resultados también si se consideran parámetros adicionales.

Capítulo 7

Conclusiones y trabajos futuros

7.1. Conclusiones

En este trabajo se adoptó una metodología de cosimulación estocástica espacial utilizando métodos metaheurísticos de optimización global y modelos de dependencia basados en cópulas. La misma está basada en un análisis de dependencia espacial de la variable de interés y de dependencia entre dicha variable con una secundaria. Dicha metodología se basa en la optimización de una función multiobjetivo (FMO) que vincula la información espacial y de dependencia. Este enfoque permite resolver casos más complejos que los abordables por métodos clásicos ya que no se restringen por dependencia lineales y por las posibles distribuciones de las variables. El resultado de la optimización global es, entonces, una simulación de la variable de interés que respeta las dependencias de los datos.

En particular se eligió la porosidad total como variable primaria e impedancia acústica como variable secundaria se debió precisamente a la estrecha relación de dependencia entre ambas, y que en la mayoría de los casos establecer un modelo lineal de dependencia se aleja del comportamiento real de dichas variables. Esta metodología es aplicable a cualquier otro par de variables con dependencia no lineal. Para ayudar a la comprensión de los fundamentos teóricos de esta metodología se unificaron las notaciones, definiciones y se sistematizaron los resultados utilizados por los autores de las diferentes teorías implicadas.

El enfoque de FMO es independiente del método de optimización global a emplear. Es por esto que se estudiaron varios métodos de optimización global. En particular solo consideramos un grupo de métodos metaheurísticos, de los cuales se escogieron el método de recocido simulado generalizado (GSA) y el de evolución diferencial (DE) de acuerdo a la naturaleza de nuestro problema y tras haber analizado sus ventajas y desventajas.

Ambos métodos fueron implementados en el lenguaje de programación Python siendo un código abierto que permite la reutilización de la biblioteca implementada en otros programas. Dicha implementación se validó de forma preliminar mediante funciones de prueba, mostrando que DE, al menos para los problemas analizados, fue más preciso y GSA más eficiente. Se realizó la simulación en tres casos de estudio y en general ambos métodos resultaron efectivos, alcanzando la precisión requerida. Las porosidades simuladas en todos los casos siguieron el comportamiento espacial de los datos, respetan las dependencias espaciales y las dependencias entre las variables, además, los errores de estimación fueron bajos. Esto garantiza la validez de la metodología e implementación propuesta.

Resultó interesante notar que en la mayoría de los casos GSA fue más eficiente desde el punto de vista computacional que DE, apoyando lo observado en la validación preliminar. El tiempo de cómputo es proporcional a la dimensión de los problemas y la diferencia en eficiencia de ambos métodos es más evidente al aumentar la dimensión. Sin embargo el cálculo de los parámetros en GSA para cada problema, especialmente la temperatura inicial, resulta más complejo; es por esto que se recomienda utilizar DE en casos donde se necesite optimizar varios problemas con diferentes naturaleza y GSA para un único problema en donde se requiera realizar varias simulaciones.

La implementación realizada en este trabajo mostró ser más flexible que implementaciones anteriores y cubre algunas de las deficiencias encontradas. Esto nos permitió abordar casos con diferentes características geológicas y petrofísicas, pudiéndose considerar la posibilidad de usar datos de condicionamiento en los casos en que se encuentren disponibles.

También se aplicó la metodología con excelentes resultados para un caso de estudio donde se desconocían los valores de la porosidad total. Las simulaciones estocásticas obtenidas satisfacen una relación inversa respecto a la impedancia acústica y respetan la dependencia espacial y entre las variables. Los resultados obtenidos fueron satisfactorios a pesar de los limitados recursos computacionales, lo que también mostró ser otra gran ventaja de la implementación.

Conclusiones resumidas

- Se desarrolló metodología de cosimulación espacial con el empleo de varios métodos de optimización global y modelos de dependencia basados en cópulas paramétricas, cuyos resultados son comparables con los existentes en la literatura.
- De los métodos de optimización estudiados se escogieron el método de recocido simulado generalizado (GSA) y el de evolución diferencial (DE) pues fueron los que mejor se adaptaban a nuestro problema.
- La implementación realizada en Python tiene características tales como código abierto y reutilizable, flexibilidad, generalidad y aprovecha de una mejor manera las potencialidades de dicha metodología.
- Se realizó la simulación en tres casos de estudio y en general ambos métodos resultaron efectivos validándose así la metodología propuesta. Sin embargo se recomienda utilizar DE en casos donde se necesite optimizar varios problemas con diferentes naturaleza y GSA para un único problema en donde se requiera realizar varias simulaciones.
- La porosidad total simulada en un cuarto caso de estudio respeta la dependencia espacial y entre las variables de los datos de referencia, por lo cual la simulación se considera satisfactoria.

7.2. Trabajos futuros

Esta investigación puede ser extendida en dos líneas de trabajo fundamentales:

1. Utilizar modelos de distribución bivariado no paramétricos: la implementación actual admite dicha variante, pero la principal dificultad en este caso son los limitados recursos computacionales que impidieron su utilización.

2. Se puede generalizar a modelos de dependencia multivariados.
3. Extender los resultados a problemas con dos o tres dimensiones: si bien la implementación actual permite considerar más dimensiones, se consideró suficiente el caso unidimensional para validar la metodología y comparar los métodos de optimización.

Apéndices

Apéndice A

Simulación estocástica

A.1. Fundamentos teóricos

A continuación se presentan algunas definiciones fundamentales de la teoría de probabilidades y geoestadística que se utilizan en la simulación estocástica.

Un *espacio de probabilidad* $(\Omega, \mathcal{F}, \mathbb{P})$ está definido por: un espacio muestral o conjunto de estado Ω , un conjunto de sucesos aleatorios \mathcal{F} con estructura de σ -álgebra que es la combinación de todos los subconjuntos que se pueden formar con Ω y una medida de probabilidad \mathbb{P} asociada a la frecuencia de ocurrencia de los sucesos y está entre 0 y 1. Los eventos con probabilidad positiva se suelen llamar *probables*, aunque no significa que los eventos de probabilidad nula sean imposibles.

Dado un espacio de probabilidad, llamaremos *variable aleatoria* a cada una de las funciones reales medibles en dicho espacio ($X : \Omega \rightarrow \mathbb{R}$). Un *vector aleatorio* $X = (X_1, X_2, \dots, X_d)$ está formado por variables aleatorias.

Llamaremos *función de distribución* $F_X : \mathbb{R}^d \rightarrow [0, 1]$ del un vector aleatorio X a

$$F_{\mathbf{X}}(x_1, x_2, \dots, x_d) = \mathbb{P} \left(\bigcap_{i=1}^d \{X_i \leq x_i\} \right). \quad (\text{A-1})$$

Para un conjunto de índices $\sigma = (j_1, \dots, j_m) \subset (1, \dots, d)$ llamaremos σ -*marginal* de F a la función obtenida igualando a $+\infty$ las variables con los índices restantes:

$$F_{\sigma}(x_1, \dots, x_m) = F(y_1, \dots, y_d) \quad (\text{A-2})$$

con $y_{j_i} = x_i$ y $y_i \notin \sigma = +\infty$. A las marginales 1-dimensionales ($\sigma = j$) se les conoce simplemente como marginales asociadas a la variable aleatoria (X_j).

Un vector aleatorio \mathbf{X} y la función de distribución asociada $F_{\mathbf{X}}$ se dicen *absolutamente continuos* si existe una función medible $f_X : \mathbb{R}^d \rightarrow \mathbb{R}$ tal que

$$F_{\mathbf{X}}(\mathbf{x}) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} \dots \int_{-\infty}^{x_d} f_X(t_1, t_2, \dots, t_d) dt_1 dt_2 \dots dt_d. \quad (\text{A-3})$$

A f_X se la llama *densidad de probabilidad* del vector aleatorio \mathbf{X} y coincide con (2-2) para una variable aleatoria.

Dado un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$, si $A, B \in \mathcal{F}$ y B es probable ($\mathbb{P}(B) > 0$) entonces la *probabilidad condicional* del evento A dado el evento B se define como:

$$\mathbb{P}_B(A) = \mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}. \quad (\text{A-4})$$

La probabilidad condicional es también una medida de probabilidad que sustituye (o actualiza) a la medida original P , dado el hecho de que ha ocurrido el evento B (Erdely Ruiz y Gutiérrez Peña, 2012), definiendo un nuevo espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P}_B)$.

Dos sucesos son *independientes* si la probabilidad de que ocurran ambos simultáneamente es el producto de sus probabilidades, es decir

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B). \quad (\text{A-5})$$

En particular, si $\mathbb{P}(B) > 0$ entonces $\mathbb{P}_B(A) = \mathbb{P}(A)$. Dos vectores aleatorios $X = (X_1, \dots, X_m)$ e $Y = (Y_1, \dots, Y_n)$ se dicen *independientes* (se denota $X \perp Y$) si

$$F_{X,Y}(x, y) = F_X(x) \cdot F_Y(y) \forall x, y \quad (\text{A-6})$$

donde F_X y F_Y son las funciones de distribución de X y Y respectivamente y $F_{X,Y}$ es la función de distribución del vector $(X_1, \dots, X_m, Y_1, \dots, Y_n)$.

Partiendo del concepto de probabilidad condicional se define la *función de distribución condicional*, de la siguiente manera:

$$F_{X_j|X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_d}(x_j) = \mathbb{P}(X_j \leq x_j | X_1 = x_1, \dots, X_{j-1} = x_{j-1}, X_{j+1} = x_{j+1}, \dots, X_d = x_d). \quad (\text{A-7})$$

Para $d = 2$ la ecuación anterior es simplemente (2-21).

Si $X = (X_1, X_2, \dots, X_d)$ son todas variables aleatorias continuas y sus respectivas densidades son f_1, \dots, f_d , entonces las funciones de distribuciones condicionales se pueden calcular mediante límites como:

$$F_{S_1|k}(x_{S_1}|x_k) = \lim_{\epsilon \rightarrow 0^+} \frac{\mathbb{P}(X_j \leq x_j, j \in S_1, x_k \leq X_k < x_k + \epsilon)}{\mathbb{P}(x_k \leq X_k < x_k + \epsilon)} = \frac{\partial F_{S_1 \cup \{k\}} / \partial x_k}{f_k(x_k)}. \quad (\text{A-8})$$

Para $d = 2$ esto se reduce a la ecuación (2-22).

A.1.1. Funciones aleatorias

Fijado un espacio de probabilidad, se entiende como *función aleatoria* al conjunto de variables aleatorias espacialmente distribuidas, mientras que una *variable aleatoria regionalizada* será una realización o muestra de esta función aleatoria (Díaz Viera, 2002). En este sentido las observaciones son realizaciones que provienen de una función aleatoria, y de esta función aleatoria es posible obtener realizaciones diferentes.

Se dice que $Z(\mathbf{x})$ es estrictamente o *fuertemente estacionaria* si la función de distribución de probabilidades de las $Z(\mathbf{x}_i)$ son iguales entre sí e independientes. Esto requiere que los momentos de distinto orden sean completamente independientes del punto \mathbf{x}_i . Esta condición como su nombre lo indica es demasiado restrictiva al estudiar la mayoría de los fenómenos

encontrados en la práctica.

Se dice que $Z(\mathbf{x})$ es estacionaria de segundo orden o *devilmente estacionaria* si

1. $E(Z(\mathbf{x}_i)) = m$ es constante y no depende del punto \mathbf{x}_i .
2. $\text{Cov}(Z(\mathbf{x}_i) - Z(\mathbf{x}_i + h))$ exista y sólo dependa de la longitud h del vector de desplazamiento:

$$C(h) = \text{Cov}[Z(\mathbf{x}_i), Z(\mathbf{x}_j)] = E\{Z(\mathbf{x}_i), Z(\mathbf{x}_i + h)\} - m^2.$$

En particular si $Z(\mathbf{x})$ es devilmente estacionaria $\text{var}[Z(\mathbf{x}_i)] = E\{[Z(\mathbf{x}_i) - m]^2\} = C(0)$.

Una función aleatoria $Z(\mathbf{x})$ se dice intrínseca, o que cumple la *hipótesis intrínseca*, cuando:

1. $E(Z(\mathbf{x}_i)) = m$ es constante y no depende del punto X_i .
2. Todo vector de incremento $[Z(\mathbf{x}_i + h) - Z(\mathbf{x}_i)]$ tiene varianza finita y no depende de la localización \mathbf{x}_i , es decir $\text{var}[Z(\mathbf{x}_i) - Z(\mathbf{x}_i + h)] = \gamma(h)$ está bien definida. Cuando se cumple esta condición se dice que la varianza es homogénea.

La estacionaridad débil implica la condición intrínseca (homogeneidad), sin embargo la inversa no siempre se cumple.

La *simulación estocástica* consiste en que dada una función aleatoria $Z(\mathbf{x})$ se obtengan nuevas realizaciones o muestra de la misma que denotaremos por $Z^*(\mathbf{x})$, con las mismas propiedades estadísticas que la función aleatoria. En la práctica no se conocen estas propiedades estadísticas sino que son inferidas a través de una muestra y se intenta obtener realizaciones que sean estadísticamente equivalentes a la muestra. Esta equivalencia estadística significa que todas las realizaciones $Z^*(\mathbf{x})$ tengan la misma distribución de probabilidad de la función aleatoria $Z(\mathbf{x})$, sin embargo en ocasiones tenemos que conformarnos con que al menos posean los mismos momentos de primer y segundo orden que la muestra (Díaz Viera, 2002).

A.2. Métodos de simulación estocástica espacial

Los métodos de simulación espacial son ampliamente utilizados en la modelación de yacimientos. Según su geometría pueden clasificarse en basados en celdas o en objetos. Los modelos de facies basados en objetos imitan geometrías idealizadas interpretadas geológicamente. Dichos modelos muestran facies que pertenecen a formas geológicas. Los más conocidos son: el modelo de contadores aleatorios, de función aleatorias booleanas y de hojas muertas. Nos concentraremos solo en los métodos basados en celdas ya que están más relacionados con la metodologías de cosimulación presentada.

Una teselación es una división del espacio en pequeñas unidades o celdas. Las celdas suelen ser polígonos en el caso bidimensional o poliedros en el caso tridimensional, pero esto no es estrictamente necesario. Dependiendo de la aplicación considerada se pueden interpretar como una función aleatoria (véase el capítulo 2), como una partición de espacio o como una población.

Cuando las celdas de una teselación están asociadas a puntos del espacio se le suele denominar malla o mallado, por lo que es común utilizar estos términos indistintamente. A cada celda se le asigna un valor extraído de una función de distribución. Los valores no dependen de las celdas y las diferentes celdas tienen valores independientes. Debido a que se supone que

todos los puntos pertenecen a una celda casi con seguridad, a cada punto x se le puede asignar un valor aleatorio $Z(x)$. Esto define una función aleatoria Z cuyas propiedades estadísticas se pueden describir por su distribución espacial

A.2.1. Método Secuencial Gaussiano

Antes de aplicar este método es necesario tener en cuenta que se supone que la distribución conjunta es normal. Se trata de un modelo matemático bien definido pero muy restrictivo para las aplicaciones.

Se parte de un conjunto de datos conocidos y de una estimación por *kriging simple* en una malla, es decir

$$Z_{KS}(\mathbf{x}) = \sum_{i=1}^n \lambda_i \cdot Z(\mathbf{x}_i)$$

donde los pesos se obtienen de resolver el sistema:

$$\sum_{i=1}^n \lambda_i C(\mathbf{x}_j - \mathbf{x}_i) = C(\mathbf{x} - \mathbf{x}_j), \quad \mathbf{x}_j = 1, \dots, n.$$

Este es un método de interpolación por lo que estima de manera exacta los valores en los puntos de datos, además, si calculamos la covarianza se obtiene que

$$Cov(Z_{KS}(\mathbf{x}), Z(\mathbf{x}_k)) = \sum_{i=1}^n \lambda_i C(\mathbf{x}_k - \mathbf{x}_i) = C(\mathbf{x} - \mathbf{x}_k).$$

Lo anterior quiere decir que el kriging obliga a que la covarianza entre los valores de los datos y la estimación sea correcta. Sin embargo, la varianza tiene la forma

$$\text{Var}(Z_{KS}(\mathbf{x})) = C(0) - \sigma_{KS}^2(\mathbf{x})$$

cuando para una función aleatoria estacionaria debería ser constante $\sigma^2 = C(0)$; siendo entonces más pequeña de lo que debería. El mismo fenómeno sucede con la covarianza entre diferentes estimaciones.

Para corregir el problema anterior el método de simulación secuencial gaussiana propone sumarle a la estimación del kriging un término independiente con media cero y sin la varianza

$$Z(\mathbf{x}) = Z_{KS}(\mathbf{x}) + R(x), \quad R(x) \sim N(0; \sigma_{KS}^2(\mathbf{x})).$$

De esta forma se mantiene que $Cov(Z(\mathbf{x}), Z(\mathbf{x}_k)) = Cov(Z_{KS}(\mathbf{x}), Z(\mathbf{x}_k)) = C(\mathbf{x} - \mathbf{x}_k)$, y además: $\text{Var}(Z(\mathbf{x})) = C(0)$.

El nombre de *secuencial* se refiere a que el algoritmo visita todos los nodos a través de una función aleatoria y una vez calculados incorpora el valor simulado al conjunto de datos para asegurar que las covarianzas calculadas sean correctas, teniendo en cuenta este valor y todas las simulaciones futuras.

Las propiedades matemáticas claves que hacen que la simulación secuencial gaussiana funcione no se limitan a la distribución normal. La reproducción de covarianza de kriging es independiente de la distribución de datos, bastaría con que el residuo tenga una distribución

adecuada (media cero y varianza igual a la varianza kriging). Sin embargo el uso de otras conduce en la mayoría de los casos a una distribución global incorrecta de los valores simulados (Deutsch, 2002). Otra ventaja de esta distribución es que el teorema central del límite implica que la suma secuencial de residuos aleatorios i.i.d. conduce a una distribución gaussiana.

A.2.2. Método Secuencial Indicador

Uno de las principales críticas a la simulación secuencial gaussiana es que no refleja la posible conectividad de valores extremos (Deutsch, 2002), este problema puede ser resuelto para variables categóricas. El presente método, como los que se verán posteriormente, fue diseñado para la modelación de variables discretas o categóricas, aunque puede ser extendido a variables continuas.

El objetivo del formalismo de indicadores es estimar directamente la distribución de las incertidumbres. Para variables continuas se necesita primeramente codificar los datos como indicador de cierta propiedad de la manera siguiente:

$$\begin{aligned} I_{z_k}(\mathbf{x}) &= \mathbb{P}\{Z(\mathbf{x}) \leq z_k\} \\ &= \begin{cases} 1, & \text{si } Z(\mathbf{x}) \leq z_k \\ 0, & \text{en caso contrario;} \end{cases} \end{aligned}$$

y para variables categóricas la codificación es muy similar:

$$\begin{aligned} I_{z_k}(\mathbf{x}) &= \mathbb{P}\{\text{el valor } z_k \text{ esté presente}\} \\ &= \begin{cases} 1, & \text{si el valor } z_k \text{ se manifiesta en el punto } \mathbf{x} \\ 0, & \text{en caso contrario} \end{cases} \end{aligned}$$

El valor esperado de las funciones indicadoras es la probabilidad estacionaria a priori de dicha categoría, por lo que realizar un kriging de los datos residuales permite derivar la probabilidad de la misma. El resultado del kriging indicador es un modelo de incertidumbre en cada ubicación de la forma

$$F_{\text{IK}}(\mathbf{x}; k) = \sum_{i=1}^n \lambda_i(k) \cdot [I_{z_k}(\mathbf{x}_i) - F(k)] + F(k).$$

El kriging indicador estima directamente la distribución de probabilidad condicional en una ubicación no muestreada sin necesidad de realizar ninguna suposición sobre la distribución de las variables.

La simulación secuencial indicador se basa en utilizar la distribución así obtenida, en cada punto, para generar un nuevo valor de la variable codificada. Es decir, se realiza un análisis estructural (variograma):

$$\gamma_i^*(h, z_k) = \frac{1}{2} \frac{\sum_{i=1}^{m(b)} [I_{z_k}(\mathbf{x}_i + h) - I_{z_k}(\mathbf{x}_i)]^2}{n(h)};$$

y luego se realiza el kriging indicador para construir una distribución y se obtiene un valor simulado a partir de dicha distribución de la incertidumbre (véase 2.4). Para ello se genera un valor de forma uniforme y se le identifica con la clase correspondiente de acuerdo a F_{IK} (figura

A-1).

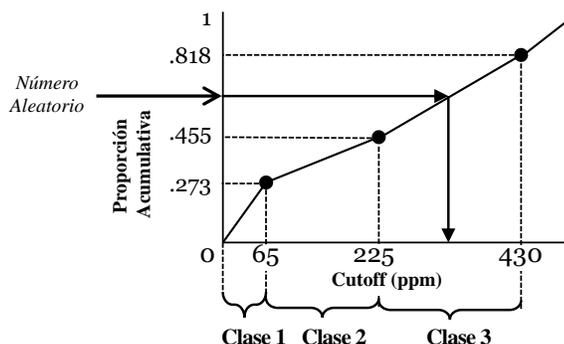


Figura A-1: Inversión de la función de distribución y selección de la categoría. Tomado de (Deutsch, 2002).

Estos pasos se realizan de forma secuencial, recorriendo todos los nodos a través de una función aleatoria e incorporando cada nuevo valor simulado al conjunto de datos.

Se trata de un método flexible basado en variogramas muy útil para variables categóricas que tiene, además, la capacidad de restringir la continuidad espacial para cada categoría. Sin embargo, el enfoque de indicadores no tiene en cuenta las relaciones de ordenación entre categorías, y el variograma indicador es solo una medida de la probabilidad de transición desde la categoría actual a cualquier otra categoría. Se necesita un variograma para cada categoría. Se suelen considerar pocas para simplificar el procedimiento y se podrían agregar categorías adicionales para proporcionar una mayor resolución. Elegir demasiadas categorías da como resultado violaciones significativas de la relación de orden de las variables, provocando efectos de ruido.

A.2.3. Método Gaussiano Truncado

Este método consiste en generar realizaciones de una variable aleatoria gaussiana continua y para luego truncarla en una serie de umbrales y^t y crear realizaciones de facies categóricas.

En la figura A-2 se muestra un ejemplo esquemático 1D. A lo largo del eje inferior se observa la simulación categórica que se deriva de una simulación gaussiana continua mostrada por la curva negra.

El *orden de facies* es una característica importante. En la figura A-3 se muestra un ejemplo de tres facies. Se observa con mayor frecuencia las facies 2 entre 1 y 3. Raramente se verá la facies 1 junto a la facies 3, solo cuando y cambia muy rápidamente de valores bajos a altos.

Este orden es una ventaja significativa de la simulación gaussiana truncada cuando se trata de facies ordenadas y una desventaja cuando dicho orden no es natural. Por ejemplo las facies pueden definirse por la cantidad de arcilla: 1 = arcilla, 2 = arena con intercalación de arcillas y 3 = arena limpia.

Las *proporciones* de facies pueden tomarse como constantes o modelarse como variables. En cualquier caso, supongamos que la proporción p de cada una de las facies ordenadas k se conoce en cada ubicación \mathbf{x} en la capa A , es decir: $p_k(\mathbf{x})$ con $k = \{1, \dots, K\}$ y $\mathbf{x} \in A$. Estas

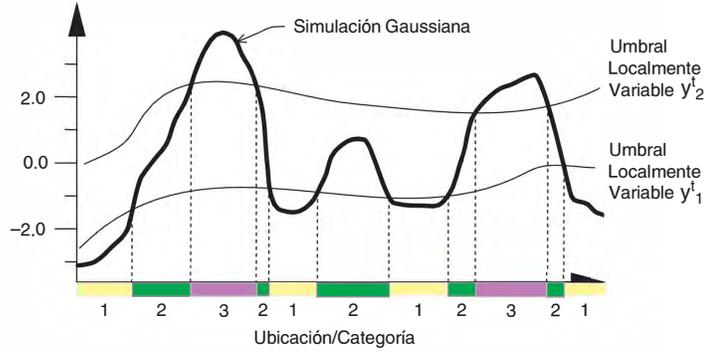


Figura A-2: Simulación gaussiana truncada: la variable de gaussiana continua se trunca en una serie de umbrales para crear una realización de la variable categórica. Tomado de (Deutsch, 2002).

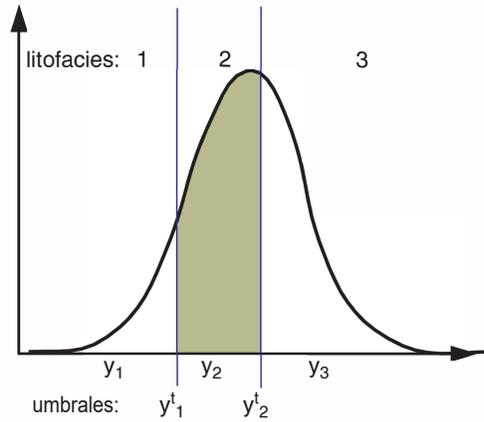


Figura A-3: Ejemplo de distribución normal con dos umbrales (y_1^t y y_2^t), tres códigos de facies (1, 2 y 3) y tres valores de y continuos para cada facies (y_1 , y_2 , y y_3). La proporción de cada facies es el área bajo la curva en cada intervalo. Tomado de (Deutsch, 2002).

proporciones pueden convertirse en proporciones acumulativas

$$cp_k(\mathbf{x}) = \sum_{j=1}^k p_j(\mathbf{x}) \text{ con } k = \{1, \dots, K\} \text{ y } \forall u \in A,$$

donde $cp_0 = 0$ y $cp_K = 1$ ya que las proporciones suman 1 en todas las localizaciones.

Los umbrales $K - 1$ para transformar y en facies están dados por

$$y_k^t(\mathbf{x}) = G^{-1}(cp_k(\mathbf{x})) \text{ con } k = \{1, \dots, K\} \text{ y } \forall u \in A, \quad (\text{A-9})$$

donde $y_k^t(\mathbf{x})$ son los umbrales ($y_0^t = -\infty$ y $y_K^t = +\infty$) y G^{-1} es la función cuantil de una distribución normal.

Dada y , estos umbrales pueden usarse para asignar un código de facies

$$\text{facies en } \mathbf{x} = k \text{ si } y_{k-1}^t(\mathbf{x}) < y(\mathbf{x}) \leq y_k^t(\mathbf{x}).$$

Se pueden usar umbrales variables para introducir la información de tendencia.

Los datos de facies categóricas deben transformarse en *datos de condicionamiento* gaussianos continuos para la simulación condicional de y . Para transformarlos se asigna el centroide de cada tramo en que la variable continua está asociada a una categoría determinada o a través de los percentiles. Por ejemplo, para una distribución $Normal(0, 1)$, si la facies 1 incluye la cola inferior hasta el percentil 40(P_{40}) asociado a un valor de -0.25 entonces esta facies puede codificarse como el valor -0.84 asociado al percentil 20(P_{20}).

Se deben tomar algunas decisiones sobre los *picos* causados por los datos categóricos. Para quitarlos la solución más simple es establecer la transformación de puntaje normal de cada facies en el centro de la clase en la distribución normal, es decir, reemplazar la ecuación (A-9) por

$$y(\mathbf{x}) = G^{-1} \left(\frac{cp_{k-1}(\mathbf{x}) + cp_k(\mathbf{x})}{2} \right).$$

La simulación gaussiana truncada requiere un solo variograma para la variable y . Esto es una ventaja en el sentido de que no necesitan ser calculados $K - \text{variogramas}$ indicadores y modelarlos de manera consistente, pero también es una desventaja significativa ya que no existe la capacidad de controlar los diferentes patrones de variabilidad para diferentes facies.

En general, no es una buena idea calcular y ajustar directamente el $y - \text{variograma}$ con los $y - \text{datos}$ debido a los picos. Una solución pudiera ser calcular el modelo de covarianza de un promedio ponderado de los $K - \text{variogramas}$ indicadores. Este sería el variograma indicador que se reproduciría mediante la simulación.

El enfoque recomendado es tomar el variograma indicador más importante e *invertirlo* numéricamente en el variograma de puntaje normal correspondiente. El variograma de puntaje normal tiene un rango y anisotropía similar al variograma indicador. La *simulación pluri-gaussianas* extiende el método gaussiano truncado para la simulación de facies múltiples utilizando varias funciones aleatorias gaussianas. Este es un problema mal planteado.

Los pasos del algoritmo de esta simulación incluyen:

1. Calcular los umbrales: se trunca una distribución gaussiana continua para respetar las proporciones de las categorías y las relaciones de orden.
2. Inferir un solo variograma continuo para representar la continuidad espacial de todas las facies.
3. Calcular una simulación continua condicionada a los datos categóricos interpretados como una variable continua.
4. Truncar la simulación continua en una simulación categórica.

Luego se generan múltiples realizaciones repitiendo todo el procedimiento con una semilla diferente.

La simulación gaussiana truncada debe usarse cuando hay un orden claro y no hay una diferencia significativa en la anisotropía de las diferentes geometrías de facies (Deutsch, 2002).

Apéndice B

Modelos de dependencia bivariada

B.1. Modelos de dependencia usando cópulas

Toda la información sobre la dependencia entre variables aleatorias está contenida en su correspondiente cópula (Erdely Ruiz y Gutiérrez Peña, 2012). Los conceptos y resultados que se presentarán en lo adelante tienen la intención de complementar lo expuesto en la sección 2.3.1.

Se pueden derivar varias propiedades para las cópulas, por ejemplo un corolario inmediato del teorema de Sklar es el siguiente:

Corolario B.1.1. *X e Y son variables aleatorias continuas independientes si y solo si su cópula subyacente es la cópula de independencia C^\perp*

$$C^\perp = C(F_1(x_1), F_2(x_2)) = F_1(x_1) \cdot F_2(x_2) = F(x_1, x_2), \quad (\text{B-1})$$

Para observaciones conocidas de una variable aleatoria, podemos obtener estimaciones empíricas para su distribución por medio de las distribuciones empíricas (sección 2.1.1), y esta converge casi seguramente cuando aumenta el tamaño muestral. De manera similar, tenemos la *cópula empírica* (Deheuvels, 1979), una función con dominio discreto $\left\{\frac{i}{n} : i = 0, 1, \dots, n\right\}^2$ definida como

$$C_n\left(\frac{i}{n}, \frac{j}{n}\right) = \frac{1}{n} \sum_{k=1}^n \mathbf{I}\{\text{rank}(x_k) \leq i, \text{rank}(y_k) \leq j\}.$$

La cópula empírica no es una cópula, por su dominio finito, puede extenderse en el sentido del teorema de Sklar mediante una función escalonada. También se ha demostrado su convergencia a la verdadera cópula (Erdely Ruiz y Gutiérrez Peña, 2012).

Para cópulas paramétricas una manera para estimar la cópula es el método de *funciones de inferencia por marginales* (IFM). La ventaja de este método es que simplifica la estimación conjunta de todos los parámetros y se basa en la minimización del logaritmo de la función de máxima verosimilitud (Ceballos López *et al.*, 2015). Este método se encuentra implementado en la biblioteca de `OpenTURNS` (Baudin *et al.*, 2015), utilizada en este trabajo.

Cópulas arquimedianas: Una importante familia de cópulas paramétricas son las *arquimedianas*, se trata de una clase asociativa de cópulas que admiten una fórmula explícita utilizando

una función auxiliar.

Se definen a partir de una función positiva φ de la siguiente manera:

$$C_a(u_1, \dots, u_d) = \varphi^{-1}(\varphi(u_1) + \dots + \varphi(u_d)). \quad (\text{B-2})$$

Para que C_a sea una cópula: $\varphi \in C((0, 1])$ tiene que ser estrictamente decreciente, convexa, y cumplir $\varphi(0) = +\infty$, $\varphi(1) = 0$. La función φ es conocida como el generador de la cópula (Nelsen, 2006; Joe, 2014) y suelen depender de parámetros que regulan la fuerza de la dependencia.

Esta clase de cópulas logran capturar un mayor número de estructuras de dependencia y pueden tener más de un parámetro. Ejemplos de estas cópulas son las siguientes:

- Cópula de Gumbel: $C(u_1, \dots, u_d) = \exp\left(-\left(\tilde{u}_1^{-\theta} + \dots + \tilde{u}_d^{-\theta}\right)^{\frac{1}{\theta}}\right)$ $\tilde{u}_i = -\log u_i$.

Su función generadora es: $(-\log(t))^\theta$ y su inversa: $\exp(-t^{1/\theta})$.

- Cópula de Clayton: $C(u_1, \dots, u_d) = \left(u_1^{-\theta} + \dots + u_d^{-\theta} - d + 1\right)^{\frac{1}{\theta}}$.

Su función generadora es: $\frac{1}{\theta}(t^{-\theta} - 1)$ y su inversa: $(1 + \theta t)^{-1/\theta}$.

- Cópula de Ali-Mikhail-Haq: $C(u, v) = \frac{uv}{1 - \theta(1-u)(1-v)}$.

Su función generadora es: $\log\left[\frac{1 - \theta(1-t)}{t}\right]$ y su inversa: $\frac{1 - \theta}{\exp(t) - \theta}$.

- Cópula de Frank: $C(u_1, \dots, u_d) = -\frac{1}{\theta} \log\left[1 + \frac{\prod_{i=1}^d (e^{\theta u_i} - 1)}{(e^{-\theta u_i} - 1)^{d-1}}\right]$.

El caso de la cópula de Frank bivariado fue analizado con más detalle en la sección 2.3.1. La cópula de Gumbel es asimétrica y puede captar variaciones bruscas o repentinas ya que las distribuciones que generan poseen colas anchas. Por otra parte, la cópula de Clayton es adecuada para describir la dependencia negativa entre las variables, debido a que asigna mayor probabilidad a eventos ubicados en la cola izquierda de la distribución (Ceballos López *et al.*, 2015). Por último la cópula Ali-Mikhail-Haq se suele utilizar para modelar dependencias débiles (Flores de la Fuente, 2009).

B.2. Otros modelos dependencia

La herramienta principal para estudiar como varían de forma conjunta dos variables aleatorias es la *covarianza*. Esta es una medida que indica el grado de variación conjunta de dos variables aleatorias alrededor de sus medias. Formalmente se define como

$$\begin{aligned} \text{cov}(X, Y) &= \text{E}[(X - \text{E}[X])(Y - \text{E}[Y])] \\ &= \text{E}[XY] - \text{E}[X]\text{E}[Y]. \end{aligned} \quad (\text{B-3})$$

Notemos que la varianza no es más que la covarianza de una variable consigo misma.

El principal problema de la covarianza como medida de variación conjunta es que, por su propia definición, sus unidades son producto de las unidades de las variables. Esto hace imposible

establecer comparaciones sobre la variación conjunta de variables diferentes, por ejemplo, si se tienen tres variables aleatorias X, Z e Y : $\text{cov}(X, Y) > \text{cov}(X, Z)$ no implica necesariamente que la relación de dependencia entre X e Y sea más fuerte que entre X y Z . Es por esto que se necesitan medidas normalizadas de las relaciones de dependencia, en particular presentaremos un grupo de medidas de asociación monótona de gran utilidad.

Existen otras caracterizaciones de la dependencia entre variables tales como máxima divisibilidad infinita (max-id), dependencia positiva del cuadrante (PQD) y otras (Joe, 2014) que no serán analizadas en este trabajo. En particular nos interesan las conocidas como medidas de asociación monótona.

B.2.1. Medidas de asociación monótona

El término asociación monótona se refiere a la dependencia donde si una variable aumenta, la otra tiende a aumentar (o disminuir); es decir, existe una relación más o menos monótona, no necesariamente lineal, en términos de esperanza condicional o mediana condicional de una variable dada la otra. Las medidas de asociación monótona bivariada, también llamadas medidas de concordancia bivariada, son mayores en valor absoluto si hay más probabilidad de que exista una función monótona que relaciona dos variables (Joe, 2014; McClarren, 2018).

Especial mención merece el coeficiente de correlación de Pearson. Este, si bien no es una medida de asociación monótona, es una de las formas más usuales de medir la relación entre las variables. El coeficiente de correlación de Pearson se encuentra, además, estrechamente relacionado con las medidas de asociación monótona más usuales: ρ de Spearman, τ de Kendall, β de Blomqvist y correlación de puntuaciones normales; algunas de las cuales serán presentadas a continuación.

Coeficiente de correlación de Pearson: se trata de una medida de dependencia lineal entre dos variables aleatorias cuantitativas absolutamente continuas. También se le conoce como ρ de Pearson, coeficiente de correlación momento-producto o simplemente correlación, debido a su uso generalizado.

Para dos variables aleatorias con varianza positiva se define como

$$\rho = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y},$$

Donde $\text{cov}(X, Y)$ es la covarianza del vector aleatorio (X, Y) y σ_X, σ_Y son las desviaciones estándar de las variables X e Y respectivamente.

A pesar de ser la medida de dependencia más utilizada no puede considerarse como una medida de asociación monótona, ya que no cumple con todas las propiedades de otras medidas de asociación.

La principal ventaja del coeficiente de correlación de Pearson es que es fácil de calcular e interpretar. Su valor varía en el intervalo $[-1, 1]$, su signo coincide con el sentido (positivo o negativo) de la correlación y su valor absoluto con la linealidad de dicha correlación.

La mayor desventaja de este coeficiente es que si se transforma una de las variables con una función no lineal, aún siendo estrictamente creciente, el coeficiente será diferente. Esto significa que si existe una relación entre X e Y pero no es lineal, entonces el coeficiente de correlación de Pearson puede subestimar o sobrestimar la relación real entre las variables.

Coefficiente de correlación de rangos de Spearman: esta medida de asociación monótona busca relaciones generales de monotonía entre dos variables. Para esto se observa la correlación entre las funciones de distribución de probabilidad marginales de cada variable:

$$\rho_S(X, Y) = \rho(F_X(x), F_Y(y)) \quad (\text{B-4})$$

Una de las propiedades más importantes de la correlación de Spearman es que si existe una función estrictamente creciente g que relaciona X con Y de la forma

$$Y = g(X),$$

entonces

$$\rho_S(X, Y) = 1.$$

Además, una transformación estrictamente monótona de X o Y no afectará la correlación de Spearman. Esta última propiedad es fundamental para todas las medidas de asociación monótona.

Si no se conocen las funciones de distribución marginales, pero tenemos muestras de las variables aleatorias, se puede estimar la correlación de Spearman. Dados N muestras de X y Y , se define una función que toma la muestra x_i o y_i y devuelve el rango de esa muestra entre las muestras de N :

$$\text{rank}(x_i) = \text{valor de ordenamiento de } x_i \text{ en la muestra de población.}$$

Para esto se ordenarán de menor a mayor los valores y se asignarán el rango según la posición que ocupa en dicho ordenamiento. Para el caso en que el valor se repita más de una vez, se calcula el promedio de su posición.

Usando esta función, aproximamos el coeficiente de correlación de Spearman para las muestras mediante

$$\rho_S(X, Y) = \frac{\text{cov}(\text{rank}(X), \text{rank}(Y))}{\sigma_{\text{rank}(X)}\sigma_{\text{rank}(Y)}}. \quad (\text{B-5})$$

Cuando se conoce la cópula con la cual se obtiene la distribución conjunta, entonces la ecuación (B-4) se transforma en

$$\rho_S = 12 \int F_X(x)F_Y(y)dF(x, y) - 3 = 12 \int_0^1 \int_0^1 uv dC(u, v) - 3.$$

Coefficiente de correlación de rangos de Kendall: conocido también como τ de Kendall, es una medida de asociación monótona que permite estudiar la correlación en una población de muestras de variables aleatorias a través de pares de dichas muestras.

Sean (X_1, Y_1) y (X_2, Y_2) vectores aleatorios *independientes e igualmente distribuidos* con función de densidad conjunta F , diremos que son *pares concordantes* si $(X_1 - X_2)(Y_1 - Y_2) > 0$

y que son *pares discordantes* si $(X_1 - X_2)(Y_1 - Y_2) < 0$. El τ de Kendall se define como

$$\begin{aligned}\tau &= P((X_1 - X_2)(Y_1 - Y_2) > 0) - P((X_1 - X_2)(Y_1 - Y_2) < 0) \\ &= 2P((X_1 - X_2)(Y_1 - Y_2) > 0) - 1 = 4P((X_1 > X_2) \cap (Y_1 > Y_2)) - 1 \\ &= 4 \int F dF - 1.\end{aligned}\tag{B-6}$$

Definiendo de igual forma la concordancia de las muestras, entonces podemos estimarla como

$$\tau = 2 \frac{(\# \text{ de pares concordantes}) - (\# \text{ de pares discordantes})}{N(N-1)}.$$

Es una de las medidas más comunes cuando se quiere relacionar dos variables mediante cópulas. Si se conoce dicha cópula, entonces la fórmula (B-6) se transforma en

$$\begin{aligned}\tau &= 4 \int C dC - 1 \\ &= 1 - 4 \int_0^1 \int_0^1 C_{2|1}(v | u) C_{1|2}(u | v) du dv.\end{aligned}$$

En particular, si el vector aleatorio tiene una distribución normal bivariada:

$$\tau = \frac{2}{\pi} \arcsin \rho(X, Y).$$

B.2.1.1. Propiedades

Todas las medidas de asociación monótona bivariada descritas cumplen con un grupo de propiedades deseables para cualquier medida de asociación. Es por esto que estas propiedades son tomadas como axiomas para definir cualquier medida de asociación monótona. Entonces, una función δ es una *medida de asociación monótona* si y solo si (Joe, 2014):

1. Dominio general: $\delta(X, Y)$ está bien definido para cualesquiera dos variables aleatorias X, Y .
2. Simetría: $\delta(X, Y) = \delta(Y, X)$.
3. Coherencia: $\delta(X, Y) \leq \delta(X', Y')$ si

$$(F_X(X), F_Y(Y)) \prec_c (F_{X'}(X'), F_{Y'}(Y')).$$

Lo anterior es equivalente a decir que la cópula asociada a (X, Y) es menor que la de (X', Y') en el orden de concordancia (Taylor, 2006).

4. Rango: $-1 \leq \delta \leq 1$ y se cumple que $\delta(X, Y) = 1$ o $\delta(X, Y) = -1$ cuando X e Y tienen comonotonicidad o contramonotonicidad respectivamente.
5. Independencia: $\delta(X, Y) = 0$ si $X \perp Y$.
6. Reversión de signo: $\delta(-X, Y) = -\delta(X, Y)$.

7. Continuidad: Si $(Y_{m1}, Y_{m2}) \sim F_m$ para $m = 1, 2, \dots$, $(X, Y) \sim F$, y $F_m \rightarrow_d F$, entonces $\lim_{m \rightarrow \infty} \delta(Y_{m1}, Y_{m2}) = \delta(X, Y)$.
8. Invarianza: Para cualesquiera dos funciones crecientes h_1, h_2 :

$$\delta(X, Y) = \delta(h_1(X), h_2(Y)). \quad (\text{B-7})$$

Posiblemente la más importante de todas las propiedades sea la de invarianza (B-7), aunque a veces no es considerado como uno de los axiomas (Taylor, 2006), ya que siempre puede obtenerse si se satisfacen el resto de los mismos (Joe, 2014).

B.2.2. Dependencia de colas

El concepto de dependencia de la cola bivariada se relaciona con la cantidad de dependencia en la cola de los cuadrantes superior (λ_U) o inferior (λ_L). Se utiliza cuando queremos medir la correlación entre las variables a medida que se acercan sus límites inferior y superior (McClarren, 2018). En este sentido los coeficientes de dependencia de la cola son medidas de la fuerza de dicha dependencia en la cola inferior o superior de una distribución multivariada.

El coeficiente de dependencia de la cola se deriva como una probabilidad condicional por lo que estará entre 0 y 1. Por lo tanto, no es una medida de dependencia monótona. La dependencia de la cola es invariante para funciones crecientes (B-7) (Joe, 2014).

La dependencia de la cola es diferente de otras medidas de correlación típicas en el sentido de que solo está interesada en valores extremos. Por ejemplo, dos variables podrían tener una correlación de Pearson de 0.5, pero una dependencia de la cola mucho mayor si sus valores extremos están relacionados.

Apéndice C

Métodos de Monte Carlos con Cadenas de Markov

C.1. Cadenas de Markov

Una *cadena de Markov* es un espacio de probabilidades que cumple con la propiedad de Markov, es decir; que la probabilidad de que el proceso se encuentre en un estado depende solo del estado anterior.

Formalmente, sea $\mathcal{M} = \{X_n\}_{n=0}^{\infty}$ una sucesión de variables aleatorias correlacionadas donde cada variable aleatoria X_n puede tomar valores (estados) en un conjunto Ω conocido como el espacio de estados. Se supone además que Ω es discreto. Si $X_n = i$, diremos que el proceso está en estado i en el tiempo n .

El proceso es una *cadena de Markov* si este satisface la *condición de Markov*, es decir:

$$P(X_{n+1} = j | X_n = i, X_{n-1} = x_{n-1}, \dots, X_0 = x_0) = P(X_{n+1} = j | X_n = i) = P_{ij}(n). \quad (\text{C-1})$$

A $P_{ij}(n)$ lo llamaremos *probabilidad de transición*, si no depende de n se dice que la cadena de Markov es *homogénea*, es decir:

$$P_{ij} = P(X_{n+1} = j | X_n = i) = P(X_1 = j | X_0 = i). \quad (\text{C-2})$$

Definiremos una *matriz de transición* a la matriz formada por los elementos $P_{ij}(n)$ para un número finitos de estados (N). En el caso homogéneo será $\mathbf{P} = [P_{ij}]$, es decir:

$$\mathbf{P} = \begin{bmatrix} P_{00} & P_{01} & \cdots & P_{0N} \\ P_{10} & P_{11} & \cdots & P_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N0} & P_{N1} & \cdots & P_{NN} \end{bmatrix},$$

donde para todo estado i

$$\sum_{j=1}^N P_{ij} = 1. \quad (\text{C-3})$$

Esta propiedad es conocida como *condición de normalización* y las matrices que las cumplen son llamadas estocásticas. Es importante destacar que dicha condición también se cumple para un número infinito de estados $\sum_{j=1}^{\infty} P_{ij} = 1$. Observemos que la suma se realiza por j , esta condición no se cumple necesariamente al sumar por i (dicha suma no es interpretada como una probabilidad) (Sen y Stoffa, 2015).

Ejemplo C.1.1. Supongamos que $\Omega = \{1, 2, 3\}$, la matriz de transición es

$$\mathbf{P} = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix}$$

y definamos una distribución inicial de $P(X_0 = i) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) = \pi^{(0)}$.

Esto se puede representar como un grafo, donde cada vértice corresponde a un estado con enlaces orientados de los vértices i a los vértices j . A cada uno de estos enlaces les corresponde la probabilidad de transición P_{ij} cuando esta sea diferente de cero. Esta representación se muestra en la figura C-1.

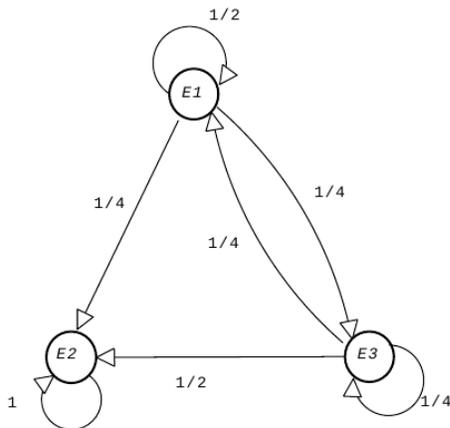


Figura C-1: Grafo orientado correspondiente al ejemplo 1. Tomado de (Nicholls y Tan, 2006).

Definiremos el *vector de probabilidades* $\pi^{(n)}$ a partir de las probabilidades de que se alcance cada estado j en un tiempo n como:

$$\pi_j^{(n)} = P(X_n = j). \quad (\text{C-4})$$

Un ejemplo de vector de probabilidades es la distribución inicial de $\pi^{(0)}$ del ejemplo C.1.1.

Este vector puede ser calculado a partir del vector en el tiempo anterior como:

$$\pi_j^{(n)} = \sum_{i \in \Omega} \pi_i^{(n-1)} P_{ij}(n) \quad (\text{C-5})$$

y en el caso homogéneo

$$\pi^{(n)} = \pi^{(n-1)} \mathbf{P} = \pi^{(0)} \mathbf{P}^n. \quad (\text{C-6})$$

El vector de probabilidad es propio con valor propio igual a 1 si se cumple que $\pi = \pi \mathbf{P}$ y es además una *distribución estacionaria*.

Por otra parte si se cumple que el límite $\lim_{n \rightarrow \infty} \pi^{(n)} = \pi$ existe para cualquier $\pi^{(0)}$ entonces π es la *distribución de equilibrio* de la cadena.

Ejemplo C.1.2. Retomando la matriz de transición del ejemplo C.1.1, se puede observar que

$$\lim_{n \rightarrow \infty} \mathbf{P}^n = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Esto implica que

$$\lim_{n \rightarrow \infty} \pi^{(n)} = \lim_{n \rightarrow \infty} (\pi^{(0)} \mathbf{P}^n) = \pi^{(0)} \lim_{n \rightarrow \infty} (\mathbf{P}^n) = (0 \quad 1 \quad 0) = \pi,$$

para cualquier distribución inicial $\pi^{(0)}$.

Además $\pi = \pi \mathbf{P}$. Es decir, $\pi = (0 \quad 1 \quad 0)$ es una distribución estacionaria que coincide con la distribución de equilibrio de la cadena.

Propiedades de las cadenas de Markov Si el proceso puede pasar del estado i al estado j se dice que j es *accesible* desde i ($P_{ij}(n) > 0$). Si los estados i y j son mutuamente accesibles se dice que se *comunican*. Una cadena de Markov es *irreducible* si todos los estados pueden comunicarse entre sí (figura C-2) (Jiménez Luna, 2015).

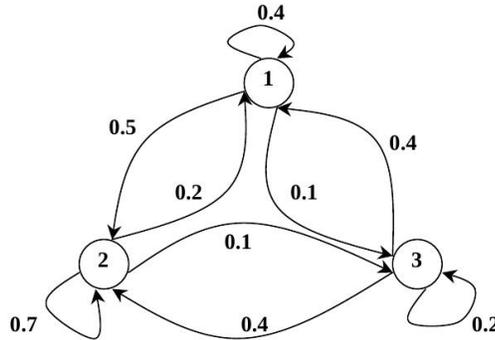


Figura C-2: Grafo que representa una cadena irreducible. Tomado de (Nicholls y Tan, 2006).

Sea una cadena de Markov irreducible con matriz de transición \mathbf{P} y sea i un estado fijo. Se define el conjunto

$$T_i = \{k : (\mathbf{P}^k)_{ii} > 0, k > 0\}. \quad (\text{C-7})$$

El mayor común divisor de los enteros en T_i es conocido como el *período* del estado i . Es posible demostrar que el período de una cadena irreducible para cada estado i es una constante independiente de i . La cadena se dice que es *periódica* si el período de cualquiera de sus estados es mayor que uno (Lantuéjoul, 2002).

Los estados pueden clasificarse como:

- Recurrente si $P_{ii}(n) = 1$ para algún n .
- Transitorio si $P_{ii}(n) < 1$ para todo n .
- Absorbente si $P_{ii}(n) = 1$ para todo n .

La cadena se dice *positiva recurrente* si cumple que cuando un estado de la cadena pertenece a una distribución estacionaria, la cadena converge a dicha distribución.

La cadena de Markov debe ser irreducible para que la distribución de equilibrio sea única; de lo contrario, el proceso puede caer en un estado desde el cual no puede ingresar a otros estados, y ya no puede ser independiente de su configuración inicial. Todos los estados también deben ser aperiódicos para que exista la distribución de equilibrio; de lo contrario, el sistema no exhibirá una distribución de estados que es independiente del tiempo.

Teorema C.1.1. *Sea una cadena de Markov irreducible, aperiódica y positiva recurrente, entonces convergerá a una distribución estacionaria si existe este límite:*

$$\pi_j = \lim_{n \rightarrow \infty} P_{ij}^n \quad (\text{C-8})$$

y además es independiente del estado inicial.

El siguiente teorema como el anterior, es fundamental para la aplicación de los métodos de Monte Carlo, que explicaremos a continuación (Pérez, 2014).

Teorema C.1.2 (Teorema ergódico). *Sea una cadena de Markov X_n positivo recurrente aperiódica con distribución estacionaria $f(x)$. Si $E[f(X)] < \infty$, entonces $P(\bar{f}_n \rightarrow E[f(X_n)]) = 1$, donde $\bar{f}_n = \frac{1}{n} \sum_t X_t$ es la media ergódica. Es decir, que converge en media cuadrática a su esperanza.*

C.2. Método de Monte Carlo con cadenas de Markov

Los métodos de Monte Carlo con cadena de Markov (MCMC por sus siglas en inglés) comprenden algoritmos para el muestreo de una distribución de probabilidad. Se basan en construir una cadena de Markov ergódica que tenga a la distribución deseada (a posteriori) como su distribución estacionaria (de equilibrio). Las muestras creadas a partir de la cadena son correlacionadas.

Definición C.2.1. *Un método de Monte Carlo basado en Cadenas de Markov (MCMC) usado para simular una distribución a posteriori π es cualquier método que genere una cadena de Markov ergódica (X_n) con distribución estacionaria π .*

Dado un valor inicial x_0 , se construye una cadena de Markov (X_n) a través de una densidad condicional $q(j|i)$ cuya densidad invariante es π . Para que esta cadena converja a la distribución a posteriori basta solo demostrar que es irreducible (según el Teorema C.1.1). Esto asegura la convergencia en distribución de (X_n) a una variable aleatoria con distribución π . Por lo tanto, para una n suficientemente grande los valores X_n, X_{n+1}, \dots pueden tomarse como una muestra de π . Debemos recordar que estas muestras no son independientes.

La mayoría de estimaciones que se realizan sobre la cadena construida con estas condiciones se relacionan con una media ergódica, de acuerdo al Teorema C.1.2. Sin embargo este teorema no proporciona información sobre cuán larga debe ser la cadena para poder realizar estimaciones precisas. Generalmente en estas cadenas se suele descartar un número alto de valores al principio de la misma, en lo que se denomina periodo de calentamiento.

Los ejemplos de métodos de Monte Carlo con Cadenas de Markov incluyen los siguientes:

- **Algoritmo de Metrópolis-Hastings (MH):** el cuál veremos con más detalles a continuación, genera una cadena de Markov utilizando una distribución a priori (q) y un método de aceptación rechazo. En realidad, es un marco general que incluye como casos especiales a los siguientes:
 - **Algoritmo de Metrópolis:** es un caso particular del algoritmo MH para una distribución a priori (q) simétrica.
 - **Muestreo de Gibbs:** es aplicable cuando la distribución conjunta no se conoce explícitamente o es difícil tomar muestras directamente, pero la distribución condicional de cada variable es conocida y más fácil tomar muestras. Este método requiere que todas las distribuciones condicionales de la distribución objetivo se muestreen exactamente ($\alpha = 1$).
 - **Metrópolis-Hastings pseudo-marginal:** este método reemplaza la evaluación de la densidad de la distribución objetivo con una estimación imparcial y es útil cuando la densidad objetivo no está disponible analíticamente, por ejemplo, modelos de variables latentes (variables que no se observan directamente sino que son inferidas a partir de otras, por ejemplo la densidad).
- **Muestreo mediante cortes:** este método depende del principio de que se puede muestrear de una distribución mediante un muestreo uniforme de la región bajo el gráfico de su función de densidad. Alterna el muestreo uniforme en la dirección vertical con el muestreo uniforme del *corte* horizontal definido por la posición vertical actual.
- **Metrópolis de múltiples intentos:** este método es una variación del algoritmo de MH que permite múltiples ensayos en cada punto. Al hacer posible dar pasos más grandes en cada iteración, ayuda a abordar el fenómeno de la *maldición de la dimensionalidad*.
- **Monte Carlo hamiltoniano (o híbrido):** intenta evitar el comportamiento aleatorio de la caminata introduciendo un vector de impulso auxiliar e implementando la dinámica hamiltoniana, por lo que la función de energía potencial es la densidad objetivo. Las primeras muestras se descartan después del muestreo. El resultado final es que las propuestas se mueven a través del espacio muestral en *saltos* más grandes; por lo tanto, están menos correlacionados y convergen a la distribución objetivo más rápidamente.

A continuación se detallarán algunos de estos métodos.

C.2.1. Algoritmo de Metropolis-Hastings

Primero que todo seleccionaremos una distribución a priori q que cumpla con las características siguientes:

- a) $q(i|j) = 0 \Rightarrow q(j|i) = 0$ (no poder ir hacia delante implica que no se puede ir hacia atrás),
 b) $q(j|i)$ es la matriz de transición de una cadena de Markov irreducible en el conjunto de estados Ω .

Partiendo de un estado inicial X_0 , para pasar del estado X_n al X_{n+1} con una distribución a posteriori π

1. Generación: generar un estado candidato j a partir de $X_n = i$ utilizando la distribución a priori $q(j|i)$.
2. Aceptación: j se acepta ($X_{n+1} = j$) con probabilidad

$$\alpha(j|i) \equiv \min \left(1, \frac{\pi_j q(i|j)}{\pi_i q(j|i)} \right), \quad (\text{C-9})$$

en caso contrario se rechaza, es decir, $X_{n+1} = X_n = i$.

Observación: para la aceptación es necesario generar una distribución uniforme $u \in [0, 1]$, si $u \leq \alpha(j|i)$ se acepta, si $u > \alpha(j|i)$ se rechaza.

Note que debido a que los pasos de generación y aceptación son independientes, la probabilidad de transición para ir de i a j es

$$P(X_{n+1} = j | X_n = i) = q(j|i)\alpha(j|i) \quad \text{para } i \neq j. \quad (\text{C-10})$$

La probabilidad P_{ii} puede ser hallada a partir de la condición de normalización C-3.

Cualquier distribución a priori es suficiente para que la cadena converja a su distribución estacionaria. La distribución a priori debe tener la misma dimensión que la a posteriori.

En otro caso la cadena puede pasar periodos largos de tiempo en un mismo estado. Por ello, escoger una distribución a priori adecuada se convierte en un problema, de esta forma se obtienen algunos casos particulares (Lantuéjoul, 2002).

C.2.2. Algoritmo de Metropolis

Supone que utilizamos distribuciones a priori simétricas (esto es $q(i|j) = q(j|i)$). Con esta condición, la probabilidad de aceptación pasa a ser:

$$\alpha(j|i) = \min \left(1, \frac{\pi(j)}{\pi(i)} \right). \quad (\text{C-11})$$

C.2.3. Muestreo de Gibbs

El muestreo de Gibbs es un algoritmo que permite generar muestras aleatorias a partir de una distribución de probabilidad multivariada específica. Se puede usar para aproximar la distribución conjunta y la distribución marginal de una de las variables o algún subconjunto de ellas.

Es aplicable cuando la distribución conjunta no se conoce explícitamente o es difícil tomar muestras directamente, pero la distribución condicional de cada variable es conocida y es fácil (o al menos, más fácil) tomar muestras.

Se considera un caso particular del algoritmo de MH por bloques en cual las densidades propuestas (q) coinciden con las distribuciones a posteriori condicionadas de modo que la probabilidad de aceptación es siempre uno.

Supongamos que el conjunto de parámetros es $\theta = (\theta_1, \dots, \theta_k)$ y que para todo $i = 1, \dots, k$, es fácil simular la distribución a posteriori condicional, $\pi(\theta_i | \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_k)$. El algoritmo consiste en fijar un valor inicial $(\theta_{1,0}, \dots, \theta_{k,0})$, y repetir:

1. Simular $\theta_{1,t+1} \sim \pi(\theta_1 | \theta_{2,t}, \dots, \theta_{k,t})$.
2. Simular $\theta_{2,t+1} \sim \pi(\theta_2 | \theta_{1,t+1}, \theta_{3,t}, \dots, \theta_{k,t})$.
3. Simular $\theta_{3,t+1} \sim \pi(\theta_3 | \theta_{1,t+1}, \theta_{2,t+1}, \theta_{4,t}, \dots, \theta_{k,t})$
- \vdots
- k . Simular $\theta_{k,t+1} \sim \pi(\theta_k | \theta_{1,t+1}, \dots, \theta_{k-1,t+1})$.

Nótese que en cada paso se utilizan los parámetros que fueron actualizados en pasos anteriores (Ausín, 2012). En la figura C-3 se muestra el ejemplo del algoritmo para dos parámetros.

Algoritmo 1.

Entrada: • Distribuciones condicionales, $f_{X|Y}(x | y)$ y $f_{Y|X}(y | x)$.

- Número máximo de iteraciones k .
- Valor inicial de $Y'_0 = y'_0$.

Salida: Sucesión de Gibbs

Paso 1. Sea $i = 0, Y'_0 = y'_0$.

Paso 2. Mientras $i + 1 \leq k$ realizar pasos 3 y 4.

Paso 3. Obtener X'_i mediante $f(x | Y'_i = y'_i)$.

Paso 4. Obtener Y'_{i+1} mediante $f(y | X'_i = x'_i)$.

- **Salida** (X'_i, Y'_{i+1}) (Algoritmo culminado satisfactoriamente).
- **PARAR.**

Paso 5. Salida (Se pidió la implementación del método como máximo para k iteraciones).

- **PARAR.**

Figura C-3: Ejemplo de un algoritmo de Gibbs para dos parámetros. Tomado de (Ausín, 2012).

Apéndice D

Código básico de la implementación computacional

A continuación se muestra el *script* básico de la implementación. Los códigos que verán a continuación se encuentran escritos en el lenguaje Python y se han omitido los detalles que fueron considerados irrelevantes.

En esta implementación se pueden diferenciar claramente tres etapas: una etapa de inicialización, una donde se efectúa la optimización con el algoritmo seleccionado y una etapa donde se muestran los resultados.

D.1. Etapas inicial y final

En una primera etapa se cargan las bibliotecas necesarias para el funcionamiento del programa, algunas son bibliotecas estándar (`numpy`, `scipy.optimize` y `openturns`) y otras creadas y modificadas por la autora. En esta etapa se cargan los datos, se definen los parámetros del variograma y la distribución bivariada y, además, se elimina la pendiente en caso de ser necesario. Por último se generará la función objetivo que será utilizada en los algoritmos de optimización (línea 52).

```
1  ##Modules##
2  import numpy as np
3  from scipy.optimize import differential_evolution
4  import dual_annealing as da #algoritmo de dual_annealing modificado
5  import variograms
6  import LoadSave_data as lsd
7  import funciones0 as fo
8  import Grafics as graf
9  import openturns as ot
10 import ot_copula_conditional_YE as cond
11
12 ##Load data##
13 Data=lsd.Load_columns(7,[1,2,3,4,5,6,7],windowname='Select_Data_File')
14 X=Data[:,0]
15 Z=Data[:,2]
16 Ip=Data[:,4]
17 Phit=Data[:,3]
```

```

18 Data_Grid=Stack(X,Z,Ip) #Alias of real code
19 Pared=lsd.Load_columns(2,[1,2],windowname='Select_Pared_Data_File')
20
21 ##Variogram##
22 P=Stack(X,Z,Phit) #Alias of real code
23 #Input parameters (Lakachi-Example)#
24 model='spherical';sill=0.00112;nugget_var=0.0007;a_range=52;amplitude=0
25 w0=[1,0.001]
26 Nobs=len(Data_Grid)
27 lag_number= 10
28 lag_size=0 #If lag_size=0 it is calculated automatically.
29 Phit_detrend,pend,zero=variograms.detrend(P,amplitude=amplitude)###Trend
    amplitude=0, Detrend amplitude=1
30 #Teorical input semivariogram calculation#
31 svt,dist_max,lag_size,lag_tolerance,lags=fo.variogram_parameter(Data_Grid,sill,
    nugget_var,a_range,lag_number,var_model=model,lag_size=lag_size)
32
33 ##Bivariate input distribution##
34 muLog = 7.43459;sigmaLog = 0.555439;gamma = 4977.04
35 marginal1 =ot.LogNormal(muLog, sigmaLog, gamma)
36 mu = 0.165352;beta = 0.0193547;
37 marginal2 =ot.Logistic(mu, beta)
38 theta = -4.2364
39 copula = ot.FrankCopula(theta)
40 #Bivariate distribution calculation#
41 bivariate_distribution_data = ot.ComposedDistribution([marginal1,marginal2],
    copula)
42 marginal_data=[bivariate_distribution_data.getMarginal(i) for i in [0,1]]
43 copula_data=bivariate_distribution_data.getCopula()
44
45 #Weights#
46 w1=[1,1]
47 w=[w0[0]*w1[0],w0[1]*w1[1]]
48 Phit_min=min(Phit)
49 Phit_max=max(Phit)
50
51 ##Objective function##
52 F=fo.funcobj(w, Data_Grid,svt,lag_tolerance,lags, bivariate_distribution_data,
    trend_coef=pend)
53
54 #####
55 #
56 #     Here goes the code for the selected Optimization Algorithm     #
57 #
58 #####

```

En este punto del proceso es necesario elegir el método de optimización. La implementación de la optimización se describe para DE en la sección D.2 y para GSA en la sección D.3. Una vez ejecutado dicho método se procede a mostrar los resultados, el script de esta última etapa se muestra a continuación.

```

1 #####
2 #
3 #     Here goes the code for the selected Optimization Algorithm     #
4 #
5 #####
6
7 ##Analysis of the result##
8 #Variable#
9 Ps = Stack(Data_Grid[:, :2], ResultX) #Alias of real code
10 Phits=Ps[:, 2]
11 Pared_s=Stack(Ip, Phits) #Alias of real code
12 bivariate_distribution_s=cond.ot_copula_fit(Pared_s)
13 marginal_s=[bivariate_distribution_s.getMarginal(i) for i in [0,1]]
14 copula_s=bivariate_distribution_s.getCopula()
15
16 #Save variable#
17 lsd.Save_Data('Results/Result.dat', Ps, columns=["X", "Z", "Phits"])
18
19 #Descriptive Univariate Statistics#
20 T_Ip=graf.Stats_Univariate(Ip)
21 T_Ip.histogram_boxplot(Ip, xlabel='Ip', marginal=marginal_data[0])
22 T_Phite=graf.Stats_Univariate(Phit)
23 T_Phite.histogram_boxplot(Phit, xlabel='Phit', marginal=marginal_data[1])
24 T_Phits=graf.Stats_Univariate(Phits)
25 T_Phits.histogram_boxplot(Phits, xlabel='Phits', marginal=marginal_s[1])
26
27 #Descriptive Bivariate Statistics#
28 graf.Scater1(Ip, Phit, labelx='Ip', labely='Phit')
29 graf.pseudo_obs_scater1(marginal_data, Ip, Phit)
30 Ip_Phite=Stack(Ip, Phite) #Alias of real code
31 TB=graf.Stats_Bivariate(Ip_Phite)
32 Ip_Phits=Pared_s
33 TBs=graf.Stats_Bivariate(Ip_Phits)
34
35 #Variogram calculation#
36 #Variogram initial#
37 svt_smooth,_,_,_, lag_smooth=fo.variogram_parameter(Data_Grid, sill, nugget_var,
    a_range, lag_number*4, var_model=model, lag_size=lag_size/4)
38 vdata = variograms.semivariogram(P, lags, lag_tolerance)
39 h, sv = vdata[0], vdata[1]
40 sv=sv-0.5*(h*pend)**2
41 graf.Experimental_Variogram(lags, [svt_smooth, sv], sill, a_range, var_model=
    model, variance=T_Phite.variance_value, color_svt='red', lags_svt=lag_smooth)
42 #Variogram simulado#
43 vdatas = variograms.semivariogram(Ps, lags, lag_tolerance)
44 hs, sv_s = vdatas[0], vdatas[1]
45 sv_s=sv_s-0.5*(hs*pend)**2
46 graf.Experimental_Variogram(lags, [svt_smooth, sv_s], sill, a_range, variance=
    T_Phite.variance_value, var_model=model, lags_svt=lag_smooth)
47
48 #Porosity#
49 graf.Scater(Ip, Phits, Phite)
50 graf.pseudo_obs_scater(marginal_data, marginal_s, Ip, Phite, Phits)
51
52 #Error of porosity#

```

```

53 error=(Phits-Phit)
54 Te=graf.Stats_Univariate(error)
55 Te.histogram_boxplot(error,xlabel='Error')
56
57 #Log_well
58 Data_well=Data[:,2:]
59 tracks=[[2],[1]]
60 limits=[[max(Ip),min(Ip)],[max(Phit),min(Phit)]]
61 labels=[[ 'Ip' ],[ 'Phit' ]]
62 graf.logview(Data_well,tracks,labels,limits=limits)
63
64 #Log_porosity
65 Data_log=np.array([Z,Phit,Phits,error]).T
66 tracks=[[1,2],[3]]
67 limits=[[min(Phit),max(Phit)],[min(Phit),max(Phit)],[min(error),max(error)]
68         ]]
69 labels=[[ 'Phit' ],[ 'Phits' ],[ 'Error' ]]
70 graf.logview(Data_log,tracks,labels,limits=limits)
71 #Marginal,copula and bivariate distributions plots
72 graf.four_axis (marginal_data,Ip,Phit,copula_data, bivariate_distribution_data)
73 graf.four_axis (marginal_s,Ip,Phits,copula_s, bivariate_distribution_s)
74 graf.cumul_four_axis (marginal_data,Ip,Phit,copula_data,
75                       bivariate_distribution_data)
76 graf.cumul_four_axis (marginal_s,Ip,Phits,copula_s, bivariate_distribution_s)
77 #Conditional_PDF#
78 conditioned_pdf_s=cond.ot_compute_conditional(Phits,bivariate_distribution_s,
79                                               Data_Grid)
80 conditioned_pdf=cond.ot_compute_conditional(Phits,bivariate_distribution_data,
81                                               Data_Grid)
82 graf.Scater(Phits,conditioned_pdf_s,conditioned_pdf)
83 #Conditional_CDF#
84 conditioned_cdf_s=cond.ot_compute_conditional_cdf(Phits,
85                                                    bivariate_distribution_s,Data_Grid)
86 conditioned_cdf=cond.ot_compute_conditional_cdf(Phits,
87                                                    bivariate_distribution_data,Data_Grid)
88 graf.Scater(Phits,conditioned_cdf_s,conditioned_cdf)
89 #Empirical CDF
90 graf.emprical_CDF([Phit,Phits],[marginal_data[1],marginal_s[1]])

```

D.2. Implementación para DE

Esta etapa requiere que con anterioridad se haya cargado el módulo correspondiente al algoritmo `differential_evolution` de la biblioteca `scipy`. En este caso se mantuvo el nombre original: `from scipy.optimize import differential_evolution` (véase línea 3 de la etapa inicial).

```

1  """-----Method-Dif-Evol-----"""
2
3  ##Method input parameters##
4  bounds=[(Phit_min,Phit_max) for i in range(Nobs)] #Variable search space
5  strategy='best1bin' #Default strategy
6  max_generations=4000
7  population_size=35 #Minimum is 5
8  tolerance=1e-10
9  mutation=0.5 #between (0,2)
10 recombination=0.2 #Probability
11 disp=False
12 epsilon=1e-3 #Value from which the optimization is stopped
13 def callback_epsilon(xk,convergence):
14     if F.func0(xk)<epsilon:
15         return True
16 #Create a random array wich preserve conditional distribution from Pared values
17 Phit_ini=Pared[:,:,1]
18 initial=Phit_ini#It can be 'latinhypercube' (default), 'random' or array list.
19 for elem in initial:
20     elem[elem > Phit_max] = Phit_max
21     elem[elem < Phit_min] = Phit_min
22 updating='deferred' #'deferred' or 'immediate'
23 workers=3 #-1 to use all available CPU cores
24
25 ##Result##
26 ResultDE = differential_evolution(F.func0,bounds,strategy=strategy, maxiter=
    max_generations,popsize=population_size, tol=tolerance,mutation=mutation,
    recombination=recombination,disp=disp, callback=callback_epsilon,polish=
    False, init=initial,updating=updating,workers=workers)
27 ResultX=ResultDE.x

```

D.3. Implementación para GSA

Esta etapa requiere que con anterioridad se haya cargado el módulo correspondiente al algoritmo `dual_annealing`, en este caso, en lugar de cargarlo directamente de la biblioteca `scipy` se le realizaron varias modificaciones para lo que se creó un módulo independiente. Para la carga de dicho módulo se utilizó: `import dual_annealing as da` (véase línea 4 de la etapa inicial).

```

1  """-----Method-Dual-Ann-----"""
2
3  ##Method input parameters##
4  bounds=[(Phit_min,Phit_max) for i in range(Nobs)] #Variable search space
5  maxiter=500
6  initial_temp=16 #Temperature for Dreo
7  restart_temp_ratio=2e-5
8  visit=2.7 #The value range is (0, 3]. The value 1 is classic SA and 2 is fast
9           SA.
10 accept=-5 #Default value is -5.0 with a range (-1e4, -5!!]. Deberia ser hasta 1
11 maxfun=1e7
12 no_local_search=True
13 epsilon=1e-3 #Value from which the optimization is stopped
14 def callback_epsilon(x, f, context):
15     if f<epsilon:
16         return True
17 #Create a random array wich preserve conditional distribution from Pared values
18 Phit_ini=Pared[0,:,1]
19 x0=Phit_ini
20 x0[x0 > Phit_max] = Phit_max
21 x0[x0 < Phit_min] = Phit_min
22
23 ##Result##
24 ResultDA = da.dual_annealing(F.func0,bounds,maxiter=maxiter, initial_temp=
    initial_temp,restart_temp_ratio=restart_temp_ratio, visit=visit,accept=
    accept,maxfun=maxfun,no_local_search=no_local_search, callback=
    callback_epsilon,x0=x0)
25 ResultX=ResultDA.x

```

Apéndice E

Figuras del capítulo 5

En este apéndice se mostrarán las figuras que se omitieron en el capítulo 5.

E.1. Figuras del caso de estudio I

En las figuras E-1 y E-2 se muestran los gráficos de dispersión de (Phit,Phits) para las variantes de cosimulación realizadas al pozo *Lakach 1*.

E.2. Figuras del caso de estudio II

E.2.1. Análisis estadístico y variográfico de los datos

Análisis estadístico univariado: en la figura E-3 se muestra la estadística univariada del pozo.

Variografía: en la figura E-4 se observa el variograma experimental y teórico de los datos.

Análisis bivariado utilizando cópulas: la figura E-5 ilustra la correlación entre las variables. Las funciones de distribución y densidad descritas se muestran en la figura E-6.

E.2.2. Cosimulación con cópulas paramétricas

En la figura E-7 se muestra la estadística descriptiva del error de estimación. La figura E-8 muestra la cercanía de las componentes teóricas y resultado de la simulación de la función objetivo. En la figura E-9 se muestra la distribución univariada de los datos. Además en la figura E-10 se observan las distribuciones empíricas. La figura E-11 muestra la dependencia bivariada de los datos. La figura E-12 muestra las funciones de distribución y densidad para las simulaciones.

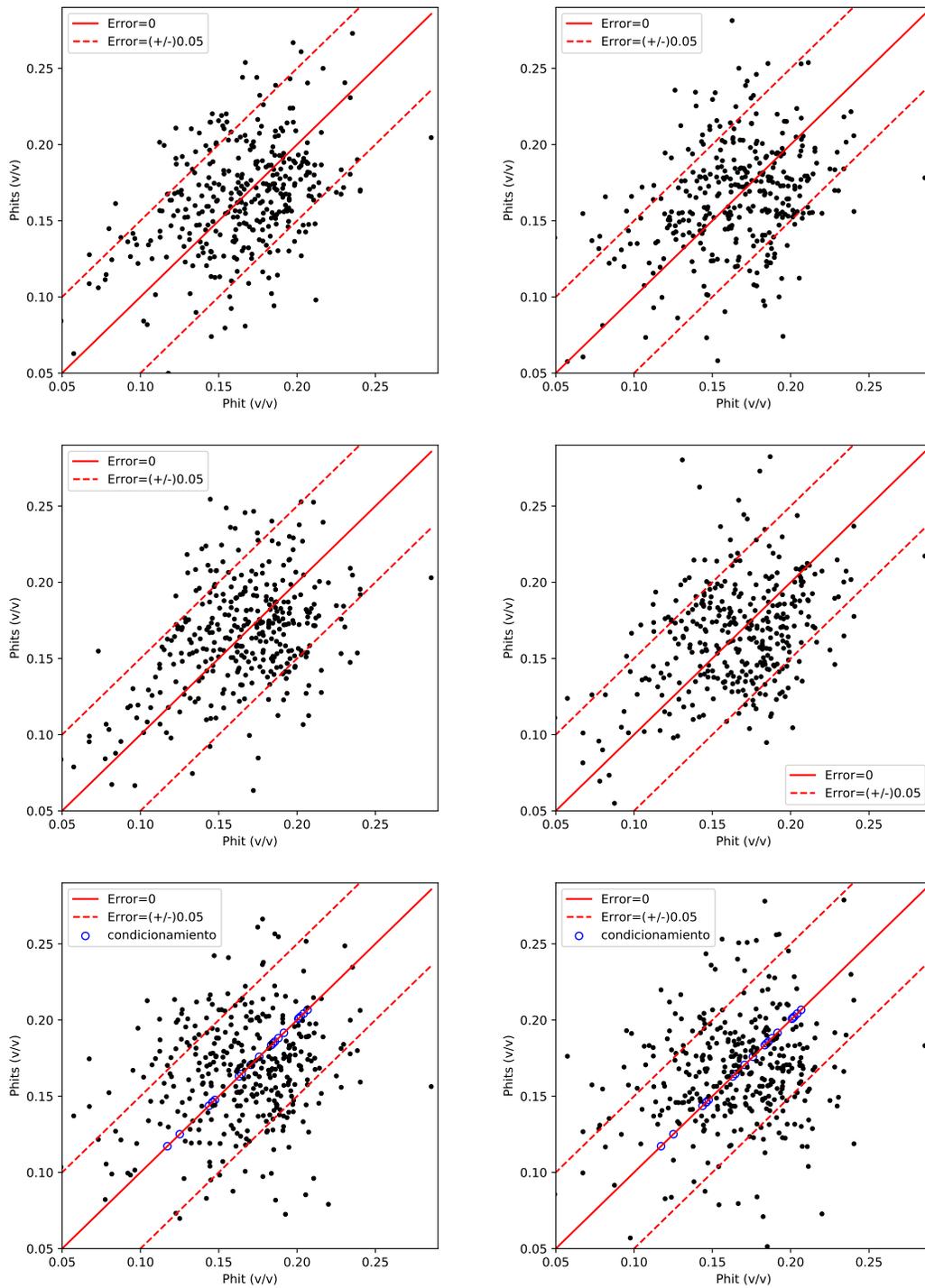


Figura E-1: Gráfico de dispersión (Phit,Phits) para DE a la izquierda y GSA a la derecha para el pozo *Lakach 1*. Parte superior: cosimulación paramétrica, parte central: variante híbrida y parte inferior: considerando datos de condicionamientos.

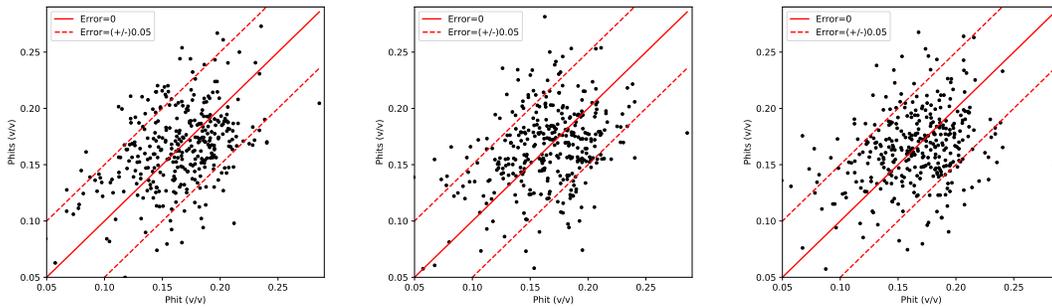
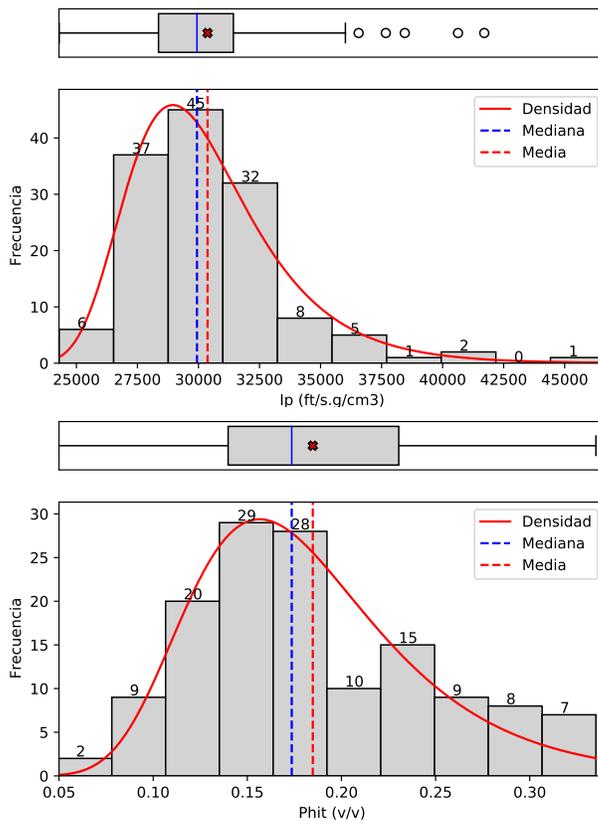


Figura E-2: Gráfico de dispersión (Phit,Phits) para DE, GSA y SA de izquierda a derecha en el pozo *Lakach 1*.



(a) Histograma y gráfico de caja para la impedancia acústica (arriba) y porosidad total (abajo).

(b) Tabla de la estadística descriptiva.

Estadística	Ip(ft.s.g/cm ³)	Phit(v/v)
Tamaño	137.0	137.0
Mínimo	24289.49	0.049478
Máximo	46654.02	0.335142
Rango	22364.53	0.285664
Media	30372.55	0.184786
Mediana	29938.86	0.173585
Primer cuantil	28363.42	0.139798
Tercer cuantil	31430.91	0.230432
Rango intercuantil	3067.488	0.090634
Varianza	9869608	0.003920
Desviación estándar	3141.593	0.062610
Simetría	1.756236	0.549447
Curtosis	5.830017	-0.435232

Figura E-3: Estadística descriptiva para el pozo *F03-FB-102*

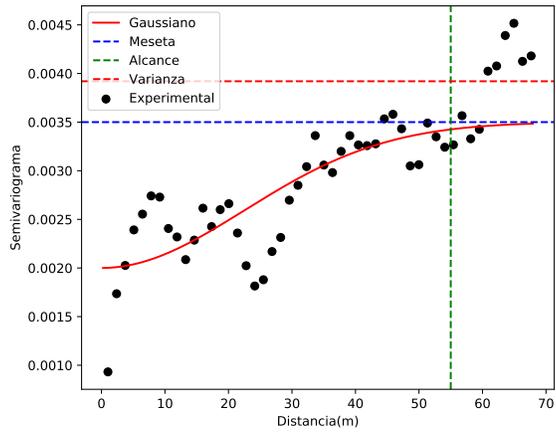


Figura E-4: Variograma de Phit para el pozo *F03-FB-102*.

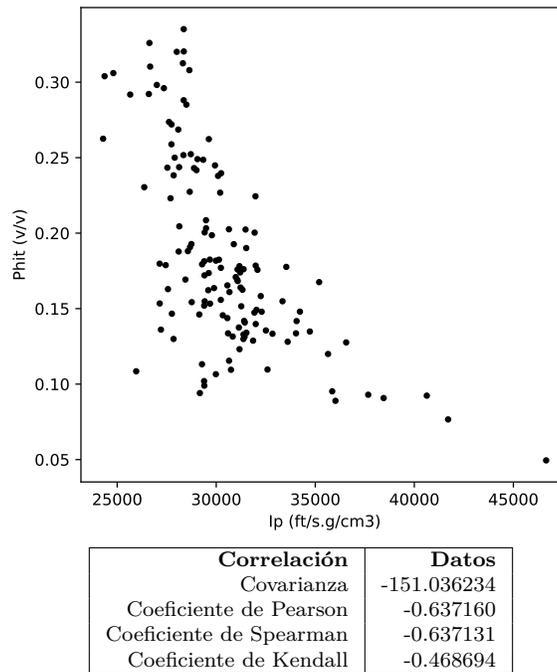


Figura E-5: Gráfico de dispersión de I_p contra $Phit$ y estadística descriptiva bivariada de los datos.

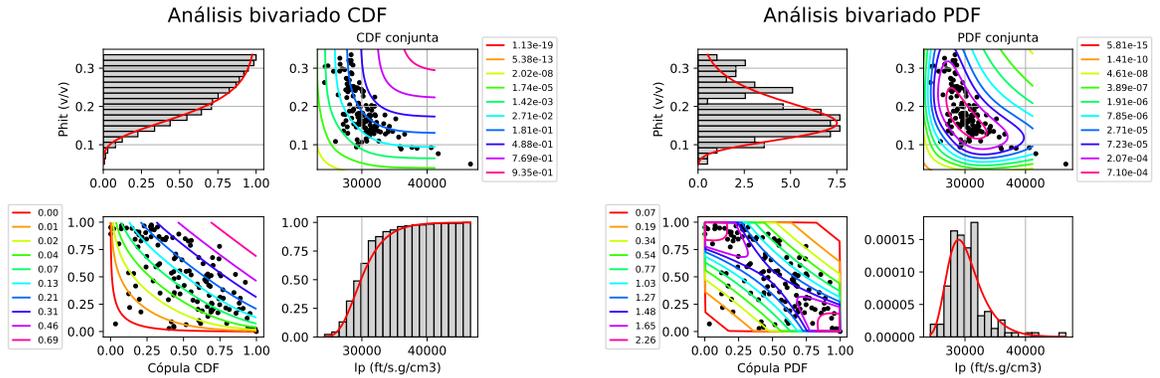
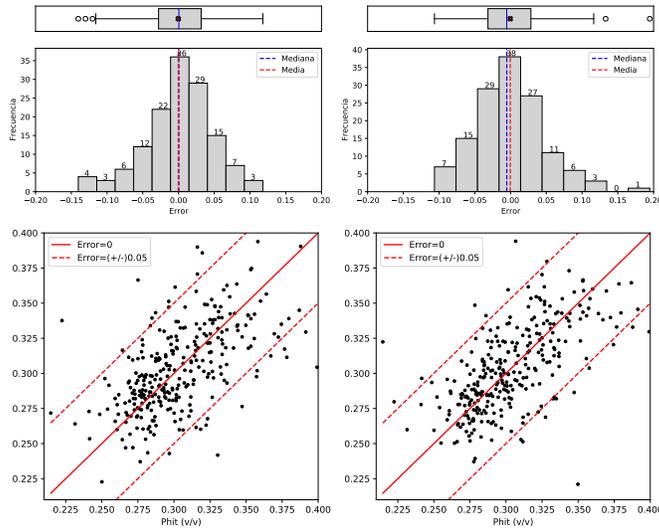


Figura E-6: Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de I_p y $Phit$ respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.



(a) Histograma y gráfico de caja para los simulados (arriba) y gráfico de dispersión ($Phit, Phits$) (abajo) para DE a la izquierda y GSA a la derecha.

(b) Tabla de la estadística descriptiva de los errores.

Estadística	Error	(DE)	(GSA)
Tamaño		137.0	137.0
Mínimo		-0.140212	-0.106456
Máximo		0.117940	0.194707
Rango		0.258151	0.301163
Media		-0.000372	-0.000496
Mediana		0.000466	-0.005371
Primer cuantil		-0.027924	-0.031651
Tercer cuantil		0.031671	0.028424
Rango intercuantil		0.059595	0.060076
Varianza		0.002237	0.002354
Desviación estándar		0.047296	0.048520
Simetría		-0.459572	0.591355
Curtosis		0.470737	1.299132

Figura E-7: Estadística descriptiva del error en la simulación para el pozo $F03-FB-102$.

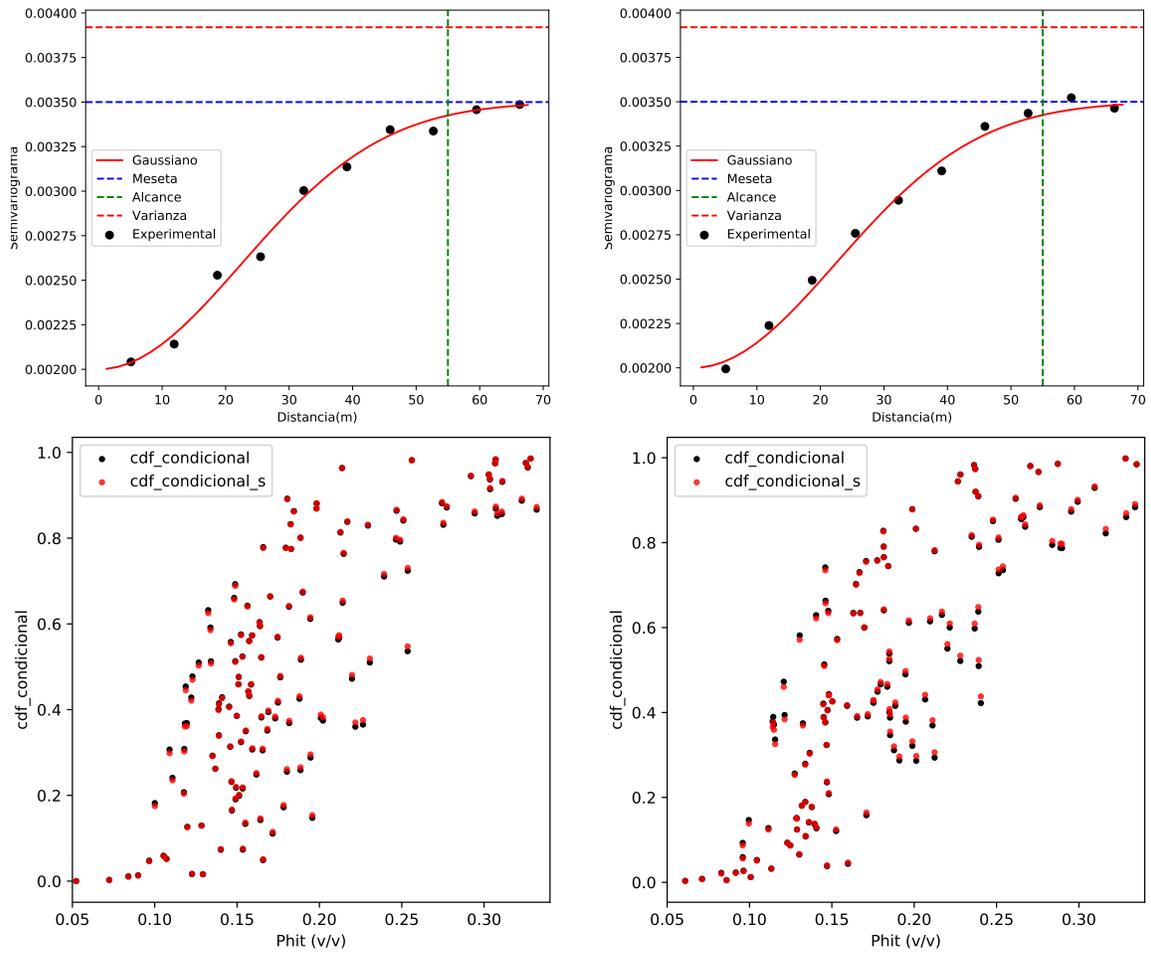
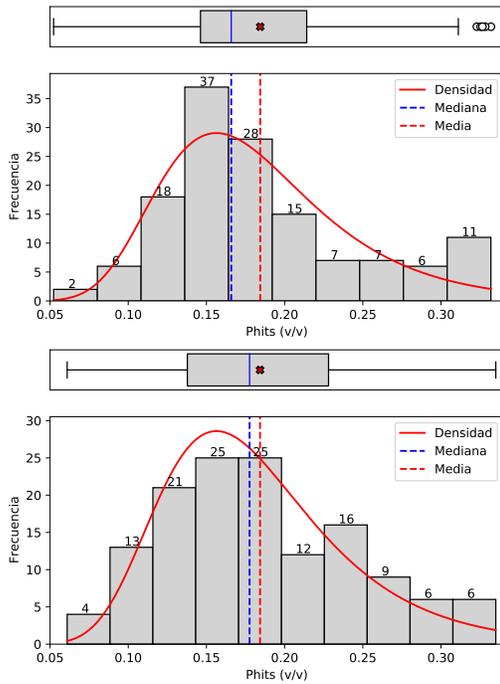


Figura E-8: Variogramas y función de distribución condicional de la simulación para el pozo *F03-FB-102* (DE a la izquierda, GSA a la derecha).



(a) Histograma y gráfico de caja para los simulados por DE (arriba) y GSA (abajo).

(b) Tabla de la estadística descriptiva.

Estadística	Phit(v/v)	Datos	(DE)	(GSA)
Tamaño		137.0	137.0	137.0
Mínimo		0.049478	0.052278	0.060941
Máximo		0.335142	0.331924	0.335050
Rango		0.285664	0.279646	0.274110
Media		0.184786	0.184414	0.184291
Mediana		0.173585	0.165859	0.177583
Primer cuantil		0.139798	0.146189	0.137789
Tercer cuantil		0.230432	0.214129	0.228084
Rango intercuantil		0.090634	0.067940	0.090295
Varianza		0.003920	0.003843	0.003811
Desviación estándar		0.062610	0.061995	0.061735
Simetría		0.549447	0.763285	0.505186
Curtosis		-0.435232	-0.077561	-0.410200

Figura E-9: Estadística descriptiva de la simulación para el pozo *F03-FB-102*.

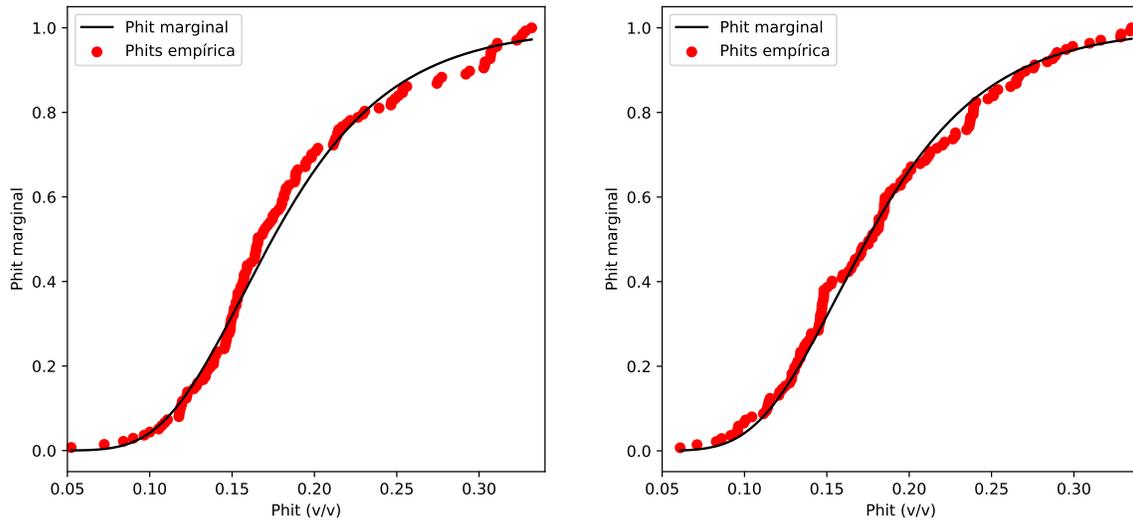


Figura E-10: Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).

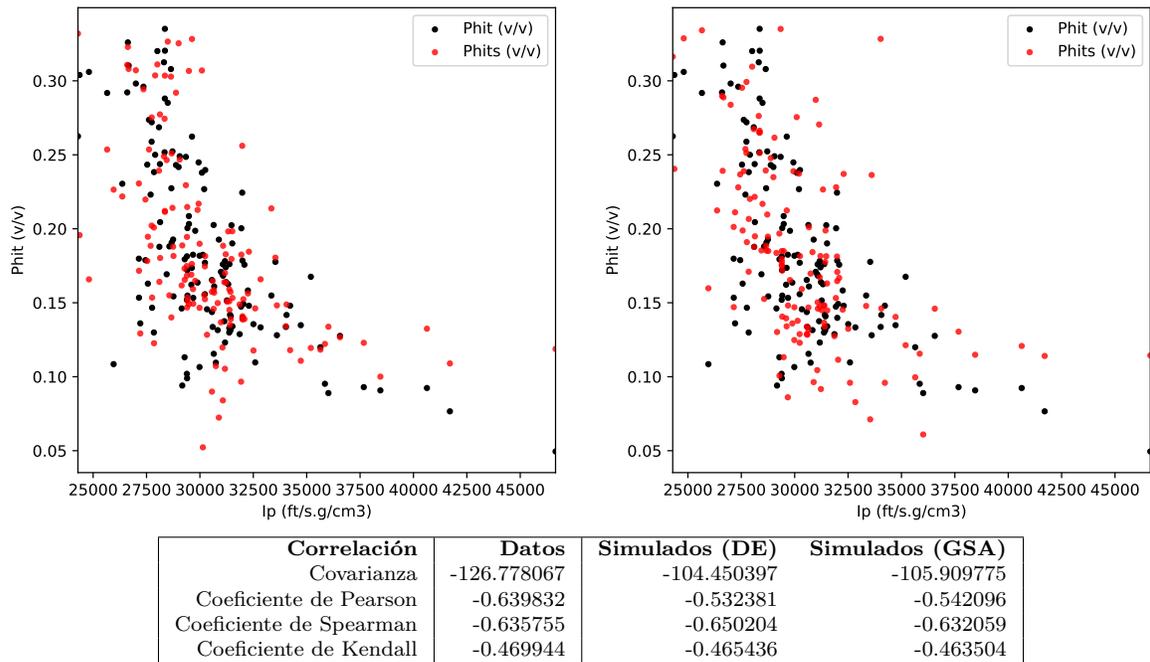


Figura E-11: Gráfico de dispersión de Ip contra Phits (DE a la izquierda, GSA a la derecha) y estadística descriptiva bivariada de los datos simulados.

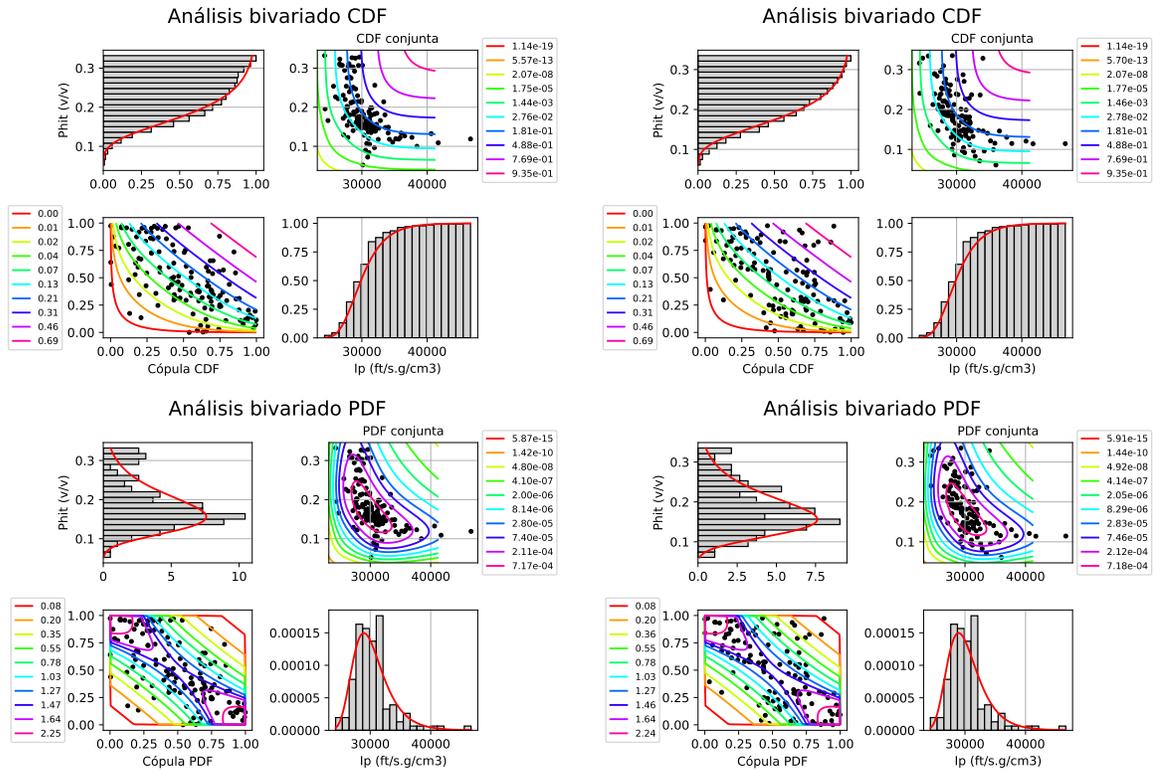
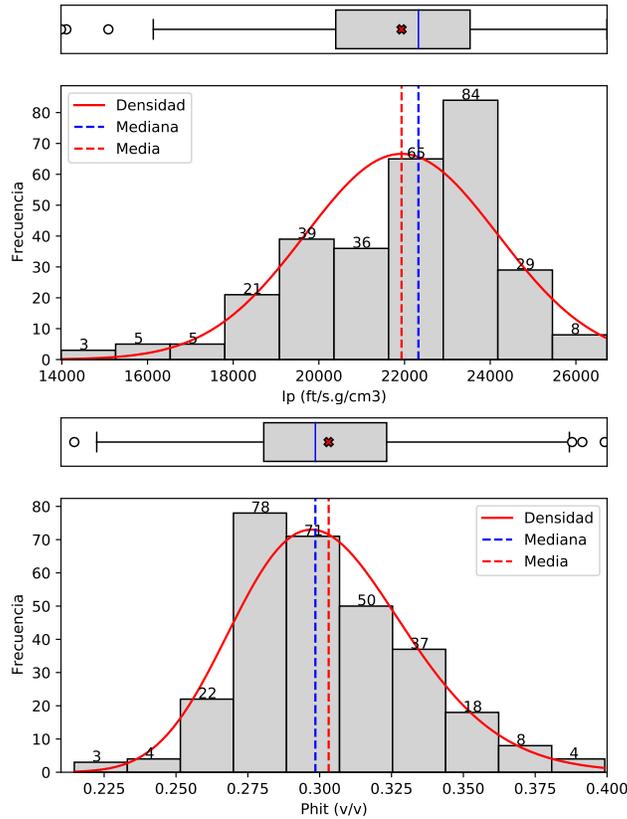


Figura E-12: Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de I_p y $Phit$ respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.

E.3. Figuras del caso de estudio III

E.3.1. Análisis estadístico y variográfico de los datos

Análisis estadístico univariado: en la figura E-13 se muestra la estadística univariada del pozo.



(b) Tabla de la estadística descriptiva.

Estadística	$I_p(ft/s.g/cm^3)$	$Phit(v/v)$
Tamaño	295.0	295.0
Mínimo	13980.60	0.214617
Máximo	26726.81	0.399200
Rango	12746.21	0.184583
Media	21929.52	0.303132
Mediana	22323.91	0.298500
Primer cuantil	20393.65	0.280540
Tercer cuantil	23529.11	0.323314
Rango intercuantil	3135.459	0.042774
Varianza	5050505	0.000932
Desviación estándar	2247.333	0.030535
Simetría	-0.7586385	0.478251
Curtosis	0.4954684	0.340092

(a) Histograma y gráfico de caja para la impedancia acústica (arriba) y porosidad total (abajo).

Figura E-13: Estadística descriptiva para el *Pozo 2*.

Variografía: en la figura E-14 se observa el variograma experimental y teórico de los datos.

Análisis bivariado utilizando cópulas: la figura E-15 ilustra la correlación entre las variables. Las funciones de distribución y densidad descritas se muestran en la figura E-16.

E.3.2. Cosimulación con cópulas paramétricas

En la figura E-17 se muestra la estadística descriptiva del error de estimación. La figura E-18 muestra la cercanía de las componentes teóricas y resultado de la simulación de la función objetivo. En la figura E-19 se muestra la distribución univariada de los datos. Además en la

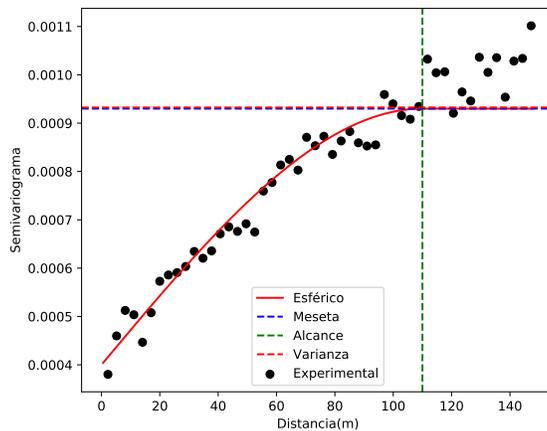


Figura E-14: Variograma de Phit para el *Pozo 2*.

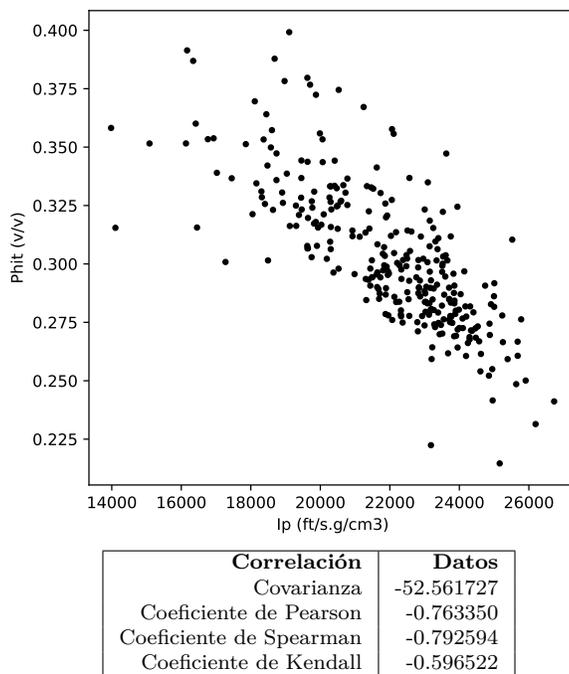


Figura E-15: Gráfico de dispersión de I_p contra $Phit$ y estadística descriptiva bivariada de los datos.

figura E-20 se observan las distribuciones empíricas. La figura E-21 muestra la dependencia bivariada de los datos. La figura E-22 muestra las funciones de distribución y densidad para las simulaciones.

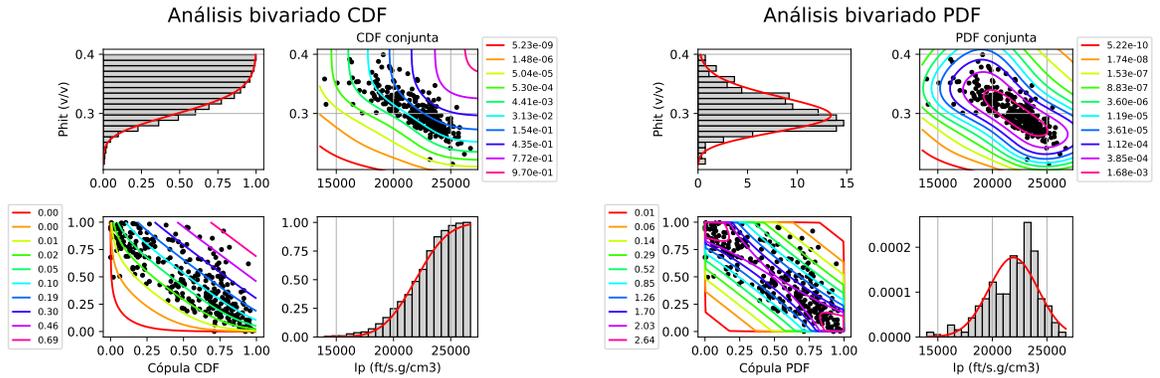
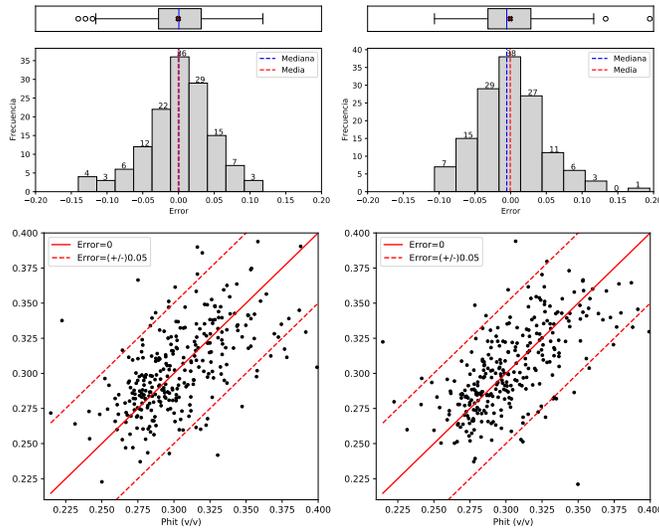


Figura E-16: Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de I_p y $Phit$ respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.



(a) Histograma y gráfico de caja para los simulados (arriba) y gráfico de dispersión ($Phit, Phits$) (abajo) para DE a la izquierda y GSA a la derecha.

(b) Tabla de la estadística descriptiva de los errores.

Estadística	Error	(DE)	(GSA)
Tamaño		295.0	295.0
Mínimo		-0.094847	-0.128801
Máximo		0.115205	0.107822
Rango		0.210052	0.236623
Media		0.000015	-0.000019
Mediana		0.001162	0.000002
Primer cuantil		-0.017258	-0.014447
Tercer cuantil		0.017278	0.014224
Rango intercuantil		0.034536	0.028671
Varianza		0.000788	0.000702
Desviación estándar		0.028074	0.026496
Simetría		0.018203	-0.178598
Curtosis		1.327909	2.445566

Figura E-17: Estadística descriptiva del error en la simulación para el *Pozo 2*.

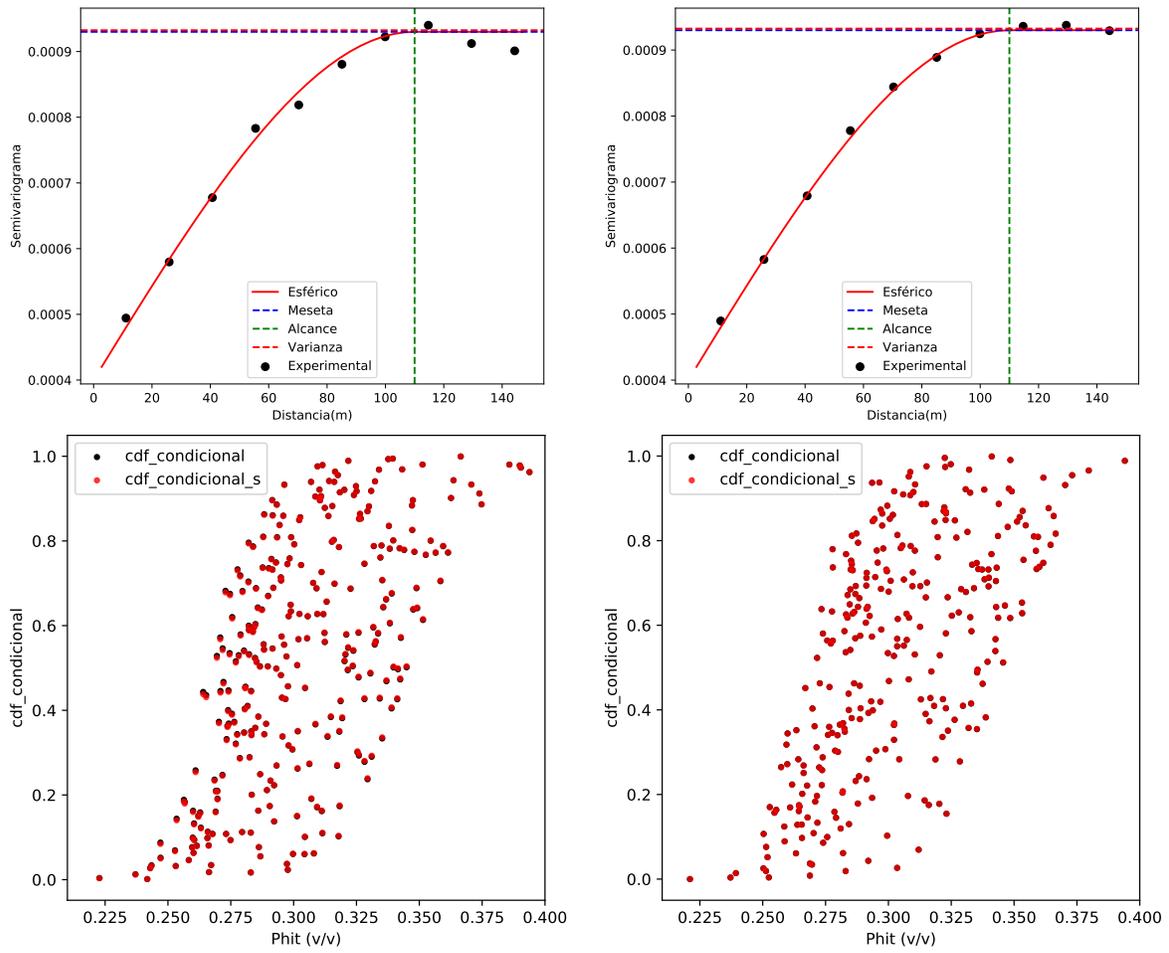
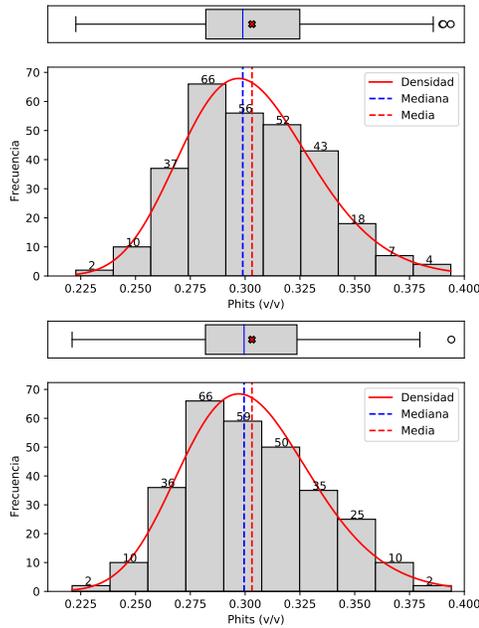


Figura E-18: Variogramas y función de distribución condicional de la simulación para el *Pozo 2* (DE a la izquierda, GSA a la derecha).



(a) Histograma y gráfico de caja para los simulados por DE (arriba) y GSA (abajo).

(b) Tabla de la estadística descriptiva.

Estadística	Phit(v/v)	Datos	(DE)	(GSA)
Tamaño		295.0	295.0	295.0
Mínimo		0.214617	0.222761	0.221057
Máximo		0.399200	0.393827	0.394076
Rango		0.184583	0.171066	0.173019
Media		0.303132	0.303147	0.303113
Mediana		0.298500	0.298916	0.299518
Primer cuantil		0.280540	0.282062	0.281899
Tercer cuantil		0.323314	0.324934	0.323649
Rango intercuantil		0.042774	0.042872	0.041751
Varianza		0.000932	0.000919	0.000919
Desviación estándar		0.030535	0.030308	0.030314
Simetría		0.478251	0.361120	0.294244
Curtosis		0.340092	-0.077056	-0.426364

Figura E-19: Estadística descriptiva de la simulación para el *Pozo 2*.

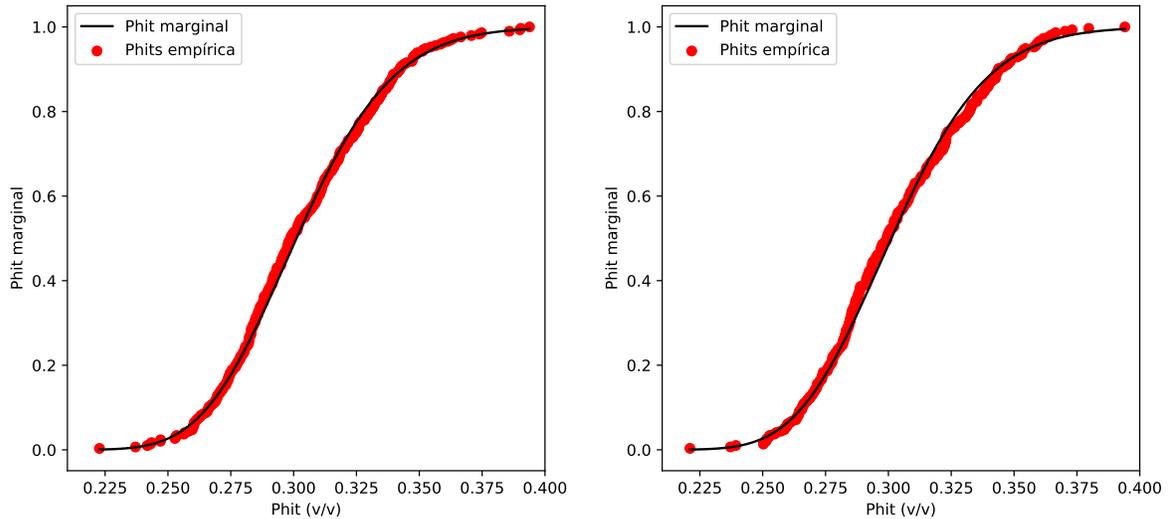


Figura E-20: Función de distribución empírica de probabilidad para las variables simuladas (DE a la izquierda, GSA a la derecha).

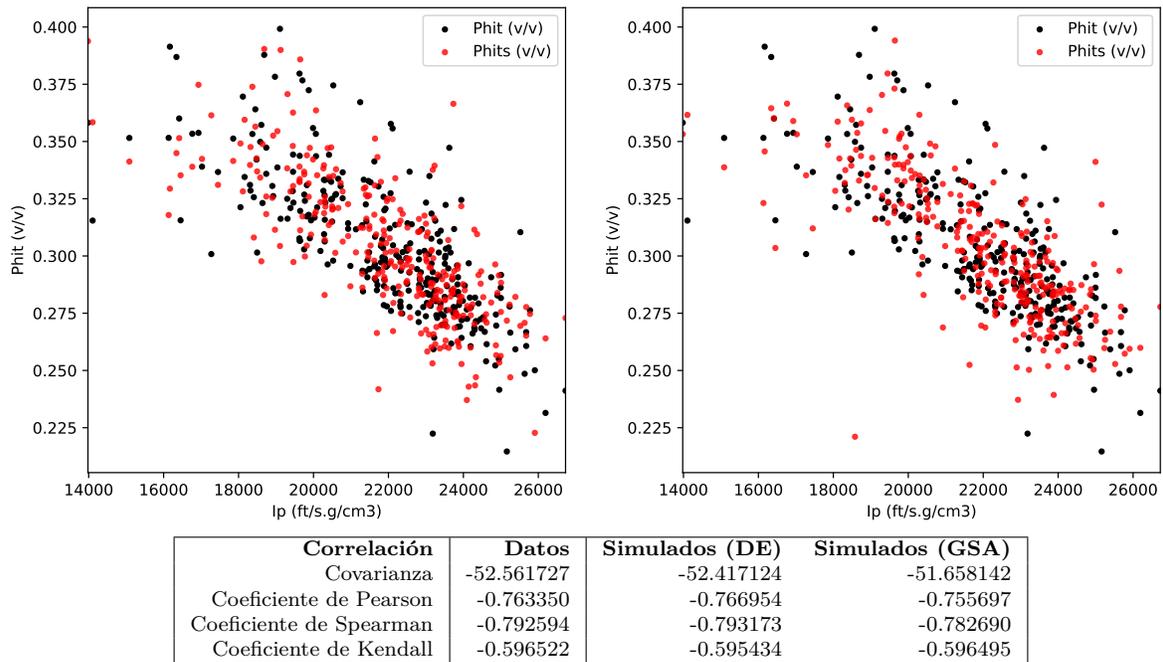


Figura E-21: Gráfico de dispersión de Ip contra Phits (DE a la izquierda, GSA a la derecha) y estadística descriptiva bivariada de los datos simulados.

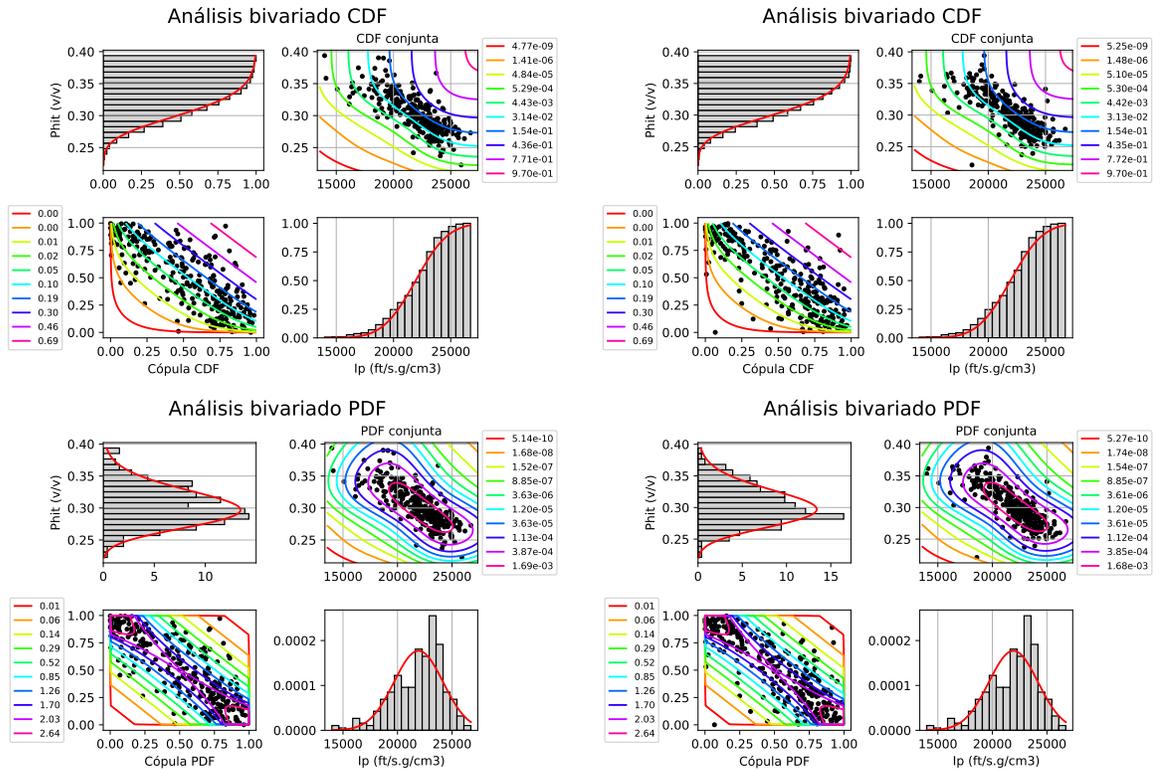


Figura E-22: Análisis de la función de distribución (a la izquierda) y análisis de la densidad de probabilidad (a la derecha) de las variables, arriba se encuentran los resultados para DE y abajo para GSA. En las esquinas superior izquierda e inferior derecha se encuentran las marginales de Ip y $Phits$ respectivamente, en las esquinas inferior derecha y superior izquierda se encuentran las curvas de nivel asociadas a la cópula y la conjunta respectivamente.

Bibliografía

- ACKLEY, D. A connectionist machine for genetic hillclimbing. *Kluwer International Series In Engineering And Computer Science* **28**:39–42 (1987) (Citado en la página 46.)
- AKHMEDOVA, S.A. Development and investigation of the effectiveness of the particle swarm optimization algorithm. En *1'st International Workshop on Mathematical Models and its Applications*, págs. 78–79 (2012) (Citado en la página 44.)
- AMBROSINO, D., ANGHINOLFI, D., PAOLUCCI, M., Y SCIOMACHEN, A. An experimental comparison of different heuristics for the master bay plan problem. En *International Symposium on Experimental Algorithms*, págs. 314–325. Springer (2010) (Citado en la página 31.)
- ARITO, F.L.A. *Algoritmos de Optimización basados en Colonias de Hormigas aplicados al Problema de Asignación Cuadrática y otros problemas relacionados*. candthesis, Universidad Nacional de San Luis, Facultad de Ciencias Físico Matemáticas y Naturales, Departamento de Informatica, Trabajo Final para alcanzar el grado de Licenciado en Ciencias de la Computacion (2010) (Citado en la página 30.)
- ASIF, A.A. Evolution of Ant Colony Optimization Algorithm—A Brief Literature Review. *arXiv preprint arXiv:1908.08007* (2019) (Citado en la página 31.)
- AUSÍN, C. Metodos de cadenas de Markov Monte Carlo. En *CESGA 2012*. Departamento de Estadstica, Universidad Carlos III de Madrid (2012) (Citado en las páginas XIII y 116.)
- AVNER, S.H. *INTRODUCCION A LA METALURGIA FISICA*, tomo 1. Mac Graw Hill (1988) (Citado en la página 20.)
- AVSETH, P., MUKERJI, T., Y MAVKO, G. *Quantitative seismic interpretation: Applying rock physics tools to reduce interpretation risk*. Cambridge university press (2010) (Citado en las páginas 81 y 85.)
- BAUDIN, M., DUTFOY, A., IOOSS, B., Y POPELIN, A.L. *OpenTURNS: An industrial software for uncertainty quantification in simulation*, capítulo 58, págs. 2001–2038. Springer-Verlag (2015) (Citado en las páginas 41 y 104.)
- BIANCHI, L., DORIGO, M., GAMBARDILLA, L.M., Y GUTJAHR, W.J. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing* **8**(2):239–287 (2009) (Citado en la página 19.)

- BLUM, C. Y ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)* **35**(3):268–308 (2003) (Citado en la página 20.)
- BRINKS, R. On the convergence of derivatives of B-splines to derivatives of the Gaussian function. *Computational & Applied Mathematics* **27**(1) (2008) (Citado en la página 8.)
- BÄCK, T. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. 1ª edición. Oxford University Press (1996) (Citado en las páginas 46 y 47.)
- CAMPOS SOBERANIS, M. Algoritmos Genéticos en Inteligencia Artificial. Parte I: Introducción. *SoldAI Research* (2019) (Citado en las páginas x y 33.)
- CASAR GONZÁLEZ, R., DÍAZ VIERA, M.A., HERNÁNDEZ MALDONADO, V.M., MÉNDEZ VENEGAS, J., MENDOZA TORRES, F., Y CHECA ROJAS, P. *Metodología para obtener modelos geológicos petrofísicos basados en la aplicación de métodos geoestadísticos*. Instituto Nacional del Derecho de Autor (2012) (Citado en la página 1.)
- CEBALLOS LÓPEZ, Á. *et al. Implementación de cópulas para la estimación del valor en riesgo*. Tesis Doctoral, Universidad del Rosario (2015) (Citado en las páginas 104 y 105.)
- CEDIEL, Y.F. Regular vine cópulas: una aplicación al cálculo de valor al riesgo. *Revista de Investigación en Modelos Financieros* **2**:30–64 (2016) (Citado en la página 15.)
- CHACÓN, J.E. Y DUONG, T. *Multivariate Kernel Smoothing and its Applications*. Chapman and Hall/CRC (2018) (Citado en la página 8.)
- CLAYTON V. DEUTSCH, A.G.J. *GSLIB: Geostatistical Software Library and User's Guide (Applied Geostatistics Series)*. Applied Geostatistics Series, 2ª edición. Oxford University Press, USA (1997) (Citado en las páginas 2 y 40.)
- COSENTINO, L. *Integrated reservoir studies*. Editions Technip (2001) (Citado en la página 1.)
- CROSSON, E. Y HARROW, A.W. Simulated Quantum Annealing can be exponentially faster than classical Simulated Annealing. *Proc of FOCS* págs. 714–723 (2016). 1601.03030v2 (Citado en la página 26.)
- CUADOR GIL, J.Q. *Elementos de geoestadística*. Departamento de Informática, Universidad de Pinar del Río “Hermanos Saiz Montes de Oca”, Pinar del Río, Cuba (2004) (Citado en la página 10.)
- DAS, A. Y CHAKRABARTI, B.K. *Quantum annealing and related optimization methods*, tomo 679. Springer Science & Business Media (2005) (Citado en las páginas x y 27.)
- DE MELO E SILVA ACCIOLY, R. Y CHIYOSHI, F.Y. Modeling dependence with copulas: a useful tool for field development decision process. *Journal of Petroleum Science and Engineering* **44**(1-2):83–91 (2004) (Citado en la página 15.)
- DEHEUVELS, P. La fonction de dépendance empirique et ses propriétés. Un test non paramétrique d'indépendance. *Bulletins de l'Académie Royale de Belgique* **65**(1):274–292 (1979) (Citado en la página 104.)

- DEUTSCH, C.V. *Geostatistical Reservoir Modeling*. Oxford University Press (2002) (Citado en las páginas XIII, XIII, XIII, 2, 5, 100, 101, 102 y 103.)
- DEUTSCH, C. Y COCKERHAM, P. Geostatistical modeling of permeability with annealing co-simulation (ACS). En *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers (1994) (Citado en las páginas 2, 5 y 40.)
- DÍAZ-VIERA, M., ANGUIANO-ROJAS, P., MOUSATOV, A., KAZATCHENKO, E., Y MARKOV, M. Stochastic modeling of permeability in double porosity carbonates applying a Monte-Carlo simulation method with t-copula. En *SPWLA 47th Annual Logging Symposium* (2006) (Citado en las páginas 2 y 5.)
- DÍAZ-VIERA, M. Y CASAR-GONZÁLEZ, R. Stochastic simulation of complex dependency patterns of petrophysical properties using t-copulas. En *Proc. IAMG*, tomo 5, págs. 749–755 (2005) (Citado en las páginas 2, 3 y 5.)
- DÍAZ VIERA, M., KERDAN KURDALIOVA, T., Y DEL VALLE GARCÍA, R. Metodología para la simulación estocástica espacial de propiedades petrofísicas basada en cópulas de Bernstein usando atributos sísmicos como variables secundarias. Informe técnico, IMP (2016) (Citado en las páginas 2, 13, 68 y 70.)
- DÍAZ VIERA, M.A. *Geoestadística aplicada*. Instituto de Geofísica y Astronomía, CITMA (2002) (Citado en las páginas 9, 10, 11, 17, 97 y 98.)
- DORIGO, M. *Optimization, learning and natural algorithms*. Tesis Doctoral, Politecnico di Milano (1992) (Citado en la página 30.)
- DRÉO, J., SIARRY, P., PÉTROWSKI, A., Y TAILLARD, E. *Metaheuristics for hard optimization*. Springer-Verlag (2006) (Citado en la página 42.)
- DUDLEY, R.M. *Uniform Central Limit Theorems*. Cambridge University Press (2014) (Citado en la página 8.)
- EBERHART, R.C. Y SHI, Y. Comparison between genetic algorithms and particle swarm optimization. En *Lecture Notes in Computer Science*, págs. 611–616. Springer Berlin Heidelberg (1998) (Citado en la página 34.)
- ERDELY, A. Y DIAZ-VIERA, M. Nonparametric and semiparametric bivariate modeling of petrophysical porosity-permeability dependence from well log data. En *Copula Theory and Its Applications*, págs. 267–278. Springer (2010) (Citado en las páginas 2 y 8.)
- ERDELY RUIZ, A. Y GUTIÉRREZ PEÑA, E. Monografía de Estadística Bayesiana (2012) (Citado en las páginas 12, 13, 97 y 104.)
- FLORES DE LA FUENTE, E. *EDAs con Funciones de cópula*. Master's thesis, Centro de Investigación en Matemáticas, Guanajuato, México (2009) (Citado en la página 105.)
- FRANK, M.J. On the simultaneous associativity of $F(x, y)$ and $x+y-F(x, y)$. *Aequationes mathematicae* **19**(1):194–226 (1979) (Citado en la página 15.)

- GAVANA, A. Global optimization benchmarks and AMPGO. *Accessed Apr* (2016) (Citado en las páginas x, x, 46 y 47.)
- GEMAN, S. Y GEMAN, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence* **PAMI-6**(6):721–741 (1984) (Citado en la página 22.)
- GESTAL, M., RIVERO, D., RABUÑAL, J.R., DORADO, J., Y PAZOS, A. *Introducción a los algoritmos genéticos y la programación genética*. Universidade da Coruña (2010) (Citado en la página 32.)
- GREENHALGH, D. Y MARSHALL, S. Convergence criteria for genetic algorithms. *SIAM Journal on Computing* **30**(1):269–282 (2000) (Citado en la página 44.)
- GUERRA GUERRA, L.F. *Metodología para la interpretación estratigráfica a partir de Sísmica 3D y registros de pozos*. techreport, Universidad Pedagógica y Tecnológica de Colombia (2013) (Citado en la página 78.)
- GUERRERO, M., GÓMEZ, D., ZAPATA, D., Y CÁRDENAS, M.V. Comparación de tres metaheurísticas para la optimización de inventarios con estimación de demanda. *Revista Ingeniería Industrial* **15**(1):51–68 (2016) (Citado en la página 37.)
- HERNÁNDEZ TORRES, R., IRIZAR MESA, M., LLANES SANTIAGO, O., CÂMARA, L.D.T., DA SILVA NETO, A.J., Y ZUMALACÁRREGUI DE CÁRDENAS, L.M. Comparación de diferentes algoritmos metaheurísticos en la estimación de parámetros del modelo relacional general de cromatografía líquida en columna. *Ingeniare. Revista chilena de ingeniería* **22**(1):14–25 (2014) (Citado en la página 44.)
- HU, X., SHI, Y., Y EBERHART, R. Recent advances in particle swarm. En *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*. IEEE (2004) (Citado en la página 29.)
- HUCK, H. The road to open source: Sharing a ten years experience in building OpendTect, the open source seismic interpretation software. En *74th EAGE Conference and Exhibition-Workshops*, págs. 1–295. European Association of Geoscientists & Engineers (2012) (Citado en la página 40.)
- INGBER, L. Very fast simulated re-annealing. *Mathematical and computer modelling* **12**(8):967–973 (1989) (Citado en la página 22.)
- INGBER, L. Adaptive simulated annealing (ASA): Lessons learned. *CoRR* **cs.MS/0001018** (2000) (Citado en la página 42.)
- JAMIL, M. Y YANG, X.S. A literature survey of benchmark functions for global optimization problems. *arXiv preprint arXiv:1308.4008* (2013) (Citado en la página 45.)
- JAWORSKI, P., DURANTE, F., HÄRDLE, W.K., Y RYCHLIK, T. (editores). *Copula Theory and Its Applications*. Springer Berlin Heidelberg (2010) (Citado en la página 12.)

- JIMÉNEZ LUNA, J. Métodos Monte Carlo basados en cadenas de Markov (2015) (Citado en la página 112.)
- JOE, H. *Dependence Modeling with Copulas*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability. CRC Press (2014) (Citado en las páginas 2, 12, 14, 105, 106, 108 y 109.)
- JOHNSON, C. Geostatmodels (2019) (Citado en la página 41.)
- KACHIGAN, S.K. *Statistical analysis: An interdisciplinary introduction to univariate & multivariate methods*. Radius Press (1986) (Citado en la página 7.)
- KIRKPATRICK, S., GELATT, C.D., Y VECCHI, M.P. Optimization by simulated annealing. *science* **220**(4598):671–680 (1983) (Citado en la página 22.)
- LANTUÉJOUL, C. *Geostatistical Simulation*. Springer Berlin Heidelberg (2002) (Citado en las páginas 10, 112 y 115.)
- LE, V.H., DÍAZ-VIERA, M.A., VÁZQUEZ-RAMÍREZ, D., [DEL VALLE-GARCÍA], R., ERDELY, A., Y GRANA, D. Bernstein copula-based spatial cosimulation for petrophysical property prediction conditioned to elastic attributes. *Journal of Petroleum Science and Engineering* **193**:107382 (2020) (Citado en las páginas 2, 8, 39, 53 y 68.)
- LIANG, F., LIU, C., Y CARROLL, R.J. *Advanced Markov Chain Monte Carlo Methods*. John Wiley and Sons, Ltd (2010) (Citado en la página 19.)
- LORENZO VALDÉS, A., ARMENTA FRAIRE, L., Y DURÁN VÁZQUEZ, R. A COPULA-TGARCH approach of conditional dependence between oil price and stock market index: The case of Mexico. *Estudios Económicos (México, DF)* **31**(1):47–63 (2016) (Citado en la página 15.)
- MALDONADO, V.M.H. *Simulación estocástica espacial de propiedades petrofísicas usando cópulas de Bernstein*. techreport, Instituto Mexicano del Petróleo (2014) (Citado en las páginas 2, 3, 5, 6, 12, 20, 22, 41, 42 y 51.)
- MCCLARREN, R.G. *Uncertainty Quantification and Predictive Computational Science: A Foundation for Physical Scientists and Engineers*. 1ª edición. Springer International Publishing (2018) (Citado en las páginas 14, 15, 16, 106 y 109.)
- MENDOZA-TORRES, F., DÍAZ-VIERA, M., Y ERDELY, A. Bernstein copula modeling for 2D discrete fracture network simulations. *Journal of Petroleum Science and Engineering* **156**:710–720 (2017) (Citado en las páginas 2 y 5.)
- MIKUSIŃSKI, P., SHERWOOD, H., Y TAYLOR, M. Probabilistic interpretations of copulas and their convex sums. En *Advances in probability distributions with given marginals*, págs. 95–112. Springer (1991) (Citado en la página 13.)
- NELSEN, R.B. *An introduction to copulas*. Springer Science & Business Media (2006) (Citado en las páginas 2, 13 y 105.)

- NESMACHNOW, S. Y SALSANO, E. Evaluación del desempeño computacional del cluster Medusa. Informe técnico, Centro de Cálculo, Instituto de Computación (2007) (Citado en la página 44.)
- NICHOLLS, G. Y TAN, S. *Inverse Problems, PHYSICS 707*. The University of Auckland (2006) (Citado en las páginas XIII, XIII, 111 y 112.)
- OLIPHANT, T.E. *A guide to NumPy*, tomo 1. Trelgol Publishing USA (2006) (Citado en la página 41.)
- PEDEMONTE, M. *Ant Colony Optimization*. techreport, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay (2007) (Citado en las páginas 30 y 31.)
- POLI, R. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications* **2008** (2008) (Citado en la página 27.)
- PRICE, K., STORN, R.M., Y LAMPINEN, J.A. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media (2006) (Citado en las páginas x, 36, 37, 42 y 43.)
- PÉREZ, D.O. *Nuevas estrategias para la inversión sparse de datos sísmicos prestack*. Tesis Doctoral, Universidad Nacional de La Plata, Facultad de Ciencias Astronómicas y Geofísicas, Argentina, Tesis presentada para obtener el grado académico de Doctor en Geofísica (2014) (Citado en la página 113.)
- PÉREZ LÓPEZ, J.R. *Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo*. Tesis Doctoral, Universidad de Cantabria (2005) (Citado en las páginas x y 28.)
- QUINTERO, L.V.S. Y COELLO, C.C. Un algoritmo basado en evolución diferencial para resolver problemas multiobjetivo. *Unpublished master thesis*. Instituto Politécnico Nacional, Mexico, DF (2004) (Citado en la página 35.)
- RAMÍREZ, D.V. *Simulación estocástica conjunta de propiedades petrofísicas con cópulas de Bernstein usando atributos sísmicos como variables secundarias a escala de registros de pozo*. techreport, Universidad Nacional Autónoma de México (2018) (Citado en las páginas 2, 39, 51, 53 y 68.)
- RAMÍREZ DA SILVEIRA, V. *Retos tecnológicos en las operaciones de perforación en aguas profundas en México*. techreport, Ingeniería, Universidad Nacional Autónoma de México (2014) (Citado en la página 52.)
- RASTRIGIN, L. Extremal control systems. *Theoretical foundations of engineering cybernetics series* **3** (1974) (Citado en la página 47.)
- RIOJAS, A. Y ÁLVAREZ, M. *Búsqueda tabú: conceptos, algoritmo y aplicación al problema de las n-reinas*. candthesis, Universidad Nacional Mayor de San Marcos-Facultad de Ciencias Matemáticas. EAP. de Investigación Operativa, Trabajo Final para alcanzar el grado de Licenciada de Investigación operativa (2005) (Citado en la página 19.)

- ROSENBROCK, H. An automatic method for finding the greatest or least value of a function. *The Computer Journal* **3**(3):175–184 (1960) (Citado en la página 47.)
- RUDOLPH, G. *Globale Optimierung mit parallelen Evolutionsstrategien*. Tesis Doctoral, Diplomarbeit, Universit at Dortmund, Fachbereich Informatik (1990) (Citado en la página 47.)
- SALVADORI, G., DE MICHELE, C., KOTTEGODA, N.T., Y ROSSO, R. *Extremes in nature: an approach using copulas*, tomo 56. Springer Science & Business Media (2007) (Citado en la página 2.)
- SCHLUMBERGER. *Petrel introduction course, v*. Schlumberger (2010) (Citado en la página 40.)
- SCHLUMBERGER OILFIELD SERVICES, M. Provincias petroleras de México. *World Energy Council* **34**:106–145 (2019) (Citado en la página 52.)
- SEN, M.K. Y STOFFA, P.L. *Global Optimization Methods in Geophysical Inversion*. Cambridge University Press (2015) (Citado en las páginas 21 y 111.)
- SERRANO CÁDIZ, A. *Optimización estocástica mediante métodos de Monte Carlo*. Proyecto Fin de Carrera, Universidad Carlos III de Madrid (2011) (Citado en las páginas 19 y 21.)
- SHARMA, A. Y MAHARJAN, P. Quantum Annealing as an Optimized Simulated Annealing: a case study. *Unpublished* (2018) (Citado en la página 27.)
- SHEN, X., ZHU, Y., Y SONG, L. Linear B-spline copulas with applications to nonparametric estimation of copulas. *Computational Statistics & Data Analysis* **52**(7):3806–3819 (2008) (Citado en la página 8.)
- SHI, X., TIAN, Z., CHEN, W., SI, B., Y JIN, X. A review on building energy efficient design optimization rom the perspective of architects. *Renewable and Sustainable Energy Reviews* **65**:872–884 (2016) (Citado en la página 44.)
- SHI, Y. Y EBERHART, R. A modified particle swarm optimizer. En *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*. IEEE (1998) (Citado en la página 28.)
- SHI, Y. Y EBERHART, R.C. Fuzzy adaptive particle swarm optimization. En *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, tomo 1, págs. 101–106. IEEE (2001) (Citado en la página 29.)
- SI, B., TIAN, Z., JIN, X., ZHOU, X., Y SHI, X. Ineffectiveness of optimization algorithms in building energy optimization and possible causes. *Renewable Energy* **134**:1295–1306 (2019) (Citado en la página 44.)
- SORIA, C.L., PANDOLFI, D.R., VILLAGRA, S.M., Y VILLAGRA, N.A. Algoritmos de Búsqueda Dispersa aplicados a problemas de Optimización Discreta. *Informes Científicos Técnicos-UNPA* **8**(1):132–150 (2016) (Citado en la página 20.)
- STORN, R. Y PRICE, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* **11**(4):341–359 (1997) (Citado en las páginas 35 y 42.)

- SZU, H. Y HARTLEY, R. Fast simulated annealing. *Physics Letters A* **122**(3-4):157–162 (1987) (Citado en la página 22.)
- TAKYAMAMOTO, Y. Knowledge salad bowl (2017) (Citado en las páginas x y 21.)
- TAYLOR, M.D. Multivariate measures of concordance. *Annals of the Institute of Statistical Mathematics* **59**(4):789–806 (2006) (Citado en las páginas 108 y 109.)
- TELLEZ CORTES, E.A., CORONEL SILVA, J.A., Y ARIAS PINTO, J.C. *Uso de técnicas de optimización global para resolver problemas de inversión sísmica*. candthesis, UNIVERSIDAD INDUSTRIAL DE SANTANDER, ESCUELA DE INGENIERIAS ELECTRICA, ELECTRONICA Y TELECOMUNICACIONES, BUCARAMANGA (2015) (Citado en las páginas 28 y 29.)
- TSALLIS, C. Possible generalization of Boltzmann-Gibbs statistics. *Journal of statistical physics* **52**(1-2):479–487 (1988) (Citado en la página 24.)
- TSALLIS, C. Y STARIOLO, D.A. Generalized simulated annealing. *Physica A: Statistical Mechanics and its Applications* **233**(1-2):395–406 (1996) (Citado en las páginas 24, 25, 42 y 49.)
- VIRTANEN, P., GOMMERS, R., OLIPHANT, T.E., HABERLAND, M., REDDY, T., COURNAPEAU, D., BUROVSKI, E., PETERSON, P., WECKESSER, W., BRIGHT, J., VAN DER WALT, S.J., BRETT, M., WILSON, J., JARROD MILLMAN, K., MAYOROV, N., NELSON, A.R.J., JONES, E., KERN, R., LARSON, E., CAREY, C., POLAT, İ., FENG, Y., MOORE, E.W., VAND ERPLAS, J., LAXALDE, D., PERKTOLD, J., CIMRMAN, R., HENRIKSEN, I., QUINTERO, E.A., HARRIS, C.R., ARCHIBALD, A.M., RIBEIRO, A.H., PEDREGOSA, F., VAN MULBREGT, P., Y CONTRIBUTORS, S... SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **17**:261–272 (2020) (Citado en las páginas 40 y 42.)
- WACKERNAGEL, H. *Multivariate Geostatistics: An Introduction with Applications*. 3ª edición. Springer-Verlag Berlin Heidelberg (2003) (Citado en la página 10.)
- WAND, M. Y JONES, M. *Kernel Smoothing*. Chapman and Hall/CRC (1994) (Citado en las páginas 8 y 9.)
- WETTER, M. Y WRIGHT, J. A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization. *Building and Environment* **39**(8):989–999 (2004) (Citado en la página 44.)
- WHITLEY, D. A genetic algorithm tutorial. *Statistics and computing* **4**(2):65–85 (1994) (Citado en la página 32.)
- XIANG, Y. Y GONG, X. Efficiency of generalized simulated annealing. *Physical Review E* **62**(3):4473 (2000) (Citado en las páginas x y 26.)
- XIANG, Y., GUBIAN, S., SUOMELA, B., Y HOENG, J. Generalized Simulated Annealing for Global Optimization: The GenSA Package. *R Journal* **5**(1) (2013) (Citado en las páginas 24, 25 y 49.)

XIANG, Y., GUBIAN, S., Y MARTIN, F. Generalized simulated annealing. *Computational Optimization in Engineering-Paradigms and Applications* págs. 25–46 (2017) (Citado en las páginas 49 y 51.)

ZAHARIE, D. Critical values for the control parameters of differential evolution algorithms. En *Proc. of MENDEL 2002, 8th Int. Conf. on Soft Computing*, págs. 62–67 (2002) (Citado en la página 43.)