



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE CARRERA

Optimización de intervenciones para el desplazamiento de poblaciones

Autor: Antonio Manuel González Ramírez

Director: David Quintana Montero

Co-director: Gustavo Recio Isasi

Leganés, octubre de 2011

Agradecimientos

Me gustaría dedicar este proyecto, que es la culminación de muchos años de esfuerzo, a todas aquellas personas que me han ayudado y apoyado en mi andadura académica.

Agradecer el apoyo de mi novia, Davinia, que me ha dado optimismo en las horas más difíciles. También agradecer el apoyo constante y sincero de mis padres, María de los Ángeles y Antonio, de mi hermana Mayte y demás familiares.

Hacer una mención especial a mi tutor de proyecto de fin de carrera, David Quintana, quién me dio luz cuando estaba perdido y me ayudo todo lo posible, sin duda el responsable del buen término del proyecto.

También me gustaría recordar a todos los compañeros que han compartido su tiempo, conocimiento y amistad, en especial a Rafa, Iñigo, Toño, Sergio, Marta, Carlos, Fernando y muchos otros.

Resumen

En este proyecto se propone explorar la posibilidad de aplicar algoritmos genéticos multiobjetivo para la construcción de un sistema de apoyo a la toma de decisiones relativas a la distribución geográfica de poblaciones.

El sistema aportará tres tipos de apoyo al usuario:

- Permite simular la forma en la que se desplazará una población dependiendo de los factores relevantes.
- Permite probar posibles intervenciones sobre los factores relevantes para la población estudiada, estas intervenciones provocarán que la población se desplace de una forma diferente a la que lo haría sin la intervención propuesta.
- Proporciona un módulo de optimización que permite identificar las intervenciones de mínimo coste que supongan la presencia de población en un lugar determinado con distintas probabilidades.

Índice

Agradecimientos	2
Resumen.....	3
CAPÍTULO 1: INTRODUCCIÓN.....	17
1.1 Motivación del proyecto	18
1.2 Objetivos del proyecto	20
1.3 Metodología	21
1.4 Visión general del documento	23
CAPÍTULO 2: ESTADO DE LA CUESTIÓN.....	24
2.1 Ecological Modelling	25
2.2 Optimización mediante Algoritmos Genéticos Multiobjetivo	30
2.2.1 Introducción a la optimización multiobjetivo	30
2.2.2 Optimización evolutiva.....	32
2.2.3 Métricas.....	35
CAPÍTULO 3: ANÁLISIS	38
3.1 Alcance del sistema	39
3.1.1 Estructura general	39
3.1.2 Modelado del desplazamiento.....	40
3.1.3 Modelado de intervenciones	42
3.2 Requisitos de usuario	43
3.2.1 Requisitos de capacidad.....	44
3.2.2 Requisitos de restricción	47
3.3 Requisitos de software.....	49
3.3.1 Requisitos funcionales.....	50
3.3.2 Requisitos no funcionales	54
3.5 Casos de uso.....	58
3.5.1 Diagrama de casos de uso	59

3.5.2 Especificación textual de los casos de uso	59
3.5.3 Diagramas de secuencia	65
CAPÍTULO 4: DISEÑO	67
4.1 Plataforma y Herramientas de Desarrollo	68
4.2 Diagrama de clases.....	69
4.3 Definición de las clases.....	72
CAPÍTULO 5: EXPERIMENTACIÓN.....	79
5.1 Introducción	79
5.2 Planteamiento Optimización Automática	80
5.2.1 Objetivos	80
5.2.2 Codificación y Operadores	80
5.3 Prueba de concepto I: Plaga.....	83
5.3.1 Módulo de simulación: cambio del terreno.....	83
5.3.2 Módulo de simulación: expansión de la población	84
5.3.3 Generador de escenarios	86
5.3.4 Optimización	87
5.3.4.1 Resultados con 100 simulaciones	88
5.3.4.2 Resultados con 1000 simulaciones	99
5.4 Prueba de concepto II: Inundación	108
5.4.1 Módulo de simulación: cambio del terreno.....	109
5.4.2 Módulo de simulación: expansión de la población	111
5.4.3 Generador de escenarios	113
5.4.4 Optimización	115
5.4.4.1 Resultados con 100 simulaciones	115
5.4.4.2 Resultados con 1000 simulaciones	124
CAPÍTULO 6: CONCLUSIONES	134
6.1 Objetivos alcanzados	135

6.2	Conclusiones finales.....	135
6.3	Líneas futuras de investigación.....	138
6.4	Problemas encontrados.....	139
6.5	Opiniones personales	140
Anexo A: PLANIFICACIÓN		141
A.1 Planificación		142
Anexo B: PRESUPUESTO		143
B.1 Cálculo de costes.....		144
Anexo C: DEFINICIONES Y REFERENCIAS.....		145
C.1 Referencias.....		145
Anexo D: MANUAL DE USUARIO		148
D.1 Ficheros de entrada		149
D.2 Manual de Usuario.....		150

Índice de tablas

Tabla 1: Tabla de definición de requisitos de usuario.....	43
Tabla 2: requisito de Capacidad RUC-1	44
Tabla 3: requisito de Capacidad RUC-2	44
Tabla 4: requisito de Capacidad RUC-3	44
Tabla 5: requisito de Capacidad RUC-4	44
Tabla 6: requisito de Capacidad RUC-5	45
Tabla 7: requisito de Capacidad RUC-6	45
Tabla 8: requisito de Capacidad RUC-7	45
Tabla 9: requisito de Capacidad RUC-8	45
Tabla 10: requisito de Capacidad RUC-9	45
Tabla 11: requisito de Capacidad RUC-10	46
Tabla 12: Requisito de Capacidad RUC-11	46
Tabla 13: requisito de Capacidad RUC-12	46
Tabla 14: requisito de restricción RUR-1.....	47
Tabla 15: requisito de restricción RUR-2.....	47
Tabla 16: requisito de restricción RUR-3.....	47
Tabla 17: requisito de restricción RUR-4.....	47
Tabla 18: requisito de restricción RUR-5.....	47
Tabla 19: requisito de restricción RUR-6.....	47
Tabla 20: requisito de restricción RUR-7.....	47
Tabla 21: requisito de restricción RUR-8.....	48
Tabla 22: requisito de restricción RUR-9.....	48
Tabla 23: requisito de restricción RUR-10.....	48
Tabla 24: requisito de restricción RUR-11.....	48
Tabla 25: requisito de restricción RUR-12.....	48
Tabla 26: requisito de restricción RUR-13.....	48

Tabla 27: tabla de definición de requisitos de software	49
Tabla 28: requisito de software funcional RSF-1	50
Tabla 29: requisito de software funcional RSF-2	50
Tabla 30: requisito de software funcional RSF-3	50
Tabla 31: requisito de software funcional RSF-4	50
Tabla 32: requisito de software funcional RSF-5	51
Tabla 33: requisito de software funcional RSF-6	51
Tabla 34: requisito de software funcional RSF-7	51
Tabla 35: requisito de software funcional RSF-8	52
Tabla 36: requisito de software funcional RSF-9	52
Tabla 37: requisito de software funcional RSF-10	52
Tabla 38: requisito de software funcional RSF-11	52
Tabla 39: requisito de software funcional RSF-12	53
Tabla 40: requisito de software funcional RSF-13	53
Tabla 41: requisito de software funcional RSF-14	53
Tabla 42: requisito de software funcional RSF-15	53
Tabla 43: requisito de software funcional RSF-16	53
Tabla 44: requisito de software funcional RSF-17	54
Tabla 45: requisito de software funcional RSF-18	54
Tabla 46: requisito de software funcional RSF-19	54
Tabla 47: requisito de software no funcional RSNF-1	54
Tabla 48: requisito de software no funcional RSNF-2	54
Tabla 49: requisito de software no funcional RSNF-3	55
Tabla 50: requisito de software no funcional RSNF-4	55
Tabla 51: requisito de software no funcional RSNF-5	55
Tabla 52: requisito de software no funcional RSNF-6	55
Tabla 53: requisito de software no funcional RSNF-7	55

Tabla 54: requisito de software no funcional RSNF-8.....	55
Tabla 55: requisito de software no funcional RSNF-9.....	56
Tabla 56: requisito de software no funcional RSNF-10.....	56
Tabla 57: requisito de software no funcional RSNF-11.....	56
Tabla 58: requisito de software no funcional RSNF-12.....	56
Tabla 59: requisito de software no funcional RSNF-13.....	56
Tabla 60: tabla de definición de casos de uso.....	59
Tabla 61: caso de uso 1 – Generar una plantilla	60
Tabla 62: caso de uso 2 – Cargar mundos.....	60
Tabla 63: caso de uso 3 – Modificar número de repeticiones	61
Tabla 64: caso de uso 4 – Modificar el tipo de los simuladores.....	61
Tabla 65: caso de uso 5 – Simulación.....	61
Tabla 66: caso de uso 6 – Modificar la vista mostrando otro instante	62
Tabla 67: caso de uso 7 – Volver a la pantalla principal.....	62
Tabla 68: caso de uso 8 – Realizar un cambio en una casilla	62
Tabla 69: caso de uso 9 – Limpiar cambios	63
Tabla 70: caso de uso 10 – Guardar cambios.....	63
Tabla 71: caso de uso 11 – Cargar archivo de costes	63
Tabla 72: caso de uso 12 – Cargar archivo de cambios.....	63
Tabla 73: caso de uso 13 – Modificar propiedades.....	64
Tabla 74: caso de uso 14 – Simulación en base a las intervenciones	64
Tabla 75: caso de uso 15 – Optimización	64
Tabla 76: caso de uso 16 – Consulta de una solución resultado de la optimización	65
Tabla 77: métrica SP (Experimento “Plaga” con 100 simulaciones)	88
Tabla 78: métrica SP (Experimento “Plaga” con 100 simulaciones)	89
Tabla 79: métrica HP (Experimento “Plaga” con 100 simulaciones).....	89
Tabla 80: métrica HP (Experimento “Plaga” con 100 simulaciones).....	90

Tabla 81: métrica SP (Experimento “Plaga” con 1000 simulaciones)	99
Tabla 82: métrica SP (Experimento “Plaga” con 1000 simulaciones)	99
Tabla 83: métrica HP (Experimento “Plaga” con 1000 simulaciones).....	99
Tabla 84: métrica HP (Experimento “Plaga” con 1000 simulaciones).....	100
Tabla 85: métrica SP (Experimento “Inundación” con 100 simulaciones).....	115
Tabla 86: métrica SP (Experimento “Inundación” con 100 simulaciones).....	115
Tabla 87: métrica HP (Experimento “Inundación” con 100 simulaciones)	116
Tabla 88: métrica HP (Experimento “Inundación” con 100 simulaciones)	116
Tabla 89: métrica SP (Experimento “Inundación” con 1000 simulaciones).....	124
Tabla 90: métrica SP (Experimento “Inundación” con 1000 simulaciones).....	125
Tabla 91: métrica HP (Experimento “Inundación” con 1000 simulaciones)	125
Tabla 92: métrica HP (Experimento “Inundación” con 1000 simulaciones)	125
Tabla 93: presupuesto final del proyecto	144

Índice de figuras

Figura 1. Evolución de población utilizando un autómata celular	26
Figura 2. Resultados de la simulación después de 65 años por cada tipo de política.	27
Figura 3. Densidad de población usada para el estudio [7].	28
Figura 4. Propagación de la enfermedad utilizando autómatas celulares.....	28
Figura 5. Dominancia de Pareto.....	31
Figura 6. Frente de Pareto de una función con dos objetivos (Wikipedia).....	31
Figura 7. Pseudo-código de NSGAll.....	35
Figura 8 Ejemplo de comparación de algoritmos[25].	36
Figura 9 Definición matemática de la métrica Spread[21].....	36
Figura 10 Distancias para el cálculo de la métrica Spread[25].....	36
Figura 11 Definición matemática de la métrica Hipervolumen[26].	37
Figura 12. Modelización jerárquica.	40
Figura 13. Fórmula de Interpolación Lineal.....	40
Figura 14. Ejemplo de evaluaciones de movimiento que realiza el autómata celular.....	41
Figura 15. Expresión matemática del modelo “Logit”	41
Figura 16. Modelo conceptual.	57
Figura 17. Diagrama de caso de uso de la aplicación.....	59
Figura 18. Diagrama de secuencia: el usuario modifica parámetros y realiza una simulación ..	65
Figura 19. Diagrama de secuencia: el usuario realiza simulaciones con cambios	66
Figura 20. Diagrama de secuencia: el usuario utiliza la optimización automática	66
Figura 21. Diagrama de clases de la aplicación	70
Figura 22. Diagrama de clases del módulo de optimización (framework JMetal).....	71
Figura 23. Patrón Singleton.....	75
Figura 24. Simulación basada en algoritmo “Logit”	76
Figura 25.Código de colores empleado para representar resultados de la experimentación....	79
Figura 26. Valores discretizados para el módulo de optimización multiobjetivo	81

Figura 27. Valores de las celdas para los factores relevantes comida (izquierda) y agua (derecha)	84
Figura 28. Distribución inicial de la población para el experimento “Plaga”	84
Figura 29. Función utilizada como modelo de atractivo para el experimento “Plaga”	85
Figura 30. Atractivo para el experimento “Plaga”	85
Figura 31. Población Final del Experimento “Plaga”	86
Figura 32. Cambio en el factor relevante “Comida”. Inicial (izquierda), Mapa Intervenido (Derecha).....	86
Figura 33. Intervención en el factor relevante comida (izquierda), agua (derecha) sigue igual. 86	
Figura 34. Atractivo inicial (izquierda) y final (derecha) después del cambio	87
Figura 35. Población Final después de la intervención	87
Figura 36. Soluciones “No Dominadas” (frente de Pareto) (Experimento “Plaga” con 100 simulaciones).....	90
Figura 37. Comparativa de soluciones “No Dominadas” para 2.000 generaciones (Experimento “Plaga” con 100 simulaciones).....	91
Figura 38. Total de soluciones para 2.000 generaciones (Experimento “Plaga” con 100 simulaciones).....	91
Figura 39. Comparativa de soluciones “No Dominadas” para 20.000 generaciones (Experimento “Plaga” con 100 simulaciones).....	92
Figura 40. Comparativa de soluciones “No Dominadas” para 20.000 generaciones (Experimento “Plaga” con 100 simulaciones).....	92
Figura 41. Comparativa de soluciones “No Dominadas” para 40.000 generaciones (Experimento “Plaga” con 100 simulaciones).....	93
Figura 42. Total de soluciones para 40.000 generaciones (Experimento “Plaga” con 100 simulaciones).....	94
Figura 43. Comparativa de soluciones “No Dominadas” para población igual a 10 (Experimento “Plaga” con 100 simulaciones).....	95
Figura 44. Total de soluciones para población 10 (Experimento “Plaga” con 100 simulaciones)	95
Figura 45. Comparativa de soluciones “No Dominadas” para población igual a 50 (Experimento “Plaga” con 100 simulaciones).....	96
Figura 46. Total de soluciones para población 50 (Experimento “Plaga” con 100 simulaciones)	97

Figura 47. Comparativa de soluciones “No Dominadas” para población igual a 100 (Experimento “Plaga” con 100 simulaciones)	97
Figura 48. Total de soluciones para población 100 (Experimento “Plaga” con 100 simulaciones)	98
Figura 49. Soluciones “No Dominadas” (frente de Pareto) (Experimento “Plaga” con 1000 simulaciones).....	100
Figura 50. Comparativa de soluciones “No Dominadas” para 2.000 generaciones (Experimento “Plaga” con 1000 simulaciones).....	101
Figura 51. Total de soluciones para 2.000 generaciones (Experimento “Plaga” con 1000 simulaciones).....	101
Figura 52. Comparativa de soluciones “No Dominadas” para 20.000 generaciones (Experimento “Plaga” con 1000 simulaciones).....	102
Figura 53. Total de soluciones para 20.000 generaciones (Experimento “Plaga” con 1000 simulaciones).....	103
Figura 54. Comparativa de soluciones “No Dominadas” para 40.000 generaciones (Experimento “Plaga” con 1000 simulaciones).....	103
Figura 55. Total de soluciones para 40.000 generaciones (Experimento “Plaga” con 1000 simulaciones).....	104
Figura 56. Comparativa de soluciones “No Dominadas” para población 10 (Experimento “Plaga” con 1000 simulaciones).....	104
Figura 57. Total de soluciones para población 10 (Experimento “Plaga” con 1000 simulaciones)	105
Figura 58. Comparativa de soluciones “No Dominadas” para población 50 (Experimento “Plaga” con 1000 simulaciones).....	106
Figura 59. Total de soluciones para población 50 (Experimento “Plaga” con 1000 simulaciones)	106
Figura 60. Comparativa de soluciones “No Dominadas” para población 100 (Experimento “Plaga” con 1000 simulaciones).....	107
Figura 61. Total de soluciones para población 100 (Experimento “Plaga” con 1000 simulaciones).....	107
Figura 62. Variables iniciales: comida (izquierda) y agua (derecha)	109
Figura 63. Variables finales: comida (izquierda) y agua (derecha)	109
Figura 64. Simulación completa de la variable “comida”	110

Figura 65. Simulación completa de la variable “agua”.....	110
Figura 66. Distribución inicial de la población.....	111
Figura 67. Función utilizada para calcular el Atractivo del experimento “Inundación”	111
Figura 68. Atractivo simulado	112
Figura 69. Simulación de población del experimento “Inundación”	112
Figura 70. Cambio en valor del factor relevante “Comida”	113
Figura 71. Cambio en la variable Comida (izquierda), la variable Agua (derecha) sigue igual .	113
Figura 72. Variables finales: comida (izquierda) modificada por la intervención, agua (derecha) sigue igual.....	113
Figura 73. Atractivo simulado después de aplicar la intervención	114
Figura 74. Simulación de la población después de la intervención	114
Figura 75. Soluciones “No Dominadas” (frente de Pareto) (Experimento “Inundación” con 100 simulaciones).....	117
Figura 76. Comparativa de soluciones “No Dominadas” para 2.000 generaciones (Experimento “Inundación” con 100 simulaciones)	117
Figura 77. Total de soluciones para 2.000 generaciones (Experimento “Inundación” con 100 simulaciones).....	118
Figura 78. Comparativa de soluciones “No Dominadas” para 20.000 generaciones (Experimento “Inundación” con 100 simulaciones)	118
Figura 79. Total de soluciones para 20.000 generaciones (Experimento “Inundación” con 100 simulaciones).....	119
Figura 80. Comparativa de soluciones “No Dominadas” para 40.000 generaciones (Experimento “Inundación” con 100 simulaciones)	119
Figura 81. Total de soluciones para 40.000 generaciones (Experimento “Inundación” con 100 simulaciones).....	120
Figura 82. Comparativa de soluciones “No Dominadas” para población igual a 10 (Experimento “Inundación” con 100 simulaciones)	120
Figura 83. Total de soluciones para población 10 (Experimento “Inundación” con 100 simulaciones).....	121
Figura 84. Comparativa de soluciones “No Dominadas” para población igual a 50 (Experimento “Inundación” con 100 simulaciones)	122

Figura 85. Total de soluciones para población 50 (Experimento “Inundación” con 100 simulaciones).....	123
Figura 86. Comparativa de soluciones “No Dominadas” para población igual a 100 (Experimento “Inundación” con 100 simulaciones).....	123
Figura 87. Total de soluciones para población 100 (Experimento “Inundación” con 100 simulaciones).....	124
Figura 88. Soluciones “No Dominadas” (frente de Pareto) (Experimento “Inundación” con 1000 simulaciones).....	126
Figura 89. Comparativa de soluciones “No Dominadas” para 2.000 generaciones (Experimento “Inundación” con 1000 simulaciones)	127
Figura 90. Total de soluciones para 2.000 generaciones (Experimento “Inundación” con 1000 simulaciones).....	127
Figura 91. Comparativa de soluciones “No Dominadas” para 20.000 generaciones (Experimento “Inundación” con 1000 simulaciones)	128
Figura 92. Total de soluciones para 20.000 generaciones (Experimento “Inundación” con 1000 simulaciones).....	128
Figura 93. Comparativa de soluciones “No Dominadas” para 40.000 generaciones (Experimento “Inundación” con 1000 simulaciones)	129
Figura 94. Total de soluciones para 40.000 generaciones (Experimento “Inundación” con 1000 simulaciones).....	129
Figura 95. Comparativa de soluciones “No Dominadas” para población 10 (Experimento “Inundación” con 1000 simulaciones)	130
Figura 96. Total de soluciones para población 10 (Experimento “Inundación” con 1000 simulaciones).....	130
Figura 97. Comparativa de soluciones “No Dominadas” para población 50 (Experimento “Inundación” con 1000 simulaciones)	131
Figura 98. Total de soluciones para población 50 (Experimento “Inundación” con 1000 simulaciones).....	131
Figura 99. Comparativa de soluciones “No Dominadas” para población 100 (Experimento “Inundación” con 1000 simulaciones)	132
Figura 100. Total de soluciones para población 100 (Experimento “Inundación” con 1000 simulaciones).....	132
Figura 101. Planificación final	142

CAPÍTULO 1: INTRODUCCIÓN

El presente Proyecto de Fin de Carrera de la titulación de Ingeniería Informática tiene como finalidad diseñar, implementar y experimentar con un sistema capaz de asistir al usuario en la toma de decisiones sobre intervenciones para modificar la distribución geográfica de poblaciones.

En este capítulo se presenta los propósitos de este proyecto, así como los objetivos y la estructura interna del mismo. En primer lugar se muestra la motivación que dio origen al presente proyecto de fin de carrera, a continuación los objetivos que se pretende alcanzar, la metodología seguida para desarrollar el proyecto y, para terminar, una visión general de los distintos apartados que contiene este documento.

1.1 Motivación del proyecto

Durante los últimos años, ha sido fácil ver el incremento del interés por las cuestiones medioambientales. Este interés se ha visto reflejado en un creciente interés por mejorar la gestión de los recursos. El proyecto que se presenta se encuadra en este campo. Se trata de explorar un enfoque que facilite estos esfuerzos. Concretamente, se pretende diseñar, implementar y realizar unas primeras pruebas con una herramienta que permita simular y asistir en la toma de decisiones relacionadas con el movimiento de poblaciones. Los destinatarios serían, entre otros, personas encargadas de realizar tareas de evaluación de impacto ambiental de la construcción de infraestructuras, personas al cargo de la conservación de una especie, gestores forestales o encargados de grandes explotaciones agrícolas.

Las personas mencionadas podrían estar interesadas en predecir y tener la capacidad de actuar sobre la distribución geográfica de determinadas especies por distintas razones. Un gestor público podría desear realizar una planificación urbanística y no perjudicar con ello a un elemento de la fauna y/o flora local especialmente valiosa, el titular de una explotación agrícola podría querer proteger sus cosechas frente a una especie concreta etc. En todos estos casos, un buen punto de partida sería tener la posibilidad de predecir la forma en la que se extendería la población objeto de interés.

Una aproximación a este problema puede ser mediante una simulación de la expansión de la población en los distintos instantes de tiempo. Para hacer esto, será necesario establecer un modelo que explique el proceso por el que una población se extiende a espacios contiguos, podría mantenerse en el mismo lugar, o desaparecer. Este modelo de movimiento posiblemente estará ligado al atractivo de un lugar. Para determinar este atractivo, sería necesario identificar factores relevantes que permiten la existencia o ausencia de la población de estudio. También sería importante ver, en su caso, como se espera que cambien los factores relevantes, bien por acciones del hombre: como obras, contaminantes, etcétera; o por acciones naturales: glaciación, erupciones, estaciones del año.

En general, los factores relevantes dependerán de la población de estudio. Por ejemplo, podemos considerar relevantes para un tipo de encina: la humedad del ambiente, horas de luz solar y agua; en cambio para unos ratones podrían definirse como factores relevantes: los depredadores, la comida, el agua y la temperatura. No todos tendrán la misma importancia para una población, de hecho es posible que algunos factores relevantes puedan perjudicar a la especie, provocando su ausencia (ejemplo de esto pueden ser los depredadores en el caso de los ratones). Por este motivo, es necesario definir un “modelo de atractivo” conectado con el “modelo de desplazamiento”

El atractivo de una zona permitirá discriminar si una población de una zona adyacente puede extenderse a esta (o permanecer en ella) o no, teniendo en cuenta que una población intentará expandirse en todas las direcciones y ocupar su espacio inicial, si es posible por el valor del atractivo en estas zonas.

Este funcionamiento por sí mismo permitiría conocer qué sucederá en el futuro con el desarrollo de una determinada especie pero no permitiría intervenir. Por ejemplo, el modelo ayudaría a predecir que el granjero posiblemente perderá la cosecha por la invasión de ratones, pero no hacer nada al respecto. Es por ello que resulta necesario contar con una forma de predecir qué podría suceder al intervenir sobre los factores relevantes.

Es esperable que los cambios sobre los factores tengan un coste económico asociado. Este coste asociado de las intervenciones puede ser diferente en cada instante y para cada factor relevante (por ejemplo, cambiar las horas de luz que reciben las encinas de nuestro anterior ejemplo saldría enormemente más caro que modificar la cantidad de agua que estas reciben). También puede haber diferencias en cuanto a la zona a intervenir (no tiene el mismo coste sulfatar una granja que tiene un acceso directo a una carretera que otra que está entre montañas en una zona inaccesible, por ejemplo).

Las intervenciones, y el coste asociado de estas, pueden tener características más complejas, es posible que las intervenciones deban mantenerse en el tiempo sin que se mantenga un coste (por ejemplo, construir nidos para alguna especie de pájaro que solo requieren coste en un primer instante, pero que el factor relevante de acondicionamiento se vería afectado para todos los instantes), o es posible que la intervención se mantenga y el coste también (por ejemplo, instalar y aprovisionar un comedero).

Por último, simular de nuevo teniendo en cuenta las intervenciones, podrá servir como modelo de ayuda al usuario de nuestra granja sobre qué decisiones tomar, dependiendo de las distintas intervenciones posibles y el coste asociado a estas.

Pero existe un problema a este planteamiento, y es la complejidad que puede alcanzar un sistema de estas características, dado que una especie puede tener muchos factores relevantes distintos, con muchos tipos de intervención en cada uno, de estos factores relevantes en cada instante o lugar.

Esta complejidad, introduciría una nueva necesidad importante, que sería contar con asesoramiento sobre las intervenciones a efectuar. Este consejo se podría realizar mediante un sistema de automatización de intervenciones. Por ejemplo para una hipotética población que requiera de veinte factores relevantes y reglas complejas en el modelo de atractivo, la aplicación podría presentar varias soluciones con distintas probabilidades de presencia de población en un lugar de interés y distintos costes,

para que el usuario elija la más conveniente (por ejemplo, el usuario podría preferir un gasto de 100.000 euros con probabilidad 95% de presencia que una solución que asegure la presencia al 100% con coste 500.000 euros)

En conclusión, en este proyecto se explorará la posibilidad de ofrecer una solución que facilite la toma de decisiones en intervenciones en terrenos. El objetivo será asegurar la presencia/ausencia de poblaciones en un determinado lugar y momento. Para esto permitirá hacer simulaciones, valorar las consecuencias de posibles iniciativas de los usuarios, y ofrecerá un abanico de alternativas de acción que garanticen la presencia con distinta probabilidad y coste mínimo.

1.2 Objetivos del proyecto

El objetivo que nos planteamos es crear una herramienta que permita simular el movimiento de una población en base a los factores relevantes que influyen en el desarrollo de esta. Para ello, se requiere:

- Representar un entorno geográfico con un modelo de distribución de recursos.
- Establecer un modelo de atractivo del terreno, que permita ponderar los factores relevantes y el efecto que estos tienen sobre la población.
- Un modelo de evolución de población que permita simular la expansión de esta, dependiendo del atractivo de las zonas.

Necesitamos poder modificar los factores relevantes para evaluar el impacto de las intervenciones en el movimiento de la población, y calcular el coste asociado a estas intervenciones. Para esto necesitamos un módulo de intervenciones que permita:

- Establecer coste a las intervenciones, dependiendo del instante, del factor relevante y de la zona (o celda) intervenida.
- Considerar intervenciones puntuales o que se mantengan en el tiempo

También necesitaremos un módulo de optimización que permita automatizar el proceso de toma de decisiones de intervención teniendo en cuenta dos objetivos: el coste y la probabilidad de presencia de la población.

Por tanto, el objetivo a cumplir en este proyecto es establecer un marco de trabajo estable y completo, que ofrezca a futuros desarrolladores funcionalidades

predefinidas y la posibilidad de crear funcionalidades propias que permitan adaptar este enfoque a otras áreas como la sociología o economía.

Por último, como un objetivo personal en este proyecto pretendo aprender y practicar el diseño y la implementación de interfaces de usuarios, intentando que sean sencillas e intuitivas, además de realizar tareas de investigación y experimentación.

1.3 Metodología

Para mejorar las probabilidades de éxito del proyecto es necesario establecer una metodología de trabajo y ajustarse a ella. La metodología se divide en cuatro fases diferenciadas, las cuales se describen a continuación:

1. **Planteamiento del problema:** fase de estudio de viabilidad del proyecto en la que el tutor explica el proyecto en cuestión, el dominio del problema, así como las restricciones y posibilidades o vías de investigación que ofrece el proyecto.
2. **Fase de documentación sobre el problema:** fase de recogida de información sobre el proyecto, se extrae información tanto del dominio como de la tecnología a utilizar, también se obtiene mucha información sobre los algoritmos genético multiobjetivo para abordar el problema de la optimización de intervenciones.
3. **Desarrollo de la solución:** en esta fase se desarrollará la aplicación, esta fase será documentada en esta memoria y comprenderá análisis, diseño e implementación de la aplicación. Para un desarrollo más rápido y estandarizado se utilizarán distintos framework detallados durante la presente memoria.
4. **Experimentación y evaluación de la solución:** en esta fase se realizan las pruebas de integración de la aplicación y se experimenta con la aplicación de forma que se obtengan resultados sobre la optimización en base a la funcionalidad construida.
5. **Desarrollo de la documentación:** fase en la que se recoge el proceso de elaboración de la presente memoria, recopilando las cuestiones más importantes referentes al proyecto. Esta fase se desarrollará en paralelo a las demás actividades ya que no es necesario esperar a haber evaluado la aplicación para describir el proceso de desarrollo de la misma.

1.4 Visión general del documento

Este documento se organiza de la siguiente manera:

- En el primer apartado, **Introducción**, se aclara la motivación que ha hecho surgir el proyecto, así como los objetivos que se plantean.
- El segundo apartado, **Estado de la cuestión**, incluye una introducción al modelado de sistemas ecológicos utilizando autómatas celulares.
- El tercer apartado, **Análisis**, comienza con la definición del estudio de viabilidad del sistema que contendrá el alcance del mismo. A continuación se incluirá la especificación de requisitos, tanto de usuario como de software, y casos de uso.
- El apartado cuatro, **Diseño**, describe el diagrama de las clases que se implementarán. Para cada una de las clases, se comentan sus atributos y métodos más importantes dentro del desarrollo del proyecto. La Entrada/Salida que se proporciona y las interfaces que se requieren para el desarrollo del proyecto.
- En el apartado cinco, **Implementación**, se explican los aspectos más importantes, separado en módulos, del desarrollo desde el punto de vista del usuario.
- En el apartado seis, **Experimentación**, se detallan los experimentos que se han realizado, en este punto, además, se ofrece un análisis de los resultados obtenidos de la ejecución de los distintos experimentos.
- En el séptimo apartado **Conclusiones**, se recogen las conclusiones a las que se ha llegado tras el desarrollo del proyecto, así como un análisis para comprobar si se han cumplido los objetivos marcados.
- En el octavo apartado, **Líneas futuras**, contiene las posibles mejoras que se podrían aplicar en este producto software en futuros desarrollos, tomando este proyecto como base.
- Por último, se recogen los anexos.
 - En primer lugar tenemos el **manual de usuario**, que acompañará a la aplicación y explica el funcionamiento de la misma.
 - A continuación se incluye la **planificación** que se ha seguido, y el **presupuesto** derivado del desarrollo de este proyecto.
 - Finalmente se incluye el anexo con las **referencias**.

CAPÍTULO 2: ESTADO DE LA CUESTIÓN

En este capítulo se presenta el contexto del problema que este proyecto trata de solucionar, presentando un modelado de sistemas ecológicos de movimientos de poblaciones. Se describirán los algoritmos genéticos multiobjetivo que se han empleado para la implementación del proyecto, y se mencionarán varias aproximaciones al problema que se ha realizado usando otras técnicas.

También se hará una introducción a los autómatas celulares definiendo la necesidad de esta técnica y su utilización en proyectos similares, así como del tipo de autómata celular que se utilizará en el desarrollo del proyecto (autómata celular estocástico).

2.1 Ecological Modelling

Podemos definir “Ecological Modelling” como la *“conceptualización e implementación de simulaciones por ordenador del comportamiento de los seres vivos”*[1]. Bajo esta definición, podemos estudiar la necesidad de desarrollos de proyectos en esta área y las distintas iniciativas que se han producido en este campo y que pueden ayudar a completar el análisis y desarrollo de una aplicación de este tipo.

Las herramientas más adecuadas que se aplican para “Ecological Modelling”, más concretamente para los estudios sobre distribución geográfica de poblaciones, son los autómatas celulares, en contraposición de modelos formados por secuencias recursivas, tablas de vida, modelos de matrices y ecuaciones diferenciales ordinarias o parciales [24]. Por este motivo es utilizado en multitud de proyectos [2][4][5].

Un autómata celular (Wolfram, 1984) es una cuadrícula cartesiana de células idénticas cada una de ellas en uno de un número finito de estados. En cada paso de tiempo, el estado de una célula se actualiza de acuerdo a una función de los estados de sus células vecinas y ella misma. Podemos definir de forma más generalista un autómata celular como un modelo matemático para un sistema dinámico que evoluciona en pasos discretos [3].

Un autómata celular, tiene las siguientes características:

- El espacio está formado por un conjunto finito de subespacios homogéneos, llamadas **celdas** distribuidas en una rejilla regular n-dimensional.
- Cada celda debe estar en un único estado (definido en un conjunto finito de estados asociados al espacio del autómata), en un determinado instante de tiempo.
- Cada celda tiene asignado un estado inicial (configuración inicial).
- El estado de las celdas cambia de un instante a otro de acuerdo a un conjunto de reglas de evolución comunes a todas las celdas.
- Las reglas que definen la evolución de los estados del autómata son esencialmente una máquina de estado finito, especificadas en una tabla de reglas (también conocida como función de transición), con una entrada para todas las posibles configuraciones del vecindario.
- El vecindario de una celda está formado por las celdas adyacentes.
- Cada celda del autómata está sincronizada, de forma que se establecen los cambios de estado al mismo tiempo, según las Reglas de Evolución.

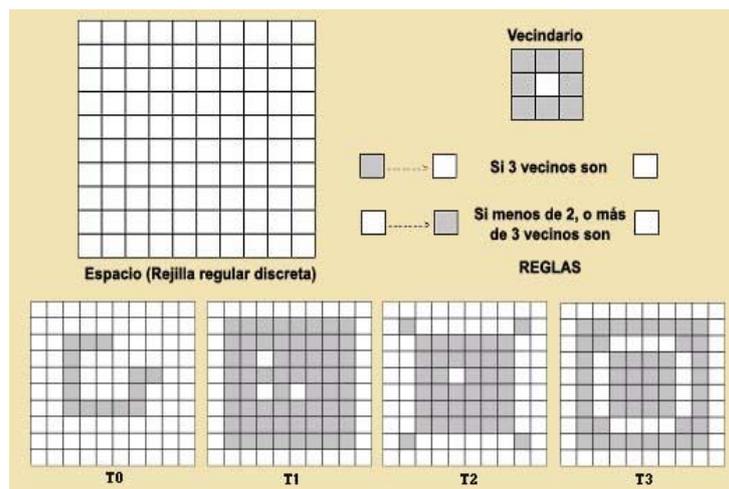


Figura 1. Evolución de población utilizando un autómata celular

El mejor ejemplo de automáta celular podemos encontrarlo en “El Juego de la Vida”, diseñado por el matemático John Horton Conway en 1970. Se caracteriza por ser un autómata celular con dos posibles estados para cada celda (activo e inerte). El estado depende de su estado actual y el de sus 8 vecinas para evolucionar. El universo es cerrado e ilimitado siendo vecinas las celdas superiores de las inferiores y viceversa. Cada generación surgirá en un instante determinado de tiempo reemplazando completamente a la anterior. La evolución es discreta y simultánea.

Las reglas que definió para la evolución de las células son:

- Muerte: una célula se inactiva (o permanece inactiva) si existen menos de 2 o más de 3 vecinas activas.
- Conservación: una célula mantiene su estado si existen solo 2 células vecinas activas.
- Nacimiento: una célula cobra actividad (o permanece activa si ya lo estaba) cuando la rodean 3 células activas.

Con estas sencillas reglas y dependiendo de la configuración inicial, la cual es aleatoria, se podrán observar comportamientos complejos tales como la extinción, oscilación, estabilización (creando fractales en algunos casos), proliferación compleja, interacciones a distancia, etcétera.

Las razones para aplicar un autómata celular en vez de ecuaciones diferenciales parciales, es que estos modelos consumen más tiempo computacional que el autómata celular correspondiente (Gerhardt et al., 1990).

En el estudio *“Celular Automata as a paradigm for Ecological Modelling”* se concluye que los autómatas celulares son un formalismo adecuado para el modelado orientado en el espacio de procesos ecológicos y se discuten algunas limitaciones de este formalismo, mencionando algunas extensiones y generalizaciones que pueden remediar estas deficiencias. En el estudio también se concluye la necesidad de máquinas altamente paralelas para este formalismo [4].

La generación de comportamientos complejos a partir de reglas muy simples conduce a la consideración de características estadísticas en la evolución de los autómatas celulares. Si se incorpora a la regla de transición algún elemento probabilístico o estocástico, el Autómata Celular se denomina estocástico.

Los autómatas celulares estocásticos proporcionan modelos más realistas de fenómenos naturales, por los que son especialmente interesantes para el *“Ecological Modelling”*.

Entre los casos de aplicaciones de esta técnica en dominios de *“Ecological Modelling”* se podría citar el caso del análisis de la bio-complejidad de la deforestación de la reserva forestal de Caparo (Venezuela) [5], en este estudio se utiliza un modelo *“multi-agente”* con tres tipos de agentes: los colonos, el gobierno y los concesionarios de madera. Las herramientas utilizadas incluyen Galatea (multi-agentes), Actilog (descripción de la regla), y SpaSim (autómatas celulares).

El estudio se realiza mediante tres simulaciones donde se comparan los distintos resultados que ofrecería una política:

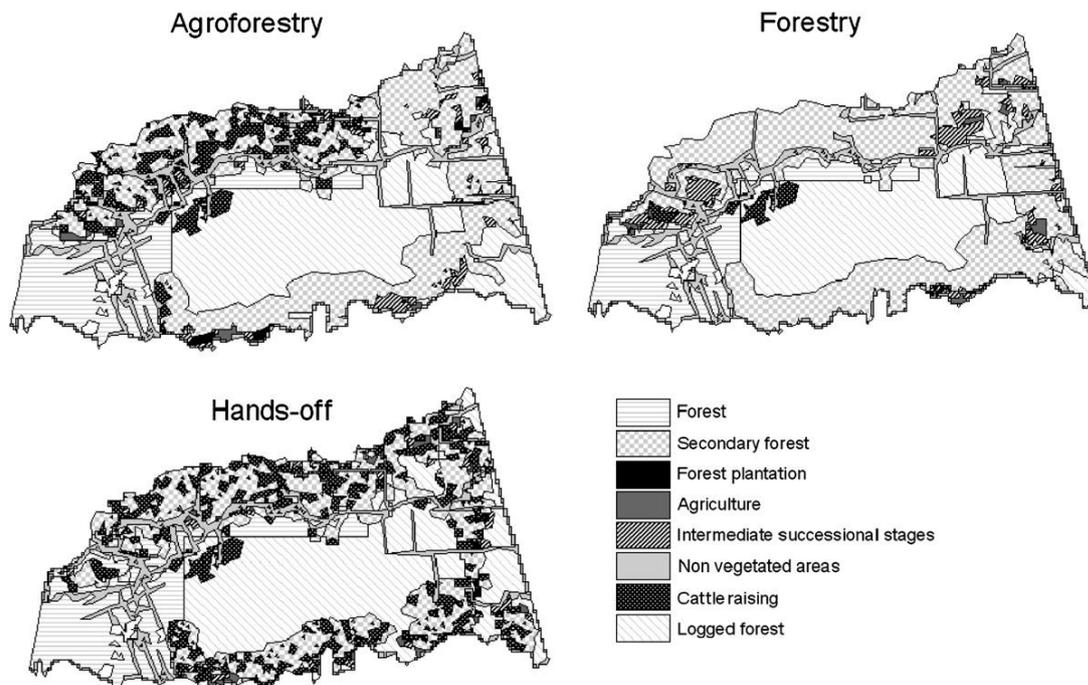


Figura 2. Resultados de la simulación después de 65 años por cada tipo de política.

En el estudio “*A Cellular Automaton Framework for Infectious Disease Spread Simulation*” [7] se trata la propagación de enfermedades infecciosas en el estado de Tyrol con 660.000 habitantes en el momento del estudio.

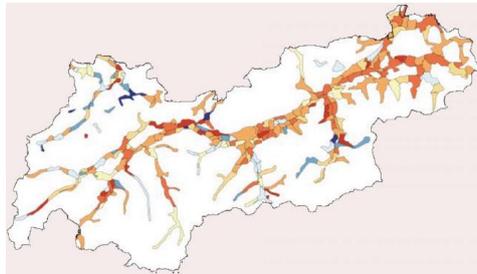


Figura 3. Densidad de población usada para el estudio [7].

Este estudio se implementa mediante un autómata celular, en el que cada celda es una porción del estado de Tyrol. En cada instante (un total de 50 que representan 3 meses reales), se computan los periodos de latencia, infección e incubación, el tiempo de recuperación y de incubación y las tasas de nacimiento, defunción e infección por contacto e infecciones espontáneas. También se tienen en cuenta los movimientos de población entre zonas y la inmigración.

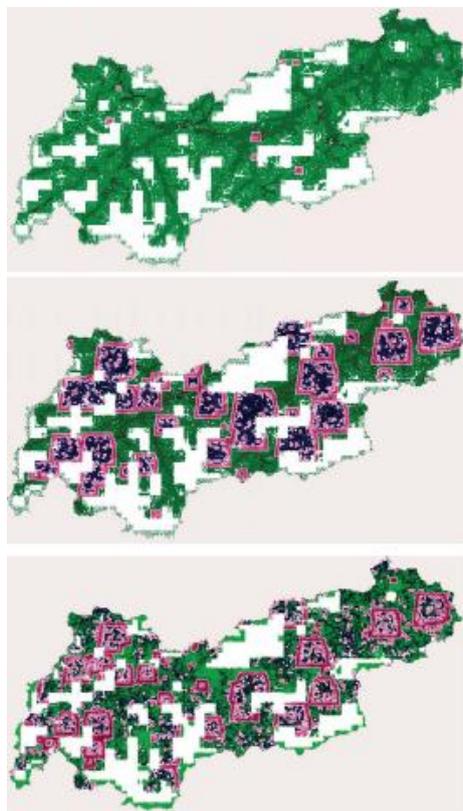


Figura 4. Propagación de la enfermedad utilizando autómatas celulares

En el gráfico se puede observar que después de 20 días hay algunos brotes en el oeste del estado, así como en el este, lo que se traduce forma el hecho de que las personas se mueven de una célula a otra. En la segunda (100 días después de la infección) se observa que en la capital la mayoría de las personas están en estado de recuperación (las personas que no muren tienen una inmunidad temporal). La última imagen muestra una instantánea en el punto de tiempo 200 días después del brote, donde las personas se infectan de nuevo en un segundo brote de menor intensidad.

Otros estudios en los que se relacionan autómatas celulares con la regeneración de suelos contaminados [6], modelización eco-hidráulica [8], propagación de incendios forestales [9], simulación de especies de plantas con estrategias de vida predefinidas [10], demuestran la adecuación de los autómatas celulares para representar la evolución de un sistema dinámico e un área geográfica.

2.2 Optimización mediante Algoritmos Genéticos Multiobjetivo

El módulo de optimización pretende ofrecer un soporte a la toma de decisiones de planificación sobre el entorno, esto se conseguirá mediante un conjunto de alternativas de relación coste - probabilidad esperada de permanencia o ausencia de población.

2.2.1 Introducción a la optimización multiobjetivo

Para explicar las diferentes técnicas empleadas para resolver el módulo de optimización, se comienza definiendo en qué consiste la optimización: *“optimizar consiste en encontrar la solución óptima de entre un conjunto finito de soluciones alternativas”* [14]. Mientras que en el contexto mono-objetivo la calidad de una determinada solución viene dada directamente por el valor de una función objetivo, en el contexto multi-objetivo se consideran varios objetivos contrapuestos. Por esta razón, lo normal es que no exista un solo óptimo global, sino varias soluciones que se consideran óptimos por tener mejores valores en alguno de los distintos objetivos marcados.

Los problemas de optimización se caracterizan por [15]:

1. Variables de decisión. Son una abstracción de los criterios relevantes del problema y que se representan como entidades numéricas, las que serán determinadas por la optimización.
2. Funciones objetivo. Son las relaciones que deben cumplir las variables de decisión y son expresadas como funciones computables.
3. Restricciones. Son cotas dentro de las que se puede asignar valor a las variables de decisión para que la solución sea factible.

La optimización vectorial, multi-objetivo se puede definir como *“el problema de encontrar un vector de variables de decisión que satisfaga ciertas restricciones y optimice una función vectorial cuyos elementos representen las funciones objetivo. Estas funciones forman una descripción matemática de los criterios de desempeño que usualmente están en conflicto entre sí y que suelen medir en unidades diferentes. Por lo tanto el término optimizar significa encontrar una solución tal que proporcione valores para todos los objetivos que resulten aceptables para el diseñador”*. [16]

El concepto de eficiencia de Pareto, también llamado óptimo de Pareto, Pareto-optimalidad u óptimo paretiano en honor de su introductor, Vilfredo Pareto, es un

concepto aplicado en economía, ciencias sociales e ingeniería. El óptimo de Pareto establece como eficiente aquella situación en la cual se cumple que no es posible beneficiar a más individuos en un sistema sin perjudicar a otros. [17]

En general, la solución en el sentido de Pareto al problema de optimización multi-objetivo no será única: la solución estará formada por el conjunto de todos los vectores no-dominados, a los que se conoce con el nombre de conjunto de no-dominados o frente de Pareto [17].

Dado un vector $\mathbf{u} = (u_1, \dots, u_k)$, se dice que domina a otro vector $\mathbf{v} = (v_1, \dots, v_k)$ si y sólo si:

$$\forall i \in \{1, \dots, k\}, u_i \leq v_i \quad \text{y} \quad \exists i_0 \in \{1, \dots, k\} \mid u_{i_0} < v_{i_0}$$

Figura 5. Dominancia de Pareto.

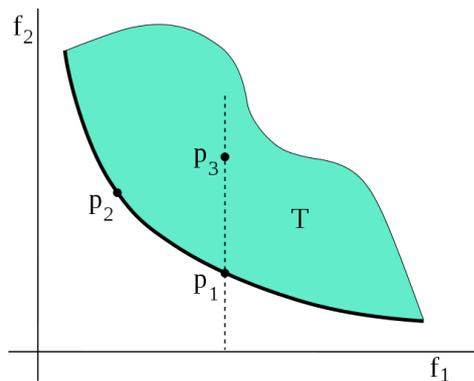


Figura 6. Frente de Pareto de una función con dos objetivos (Wikipedia).

En la figura anterior se representa, con trazo grueso, el frente de Pareto de una función con dos objetivos. El área coloreada (T) representa las distintas soluciones alcanzables de dicha función objetivo. Se puede observar que no existe ningún punto perteneciente a T que mejore las soluciones del frente de Pareto. Se puede observar a su vez que P1 siempre tendrá el mismo valor de f1 que cualquier punto que esté en vertical (ej. P3), pero siempre tendrá un valor mejor de f2. Entre P2 y P1, se puede ver que cada uno mejora uno de los objetivos a costa de empeorar el otro. Nunca habrá un punto fuera del frente de Pareto que mejore de forma simultánea los dos objetivos.

2.2.2 Optimización evolutiva

Para el desarrollo de este proyecto se hará uso de la optimización multiobjetivo evolutiva. Podemos definir los algoritmos evolutivos como métodos de optimización y búsqueda de soluciones basados en los postulados de la evolución biológica. En ellos se mantiene un conjunto de entidades que representan posibles soluciones, las cuales se mezclan, y compiten entre sí, de tal manera que las más aptas son capaces de prevalecer a lo largo del tiempo, evolucionando hacia mejores soluciones cada vez.

Los algoritmos evolutivos, y la computación evolutiva, son una rama de la inteligencia artificial. Son utilizados principalmente en problemas con espacios de búsqueda extensos y no lineales, en donde otros métodos no son capaces de encontrar soluciones en un tiempo razonable.

Siguiendo la terminología de la teoría de la evolución, las entidades que representan las soluciones al problema se denominan individuos o cromosomas, y el conjunto de estos, población. Los individuos son modificados por operadores genéticos, principalmente el sobrecruzamiento, que consiste en la mezcla de la información de dos o más individuos; la mutación, que es un cambio aleatorio en los individuos; y la selección, consistente en la elección de los individuos que sobrevivirán y conformarán la siguiente generación. Dado que los individuos que representan las soluciones más adecuadas al problema tienen más posibilidades de sobrevivir, la población va mejorando gradualmente.

Suele hablarse de tres paradigmas principales de algoritmos evolutivos:

- Programación evolutiva
- Estrategias evolutivas
- Algoritmos genéticos

En el desarrollo del presente proyecto, se tratará especialmente este último paradigma (algoritmos genéticos, John Henry Holland 1970). Son llamados así porque se inspiran en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una Selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados.

Un algoritmo genético puede presentar diversas variaciones, dependiendo de cómo se aplican los operadores genéticos (cruzamiento, mutación), de cómo se realiza

la selección y de cómo se decide el reemplazo de los individuos para formar la nueva población. En general, el pseudocódigo consiste de los siguientes pasos:

- Inicialización: se genera aleatoriamente la población inicial, que está constituida por un conjunto de cromosomas los cuales representan las posibles soluciones del problema. En caso de no hacerlo aleatoriamente, es importante garantizar que dentro de la población inicial, se tenga la diversidad estructural de estas soluciones para tener una representación de la mayor parte de la población posible o al menos evitar la convergencia prematura.
- Evaluación: a cada uno de los cromosomas de esta población se aplicará la función de aptitud para saber qué tan "buena" es la solución que se está codificando.
- Condición de término: el algoritmo genético se deberá detener cuando se alcance la solución óptima, pero ésta generalmente se desconoce, por lo que se deben utilizar otros criterios de detención. Normalmente se usan dos criterios: correr el algoritmo genético un número máximo de iteraciones (generaciones) o detenerlo cuando no haya cambios en la población. Mientras no se cumpla la condición de término se hace lo siguiente:
 - Selección: después de saber la aptitud de cada cromosoma se procede a elegir los cromosomas que serán cruzados en la siguiente generación. Los cromosomas con mejor aptitud tienen mayor probabilidad de ser seleccionados.
 - Sobrecruzamiento: el cruzamiento es el principal operador genético, representa la reproducción sexual, opera sobre dos cromosomas a la vez para generar dos descendientes donde se combinan las características de ambos cromosomas padres.
 - Mutación: modifica al azar parte del cromosoma de los individuos, y permite alcanzar zonas del espacio de búsqueda que no estaban cubiertas por los individuos de la población actual.
 - Reemplazo: una vez aplicados los operadores genéticos, se seleccionan los mejores individuos para conformar la población de la generación siguiente

La primera implementación de un algoritmo evolutivo multiobjetivo fue realizada por Schaffer en la década de los ochenta con el Vector Evaluated Genetic

Algorithm (VEGA). Hasta ese momento, la forma habitual de plantear la solución a los problemas multiobjetivo era transformarlos en variantes monoobjetivo que se pudiesen abordar con las heurísticas tradicionales (por ejemplo, ponderando los objetivos en una sola función). Con VEGA se inicia la primera generación de algoritmos evolutivos multiobjetivo que se caracterizan por el uso de mecanismos de selección basados en el concepto de optimalidad a través de la dominancia de Pareto. Los algoritmos más representativos de esta generación son NSGA de 1994, NPGA de 1994 y MOGA de 1993.

La implementación del mecanismo de elitismo en el contexto de optimización multiobjetivo dio paso a la segunda generación de algoritmos evolutivos multiobjetivo. El elitismo en este contexto normalmente es implementado a través de una población externa o secundaria, en la que se almacena a los individuos no-dominados encontrados durante el proceso de búsqueda. Los principales algoritmos de esta segunda generación son SPEA de 1999, SPEA2 de 2001, PAES de 2000, NSGAI de 2000, NPGA2 de 2001, PESA de 2000 y micro-GA de 2001.

De las diferentes técnicas de optimización multiobjetivo evolutivas que se han propuesto, para los fines de este trabajo se considera NSGAI. Este algoritmo es uno de los de referencia [11] y se ha usado con éxito en muchos dominios de aplicación. El algoritmo es una evolución de NSGA.

El "*Nondominated Sorting Genetic Algorithm*" (NSGA) fue propuesto por Kalyanmoy Deb y uno de sus estudiantes de posgrado en 1994. NSGA se basa en el uso de varias capas de clasificación de los individuos. Los individuos que son no dominados con respecto a toda la población son removidos y se les asigna una "falsa" aptitud.

Posteriormente, se obtienen los individuos no dominados de la población restante (o sea, de la segunda capa) y se les asigna una aptitud "falsa" menor que la asignada a los individuos de la primera capa. Luego se les elimina de la población. El proceso continúa hasta que todos los individuos de la población hayan sido clasificados.

Para mantener diversidad, el NSGA usa compartición de aptitud (en el espacio de las variables de decisión), la cual se aplica sobre los valores "falsos" de aptitud asignados a cada individuo.

El "*Nondominated Sorting Genetic Algorithm II*" (NSGA-II) es una versión mejorada del NSGA que fue propuesta en 2000 por Deb y sus colaboradores. En este caso, se utiliza un operador de "crowding". El NSGA-II utiliza un esquema de selección más (+) en el cual la población de padres se compara con la población de hijos. El NSGA-II además de contar con el uso de elitismo es mucho más eficiente (computacionalmente hablando) que el NSGA y es un algoritmo muy competitivo.

El pseudo-código de NSGAIi es el siguiente:

```
Initialize Population
  Generate random population (size M)
  Evaluate Objective Values
  Assign Rank (level) Based on Pareto Dominance - "sort"
  Generate Child Population
    Binary Tournament Selection
    Recombination and Mutation
For i = 1 to Number of Generations
  With Parent and Child Population
    Assign Rank (level) Based on Pareto - "sort"
    Generate sets of non-dominated fronts
    Loop (inside) by adding solutions to next generation
      starting from the "first" front until M individuals found
      determine crowding distance between points on each front
    Select points (elitist) on the lower front (with lower rank)
      and are outside a crowding distance
    Create next generation
      Binary Tournament Selection
      Recombination and Mutation
    Increment generation index
End of Outside Loop
```

Figura 7. Pseudo-código de NSGAIi.

2.2.3 Métricas

Al contrario que para el caso de la optimización monoobjetivo en la que es sencillo comparar si un resultado es mejor que otro, la comparación de frentes es menos directa. Hay casos en los que es difícil determinar qué frente es mejor que otro, especialmente cuando varios puntos del primer frente dominan al del segundo, pero otros puntos del segundo dominan a los del primero (ver siguiente figura). Por otro lado, no todos los frentes que se puedan haber conseguido tras el proceso de optimización tienen por qué tener el mismo número de soluciones o la misma distribución (las soluciones pueden estar más concentradas en determinadas zonas del frente o distribuidas de forma más uniforme. Para comparar los frentes resultantes de la optimización multiobjetivo se requieren métricas. En el contexto de este proyecto se utilizarán dos: Spread (S) e Hipervolumen (H).

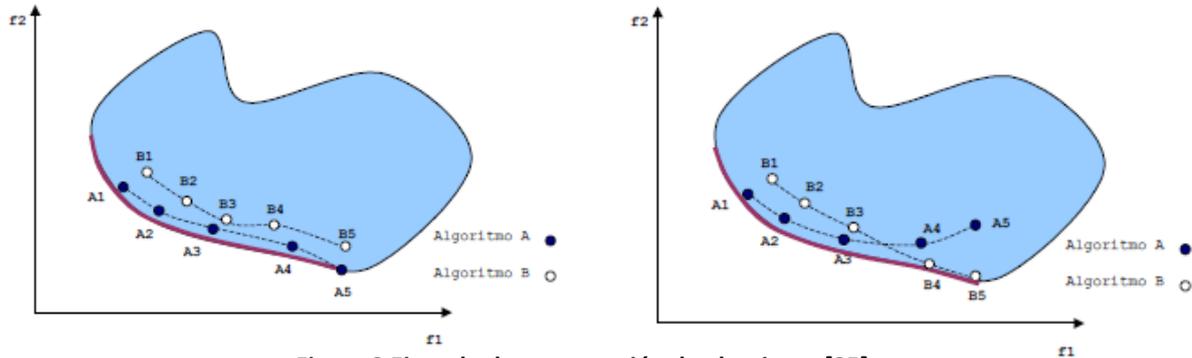


Figura 8 Ejemplo de comparación de algoritmos[25].

La métrica Spread presentada por Deb et al. (2000), propone utilizar como información adicional la distancia a los "extremos" (puntos con menores valores en cada una de las funciones objetivo) del frente de Pareto real para tener una medida más precisa de la cobertura del frente [25]. Esta métrica se define como:

$$\Delta = \frac{d_f + d_t + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_t + (N - 1)\bar{d}}$$

Figura 9 Definición matemática de la métrica Spread[21].

Donde d_i es la distancia Euclídea entre dos soluciones consecutivas y \bar{d} es la media de estas distancias, d_f y d_t son las distancias euclídeas a los puntos "extremos" del frente óptimo en el espacio objetivo.

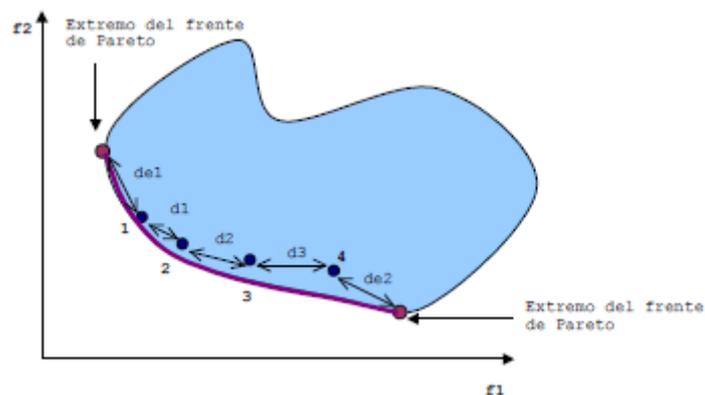


Figura 10 Distancias para el cálculo de la métrica Spread[25].

La métrica hipervolumen (Zitzler99a) es un indicador combinado de convergencia y diversidad que calcula el volumen, en el espacio de objetivos, cubierto por los miembros de un conjunto de soluciones no dominadas para problemas en los que todos los objetivos deben ser minimizados. Esta métrica se define de la siguiente manera:

$$HV = \text{volume} \left(\bigcup_{i=1}^{|Q|} v_i \right)$$

Figura 11 Definición matemática de la métrica Hipervolumen[26].

Para cada solución i que pertenece a Q , un hipercubo v_i se construye utilizando un punto de referencia W (que puede estar compuesto por la peor solución para cada objetivo, por ejemplo) y la solución i como las esquinas de la diagonal del hipercubo.

CAPÍTULO 3: ANÁLISIS

En este capítulo se va a realizar el análisis del sistema. Se empezará explicando el alcance del sistema, es decir, en qué consiste, qué funcionalidades tendrá, etcétera.

Posteriormente se incluirán los requisitos de usuario, divididos en requisitos de capacidad y de restricción, seguidos de los requisitos de software, funcionales y no funcionales.

Finalmente, se concluirá el capítulo con el análisis de los casos de uso, los diagramas de secuencia más relevantes y el diagrama de actividad del sistema.

3.1 Alcance del sistema

En esta sección se estudia el alcance de la necesidad planteada para este proyecto, realizando una descripción general de la misma. Se inicia el estudio de los requisitos y se analizan las posibles restricciones, tanto generales como específicas, que puedan condicionar el estudio y la planificación de las alternativas de solución que se propongan.

Todos los módulos serán intercambiables y se estima que habrá grandes cambios en estos a lo largo del tiempo de vida del software generado para este proyecto, por lo que se debe realizar una implementación lo suficientemente genérica para soportar todos los cambios que se puedan dar y programar defensivamente para asegurar que los módulos que se añadan con posterioridad no comprometan la integridad del código generado para el presente proyecto.

3.1.1 Estructura general

El problema que se pretende resolver con este sistema es recrear un terreno dado (en base a los **factores relevantes** que condicionan la existencia de la población en estudio) en los diversos instantes indicados por el usuario.

La modelización de los factores relevantes puede estar basada en una visión lógica de la naturaleza. La jerarquía más intuitiva es la siguiente:

- **Celda:** valor de una zona para un factor relevante, en un instante determinado. El concepto celda también puede referirse al valor de una zona que indica la presencia o ausencia de población. La granularidad para los distintos problemas dependerá del usuario, pudiendo crear celdas que representen un espacio pequeño en la realidad (metros cuadrados), o celdas que representen mucho espacio (kilómetros cuadrados). Los valores posibles de la celda serán índices normalizados del factor relevante en el rango $[0,1]$ (si nos referimos a una celda para describir la presencia o ausencia de población los valores posibles serán 0 o 1).
- **Tablero:** celdas colocadas de forma matricial, de forma que se establecen relaciones de vecindad/no vecindad entre celdas.
- **Mundo:** colección de todos los factores relevantes y la distribución de población.

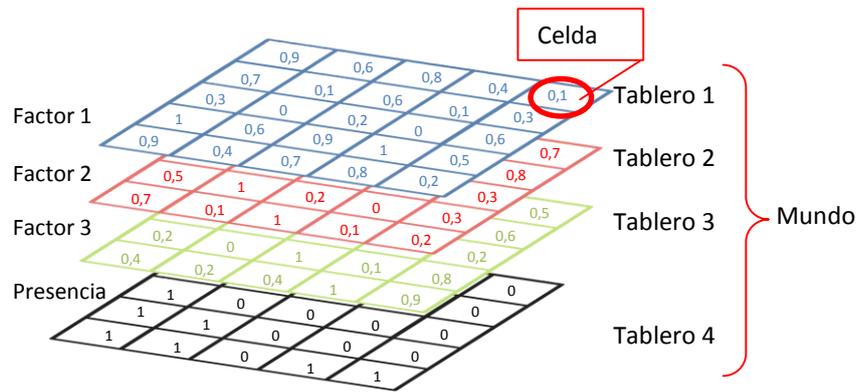


Figura 12. Modelización jerárquica.

Idealmente, el usuario debería ser capaz de facilitar información detallada sobre los valores previstos para cada celda para todos los factores relevantes y para cada una de los instantes. En caso de que el usuario no pueda o crea conveniente proporcionar esta información, al menos deberá indicar el primer y último instante. En este caso los estados intermedios se calcularán a través de una interpolación lineal, expresado matemáticamente por la siguiente expresión:

$$f(x|x_1; x_2) = f(x_1) + \frac{f(x_2) - f(x_1)}{(x_2 - x_1)}(x - x_1)$$

Figura 13. Fórmula de Interpolación Lineal

Donde x es el instante de tiempo considerado, xi el instante inicial, x2 el final y f(x) los valores del factor explicativo en el instante x.

3.1.2 Modelado del desplazamiento

Asociado a los factores relevantes, se desea que se simule la expansión de una población inicial proporcionada por el usuario. Esto se realizará mediante un autómata celular estocástico.

Aunque para el autómata celular será posible incorporar distintos esquemas de funcionamiento definido por el usuario, por defecto se propone uno basado en un índice de atractivo de terreno normalizado con rango [0,1]. En cada instante, el autómata celular tendrá que evaluar la expansión de la población, esto se realizará estimando el valor del atractivo en cada celda, después se debe recorrer todas las celdas con ocupación y se analizará si, dado el atractivo del terreno, la población se

desplazará a cualquier celda adyacente (izquierda, derecha, arriba, abajo y por las cuatro diagonales) y si se mantendrá en la celda actual.

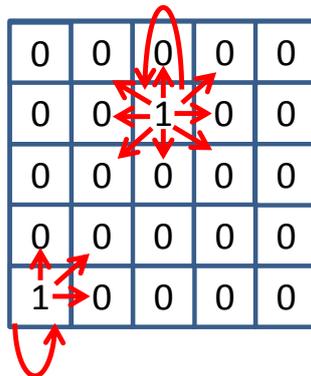


Figura 14. Ejemplo de evaluaciones de movimiento que realiza el autómata celular

Por tanto, en la función de transición del autómata celular se reflejará la población del instante anterior (una población solo podrá extenderse si la celda/célula tiene presencia) y el atractivo, de modo que sirva como entrada para determinar las preferencias para la expansión de una población. El componente estocástico se introduce al definir un número aleatorio con distribución uniforme [0,1]. En caso de que el número generado sea inferior al atractivo de la casilla adyacente la población se extenderá, este mismo comportamiento se realizará en la casilla de partida, si el número aleatorio generado es inferior al atractivo se mantendrá la presencia de la población en la celda.

El modelo de atractivo deberá ser proporcionado por el usuario y será una función que depende de los factores relevantes, si el usuario no fuera capaz de proporcionar el modelo la aplicación ofrece por defecto una forma de establecer una entrada coherente al autómata celular estocástico, generando un modelo de atractivo que depende de los factores relevantes de esa población. Se considera que el algoritmo que reúne estas capacidades es el modelo “Logistic Regression”, o “Logit”, este modelo es no lineal, y viene definido matemáticamente de la siguiente forma [13]:

$$p_i = \frac{1}{1 + \exp(-x'_i\beta)}$$

Figura 15. Expresión matemática del modelo “Logit”

Donde, la interpretación de los estimados del parámetro β_j es como los efectos aditivos en el log odds ratio (cociente de la razón de la probabilidad de que un evento suceda y la probabilidad de que no suceda bajo ciertas condiciones, y la razón de la probabilidad de que dicho evento suceda y la probabilidad de que no suceda bajo las

condiciones complementarias) para una unidad de cambio en la j -ésima variable explicativa. En el caso de una variable explicativa dicotómica, por ejemplo género, $e\beta$ es la estimación del odds ratio de tener el resultado para, por ejemplo, hombres comparados con mujeres.

El modelo de atractivo permite establecer un modelo lineal que relaciona los factores relevantes con la probabilidad de presencia a partir de observaciones.

Es importante que se realicen varias repeticiones de la simulación para evitar el factor de indeterminación propio de los algoritmos estocásticos. Será en este punto en el que pasaremos de un modelo binario de presencia correspondiente a una sola repetición de la simulación, en uno de probabilidad en presencia adecuado a partir de varias simulaciones.

3.1.3 Modelado de intervenciones

Otra funcionalidad que se desea alcanzar con este sistema es la capacidad de realizar intervenciones con coste asociado. Las intervenciones se definen como acciones que tienen por objeto el valor de un factor relevante en una casilla determinada.

Esta intervención/es pueden afectar a cualquier zona de cualquier factor relevante en cualquier instante, y con estas se podrá obtener un resultado distinto que el obtenido con la simulación sin la intervención/es. Se desea realizar las simulaciones una vez realizadas las intervenciones y calcular su coste total.

Por último, se desea crear un módulo de optimización. Dicho módulo realizará intervenciones y, usando algoritmos genéticos multiobjetivo, se determinará cuál es la mejor forma de intervenir los factores relevantes. Como aproximación inicial, este módulo solo producirá intervenciones en el primer instante de tiempo y para uno de los factores relevantes.

3.2 Requisitos de usuario

En este punto del documento se pretende identificar, clasificar y catalogar los requisitos de usuario obtenidos a través del estudio del sistema que se demandó diseñar.

Los requisitos que se extraigan deben plasmar no sólo lo que el sistema debe hacer, sino también cómo va éste a desarrollar dichas funcionalidades. De esta manera se podrá realizar una clasificación de los requisitos en dos grandes bloques:

- **Requisitos de capacidad:** son aquellos que representan lo que el sistema debe hacer, de acuerdo a las necesidades indicadas. Se trata de requisitos que hacen referencia al problema en cuestión que se desea resolver.
- **Requisitos de restricción:** indican la manera que tendrá el sistema de resolver los problemas, la forma de interactuar con el sistema y las restricciones que habrá sobre éste.

A continuación se mostrará el formato que se va a emplear para plasmar los diversos requisitos extraídos, indicando el significado de cada uno de sus campos.

<i>RUT-NN</i>	
Descripción	

Tabla 1: Tabla de definición de requisitos de usuario

Dónde:

- **Descripción:** como su propio nombre indica, describe el requisito.

3.2.1 Requisitos de capacidad

RUC-1	
Descripción	<p>La aplicación permitirá la recreación del mundo (conjunto de factores relevantes en un instante determinado), mediante los parámetros indicados por el usuario, esto son:</p> <ul style="list-style-type: none">• Número de factores relevantes para la población.• Tamaño de la matriz de terreno de los factores relevantes o número de celdas en las que se divide este.• Valores de las celdas que componen los distintos factores relevantes.

Tabla 2: requisito de Capacidad RUC-1

RUC-2	
Descripción	<p>La aplicación permitirá recrear el mundo en diferentes estados de tiempo (llamados instantes), mediante los parámetros indicados por el usuario, descritos en RUC-1.</p>

Tabla 3: requisito de Capacidad RUC-2

RUC-3	
Descripción	<p>La aplicación permitirá, para el primer instante de tiempo, recrear una matriz de presencia/ausencia de población, por cada celda que compone el terreno.</p>

Tabla 4: requisito de Capacidad RUC-3

RUC-4	
Descripción	<p>La aplicación permitirá la simulación partiendo de la población inicial y los siguientes mundos pasados por parámetros: Inicial (primer instante), final (último instante), intermedios.</p>

Tabla 5: requisito de Capacidad RUC-4

RUC-5

Descripción	Aunque la aplicación espere recibir todos los instantes, alternativamente se ofrecerán comportamientos por defecto para que el usuario no tenga que introducir estos estados, pudiéndose generar automáticamente dependiendo de los ya introducidos.
--------------------	--

Tabla 6: requisito de Capacidad RUC-5

RUC-6

Descripción	El usuario podrá insertar un modelo de atractivo (aportación para la población de los distintos factores relevantes), no obstante, por defecto la aplicación ofrecerá la capacidad de obtener un modelo de atractivo que dependerá de los valores de los factores relevantes en un instante determinado.
--------------------	--

Tabla 7: requisito de Capacidad RUC-6

RUC-7

Descripción	El desplazamiento de las poblaciones se simulará automáticamente, dependiendo del modelo de atractivo y los valores de los factores relevantes en el instante concreto.
--------------------	---

Tabla 8: requisito de Capacidad RUC-7

RUC-8

Descripción	Una vez concluida la fase de simulación inicial, se podrá realizar una serie de intervenciones que afecten a uno o varios factores relevantes en el instante/s que desee el usuario, estas intervenciones tendrán un coste asociado que dependerá del instante, factor relevante y celda a intervenir.
--------------------	--

Tabla 9: requisito de Capacidad RUC-8

RUC-9

Descripción	Mediante el coste unitario de cada intervención, la aplicación calculará el coste total de todas las intervenciones.
--------------------	--

Tabla 10: requisito de Capacidad RUC-9

RUC-10

Descripción	Una vez realizadas las intervenciones la aplicación permitirá una nueva simulación teniendo en cuenta dichas intervenciones en los factores relevantes (lo que puede provocar distintos movimientos de población con respecto a la simulación original).
--------------------	--

Tabla 11: requisito de Capacidad RUC-10

RUC-11

Descripción	La aplicación contendrá un módulo de optimización que determinará las intervenciones a realizar con el menor coste y que aseguren una probabilidad de presencia determinada, dando a escoger al usuario si esta presencia se desea aumentar o disminuir.
--------------------	--

Tabla 12: Requisito de Capacidad RUC-11

RUC-12

Descripción	La aplicación podrá guardar las intervenciones realizadas mediante la interfaz, así como los distintos resultados (simulación y optimización).
--------------------	--

Tabla 13: requisito de Capacidad RUC-12

3.2.2 Requisitos de restricción

<i>RUR-1</i>	
Descripción	La interfaz de la aplicación tendrá un diseño sencillo e intuitivo.

Tabla 14: requisito de restricción RUR-1

<i>RUR-2</i>	
Descripción	La interfaz de la aplicación será usable.

Tabla 15: requisito de restricción RUR-2

<i>RUR-3</i>	
Descripción	El idioma de la aplicación será el castellano.

Tabla 16: requisito de restricción RUR-3

<i>RUR-4</i>	
Descripción	La aplicación se comunicará con el usuario con mensajes claros y legibles.

Tabla 17: requisito de restricción RUR-4

<i>RUR-5</i>	
Descripción	La aplicación será independiente de la plataforma.

Tabla 18: requisito de restricción RUR-5

<i>RUR-6</i>	
Descripción	La aplicación contendrá un Manual de Usuario intuitivo.

Tabla 19: requisito de restricción RUR-6

<i>RUR-7</i>	
Descripción	La forma de calcular el atractivo podrá cambiar.

Tabla 20: requisito de restricción RUR-7

RUR-8

Descripción	La forma de calcular los instantes intermedios de los factores relevantes podrá cambiar.
--------------------	--

Tabla 21: requisito de restricción RUR-8

RUR-9

Descripción	La forma de calcular el movimiento de la población podrá cambiar.
--------------------	---

Tabla 22: requisito de restricción RUR-9

RUR-10

Descripción	El modelo de atractivo podrá cambiar.
--------------------	---------------------------------------

Tabla 23: requisito de restricción RUR-10

RUR-11

Descripción	La forma de realizar las intervenciones, y los efectos de estos podrán cambiar.
--------------------	---

Tabla 24: requisito de restricción RUR-11

RUR-12

Descripción	La forma de realizar la optimización podrá cambiar.
--------------------	---

Tabla 25: requisito de restricción RUR-12

RUR-13

Descripción	Este módulo de optimización consistirá inicialmente en un algoritmo genético multiobjetivo.
--------------------	---

Tabla 26: requisito de restricción RUR-13

3.3 Requisitos de software

En este punto del documento se pretenden identificar, clasificar y catalogar los distintos requisitos de software obtenidos tanto de los requisitos de usuario, como de los casos de uso especificados en el siguiente apartado de este mismo documento.

Los requisitos deben plasmar, no sólo lo que el sistema va a ser capaz de realizar, sino también como va a hacerlo, las restricciones que tendrá y sus propiedades. De esta manera, tenemos dos grandes grupos en los que catalogar los requisitos de software:

- **Requisitos funcionales:** especifican que es lo que tiene que realizar el sistema.
- **Requisitos no funcionales:** especifican como hace las cosas el sistema.

A continuación se mostrará el formato que se va a emplear para plasmar los diversos requisitos extraídos, indicando el significado de cada uno de sus campos.

<i>RST-NN</i>	
Descripción	

Tabla 27: tabla de definición de requisitos de software

Dónde:

- **Descripción:** describe el requisito.

3.3.1 Requisitos funcionales

RSF-1	
Descripción	Se permitirá modelar factores relevantes para una población, que contendrán: <ul style="list-style-type: none">• Valores de cada casilla (con rango [0,1]).• Nombre del factor relevante.• Número de filas y de columnas.

Tabla 28: requisito de software funcional RSF-1

RSF-2	
Descripción	Los mundos (factores relevantes de un instante de tiempo) contendrán: <ul style="list-style-type: none">• Número de factores relevantes que componen el mundo.• Factores relevantes.

Tabla 29: requisito de software funcional RSF-2

RSF-3	
Descripción	Los distintos mundos de los diferentes instantes se podrán cargar desde un documento Excel, en el que se ofrezcan los datos según la especificación de datos de entrada.

Tabla 30: requisito de software funcional RSF-3

RSF-4	
Descripción	Se debe guardar una relación entre un mundo y el instante de tiempo en el que pertenece.

Tabla 31: requisito de software funcional RSF-4

RSF-5	
Descripción	Se podrá realizar una simulación con los siguientes datos: <ul style="list-style-type: none">• Estado inicial de los factores relevantes (primer instante).• Estado final (instante final).• Estados intermedios (opcional).• Número de repeticiones de la simulación.

Tabla 32: requisito de software funcional RSF-5

RSF-6	
Descripción	En caso de que el usuario no informe los factores relevantes intermedios necesarios para la simulación, la aplicación los generará automáticamente mediante una función predefinida, teniendo en cuenta: <ul style="list-style-type: none">• Valor de los factores relevantes en instantes anteriores.• Valor de los factores relevantes en instantes posteriores.• Número de instantes entre el actual y el inmediatamente posterior informado.

Tabla 33: requisito de software funcional RSF-6

RSF-7	
Descripción	La función predeterminada que se permitirá usar para el cálculo de la simulación del terreno será una interpolación lineal .

Tabla 34: requisito de software funcional RSF-7

RSF-8

Descripción	Se permitirá al usuario la creación de modelos de atractivo, no obstante, por defecto la aplicación ofrecerá la capacidad de obtener un modelo de atractivo que dependerá de los valores de los factores relevantes en un instante determinado.
--------------------	---

Tabla 35: requisito de software funcional RSF-8

RSF-9

Descripción	Por defecto, se proporcionará una forma de calcular un modelo de atractivo a partir de la regresión logística (que encontrará asociaciones entre la ausencia o presencia de población y las características o valores de los factores relevantes).
--------------------	--

Tabla 36: requisito de software funcional RSF-9

RSF-10

Descripción	La propagación de las poblaciones se simulará automáticamente, dependiendo del atractivo (formado por los factores relevantes) y de la distribución de la población en cada instante.
--------------------	---

Tabla 37: requisito de software funcional RSF-10

RSF-11

Descripción	La forma predeterminada de simular la propagación de la población se realizará mediante un autómata celular estocástico que generará un número aleatorio con distribución normal [0,1], en el caso que éste sea menor que el índice de atractivo de la celda la población se extenderá, en caso contrario no lo hará. Esta operación se realizará por cada celda en la que haya presencia de población y para todas las celdas adyacentes.
--------------------	--

Tabla 38: requisito de software funcional RSF-11

RSF-12

Descripción	Una vez realizada la primera simulación, se podrá cargar una serie de intervenciones que afecten a una o varias casillas en uno o varios instantes y en uno o varios factores relevantes que indique el usuario, estas intervenciones tendrán un coste asociado.
--------------------	--

Tabla 39: requisito de software funcional RSF-12

RSF-13

Descripción	Las intervenciones podrán cargarse desde un fichero Excel o desde la interfaz, señalando la casilla a intervenir de un factor relevante e instante concretos.
--------------------	---

Tabla 40: requisito de software funcional RSF-13

RSF-14

Descripción	Se permitirán intervenciones que se repitan desde el instante introducido hasta el final, y además, esta intervención podrá tener coste en cada instante, o solo en el primero. También se podrá establecer un coste de intervención manual, o se podrá cargar desde un archivo de costes en formato Excel.
--------------------	---

Tabla 41: requisito de software funcional RSF-14

RSF-15

Descripción	Mediante el coste unitario de las intervenciones la aplicación calculará el coste total de las mismas en las distintas celdas que componen el factor relevante y en los distintos instantes temporales. Este cálculo de costes dependerá de las opciones que haya elegido el usuario.
--------------------	---

Tabla 42: requisito de software funcional RSF-15

RSF-16

Descripción	Una vez realizadas las intervenciones el sistema simulará de nuevo teniendo en cuenta dichas intervenciones.
--------------------	--

Tabla 43: requisito de software funcional RSF-16

RSF-17

Descripción	La aplicación ofrecerá un módulo de optimización que determinará las intervenciones con menor coste, que aseguren la presencia de población en un lugar concreto con una probabilidad determinada.
--------------------	--

Tabla 44: requisito de software funcional RSF-17

RSF-18

Descripción	La optimización se realizará mediante un algoritmo genético multiobjetivo, puesto que se maximizará o minimizará la probabilidad de presencia de población (a determinar por el usuario) y a la vez minimizar el coste de las intervenciones.
--------------------	---

Tabla 45: requisito de software funcional RSF-18

RSF-19

Descripción	La aplicación podrá guardar las intervenciones realizadas mediante la interfaz, así como los resultados de la optimización.
--------------------	---

Tabla 46: requisito de software funcional RSF-19

3.3.2 Requisitos no funcionales

RSNF-1

Descripción	La distribución de población del primer instante de tiempo se representará con valores de existencia o ausencia de población por cada celda que compone el factor relevante.
--------------------	--

Tabla 47: requisito de software no funcional RSNF-1

RSNF-2

Descripción	El formato para la carga de archivos será Microsoft Excel 2003.
--------------------	---

Tabla 48: requisito de software no funcional RSNF-2

RSNF-3

Descripción	La interfaz de la aplicación tendrá un diseño sencillo e intuitivo. La probabilidad de presencia de una población se identificará numérica y gráficamente.
--------------------	--

Tabla 49: requisito de software no funcional RSNF-3

RSNF-4

Descripción	La interfaz de la aplicación será usable. Se permitirá modificar valores de forma interactiva y gráfica, también se permitirá poder cargar valores desde archivo.
--------------------	---

Tabla 50: requisito de software no funcional RSNF-4

RSNF-5

Descripción	El idioma de la aplicación será el castellano.
--------------------	--

Tabla 51: requisito de software no funcional RSNF-5

RSNF-6

Descripción	La aplicación se comunicará con el usuario con mensajes claros y legibles. Todos los errores deben ser controlados y lanzados mediante una ventana de error indicando la causa.
--------------------	---

Tabla 52: requisito de software no funcional RSNF-6

RSNF-7

Descripción	La aplicación será independiente de la plataforma.
--------------------	--

Tabla 53: requisito de software no funcional RSNF-7

RSNF-8

Descripción	Existirá un Manual de Usuario que podrá usarse en la propia aplicación. Este manual explicará las acciones que el usuario puede realizar, así como los formatos de los archivos de entrada y salida.
--------------------	--

Tabla 54: requisito de software no funcional RSNF-8

RSNF-9

Descripción	La aplicación ofrecerá una forma predefinida de obtener un modelo de atractivo, aunque el usuario podrá diseñar e implementar su propio modelo e integrarlo en la aplicación fácilmente.
--------------------	--

Tabla 55: requisito de software no funcional RSNF-9

RSNF-10

Descripción	El usuario debe introducir los valores de los todos instantes de los factores relevantes, pero la aplicación podrá no requerir los datos y ofrecer un modelo predefinido para el cálculo de los instantes. No obstante, el usuario también podrá diseñar e implementar su propio modelo e integrarlo en la aplicación fácilmente.
--------------------	---

Tabla 56: requisito de software no funcional RSNF-10

RSNF-11

Descripción	La aplicación ofrecerá un autómatas celular estocástico para determinar el movimiento de la población. No obstante, el usuario también podrá diseñar e implementar su propio modelo e integrarlo en la aplicación fácilmente.
--------------------	---

Tabla 57: requisito de software no funcional RSNF-11

RSNF-12

Descripción	La forma de realizar las intervenciones podrá no ser siempre la misma, por tanto la aplicación será totalmente parametrizable en las simulaciones con intervenciones.
--------------------	---

Tabla 58: requisito de software no funcional RSNF-12

RSNF-13

Descripción	El módulo de optimización estará compuesto de un algoritmo genético multiobjetivo, los objetivos a optimizar serán: coste y probabilidad de presencia de población.
--------------------	---

Tabla 59: requisito de software no funcional RSNF-13

3.4 Modelo conceptual

El modelo conceptual define el modelo lógico de la aplicación, quedándose en el análisis y procurando no entrar en detalles de diseño e implementación. A continuación se expone el modelo lógico de la aplicación propuesta en forma de diagrama de clases.

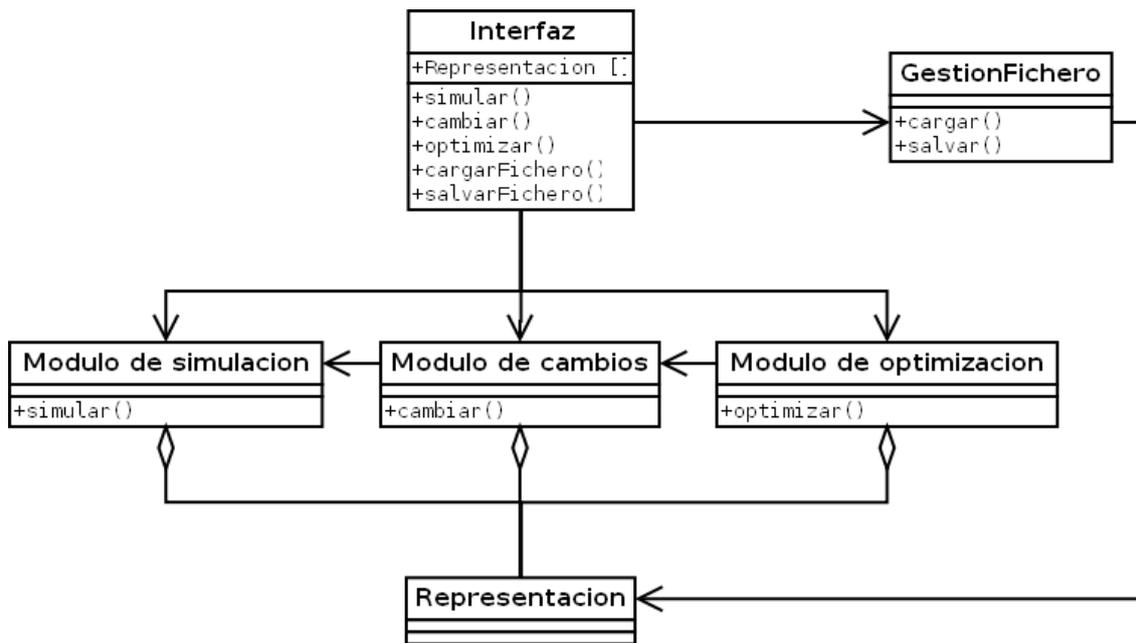


Figura 16. Modelo conceptual.

Como puede observarse, la aplicación se compondrá de una interfaz así como módulos de simulación, intervención o cambio y optimización. A partir de la interfaz se accederá a los diferentes módulos y se realizarán los procesos asociados.

El módulo de representación dará soporte a los demás módulos, permitiendo modelizar el mundo real (a través de factores relevantes que influyan en la presencia o ausencia de la población en estudio).

El módulo de simulación será el encargado de realizar las estimaciones oportunas para construir las siguientes simulaciones:

- Evolución de factores relevantes: el usuario debe proporcionar todos los valores de los instantes por cada factor relevante, no obstante la aplicación podrá estimar cómo evolucionan los factores relevantes a lo largo del tiempo si el usuario no los proporcionara.

- Modelo de atractivo: según los factores de simulación y el modelo de atractivo obtenido del usuario se obtendrá un atractivo. Por defecto, la aplicación permitirá obtener un modelo relacionando la presencia o ausencia de población y los factores relevantes del primer instante.
- Simulación de movimiento de población: se estima cómo se expandirá la población o los movimientos que ésta realice.

El módulo de intervención/cambios permitirá realizar cambios en los valores de los factores relevantes, será necesario realizar una nueva simulación una vez introducidos dichas intervenciones, respetando el modelo de atractivo. Los costes tendrán un coste asociado que debe estimarse.

El módulo de optimización permitirá buscar intervenciones en los factores relevantes de forma automática que permitan maximizar o minimizar la probabilidad de presencia de población en una casilla de terreno determinada, empleando un algoritmo de Inteligencia Artificial, como puede ser un algoritmo genético o un evolutivo. El resultado de la optimización puede interesar al usuario por ofrecer soluciones con distintas relaciones coste/porcentaje de presencia de población.

El módulo de gestión de ficheros permitirá relacionarse con los archivos externos a la aplicación y servirá tanto para realizar la carga de los factores relevantes en los distintos instantes, así como intervenciones o los costes asociados a estos.

La interfaz permitirá al usuario un entorno confortable para trabajar con la aplicación. Se deberá dar acceso controlado a los módulos antes mencionados proporcionando una visualización sencilla y gráfica.

3.5 Casos de uso

“Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.” [20]

3.5.1 Diagrama de casos de uso

En el diagrama se expresará de forma genérica los casos de uso que podrá realizar el usuario de la una vez construida la aplicación.

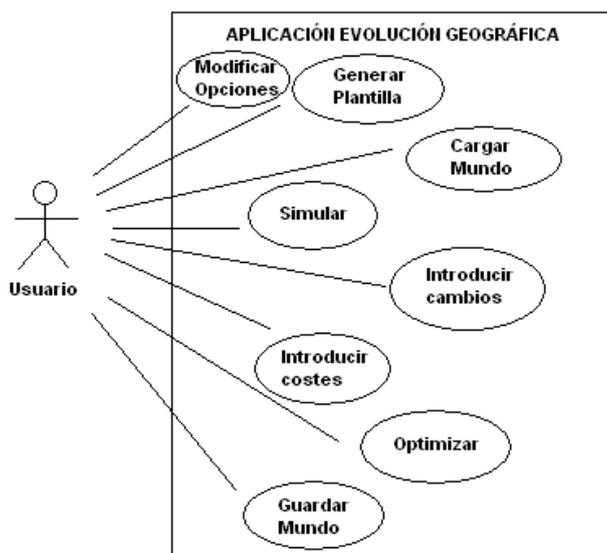


Figura 17. Diagrama de caso de uso de la aplicación

3.5.2 Especificación textual de los casos de uso

En este apartado se incluyen las tablas con la especificación textual de cada caso de uso. También se desglosarán las acciones genéricas definidas anteriormente como “Modificar opciones” o “Guardar Mundo”, así como cambios de vista que puede realizar el usuario de forma interactiva para dar más valor a su experiencia con la aplicación.

En primer lugar, se va a mostrar el formato que se va a emplear para representar los casos de uso, indicando el significado de cada uno de sus campos:

Nombre	CU X -
Actor	
Objetivo	
Precondiciones	
Postcondiciones	
Escenario básico	

Tabla 60: tabla de definición de casos de uso

Dónde:

- **CU:** iniciales que denotan que se trata de un caso de uso.
- **X:** identificador numérico y secuencial de los casos de uso. Esto hace que se pueda diferenciar unívocamente cada uno de ellos.
- **Nombre:** identificador textual del caso de uso.
- **Objetivo:** define la respuesta del sistema que se espera obtener ejecutando el caso de uso.
- **Pre-condiciones:** condiciones que deben cumplirse para poder ejecutar el caso de uso.
- **Post-condiciones:** condiciones que se darán en el sistema una vez haya finalizado la ejecución del caso de uso.
- **Escenario básico:** pasos necesarios para ejecutar el caso de uso.

Nombre	CU 1 – Generar una plantilla
Actor	Usuario
Objetivo	Generar una plantilla Excel para conocer el formato que deben tener los ficheros de entrada.
Precondiciones	Ninguna.
Postcondiciones	Se genera un Excel con un caso inicial.
Escenario básico	<ol style="list-style-type: none"> 1. Se elige un nombre de archivo. 2. Se genera el fichero en formato Excel.

Tabla 61: caso de uso 1 – Generar una plantilla

Nombre	CU 2 – Cargar mundos
Actor	Usuario
Objetivo	Se cargan los factores relevantes.
Precondiciones	Debe existir un archivo Excel con el formato adecuado para ser cargado.
Postcondiciones	Se carga en memoria los factores relevantes del instante inicial.
Escenario básico	<ol style="list-style-type: none"> 1. Se selecciona el archivo a cargar. 2. Se carga el fichero o devuelve error si el formato no es el adecuado.

Tabla 62: caso de uso 2 – Cargar mundos

Nombre	CU 3 – Modificar número de repeticiones
Actor	Usuario
Objetivo	Permite modificar el número de repeticiones de la simulación.
Precondiciones	Ninguna.
Postcondiciones	Internamente se establece un nuevo valor para las repeticiones
Escenario básico	<ol style="list-style-type: none"> 1. Se cambia el número de repeticiones 2. Se acepta o da error si es un valor incorrecto.

Tabla 63: caso de uso 3 – Modificar número de repeticiones

Nombre	CU 4 – Modificar el tipo de los simuladores
Actor	Usuario
Objetivo	Permite modificar la forma de realizar la simulación.
Precondiciones	Ninguna.
Postcondiciones	Internamente se establece unas formas distintas de realizar la simulación.
Escenario básico	<ol style="list-style-type: none"> 1. Se modifica el comportamiento por defecto de la aplicación: <ol style="list-style-type: none"> a. Se modifica el método para cumplimentar los instantes que no introduzca manualmente. b. Se modifica el modelo de atractivo a aplicar. c. Se modifica la forma en el que se calcula el movimiento de la población. 2. Se acepta o da error si es un valor incorrecto.

Tabla 64: caso de uso 4 – Modificar el tipo de los simuladores

Nombre	CU 5 – Simulación
Actor	Usuario
Objetivo	Permite realizar una simulación dadas una población inicial, factores relevantes, simuladores concretos y un número de repeticiones determinado.
Precondiciones	Tener un mundo cargado.
Postcondiciones	Obtenemos una simulación.
Escenario básico	<ol style="list-style-type: none"> 1. Con el mundo cargado se realiza la simulación. 2. Se obtiene la simulación según los parámetros, o un error si no se pudo realizar la simulación con estos parámetros.

Tabla 65: caso de uso 5 – Simulación

Nombre	CU 6 – Modificar la vista mostrando otro instante
Actor	Usuario
Objetivo	Permite ver otro instante distinto del mundo simulado.
Precondiciones	Realizado una simulación.
Postcondiciones	Cambiamos la visualización a un instante N+1 o N-1 desde un instante N.
Escenario básico	<ol style="list-style-type: none"> 1. Desde la visualizamos del instante N. 2. Cambiamos de vista con el botón indicado. 3. Visualizamos el instante N+1 o N-1 de la simulación.

Tabla 66: caso de uso 6 – Modificar la vista mostrando otro instante

Nombre	CU 7 – Volver a la pantalla principal
Actor	Usuario
Objetivo	Permite regresar a la pantalla principal
Precondiciones	Realizado una simulación.
Postcondiciones	Volvemos a la pantalla principal
Escenario básico	<ol style="list-style-type: none"> 1. Desde la visualizamos del instante N. 2. Volvemos a la pantalla inicial desechando la simulación anterior.

Tabla 67: caso de uso 7 – Volver a la pantalla principal

Nombre	CU 8 – Realizar un cambio en una casilla
Actor	Usuario
Objetivo	Permite insertar una intervención en una casilla de un factor relevante en un instante determinado.
Precondiciones	Realizado una simulación.
Postcondiciones	El valor de la casilla se ve alterado para ese instante.
Escenario básico	<ol style="list-style-type: none"> 1. Desde la visualizamos del instante N. 2. Se modifica el valor de una casilla para un instante determinado.

Tabla 68: caso de uso 8 – Realizar un cambio en una casilla

Nombre	CU 9 – Limpiar cambios
Actor	Usuario
Objetivo	Permite eliminar las intervenciones introducidas
Precondiciones	Establecida, al menos, una intervención.
Postcondiciones	Se eliminan las intervenciones establecidas.
Escenario básico	<ol style="list-style-type: none"> Desde la ventana de simulación con las intervenciones realizadas. Se eliminan las intervenciones volviendo a la simulación.

Tabla 69: caso de uso 9 – Limpiar cambios

Nombre	CU 10 – Guardar cambios
Actor	Usuario
Objetivo	Permite guardar las intervenciones introducidas.
Precondiciones	Realizado una simulación.
Postcondiciones	Se guardan las intervenciones en un fichero Microsoft Excel 2003.
Escenario básico	<ol style="list-style-type: none"> Desde la ventana de simulación con las intervenciones realizadas. Se guarda las intervenciones volviendo a la simulación.

Tabla 70: caso de uso 10 – Guardar cambios

Nombre	CU 11 – Cargar archivo de costes
Actor	Usuario
Objetivo	Permite cargar los costes unitarios por casilla desde fichero, para establecer un coste total de las intervenciones realizadas.
Precondiciones	Realizado una simulación.
Postcondiciones	Mantenemos en memoria unos costes unitarios por cada casilla.
Escenario básico	<ol style="list-style-type: none"> Desde la ventana de simulación. Se carga en memoria los costes unitarios.

Tabla 71: caso de uso 11 – Cargar archivo de costes

Nombre	CU 12 – Cargar archivo de cambios
Actor	Usuario
Objetivo	Permite cargar las intervenciones desde fichero.
Precondiciones	Realizado una simulación.
Postcondiciones	Mantenemos en memoria las intervenciones a realizar.
Escenario básico	<ol style="list-style-type: none"> Desde la ventana de simulación. Se carga en memoria las intervenciones.

Tabla 72: caso de uso 12 – Cargar archivo de cambios

Nombre	CU 13 – Modificar propiedades
Actor	Usuario
Objetivo	Permite modificar las propiedades para la simulación con intervenciones en los factores relevantes.
Precondiciones	Realizado una simulación.
Postcondiciones	Se modifican las propiedades a realizar.
Escenario básico	<ol style="list-style-type: none"> Desde la ventana de simulación. Se modifican las propiedades.

Tabla 73: caso de uso 13 – Modificar propiedades

Nombre	CU 14 – Simulación en base a las intervenciones
Actor	Usuario
Objetivo	Permite volver a simular con las intervenciones introducidas por el usuario.
Precondiciones	Realizado una simulación.
Postcondiciones	Simulación sin las intervenciones y una nueva simulación con las intervenciones.
Escenario básico	<ol style="list-style-type: none"> Desde la ventana de simulación. Se realiza una nueva simulación con las intervenciones. Se muestra la simulación sin intervenciones y la segunda simulación (con las intervenciones introducidas por el usuario) y el coste asociado de este último.

Tabla 74: caso de uso 14 – Simulación en base a las intervenciones

Nombre	CU 15 – Optimización
Actor	Usuario
Objetivo	Permite generar intervenciones automáticamente para obtener un Frente de Pareto, con las soluciones con una relación entre la probabilidad de presencia de una población máxima (o mínima), y mínimo coste asociado.
Precondiciones	Realizado una simulación.
Postcondiciones	Se muestra la 'Frente de Pareto'.
Escenario básico	<ol style="list-style-type: none"> Desde la ventana de simulación. Se realizan la optimización resultando una 'Frente de Pareto' que se muestra.

Tabla 75: caso de uso 15 – Optimización

Nombre	CU 16 – Consulta de una solución resultado de la optimización
Actor	Usuario
Objetivo	Permite mostrar las intervenciones concretas resultado de una optimización.
Precondiciones	Realizada una optimización.
Postcondiciones	Se muestra la simulación inicial y una de las soluciones que conforman un punto de la 'Frente de Pareto'.
Escenario básico	<ol style="list-style-type: none"> Desde la ventana de optimización. Se selecciona una de las soluciones óptimas. Se abre una ventana en la que se muestra la solución y se abre la simulación inicial.

Tabla 76: caso de uso 16 – Consulta de una solución resultado de la optimización

3.5.3 Diagramas de secuencia

A continuación se detallan los diagramas de secuencia de los casos de uso. En estos diagramas se muestran las interacciones que se producen entre los objetos durante la realización de los casos de uso.

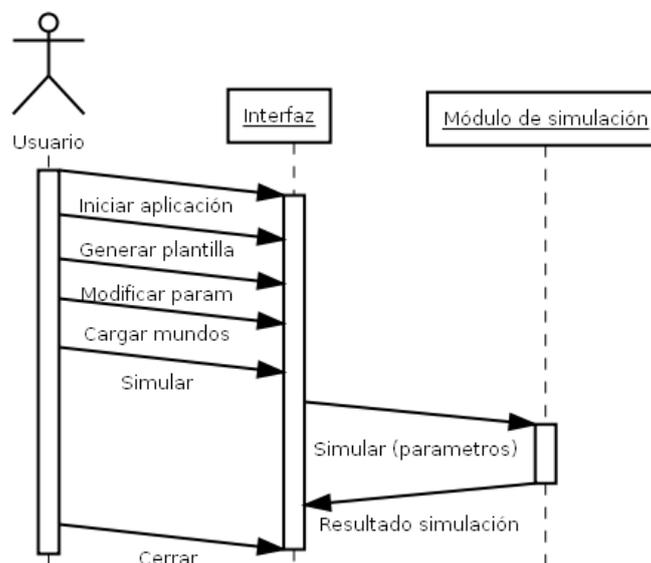


Figura 18. Diagrama de secuencia: el usuario modifica parámetros y realiza una simulación

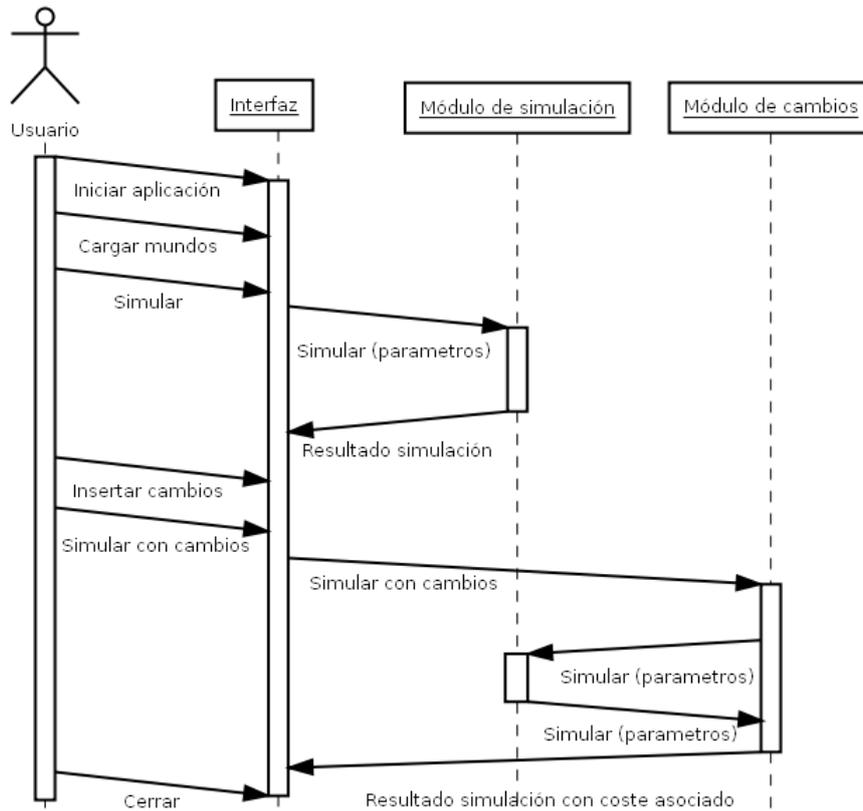


Figura 19. Diagrama de secuencia: el usuario realiza simulaciones con cambios

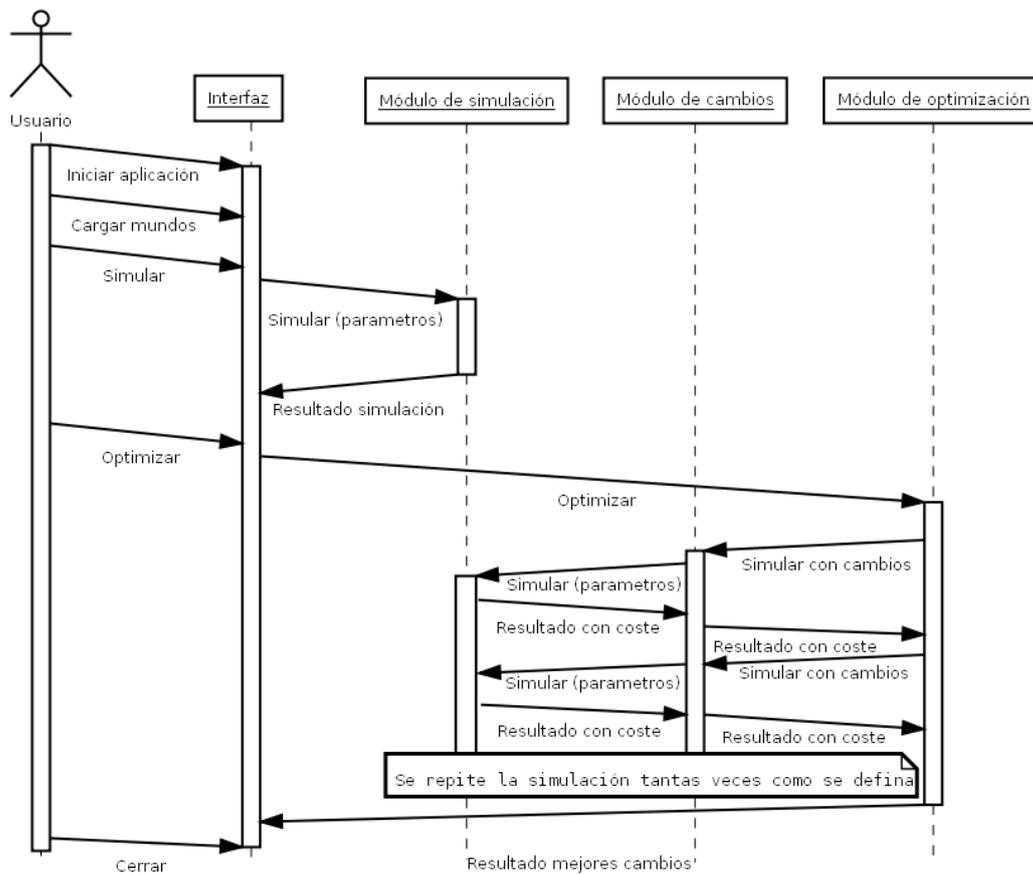


Figura 20. Diagrama de secuencia: el usuario utiliza la optimización automática

CAPÍTULO 4: DISEÑO

Una vez realizada la fase de análisis, se procederá a realizar un diseño que abarque todas las funcionalidades especificadas anteriormente. Dicho diseño, debe ser suficientemente exhaustivo y claro, de forma que no genere dudas durante la fase de implementación.

En primer lugar se va a incluir el diagrama de clases de la aplicación y, posteriormente, se irán explicando cada una de ellas por separado.

4.1 Plataforma y Herramientas de Desarrollo

Todo el desarrollo del presente proyecto se realiza en Java, debido a la familiaridad con el lenguaje y a su capacidad para poder reutilizar objetos. Este lenguaje fue desarrollado por Sun Microsystems a principios de los años 90. Fue definido dentro del paradigma de la orientación a objetos y con el objetivo de permitir ejecutar un mismo programa en diferentes sistemas operativos [Gosling, 1996].

Para ello, el código se compila en bytecodes que son interpretados por la JVM (Java Virtual Machine) [Lindholm, 1999].

Para elegir un entorno de programación en el que trabajar se han comparado dos herramientas para la implementación de código en Java de software libre *NetBeans* y *Eclipse*. Finalmente, dado que el desarrollo de la aplicación requiere una interfaz y *NetBeans* tiene un excelente soporte por defecto para tratar los paquetes `java.swing` y `java.awt` para formar interfaces en Java, se ha decidido usar esta herramienta.

Una vez elegido el lenguaje y para facilitar la implementación del proyecto y dotarlo de modularidad, se han propuestos varios framework que se definirán en este punto. Un framework es una librería que se usa como base para desarrollar aplicaciones. Normalmente contiene una arquitectura bien definida, así como un conjunto de componentes reusables.

Framework jxl

Framework que permite realizar operaciones de entrada/salida en los ficheros Excel con los que se relaciona la aplicación.

Framework freechart

Framework que permite mostrar por pantalla, de forma gráfica, los resultados de la optimización. Se necesita mostrar en forma de gráfico el “frente de Pareto”, que será las soluciones con menor coste y mayor probabilidad de presencia de población en caso de maximizar, o las soluciones de menor coste y menor probabilidad de presencia en caso de minimizar.

Framework JAMA

Framework que operar con matrices de forma eficiente. Este framework es utilizado para los cálculos intermedios necesarios para calcular el modelo de atractivo mediante el algoritmo Logit.

Framework JMetal

Framework que permite operar con algoritmos genéticos multiobjetivos de forma modulable y eficiente. Su nombre proviene de Java Metauristic Algorithms y ofrece varios algoritmos genéticos multiobjetivo, entre los que destaca NSGAII y SPEA2. También proporciona un conjunto de utilidades que ayudan a llevar a cabo estudios experimentales.

4.2 Diagrama de clases

El diagrama de clases describe las clases y objetos que componen el sistema, así como las diversas relaciones que existen entre ellos. Además de determinar las clases que compondrán el programa, es necesario incluir los atributos y métodos de cada una de ellas, así como las diversas restricciones de visibilidad de cada miembro.

Este diagrama se presentará mostrando la relación entre el código desarrollado y los distintos framework usados para dar más modularidad a la aplicación. También se debe tener en cuenta que para desarrollar el módulo de optimización se ha decidido utilizar el framework "JMetal" con el fin de aplicar un algoritmo genético. El diagrama de clases de este framework, así como el de las clases creadas para adaptarlo al problema se mostrará a continuación del diagrama de clases.

A continuación se muestra el diagrama de clases:

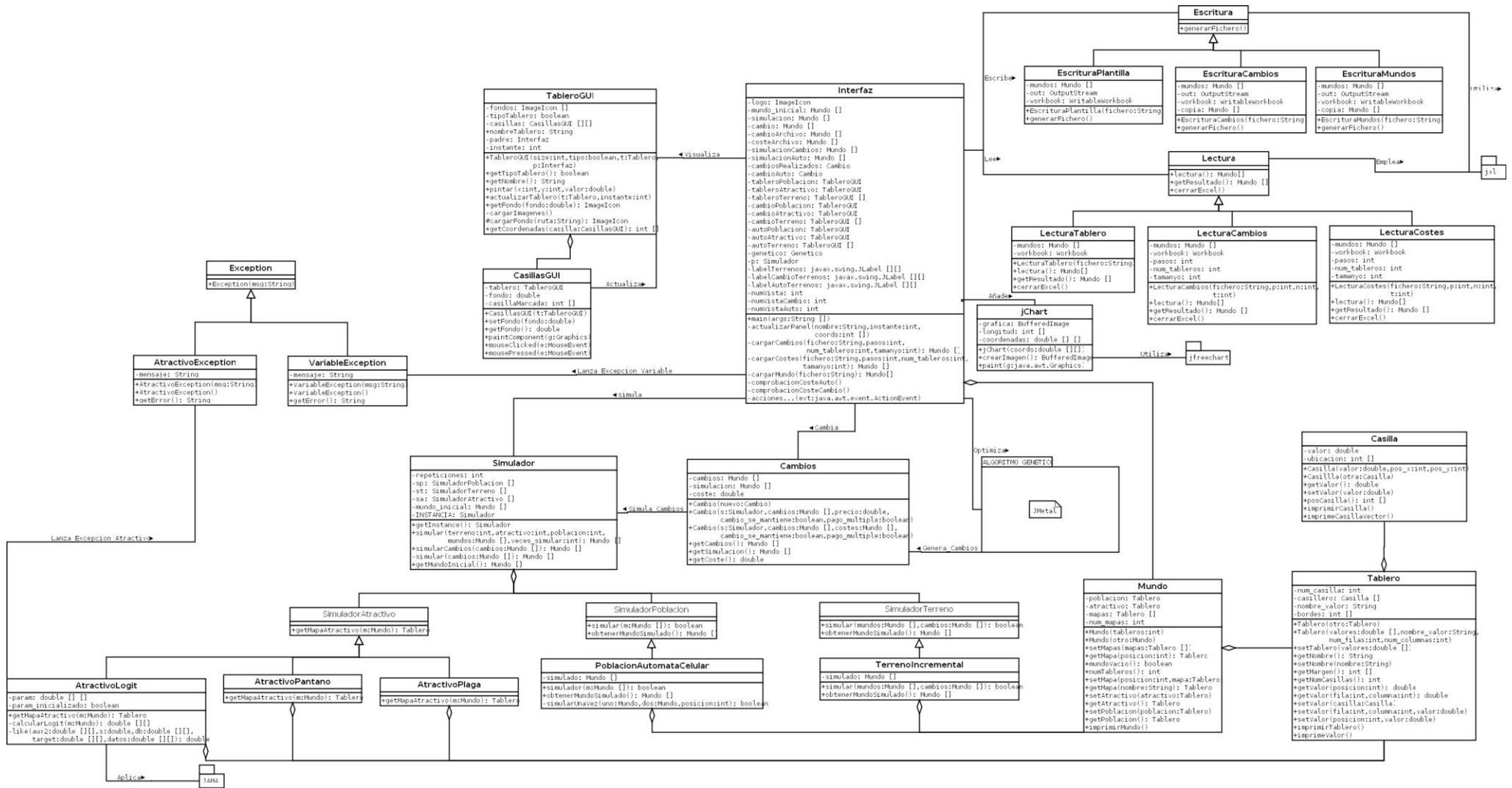


Figura 21. Diagrama de clases de la aplicación

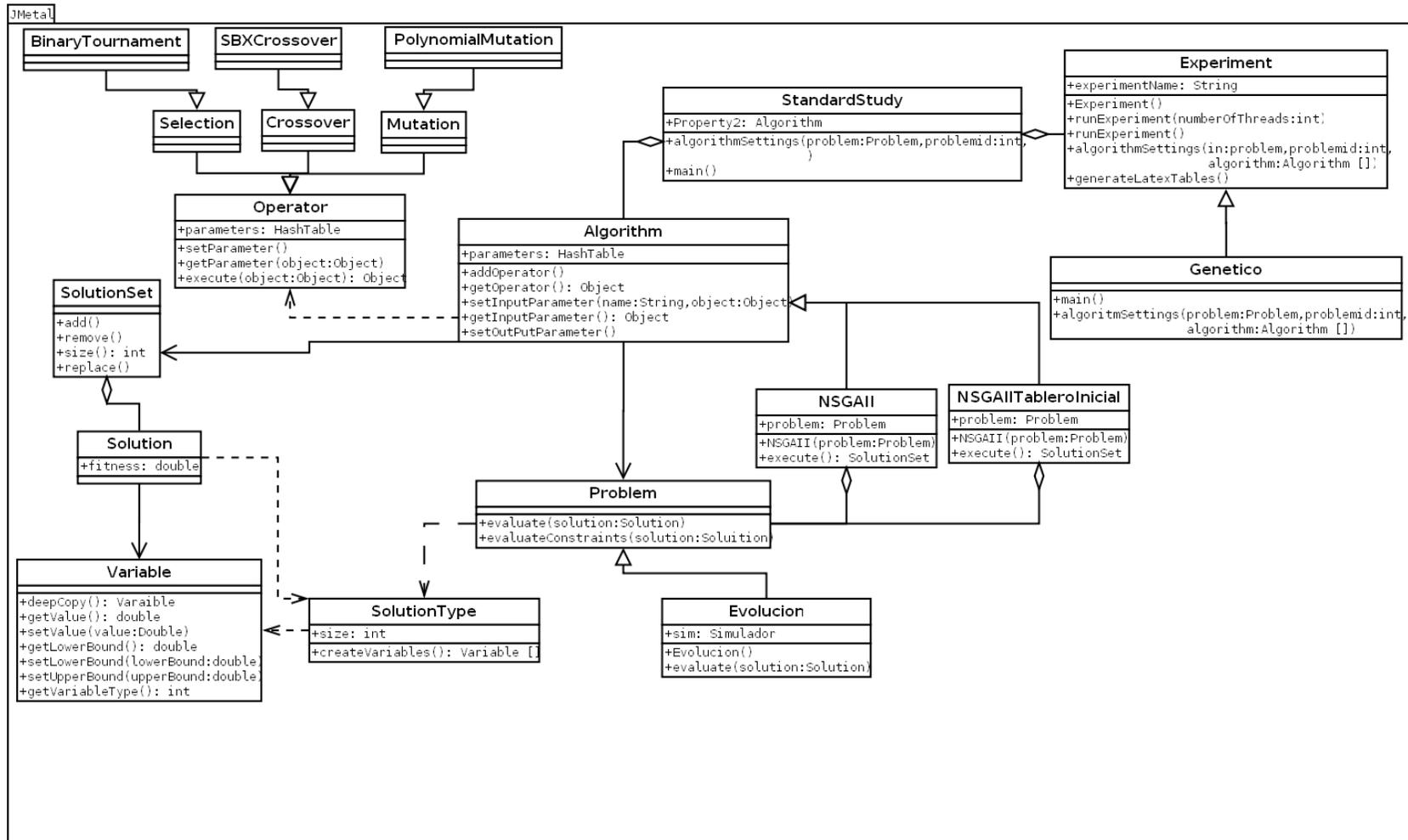


Figura 22. Diagrama de clases del módulo de optimización (framework JMetal)

4.3 Definición de las clases

Paquete evoluciongeografica

Clase Interfaz

La clase Interfaz es la responsable de comunicarse con el usuario de la aplicación y actualizar la interfaz conforme a las actividades de éste. Contiene los métodos necesarios para realizar llamadas a las clases encargadas de la carga y guardado de los archivos necesarios como entrada y salida de la aplicación.

Clase CasillasGUI

La instancia de la clase CasillasGUI se corresponde con una casilla visible por el usuario. En el caso de realizarse una acción sobre ella, informará a las clases superiores para realizar una acción. A esta clase se le asignará una imagen o color que representará el valor de la casilla.

Clase TableroGUI

La clase TableroGUI se corresponde con un conjunto de CasillasGUI que forman un mismo tablero. Desde esta clase se pueden cargar las distintas imágenes o colores que el usuario visualizará en representación del valor que adquiera la casilla.

Clase jChart

La clase jChart utiliza el framework “jfree”. Permite representar gráficamente el frente de Pareto que se obtiene al realizar una optimización multiobjetivo. Consiste en una representación de valores en un eje de coordenadas de dos dimensiones.

Paquete Cambios

Clase Cambios

La clase Cambio es la encargada de realizar la llamada a la clase Simulador cuando el usuario o el módulo de optimización requieren una simulación con intervenciones. Mediante las instancias de esta clase es posible obtener el coste de las intervenciones definidas, en función de las opciones elegidas por el usuario.

Paquete Excepcion

Clase VariableException

La clase VariableException informa a las clases superiores cuando el usuario introduce un valor incorrecto, que imposibilita realizar las operaciones.

Clase AtractivoException

La clase AtractivoException informa a las clases superiores cuando no es posible calcular el atractivo en base a los datos de entrada. Esto puede suceder, por ejemplo, al calcular el modelo de atractivo con el módulo Logit (basado en el algoritmo de Logistic Regression) sobre matrices singulares.

Paquete Fichero

Clase Escritura

La clase Escritura es una interfaz. Representa la escritura de datos en un archivo con formato Excel 2003.

Clase EscrituraCambios

La clase EscrituraCambios implementa la clase Escritura. Es utilizada para crear un archivo con formato Excel 2003 que contiene las intervenciones introducidas por el usuario para que puedan ser cargadas posteriormente.

Clase EscrituraMundos

La clase EscrituraCambios implementa la clase Escritura. Es utilizada para crear un archivo con formato Excel 2003 que contiene las intervenciones introducidas por el usuario para que se puedan cargar sobre los factores relevantes originales.

Clase EscrituraPlantilla

La clase EscrituraPlantilla implementa la clase Escritura. Es utilizada para crear plantillas, éstas serán archivos en formato Excel 2003. El objetivo de ofrecer plantillas mostrarán al usuario el formato que se debe usar, para ello la aplicación creará un archivo que la aplicación puede cargar de forma automática.

Clase Lectura

La clase Lectura es una interfaz. Representa la lectura de datos desde un archivo con formato Excel 2003.

Clase LecturaCambios

La clase LecturaCambios implementa la clase Lectura. Es utilizada para cargar las intervenciones creadas anteriormente desde archivos en formato Excel 2003(definido en el anexo A.1).

Clase LecturaCostes

La clase LecturaCostes implementa la clase Lectura. Es utilizada para cargar los costes de los terrenos desde archivos en formato Excel 2003.

Clase LecturaTablero

La clase LecturaTablero implementa la clase Lectura. Es utilizada para cargar los distintos terrenos iniciales desde archivos en formato Excel 2003.

Paquete Representación

Clase Casilla

La clase Casilla representa un área de un factor relevante concreto y en un instante determinado, que puede tomar un valor decimal que indicará, entre 0 y 1, la valoración del factor relevante. Una casilla también puede representar la presencia de población en esa área, el valor del atractivo obtenido por el modelo de atractivo.

Clase Tablero

La clase Tablero es una agrupación de casillas en un instante determinado. Su instancia corresponde con el concepto de factor relevante. Dentro del tablero, las casillas están ordenadas, representando la relación real de las distintas áreas reales.

Clase Mundo

La clase Mundo representa una agrupación factores relevantes. En concreto contiene:

- Una población, indicando la presencia/ausencia de esta en las distintas casillas.
- Un atractivo que indica cómo influyen los factores relevantes según el modelo de atractivo.
- Un número indeterminado (igual o mayor que uno) de factores relevantes (por ejemplo: agua, comida, temperatura, etc.)

Paquete Simulador

Clase Simulador

La clase simulador ha sido cumplimentada haciendo uso del patrón Singleton que se muestra a continuación:

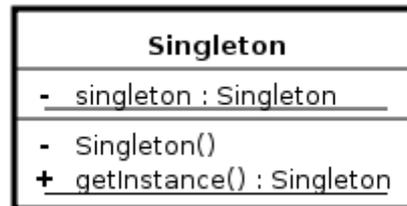


Figura 23. Patrón Singleton

El patrón Singleton es utilizado para garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Por tanto, se creará una instancia de simulador.

Esta clase ha sido creada para unificar la fase de simulación, este es uno de los tres módulos que compone la aplicación. También es el encargado de realizar el número de simulaciones que indique el usuario de forma que se obtenga una probabilidad de presencia entre todas estas simulaciones, con el fin de disminuir la aleatoriedad en la salida de la aplicación.

Paquete Simulador Atractivo

Clase SimuladorAtractivo

Clase abstracta que permite establecer los métodos necesarios para crear un modelo de atractivo que permita la existencia del módulo de simulación. La utilidad de la clase radica en la modularidad que ofrecen las clases abstractas.

Clase AtractivoLogit

Clase instanciable que hereda comportamientos de la clase SimuladorAtractivo, es un simulador que permite calcular un modelo de atractivo utilizando el algoritmo Logistic Regression que se detalla a continuación de forma esquemática (aunque puedan establecerse otros funcionamientos gracias a la modularidad que ofrece la aplicación):

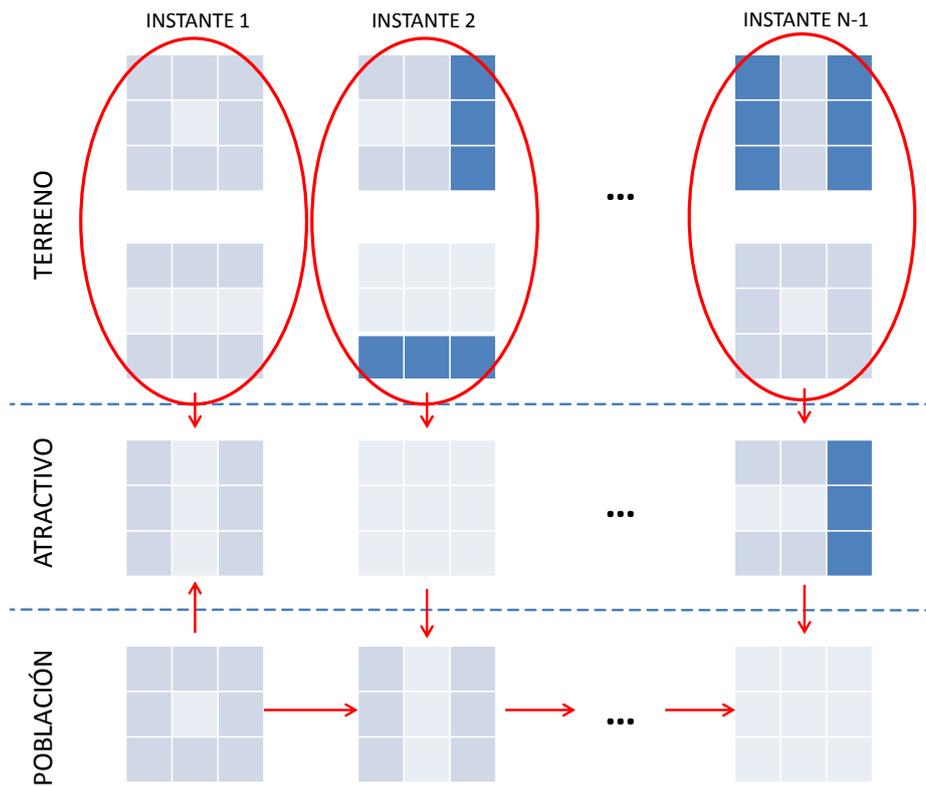


Figura 24. Simulación basada en algoritmo "Logit"

Como se puede apreciar en la figura anterior, el primer atractivo es calculado por los factores relevantes en el primer instante y la población inicial (estableciendo relaciones entre la presencia/ausencia y dichos factores relevantes). Después, con el modelo de atractivo, la aplicación seguirá su funcionamiento por defecto, obteniendo el atractivo con los factores relevantes.

Un problema que presenta este algoritmo es la asociación que se establece en áreas en los que pudiera existir población, pero donde no hay presencia debido a que, por ejemplo, no hay un lugar de paso con atractivo alto desde donde está la población. Por esta razón se ha decidido calcular el algoritmo en base a regiones concretas, limitando así la probabilidad de establecer relaciones "factores relevantes"- "población" incorrectas.

Paquete Simulador Atractivo Logit

Clase IModel

Clase abstracta que representa un modelo de simulación, tiene los métodos necesarios para modelos como, por ejemplo, el modelo Logistic Regression.

Clase LogitModel

Clase instanciable que hereda comportamiento de la clase IModel. Representa el algoritmo LogitModel, que comprende los cálculos necesarios para crear un modelo de atractivo en forma de variables que permitan determinar el atractivo en función de otros factores relevantes.

Clase NewtonRaphson

Clase que utiliza un IModel para aplicar sobre este el algoritmo de Newton Raphson necesario para el modelo Logistical Regression.

Paquete Simulador Población

Clase SimuladorPoblacion

Clase abstracta que permite establecer los métodos para crear un bloque de simulación de poblaciones necesario para la existencia del módulo de simulación. La utilidad de la clase radica en la modularidad que ofrecen las clases abstractas.

Clase PoblacionAutomataCelular

Clase instanciable que hereda el comportamiento de la clase SimuladorPoblacion. Esta clase implementa un autómata celular estocástico, cada población, de existir en una celda, se extenderá a las celdas contiguas y/o permanecerá en la suya dependiendo del atractivo de las celdas de destino (determinado por los factores relevantes).

Paquete Simulador Terreno

Clase SimuladorTerreno

Clase abstracta que permite establecer los métodos para crear un bloque de simulación de terreno necesario para la existencia del módulo de simulación. La utilidad de la clase radica en la modularidad que ofrecen las clases abstractas.

Clase TerrenoIncremental

Clase instanciable que hereda el comportamiento de la clase SimuladorTerreno. La funcionalidad de esta clase abarca la evolución de los factores relevantes en los distintos instantes mediante una función lineal, dependiendo de los instantes iniciales, finales e intermedios (si existieran). Esta funcionalidad se activará por defecto si el usuario no introduce alguno de los instantes intermedios.

Paquete Optimizador

Clase Genetico

La clase Genetico es instanciable y representa la creación de objetos necesarios para una optimización multiobjetivo basada en algoritmos genéticos. Entre otros, se instancia un algoritmo determinado, así como se determina una mutación y un algoritmo de cruce.

Clase Evolucion

Clase instanciable que representa un problema concreto y permite evaluar las soluciones que se encuentren por el algoritmo genético multiobjetivo.

Clase NSGAII Tablero Inicial

Clase instanciable que representa un algoritmo concreto (NSGAII) y permite crear soluciones a partir de los factores relevantes sin intervenciones o de forma aleatoria (funcionamiento estándar del algoritmo).

CAPÍTULO 5: EXPERIMENTACIÓN

5.1 Introducción

En este apartado se va a realizar un análisis de los resultados obtenidos al ejecutar la aplicación construida durante el desarrollo de este proyecto.

La experimentación consistirá en crear entornos sintéticos, explorar las intervenciones que se pueden realizar en los valores de los factores relevantes, y explorar la viabilidad de una optimización automática multiobjetivo de intervenciones mediante en este dominio mediante la aplicación desarrollada.

Esta experimentación estará basada en dos casos concretos. El primer experimento consistirá en minimizar la probabilidad de presencia de una especie invasora con coste mínimo de las intervenciones a realizar. El segundo experimento pretende favorecer la supervivencia de una población donde cambia el entorno.

Durante la experimentación se representarán los valores de los factores asociados a las celdas mediante un código de colores que se detalla a continuación:

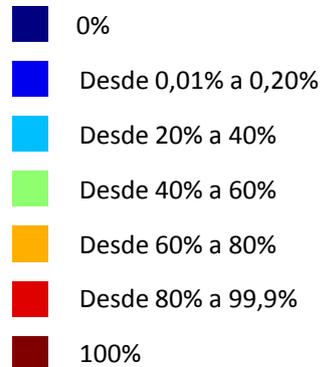


Figura 25. Código de colores empleado para representar resultados de la experimentación

Tal como se explicó en el Alcance del Sistema (**3.1.1 Estructura General**), los valores de los factores son índices normalizados en el rango $[0,1]$. En los ejemplos propuestos, y para el caso del factor agua, un 100% de agua implicará un 0% de suelo habitable. Si este porcentaje se refiere a una población, estaremos hablando de probabilidad de presencia de la población. En el caso de estar tratando la presencia inicial, solo podremos tener el color azul claro (no existe población), o rojo oscuro (existe población).

5.2 Planteamiento Optimización Automática

En este punto se detallarán los objetivos utilizados como funciones de fitness para NSGAI en las pruebas de concepto, así como todas aquellas particularidades, opciones y características que se apliquen, y los operadores que se utilicen.

5.2.1 Objetivos

Para las pruebas de concepto que se tratarán en este proyecto, se consideran dos objetivos para optimizar de forma simultánea, de modo que se pueda tener un Frente de Pareto con varias soluciones “No Dominadas” con los objetivos que se definen a continuación:

- **Presencia:** será el resultado de la acción del autómatas celular sobre la población teniendo en cuenta los valores del atractivo en ese instante. La probabilidad de presencia podría valorarse en un rango de celdas pero, a efectos de este proyecto, se limitará la optimización para un solo instante (el último) y una sola celda objetivo.
- **Coste:** será calculado a partir de las intervenciones como el valor absoluto de la diferencia entre el valor introducido en la intervención y el valor anterior.

Estos objetivos serán las funciones de fitness para nuestro algoritmo genético multiobjetivo (NSGAI), de modo que se aplicará el coste directamente (siempre se pretende minimizar el coste) y la probabilidad de presencia de población tendrá sentidos opuestos si se pretende maximizar u optimizar (será positiva en caso de intentar minimizar y negativa si el usuario quiere maximizar).

5.2.2 Codificación y Operadores

Individuos de la población (Genes)

Para reducir el espacio de búsqueda, se han discretizado los valores de las celdas, de modo que se puedan aplicar 21 niveles de intervención en cada valor del factor relevante para el primer instante. Se explica cada nivel a continuación:

Nivel 0: de 0 a 0,025	Nivel 1: de 0,025 a 0,075	Nivel 2: de 0,075 a 0,125
Nivel 3: de 0,125 a 0,175	Nivel 4: de 0,175 a 0,225	Nivel 5: de 0,225 a 0,275
Nivel 6: de 0,275 a 0,325	Nivel 7: de 0,325 a 0,375	Nivel 8: de 0,375 a 0,425
Nivel 9: de 0,425 a 0,475	Nivel 10: de 0,475 a 0,525	Nivel 11: de 0,525 a 0,575
Nivel 12: de 0,575 a 0,625	Nivel 13: de 0,625 a 0,675	Nivel 14: de 0,675 a 0,725
Nivel 15: de 0,725 a 0,775	Nivel 16: de 0,775 a 0,825	Nivel 17: de 0,825 a 0,875
Nivel 18: de 0,875 a 0,925	Nivel 19: de 0,925 a 0,975	Nivel 20: de 0,975 a 1

Figura 26. Valores discretizados para el módulo de optimización multiobjetivo

Por este motivo, los individuos que formen parte de las generaciones del algoritmo genético multiobjetivo, tendrán un número de genes igual al tamaño del tablero (en ambos casos 25), en el que cada gen podrá tener valores que serán números naturales y que estarán comprendidos entre 0 y 20, ambos inclusive.

Cuando el optimizador propone intervenciones, estas serán los niveles descritos anteriormente, para que pueda comunicar la intervención al módulo de intervención se multiplicará el número de nivel por 0,05 (ejemplo con Nivel 0 se realizaría la operación $0 * 0,05$ que resultaría una intervención con valor 0, si fuera Nivel 10 resultaría un valor 0,5 y Nivel 20 un valor de 1).

Operadores

Para todos los casos de prueba se ha utilizado el algoritmo genético multiobjetivo NSGAI, este tendrá los siguientes operadores:

- Algoritmo de mutación “BitFlipMutation”. Operador que modifica un bit según una probabilidad. En caso de ser un número natural, como en nuestro caso, se escoge un número al azar con mínimos y máximos definidos (0 a 20 en nuestro caso). La probabilidad de la mutación es de 0,33.
- Algoritmo de cruce “SinglePointCrossover”. Operador en el que se establece un único punto de cruce y, en base a este, se intercambian las cadenas entre ambos progenitores. Los organismos resultantes son los descendientes. La probabilidad de cruzamiento es de 0,9
- Algoritmo de Selección “BinaryTournament2”. Operador de selección basado en torneo binario.

Comparadores de selección suele tomar como parámetro un conjunto de soluciones, de regreso de una solución de acuerdo a un criterio. En particular, este operador se aplica un torneo binario.

Parámetros

Los parámetros utilizados como variables de experimentación son los siguientes:

- Población: 10, 50 y 100 individuos.
- Generaciones de entrenamiento: 2.000, 20.000 y 40.000 generaciones.
- Simulación: 100 simulaciones/repeticiones.
- Opción "Sostenido en el tiempo" activado, de forma que una intervención dure en el tiempo.
- La celda a considerar por el objetivo de presencia será la celda superior derecha del tablero.
- Opción "Recurrente" desactivado, una intervención solo genera gasto en el primer instante de tiempo.
- Establecer coste: se determina que el coste será la diferencia, en valor absoluto, entre el nuevo valor y el anterior.

Debe tenerse en cuenta que para esta experimentación se inicializarán los individuos de la primera generación de la siguiente forma:

- La mitad de los individuos se inicializan con valores aleatorios para cada una de sus celdas.
- Para la mitad de los individuos, un individuo tendrá los valores iniciales sin intervención para contar con los factores relevantes iniciales sin intervención y tomar la solución con mínimo coste, el segundo individuo se basará en el primero con intervenciones aleatorias en algunas de sus celdas (existirá una probabilidad de 20% de intervenir cada una de las celdas), para el tercer individuo se basará en el anterior (segundo), con intervenciones aleatorias en cada una de sus celdas (20% de probabilidad de intervenir cada celda). Así sucesivamente se construirán los N individuos que componen la mitad de la población.

5.3 Prueba de concepto I: Plaga

En este experimento, se intenta resolver un hipotético problema de los agricultores ante una plaga de ratones que devastarán los campos de arroz si no impedimos su presencia en la época de cosecha. En esta ocasión, nuestra aplicación buscará minimizar la probabilidad de presencia de esta población.

Para este experimento, se ha considerado que la especie de ratones tiene dos factores relevantes (comida y agua), que no variarán en el tiempo de forma natural (si no se introducen intervenciones).

El modelo de atractivo que determina el atractivo del terreno para la población de ratones ha sido definido exclusivamente para este experimento (en el papel de experto), y se detallará en el apartado dedicado a la expansión de la población.

El objetivo del experimento será evitar, en la medida de lo posible, la presencia de la población en una zona estratégica para nosotros al final de los periodos establecidos (10 instantes). Para lograr esto, se podrán realizar intervenciones sobre los valores del factor relevante de comida, pudiendo modificarlo (positiva o negativamente) con un coste asociado. Estas modificaciones se harán de dos formas: guiadas por un experto a través del generador de escenarios, o automáticamente mediante un algoritmo de optimización multiobjetivo. En este caso se utilizará el algoritmo NSGAII para optimizar los dos objetivos (minimizar coste y probabilidad de presencia de población).

5.3.1 Módulo de simulación: cambio del terreno

Para esta experimentación se suministra los valores de los factores relevantes en el instante de tiempo inicial y final con el que deberá empezar a trabajar la aplicación. Se muestra el instante inicial en la siguiente figura:

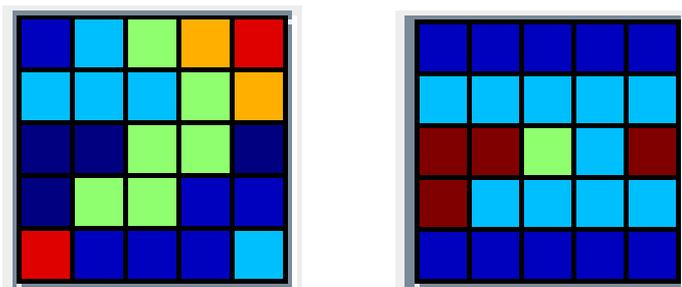


Figura 27. Valores de las celdas para los factores relevantes comida (izquierda) y agua (derecha)

A efectos de este experimento, no existirán cambios para ninguno de los valores de las celdas de los factores relevantes durante los **diez instantes** siguientes, por lo que la figura anterior también representa los valores de los factores relevantes en el instante de tiempo final e intermedios.

En este caso, dado que no hay diferencia entre los estados iniciales y finales, bastará con facilitar los instantes intermedios.

5.3.2 Módulo de simulación: expansión de la población

Para este experimento se ha partido de los datos de presencia inicial que se aportan a continuación:

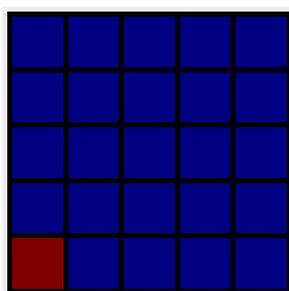


Figura 28. Distribución inicial de la población para el experimento "Plaga"

Tal como se aprecia solo hay una presencia en la esquina inferior izquierda.

La simulación de los cambios en la distribución de la población se ha realizado mediante un autómata celular estocástico definido en el Alcance del Sistema (**3.1.2 Modelado del desplazamiento**). Éste tiene como elemento central el atractivo del terreno, que marca la probabilidad de que la población se extienda hacia celdas adyacentes o incluso mantengan su presencia o no en la celda de partida.

El modelo del atractivo es una función especial que prioriza la presencia de comida si no hay un exceso de agua (dado que la especie animal de estudio es terrestre). Esta fórmula se ha ajustado de la siguiente forma:

$$\text{Atractivo} = \begin{cases} \text{Comida} & \Leftrightarrow 0.6 > \text{agua} \\ 0 & \Leftrightarrow \text{En otro caso} \end{cases}$$

Figura 29. Función utilizada como modelo de atractivo para el experimento "Plaga"

A continuación se muestran los valores del atractivo obtenido como resultado de aplicar los valores de los factores relevantes al modelo de atractivo definido en el autómatas celular. Este atractivo será igual en todos los instantes debido a que no varían las condiciones.

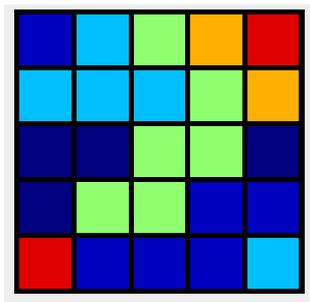


Figura 30. Atractivo para el experimento "Plaga"

Como se puede observar, hay un gran atractivo en la celda objetivo, existiendo un pasillo en el centro del mapa por el que se puede pasar, dado que en los lados derecho e izquierdo existen dos lagos que imposibilitan la vida de la especie en esa zona.

Dada la naturaleza estocástica de la propuesta de dispersión y de la propia simulación, se realizan **100 repeticiones de simulaciones de desplazamiento**. De la proporción de estas en las que una celda resulta ocupada, se deriva una probabilidad de presencia. Se muestra la distribución de población en los diferentes instantes:

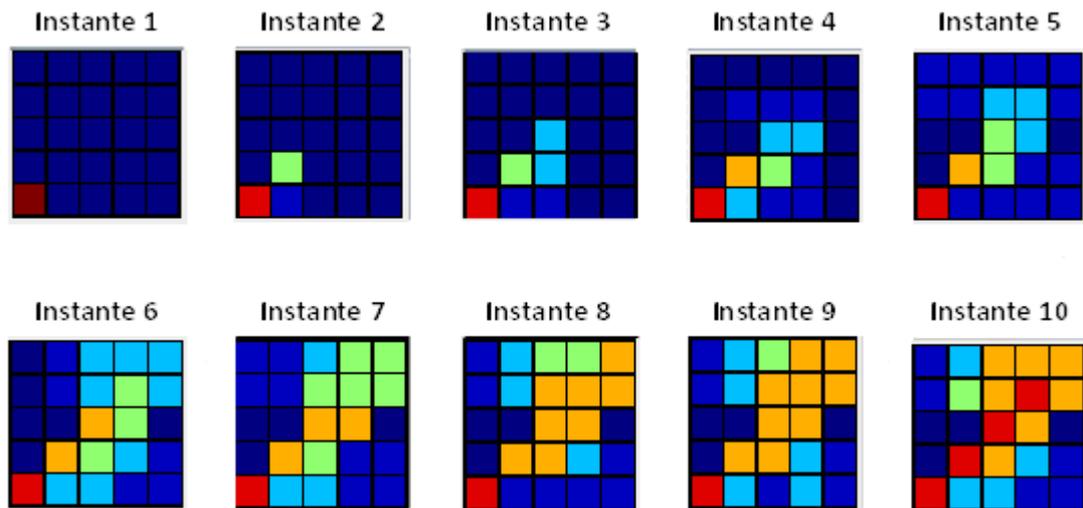


Figura 31. Población Final del Experimento “Plaga”

5.3.3 Generador de escenarios

Como se ha mencionado anteriormente, para este experimento las intervenciones se pueden realizar únicamente para el primer factor relevante (comida), esta intervención se ha considerado que tenga la característica de mantenerse en el tiempo, aunque solo tenga coste la primera vez realizada. Un ejemplo de este tipo de intervención es el que se muestra a continuación:

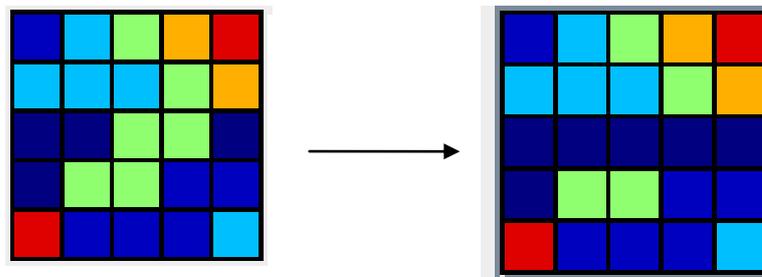


Figura 32. Cambio en el factor relevante “Comida”. Inicial (izquierda), Mapa Intervenido (Derecha)

Por tanto, las variables quedarían de la siguiente forma (se puede comparar con los valores iniciales para apreciar el cambio).

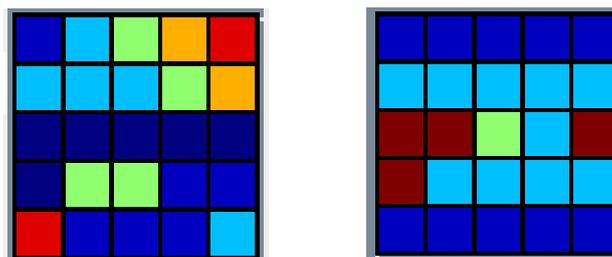


Figura 33. Intervención en el factor relevante comida (izquierda), agua (derecha) sigue igual

Por tanto, una vez realizada las intervenciones el atractivo de los terrenos, calculado a partir del mismo modelo de atractivo descrito anteriormente pero con valores distintos en la entrada (factores relevantes), es el que se muestra a continuación:

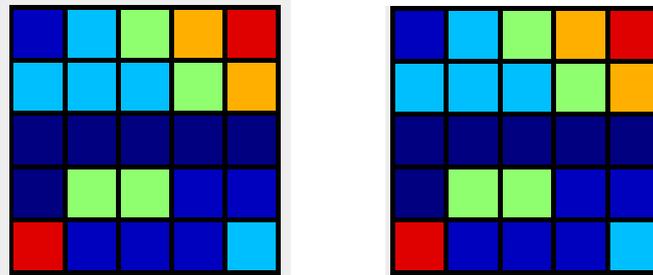


Figura 34. Atractivo inicial (izquierda) y final (derecha) después del cambio

Gracias a esta intervención (dos casillas pasan de valor 50% a 0%), obtenemos la siguiente distribución de población:

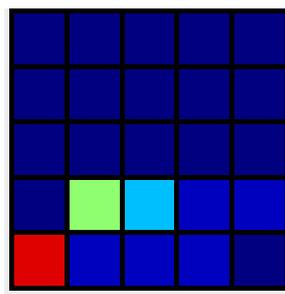


Figura 35. Población Final después de la intervención

El coste de las intervenciones, calculado en base a la diferencia entre el valor anterior y el valor nuevo en valor absoluto y dividido por 100, es de 1. En esta nueva distribución de la población se puede ver como ésta no ha conseguido pasar de la mitad inferior del tablero, logrando así nuestro objetivo de no expansión de la especie.

5.3.4 Optimización

La optimización automática se produce en base al tablero inicial visto anteriormente y las fórmulas de cálculo de atractivo exclusivo de este problema, además se utilizan los siguientes parámetros configurables:

Se debe tener en cuenta que para esta prueba de concepto, se intentará minimizar el objetivo presencia, de modo que la optimización será buscar el mínimo coste con la menor probabilidad de presencia de población posible.

5.3.4.1 Resultados con 100 simulaciones

5.3.4.1.1 Resultados globales

Como se mencionó anteriormente, en la sección **2.2.3** se ha decidido utilizar las métricas Spread (SP) e Hipervolumen (H) para comparar la calidad de los frentes obtenidos en la experimentación. En este caso concreto se obtienen los siguientes resultados para el Spread:

	Media	Mediana	Var.	Max	Min
SP_10_2000	0.676	0.696	0.038	0.993	0.281
SP_10_20000	0.763	0.752	0.033	1.079	0.450
SP_10_40000	0.708	0.705	0.021	0.949	0.364
SP_50_2000	0.716	0.734	0.020	0.984	0.422
SP_50_20000	0.780	0.800	0.022	1.117	0.468
SP_50_40000	0.709	0.727	0.031	1.055	0.347
SP_100_2000	0.636	0.650	0.030	0.967	0.183
SP_100_20000	0.742	0.795	0.025	0.976	0.372
SP_100_40000	0.709	0.698	0.024	0.996	0.404

Tabla 77: métrica SP (Experimento "Plaga" con 100 simulaciones)

Según se aprecia, el Spread mejora con el tamaño de la población. Por otro lado, vemos un repunte sistemático al pasar de 2.000 a 20.000 generaciones que se corrige al volver a incrementar el número de iteraciones hasta 40.000.

A efectos de comprobar que las diferencias que hemos observado entre las medias de la tabla anterior son estadísticamente representativas y no fruto del mero azar los resultados han sido objeto de contrastes estadísticos formales. Esto es necesario porque desde el punto de vista estadístico, podemos considerar cada grupo de 30 pruebas independientes de la relación población-número de generaciones, como muestras de una función de densidad de probabilidad cuyos estadísticos desconocemos y que no tendrían que coincidir con los observados

Primero se aplica un test de Lilliefors para determinar si los valores obtenidos siguen una distribución normal (de Gauss). Si la siguen, se utiliza el test de Levene para comprobar la homocedasticidad de las muestras (igualdad de las varianzas). Si este test es positivo (las varianzas son iguales), se realiza un t-test; en caso contrario, se usa el test de Welch. Para distribuciones no gaussianas, se emplea el test no paramétrico de Wilcoxon para comparar las medianas de los resultados. Se han establecido dos niveles de confianza para todos los test, al 99% y 95% (es decir, niveles de significación

de 1% y 5%). Todas estas pruebas se han realizado mediante un script implementado en lenguaje R.

	SP_10_2000	SP_10_20000	SP_10_40000	SP_50_2000	SP_50_20000	SP_50_40000	SP_100_2000	SP_100_20000
SP_10_20000	+							
SP_10_40000	=	=						
SP_50_2000	=	=	=					
SP_50_20000	+	=	+	+				
SP_50_40000	=	=	=	=	-			
SP_100_2000	=	--	-	-	--	=		
SP_100_20000	=	=	=	=	=	=	++	
SP_100_40000	=	=	=	=	-	=	+	=

Tabla 78: métrica SP (Experimento "Plaga" con 100 simulaciones)

Se observa que la mayor parte de las diferencias apreciadas en la tabla anterior no son significativas.

Los signos "++" y "--" representan una diferencia significativa del 1% entre los elementos contrastados, mientras que "+" y "-" simbolizan una diferencia significativa del 5%, y el signo "=" indica que no existen diferencias significativas entre los elementos.

Los contrastes se llevan a cabo comparando el elemento de cada fila con el de cada columna, de modo que el signo "++" representa que el elemento de la fila que se está comparando es mayor con una diferencia significativa al 1% con respecto al elemento de la columna comparado.

	Media	Mediana	Var.	Max	Min
HP_10_2000	0.378	0.350	0.014	0.771	0.226
HP_10_20000	0.431	0.392	0.011	0.796	0.337
HP_10_40000	0.451	0.422	0.007	0.808	0.352
HP_50_2000	0.379	0.356	0.013	0.649	0.216
HP_50_20000	0.466	0.429	0.013	0.864	0.331
HP_50_40000	0.475	0.455	0.008	0.740	0.357
HP_100_2000	0.414	0.375	0.021	0.723	0.199
HP_100_20000	0.440	0.392	0.015	0.845	0.331
HP_100_40000	0.436	0.413	0.005	0.671	0.333

Tabla 79: métrica HP (Experimento "Plaga" con 100 simulaciones)

La métrica de hipervolumen no muestra diferencias notables entre los resultados obtenidos. Mostrando que el área que abarca la solución con población 50 y 40.000 generaciones es mayor y por tanto es mejor que el resto, también es mejor en general con una población de 50. Lo que si se observa es en una mejora al aumentar las iteraciones.

	HP_10_2000	HP_10_20000	HP_10_40000	HP_50_2000	HP_50_20000	HP_50_40000	HP_100_2000	HP_100_20000
HP_10_20000	++							
HP_10_40000	++	+						
HP_50_2000	=	-	--					
HP_50_20000	++	=	=	++				
HP_50_40000	++	++	=	++	=			
HP_100_2000	=	=	-	=	=	-		
HP_100_20000	++	=	-	+	=	--	=	
HP_100_40000	++	=	=	+	=	-	=	=

Tabla 80: métrica HP (Experimento "Plaga" con 100 simulaciones)

Una vez más se observa que la mayor parte de las diferencias apreciadas en la tabla anterior no son significativas.

El "Frente de Pareto" considerado para este problema, se muestra a continuación de forma gráfica. Se ha obtenido mediante los resultados de toda la experimentación en este dominio con 100 simulaciones. Este Frente es utilizado para el cálculo de las métricas utilizadas en este proyecto y que facilita el framework JMetal.

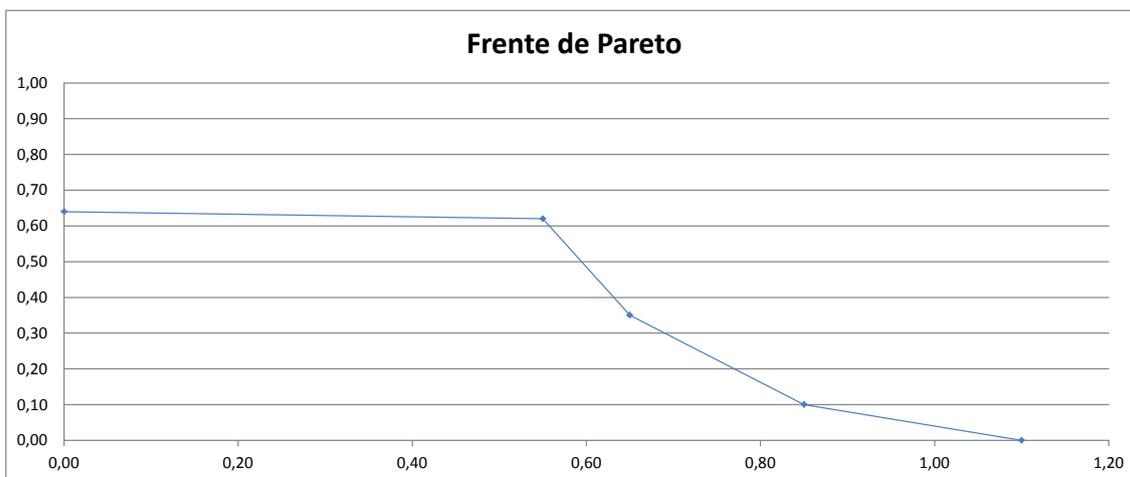


Figura 36. Soluciones "No Dominadas" (frente de Pareto) (Experimento "Plaga" con 100 simulaciones)

5.3.4.1.2 Resultados agrupados por generaciones

A continuación se mostrará la comparativa entre las distintas poblaciones para 2.000 generaciones:

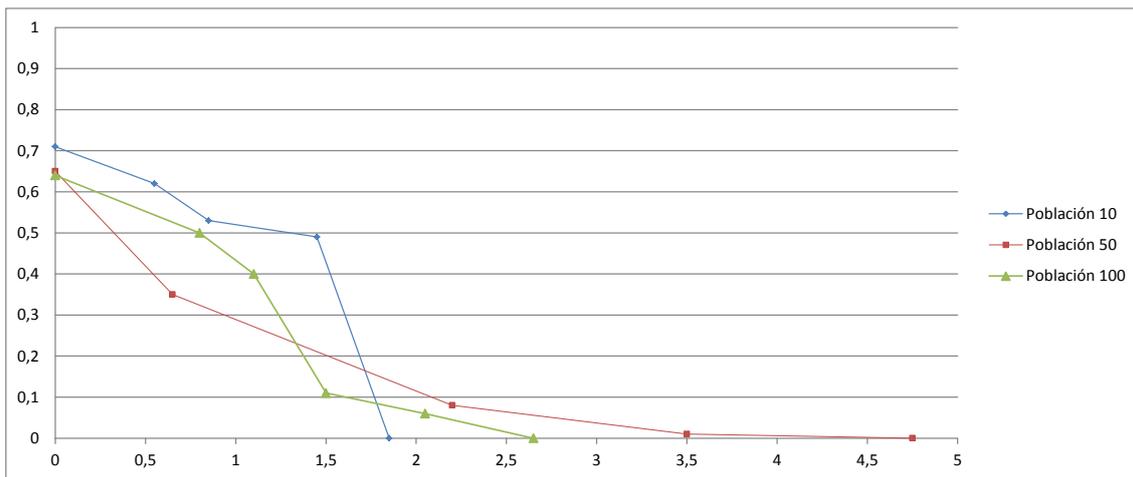


Figura 37. Comparativa de soluciones “No Dominadas” para 2.000 generaciones (Experimento “Plaga” con 100 simulaciones)

En el anterior gráfico se puede apreciar que para todas las experimentaciones para las distintas poblaciones se obtiene un conjunto de soluciones “No Dominadas” similar en número. En este caso, también se puede destacar que con población igual a 50 se obtiene una solución mejor que con población 100. Las peores soluciones son las arrojadas por población igual a 10 que sin embargo alcanza la solución de menor coste para una falta total de presencia en la casilla objeto de estudio.

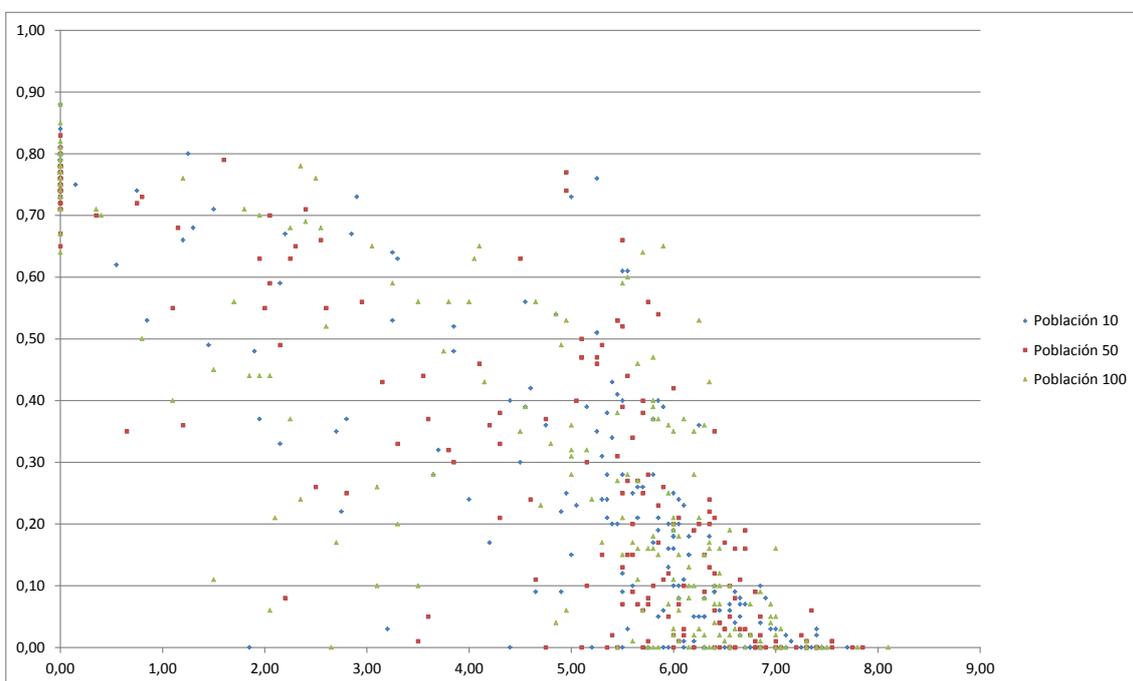


Figura 38. Total de soluciones para 2.000 generaciones (Experimento “Plaga” con 100 simulaciones)

La figura anterior detalla las soluciones obtenidas en las ejecuciones con 2.000 generaciones.

Lo primero que podemos observar es que no parece haber una relación evidente entre las mejores soluciones y el tamaño de la población.

También se puede encontrar una gran dispersión para las soluciones con unos mismos valores de entrada para coste igual a 0. Esto significa que para una misma entrada conduzca a valores muy distintos en el objetivo de permanencia.

Se presenta a continuación la comparativa con 20.000 generaciones:

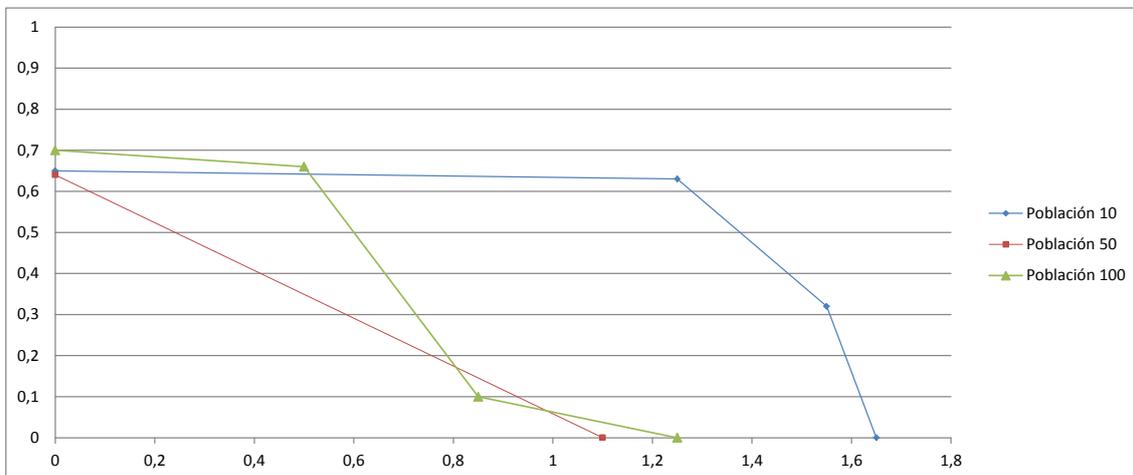


Figura 39. Comparativa de soluciones "No Dominadas" para 20.000 generaciones (Experimento "Plaga" con 100 simulaciones)

En el gráfico anterior se puede observar que para población igual a 10, se arrojan las peores soluciones, mientras que se producen soluciones mejores si en la experimentación se utiliza un tamaño de población de 100 o 50, ofreciendo este último mejores soluciones. Es muy llamativa la baja cantidad de soluciones obtenidas, en especial para población 50.

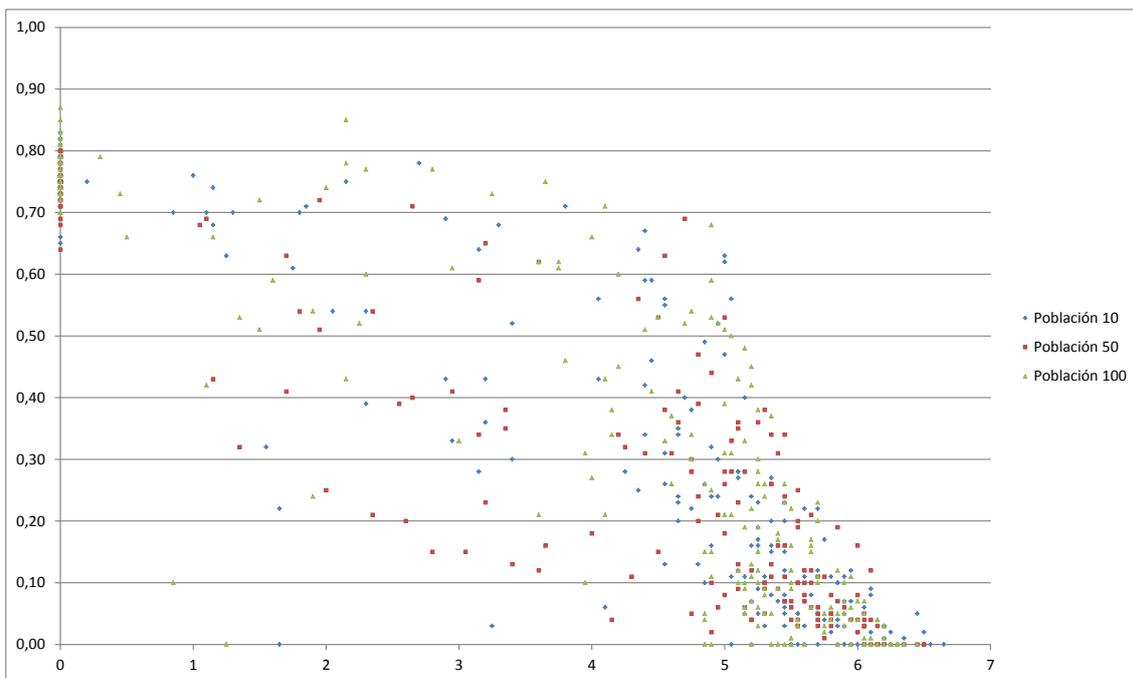


Figura 40. Comparativa de soluciones "No Dominadas" para 20.000 generaciones (Experimento "Plaga" con 100 simulaciones)

La distribución de las soluciones muestran que hay una concentración importante de puntos con probabilidad de presencia baja con coste aproximadamente 2.5, esto refuerza los resultados obtenidos anteriormente.

Por último, se muestra la comparativa para 40.000 generaciones:

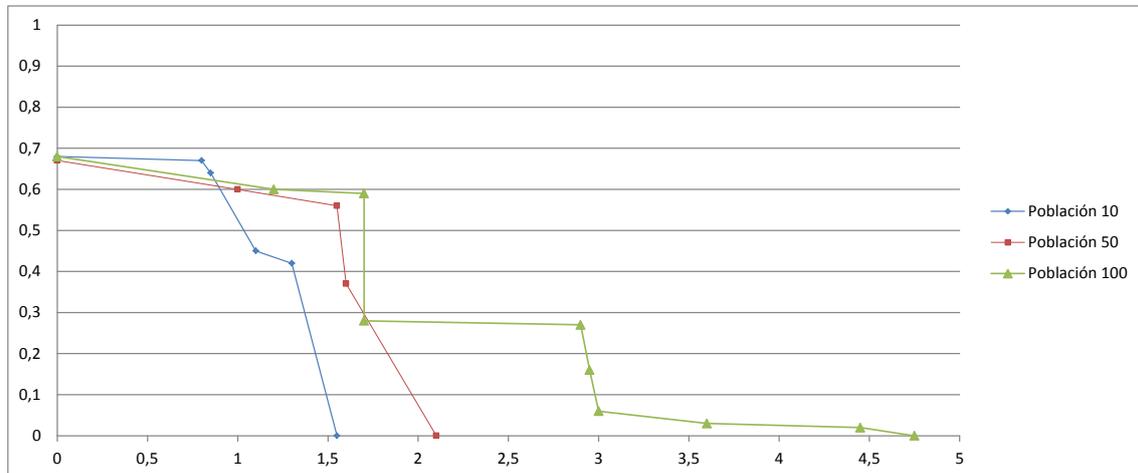


Figura 41. Comparativa de soluciones “No Dominadas” para 40.000 generaciones (Experimento “Plaga” con 100 simulaciones)

En el que se puede encontrar una incongruencia en los resultados mayor que en los anteriores resultados.

Los resultados son muy llamativos, puesto que se aprecia que la población más pequeña es la que lleva las mejores soluciones. También es destacable que las poblaciones de tamaño 10 y 100 se centren en aspectos distintos. La primera en soluciones de alta probabilidad de presencia y la segunda en intervenciones que frenan la plaga de forma más efectiva.

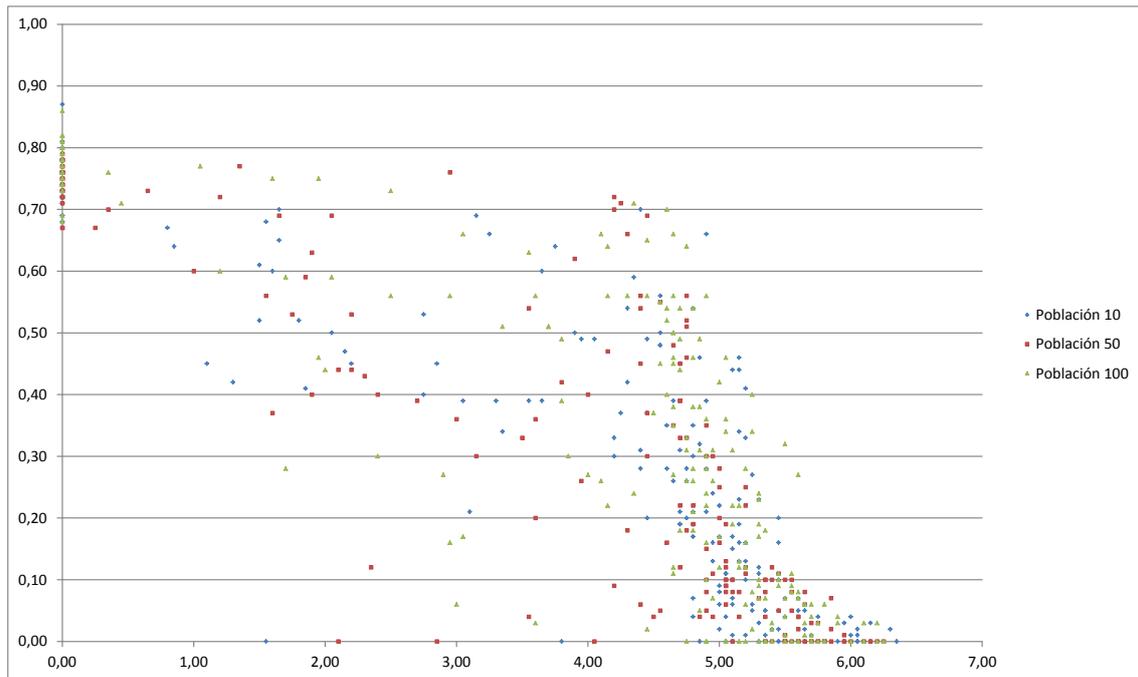


Figura 42. Total de soluciones para 40.000 generaciones (Experimento “Plaga” con 100 simulaciones)

Como en el anterior caso, se produce una gran dispersión de nuevo y volvemos a ver que al simular dos veces el mismo escenario con una misma entrada, se obtienen dos salidas distintas (en una casilla determinada en ocasiones hay presencia de población y en otra no).

La misma dispersión se observa una gran dispersión de las soluciones conforme nos alejamos de una probabilidad de presencia de 0%. La causa de no obtener mejores soluciones con 40.000 generaciones puede encontrar su génesis en esta dispersión, aunque generalmente se obtengan mejores soluciones con experimentaciones con muchas generaciones y una cantidad de población elevada, por azar se han obtenido mejores soluciones con una cantidad de generaciones baja.

5.3.4.1.3 Resultados agrupados por tamaño de población

A continuación se estudiarán las agrupaciones por tamaño de población con las soluciones agregadas por generaciones.

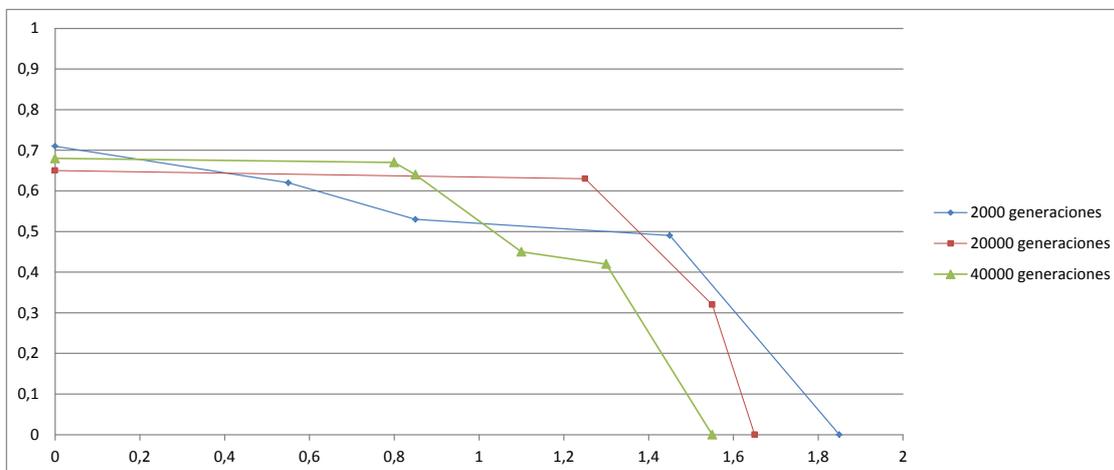


Figura 43. Comparativa de soluciones “No Dominadas” para población igual a 10 (Experimento “Plaga” con 100 simulaciones)

Aunque para probabilidades de presencia de población altas dominan las soluciones con 2.000 generaciones, para valores menores dominan las soluciones con 40.000 generaciones.

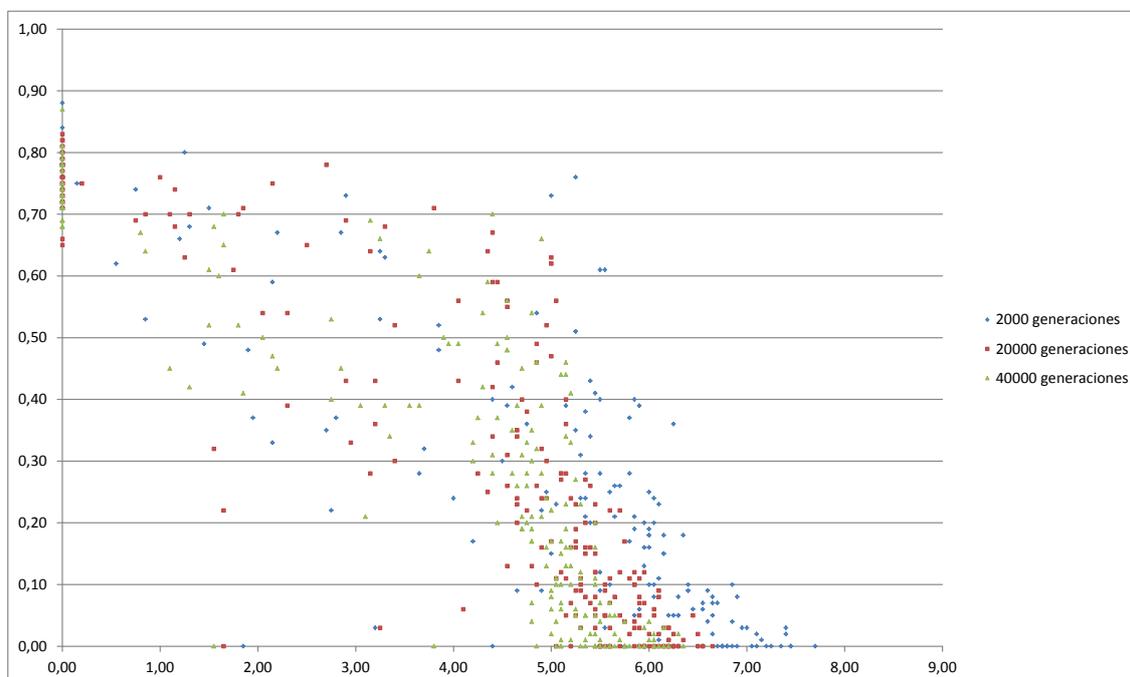


Figura 44. Total de soluciones para población 10 (Experimento “Plaga” con 100 simulaciones)

En el anterior gráfico se puede observar que se está produciendo una evolución de las soluciones al aumentar el número de generaciones. Para 2.000 generaciones se ocupan los costes más altos con probabilidad de presencia 0%, mientras que con

20.000 generaciones se ocupan costes más bajos (coste 6 de media) y con 40.000 aun más (aproximadamente coste 5,5 de media).

También puede destacarse la gran dispersión en las soluciones que puede apreciarse con las soluciones con coste 0. La misma solución puede ofrecer una probabilidad de presencia desde 6,5 a 8,5.

Se muestra a continuación la comparativa con población igual a 50:

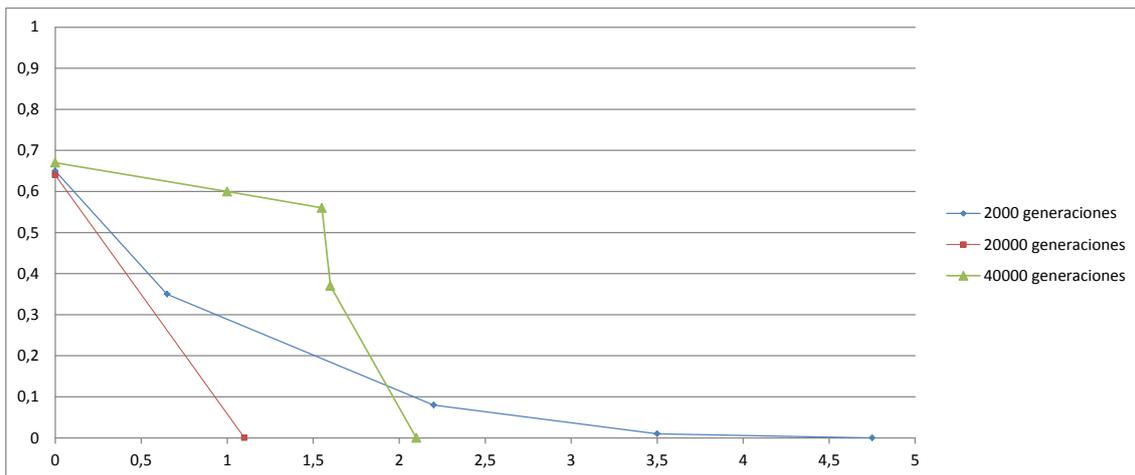


Figura 45. Comparativa de soluciones “No Dominadas” para población igual a 50 (Experimento “Plaga” con 100 simulaciones)

En la figura anterior puede verse como con 20.000 generaciones se producen pocas soluciones, aunque mejoran todas el resto. También puede verse como con 40.000 generaciones se explora las soluciones con probabilidad de presencia alta, dando peores resultados que el resto, y con 2.000 generaciones se exploran las soluciones con probabilidad de presencia baja siendo los peores resultados con respecto a las demás en este tipo de soluciones.

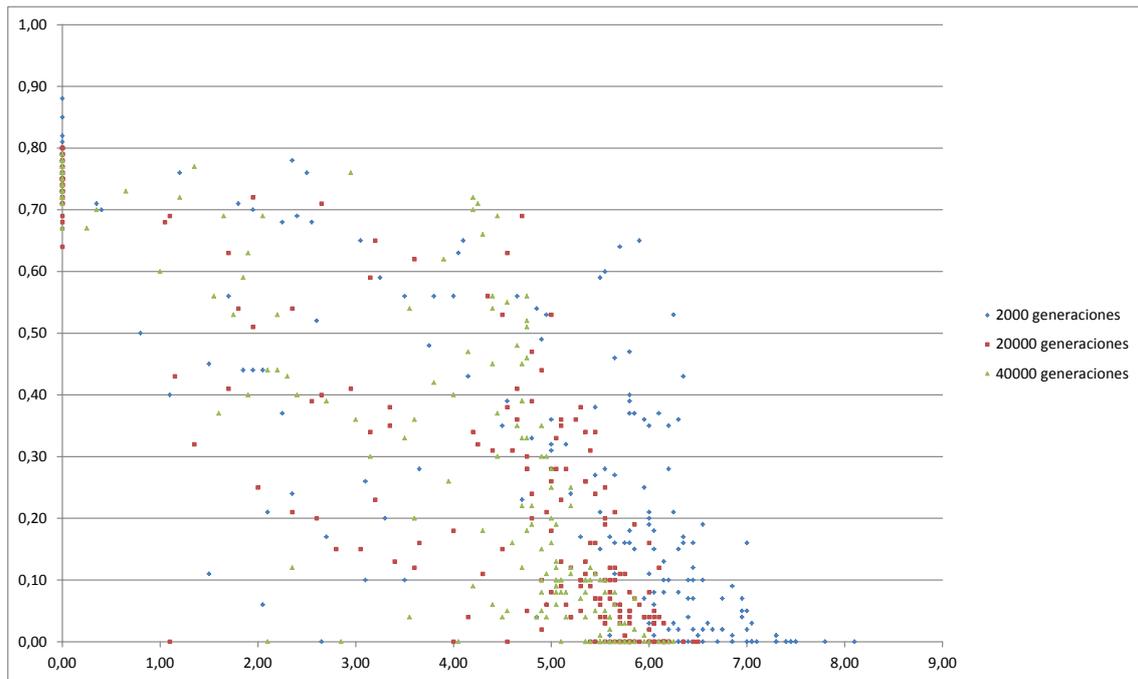


Figura 46. Total de soluciones para población 50 (Experimento “Plaga” con 100 simulaciones)

En la figura anterior se puede apreciar que, generalmente, con 2.000 generaciones se obtienen soluciones con mayor coste. No obstante, con la distribución de datos resultante no se puede asegurar que con 20.000 generaciones se puedan obtener mejores soluciones que con 40.000, incluso los resultados parecen sugerir lo contrario por poca diferencia.

A continuación se muestra la comparativa con población igual a 100:

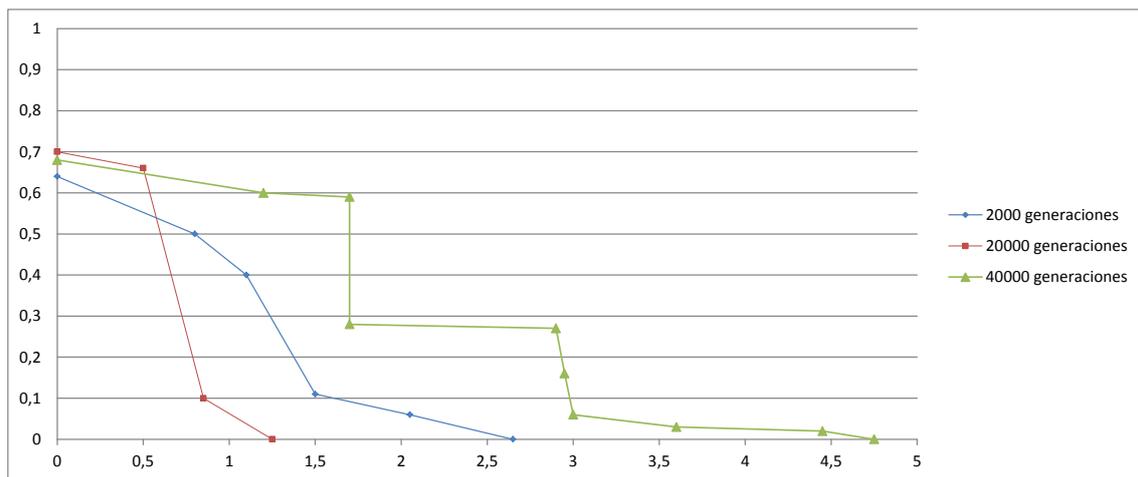


Figura 47. Comparativa de soluciones “No Dominadas” para población igual a 100 (Experimento “Plaga” con 100 simulaciones)

Para población igual a 100 se encuentra que con 40.000 generaciones se obtiene muchas soluciones, pero todas son peores comparadas con las obtenidas con 20.000 y 2.000 generaciones. Las mejores soluciones son las obtenidas por las ejecuciones con 20.000 generaciones.

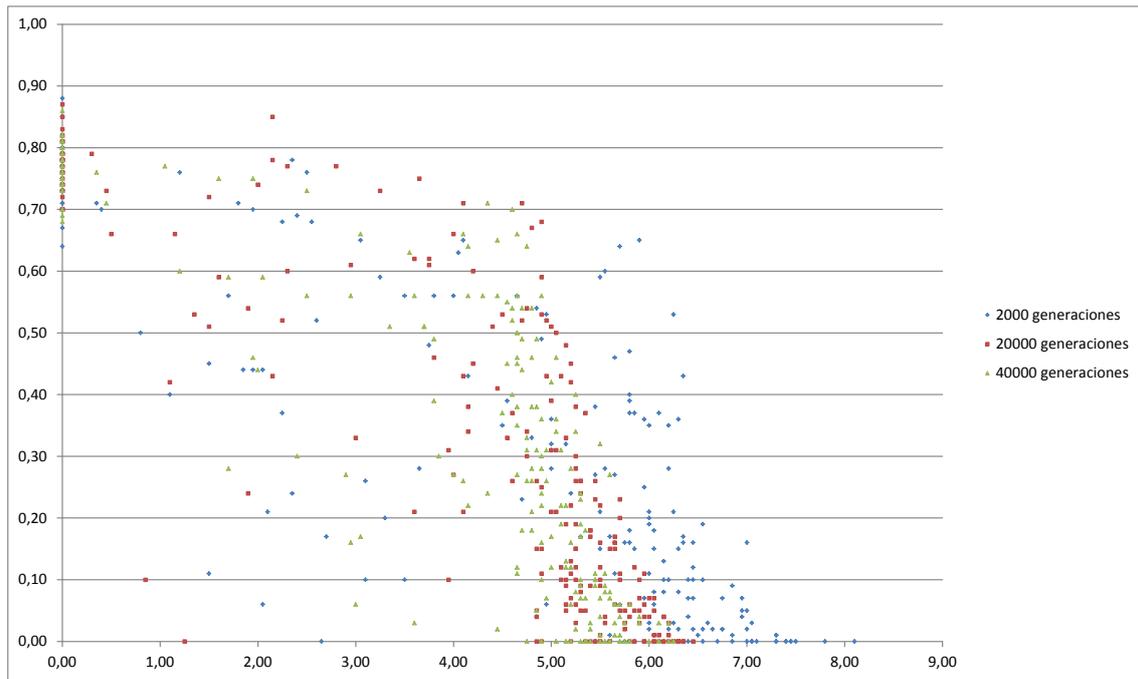


Figura 48. Total de soluciones para población 100 (Experimento “Plaga” con 100 simulaciones)

Como en los anteriores casos, se ve un progreso de las soluciones en la relación coste - probabilidad de presencia conforme se aumenta el número de iteraciones. No obstante, los resultados parecen indicar que las soluciones son similares en calidad con 20.000 y 40.000 generaciones al ser conjuntos no tan distantes como las resultantes con 2.000 generaciones.

Una posible causa de la variabilidad que se puede ver con coste 0 es la variabilidad que introduce el autómata celular estocástico, para demostrar esto se debe reducir la variabilidad y comprobar que los resultados mejoran. Por esta razón se volverá a realizar la experimentación con un número de repeticiones mayor que rebaje la variabilidad de probabilidad de presencia, para esto se propone subir este parámetro a 1.000 repeticiones.

5.3.4.2 Resultados con 1000 simulaciones

5.3.4.2.1 Resultados globales

Se muestran ahora las distintas métricas utilizadas:

	Media	Mediana	Var.	Max	Min
SP_10_2000	0.671	0.645	0.060	1.089	0.204
SP_10_20000	0.723	0.710	0.022	1.134	0.368
SP_10_40000	0.755	0.763	0.026	1.133	0.37
SP_50_2000	0.707	0.694	0.025	1.045	0.459
SP_50_20000	0.753	0.752	0.029	1.127	0.377
SP_50_40000	0.713	0.732	0.022	1.029	0.338
SP_100_2000	0.688	0.664	0.026	1.078	0.433
SP_100_20000	0.724	0.744	0.023	1.149	0.392
SP_100_40000	0.729	0.726	0.014	0.921	0.395

Tabla 81: métrica SP (Experimento "Plaga" con 1000 simulaciones)

En la tabla anterior se puede ver que para poblaciones 10 y 100 los resultados empeoran al aumentar las iteraciones, con población 50 la tendencia cambia y es inversa de 40.000 generaciones a 20.000 generaciones.

	SP_10_2000	SP_10_20000	SP_10_40000	SP_50_2000	SP_50_20000	SP_50_40000	SP_90_2000	SP_100_20000
SP_10_20000	=							
SP_10_40000	=	=						
SP_50_2000	=	=	=					
SP_50_20000	=	=	=	=				
SP_50_40000	=	=	=	=	=			
SP_100_2000	=	=	=	=	=	=		
SP_100_20000	=	=	=	=	=	=	=	
SP_100_40000	=	=	=	=	=	=	=	=

Tabla 82: métrica SP (Experimento "Plaga" con 1000 simulaciones)

En este caso, no pueden establecerse diferencias significativas entre los diferentes elementos contrastados.

	Media	Mediana	Var.	Max	Min
HP_10_2000	0.419	0.388	0.015	0.853	0.272
HP_10_20000	0.477	0.459	0.007	0.669	0.364
HP_10_40000	0.494	0.464	0.007	0.720	0.381
HP_50_2000	0.431	0.411	0.015	0.683	0.295
HP_50_20000	0.479	0.456	0.007	0.67	0.36
HP_50_40000	0.497	0.472	0.009	0.777	0.394
HP_100_2000	0.416	0.407	0.008	0.588	0.285
HP_100_20000	0.476	0.449	0.007	0.736	0.368
HP_100_40000	0.493	0.461	0.008	0.77	0.389

Tabla 83: métrica HP (Experimento "Plaga" con 1000 simulaciones)

Observando la tabla anterior, podemos ver que los resultados mejoran al aumentar las iteraciones, puesto que los valores de hipervolumen indican que las soluciones abarcan un área mayor.

	HP_10_2000	HP_10_20000	HP_10_40000	HP_50_2000	HP_50_20000	HP_50_40000	HP_100_2000	HP_100_20000
HP_10_20000	++							
HP_10_40000	++	=						
HP_50_2000	=	-	--					
HP_50_20000	++	=	=	+				
HP_50_40000	++	=	=	++	=			
HP_100_2000	=	-	--	=	--	--		
HP_100_20000	++	=	=	=	=	=	++	
HP_100_40000	++	=	=	++	=	=	++	=

Tabla 84: métrica HP (Experimento "Plaga" con 1000 simulaciones)

El contraste indica una diferencia significativa al 5% entre los valores con 20.000 y 2.000 generaciones y los de 40.000 generaciones y 2.000 generaciones. El área que abarca las soluciones con 40.000 y 20.000 generaciones es mayor y por tanto es mejor, que las de 2.000 generaciones. El contraste indica que la diferencia entre 20.000 y 40.000 generaciones no son significativas.

A continuación se muestra el Frente de Pareto para el conjunto experimental con 1.000 simulaciones:

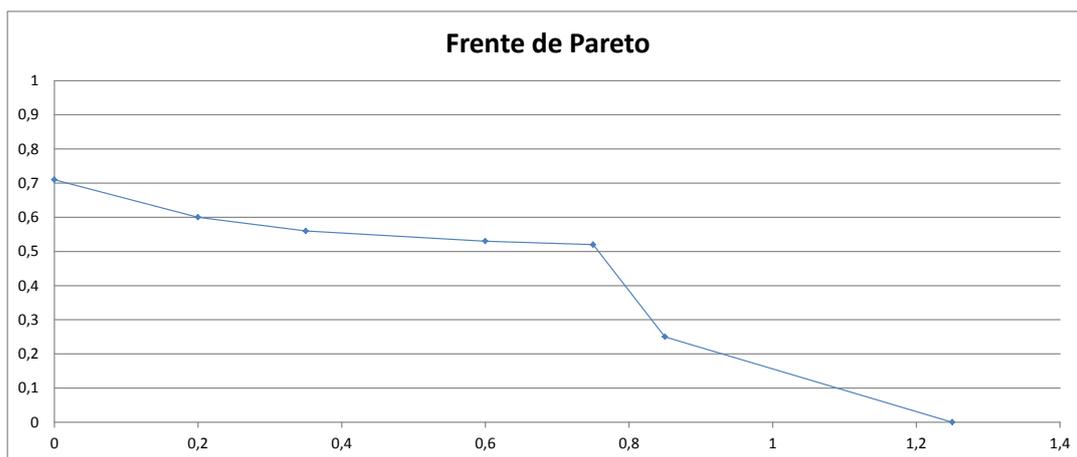


Figura 49. Soluciones "No Dominadas" (frente de Pareto) (Experimento "Plaga" con 1000 simulaciones)

Se puede apreciar que este frente de Pareto tiene soluciones peores que el resultante de la experimentación con 100 simulaciones, por ejemplo, el punto con menor probabilidad de presencia tiene coste 1,2, mientras que con 100 simulaciones se tenían soluciones con coste menor que 1,1.

5.3.4.2.2 Resultados agrupados por generaciones

A continuación se muestra un análisis de las soluciones “no dominadas” agrupadas por generaciones. Se muestran los frentes para 2.000 generaciones:

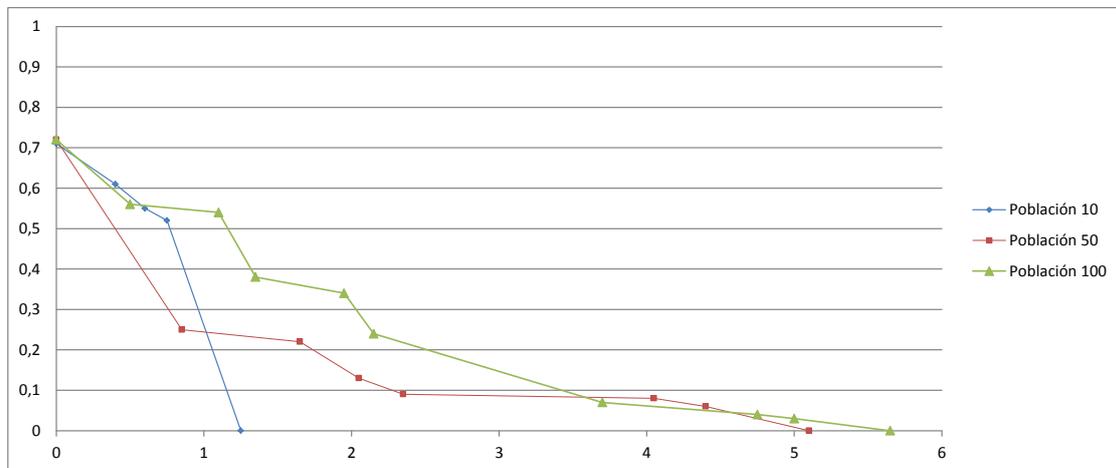


Figura 50. Comparativa de soluciones “No Dominadas” para 2.000 generaciones (Experimento “Plaga” con 1000 simulaciones)

Se pueden observar que para poblaciones 50 y 100 se exploran las soluciones con probabilidad de presencia baja, mientras que, aunque para población 10 se generan menos puntos, estos exploran las soluciones con probabilidades de presencias altas. También se puede apreciar que con población 10 se encuentra la mejor solución con coste bajo y con probabilidad de presencia de 0. El segundo mejor punto es resultado de la ejecución con población 50.

Vemos a continuación el gráfico que muestra todas las soluciones:

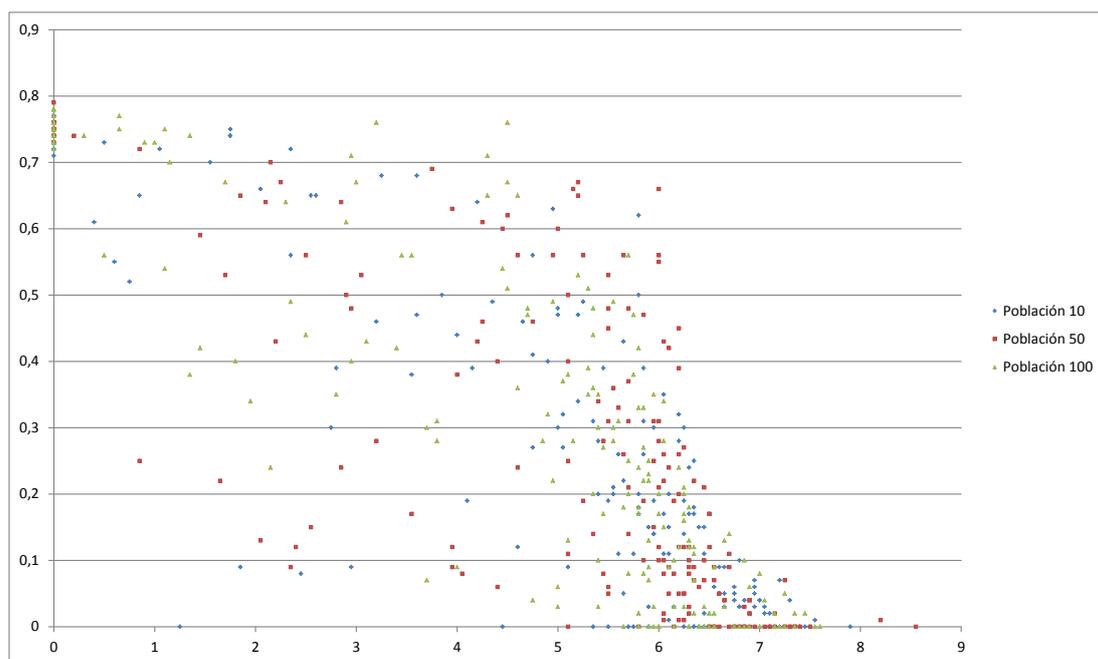


Figura 51. Total de soluciones para 2.000 generaciones (Experimento “Plaga” con 1000 simulaciones)

Como puede apreciarse, la dispersión para las soluciones con coste 0 y probabilidad de presencia alta ha descendido a la mitad con respecto a la experimentación con 100 simulaciones.

Estudiamos ahora la agrupación por población para los experimentos con 20.000 generaciones:

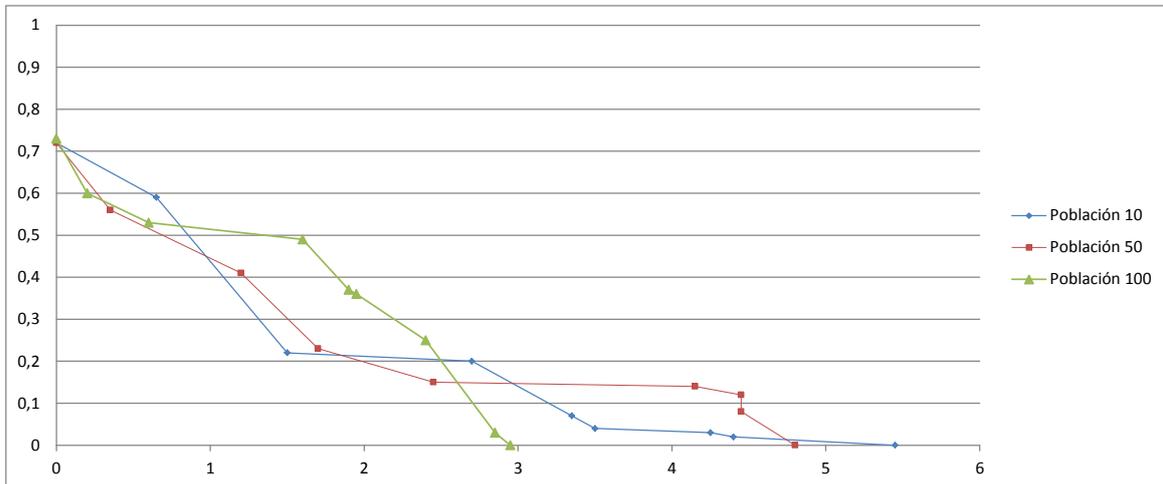


Figura 52. Comparativa de soluciones "No Dominadas" para 20.000 generaciones (Experimento "Plaga" con 1000 simulaciones)

En la figura anterior se muestra que con población 100 se encuentran mejores soluciones con probabilidad de presencia baja. Para probabilidad de presencias medias se obtienen mejores soluciones con poblaciones de 10 y 50 y para probabilidades de presencia bajas las mejores soluciones son las de población 50 y 100. Aunque para población 100 se encuentre las mejores soluciones con mayor y menor coste (y con menor probabilidad de presencia y mayor probabilidad de presencia respectivamente), las soluciones intermedias son peores que con las poblaciones de 10 y 50.

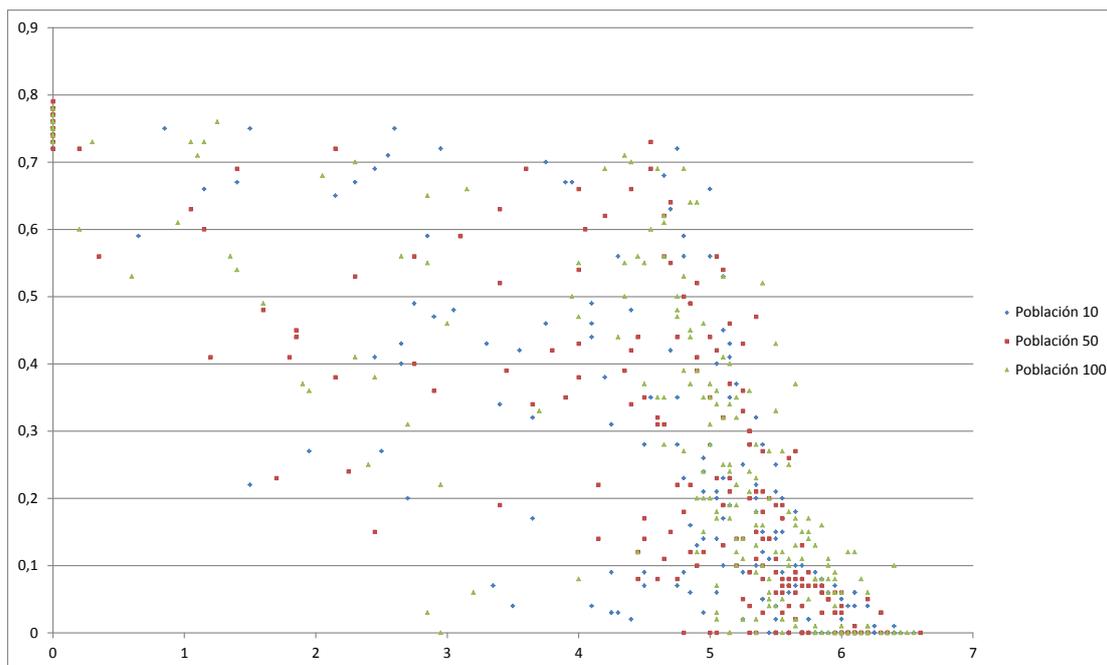


Figura 53. Total de soluciones para 20.000 generaciones (Experimento “Plaga” con 1000 simulaciones)

No se encuentra que destaquen las soluciones con población 100 como sugiere el anterior gráfico que muestra soluciones mejores para población 100 en los extremos.

Como último estudio de la agrupación por población se estudia las generadas para 40.000 generaciones:

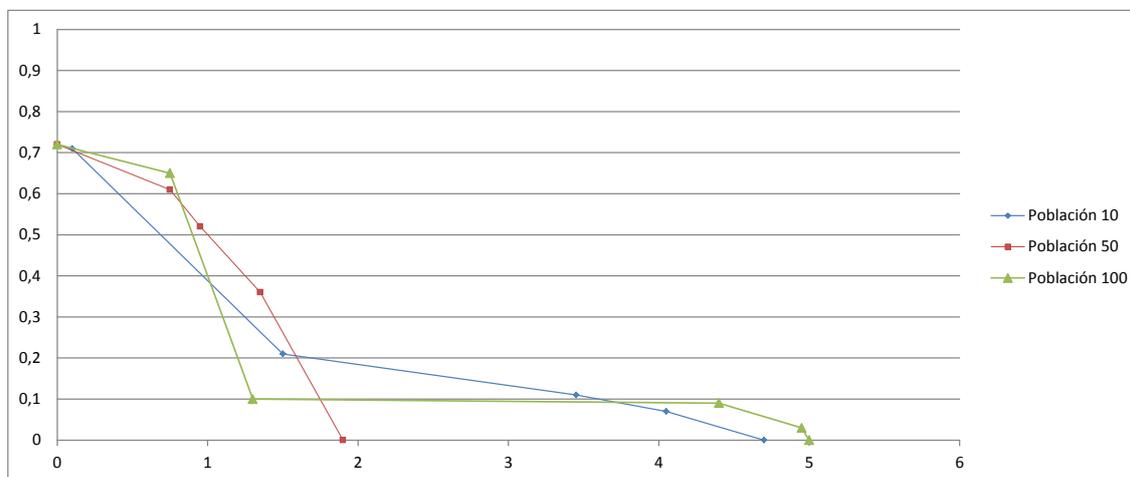


Figura 54. Comparativa de soluciones “No Dominadas” para 40.000 generaciones (Experimento “Plaga” con 1000 simulaciones)

Aunque para población 100 exista una solución con probabilidad de presencia de 10% que es mejor que cualquier punto con la misma probabilidad, cuando la probabilidad de presencia disminuye el coste aumenta mucho, permitiendo que tanto la población 10 como la 50 (esta última alcanzando un valor que supera por mucha diferencia al resto) superen a la solución con población 100.

A continuación se muestran todos los datos para 40.000 generaciones:

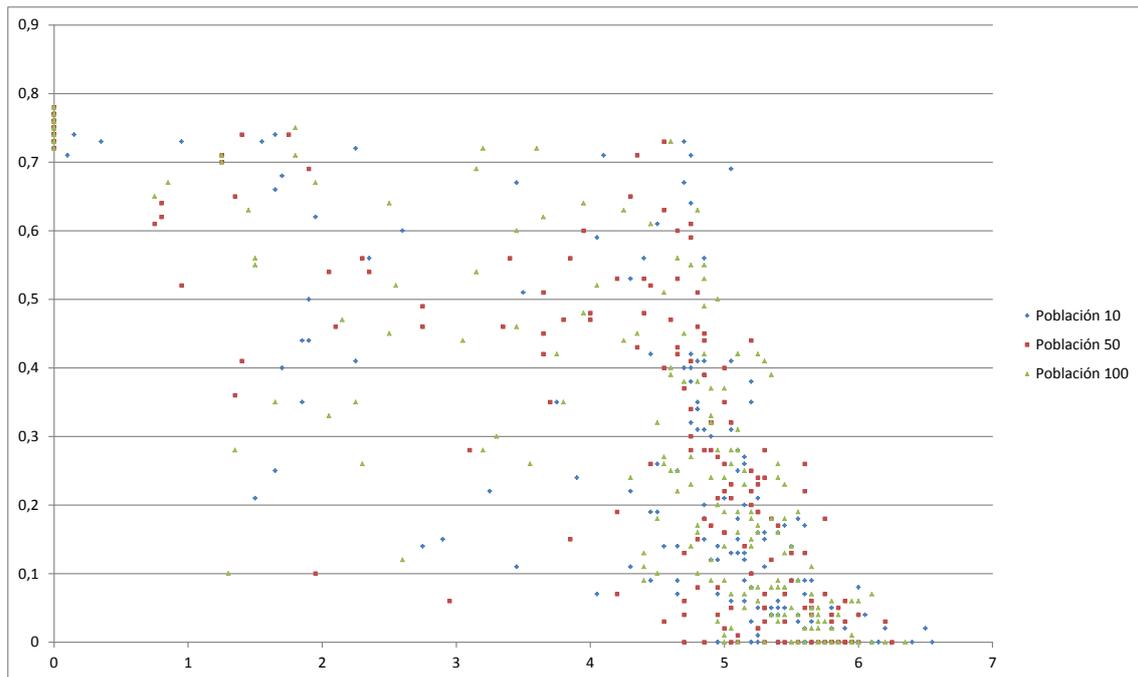


Figura 55. Total de soluciones para 40.000 generaciones (Experimento "Plaga" con 1000 simulaciones)

Se vuelve a observar que no existe una clara evolución en los resultados al aumentar el tamaño de la población.

5.3.4.2.3 Resultados agrupados por tamaño de población

Como en el anterior experimento, se concluye con las agrupaciones en generaciones de tamaño de población, se comienza por la variable con población igual a 10:

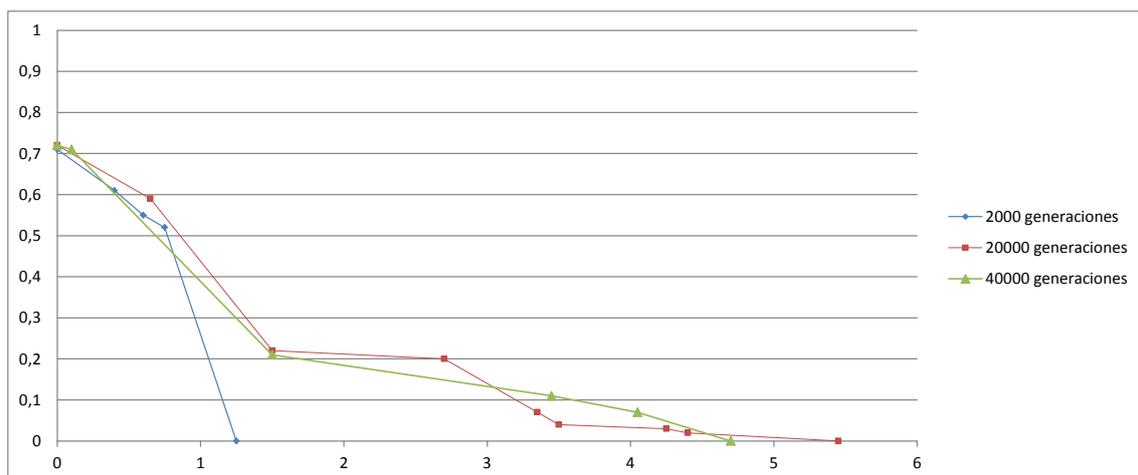


Figura 56. Comparativa de soluciones "No Dominadas" para población 10 (Experimento "Plaga" con 1000 simulaciones)

Se puede ver que las soluciones con 20.000 y 40.000 generaciones son parecidas, mientras que las encontradas para 2.000 generaciones ofrece soluciones mejores que las encontradas para el resto de generaciones.

También puede destacarse la concentración de puntos para 20.000 y 40.000 generaciones para probabilidades de presencia bajas, para 2.000 generaciones esta concentración se produce con probabilidades de presencia más altas.

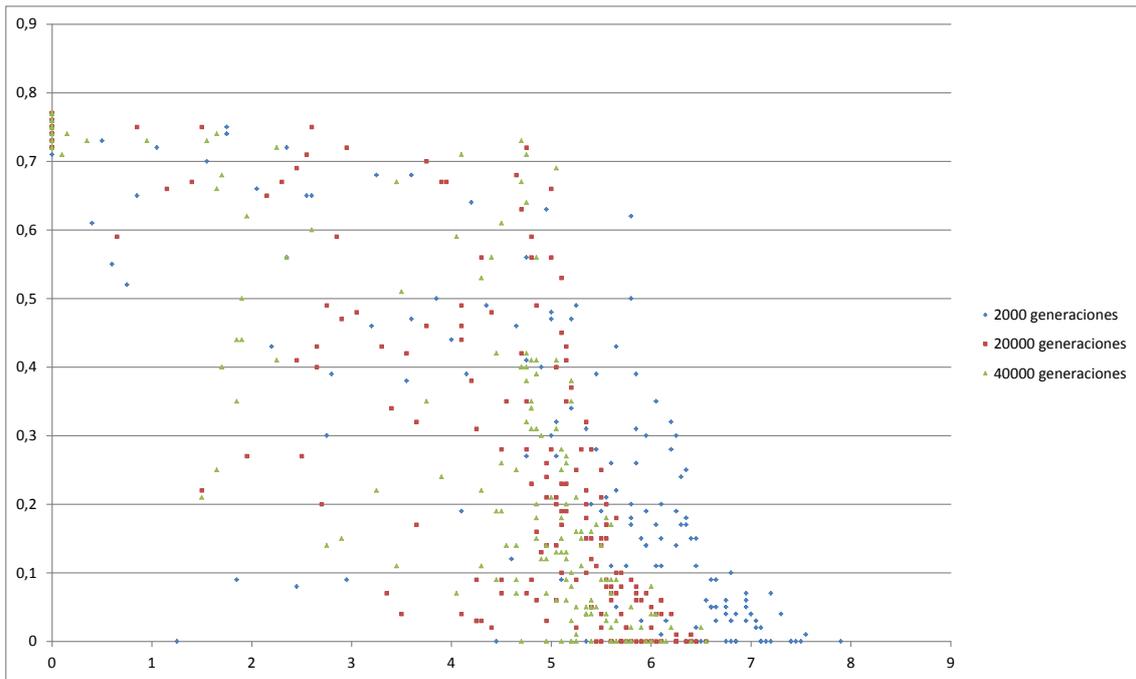


Figura 57. Total de soluciones para población 10 (Experimento "Plaga" con 1000 simulaciones)

En este nuevo gráfico del total de soluciones, se puede volver a encontrar la evolución en las soluciones al aumentar el número de generaciones, aunque ese aumento no es tan evidente al aumentar de 20.000 a 40.000 generaciones.

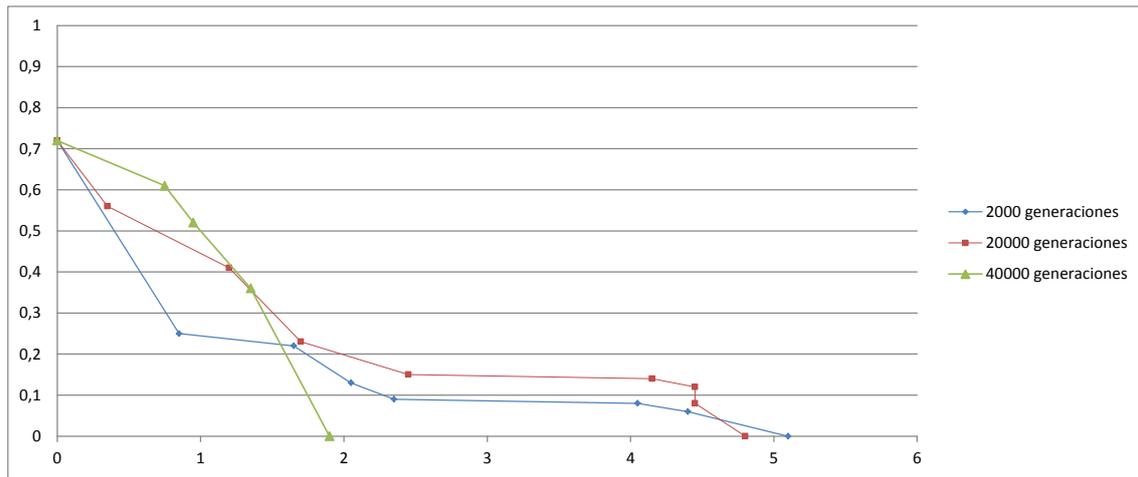


Figura 58. Comparativa de soluciones “No Dominadas” para población 50 (Experimento “Plaga” con 1000 simulaciones)

Se estudia los resultados con población 50, encontrándose un punto con 40.000 generaciones que destaca sobre el resto para probabilidad de presencia de 0%, también cabe destacar el punto de 2.000 generaciones con probabilidad de presencia del 25% y la acumulación de puntos para 2.000 y 20.000 generaciones y probabilidad de presencia baja.

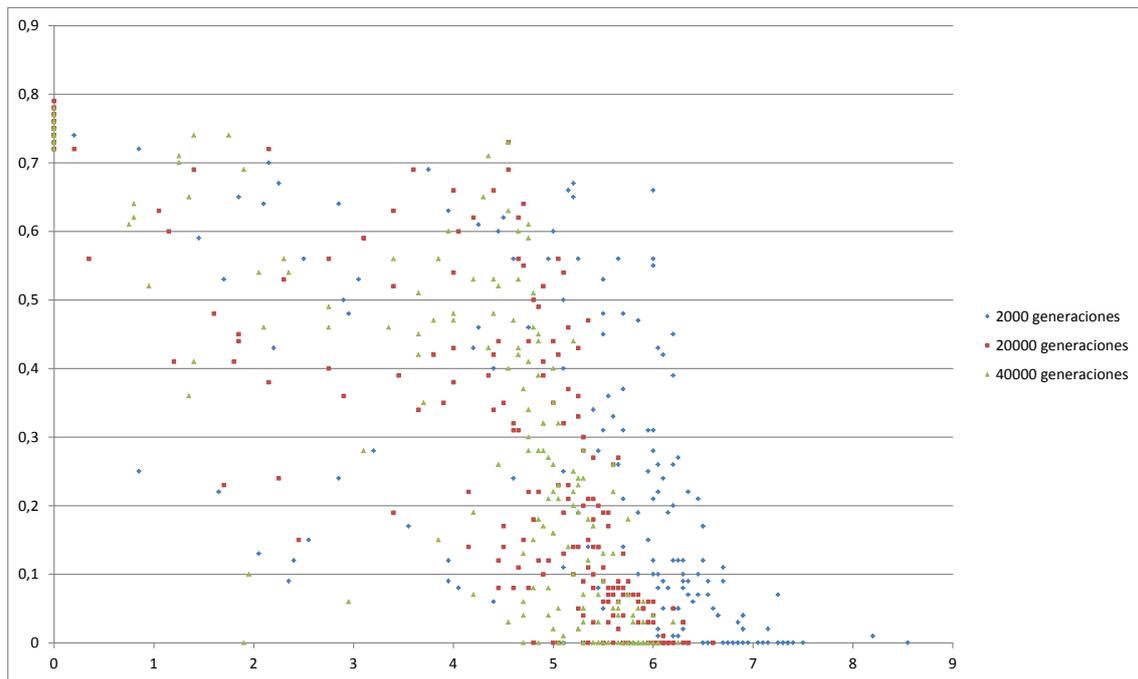


Figura 59. Total de soluciones para población 50 (Experimento “Plaga” con 1000 simulaciones)

La información que se puede extraer de la figura anterior es, como se indica en las anteriores gráficas en las que se agrupa por generación, que existe una evolución al aumentar el número de generaciones y que se ha reducido la variabilidad al aumentar el número de simulaciones.

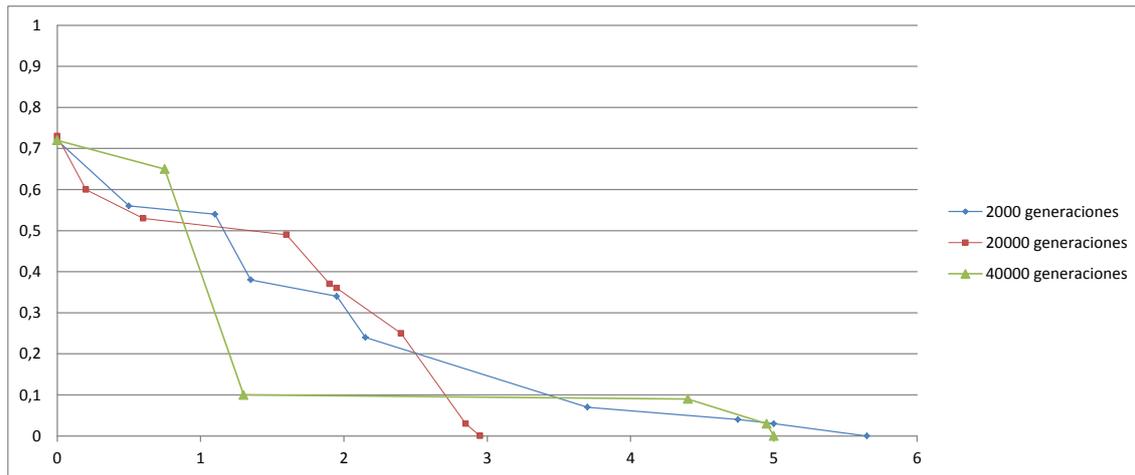


Figura 60. Comparativa de soluciones “No Dominadas” para población 100 (Experimento “Plaga” con 1000 simulaciones)

Las conclusiones para este gráfico son las mismas que las explicadas para 40.000 generaciones con la agrupación por generaciones. Se puede observar que existe una solución con probabilidad de presencia baja y con un coste razonablemente bajo 40.000 generaciones. Este punto es el único que mejora las ejecuciones con 2.000 y 20.000 generaciones. El coste menor con probabilidad de presencia baja lo encuentra la ejecución con 20.000 generaciones.

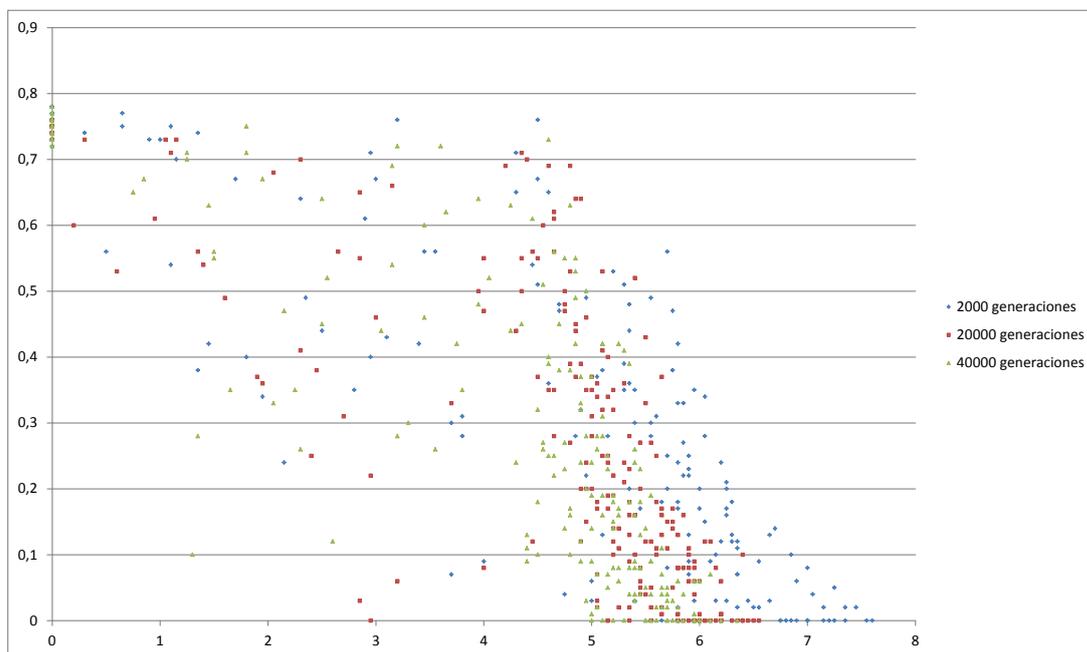


Figura 61. Total de soluciones para población 100 (Experimento “Plaga” con 1000 simulaciones)

En general, para este conjunto de pruebas con 1.000 simulaciones se puede observar que existe una mejor dispersión en las soluciones, lo que provoca que se evite que por azar, unas soluciones con menos generaciones obtengan mejores resultados. No obstante, aunque se hayan obtenido resultados más positivos con 1.000 simulaciones, todavía se produce cierta dispersión que requiere estudio.

5.4 Prueba de concepto II: Inundación

En este experimento ficticio se pretende extender una población de lince ibérico que habita una zona relativamente cercana a una zona habitada. Esta población, está en peligro de extinción y, debido a la aprobación de la construcción de un pantano destinado para consumo de agua humano, es necesaria la reubicación de esta población asegurando su existencia al ser una especie que es necesario conservar.

En este problema, debido a la topología del terreno y dispersión de los individuos que forman parte de la población se encuentra inviable reunir toda la población y transportarla al lugar de destino, en vez de ello, se pretende aprovechar los fondos destinados para la conservación de fauna y flora para que la población se extienda a esa zona con ayuda humana (por ejemplo, añadiendo una especie en el entorno de la que se pueda alimentar el lince)

Los factores relevantes que se consideran para este experimento (haciendo labores de experto en poblaciones de linces ibéricos), son la comida y el agua.

Se considera a efectos del experimentos, que la intervención “soltar conejos” en una zona (alimento del lince), tendrá un coste asociado que no requerirá de un coste futuro, además, se parte de la base que no decrece el alimento una vez que éste sea proporcionado.

Desde la constructora del pantano, nos ofrecen un plazo de tiempo que modelizaremos como 10 instantes e indican que un retraso en la expansión de la población hará inviable dicha expansión porque la masa de agua inundará las vías naturales para ir desde la zona en la que se encuentran a la zona final. Por tanto, para que la población sobreviva en la casilla de destino, será necesario proveer de alimento en algún punto intermedio para facilitar el desplazamiento antes que la crecida del pantano les afecte.

5.4.1 Módulo de simulación: cambio del terreno

Para esta experimentación se suministran los valores de los factores relevantes en el instante de tiempo inicial con el que deberá empezar a trabajar la aplicación:

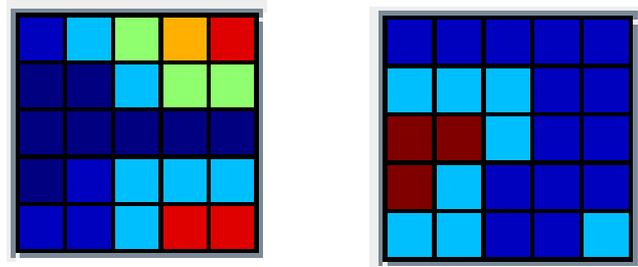


Figura 62. Variables iniciales: comida (izquierda) y agua (derecha)

Del mismo modo, en la siguiente figura se presentan los valores de esos mismos factores relevantes pasados **diez periodos de tiempo**. A efectos de este experimento, los valores de los factores relevantes intermedios no han sido informados, por lo que se calculan en la aplicación a través de interpolación lineal, pero si se contase con modelos del cambio de terreno más sofisticados, se podrían incorporar sin ninguna complicación.

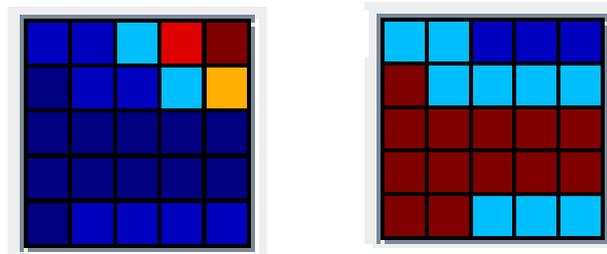


Figura 63. Variables finales: comida (izquierda) y agua (derecha)

A continuación se presenta la simulación completa para los factores relevantes de este problema: comida y agua.

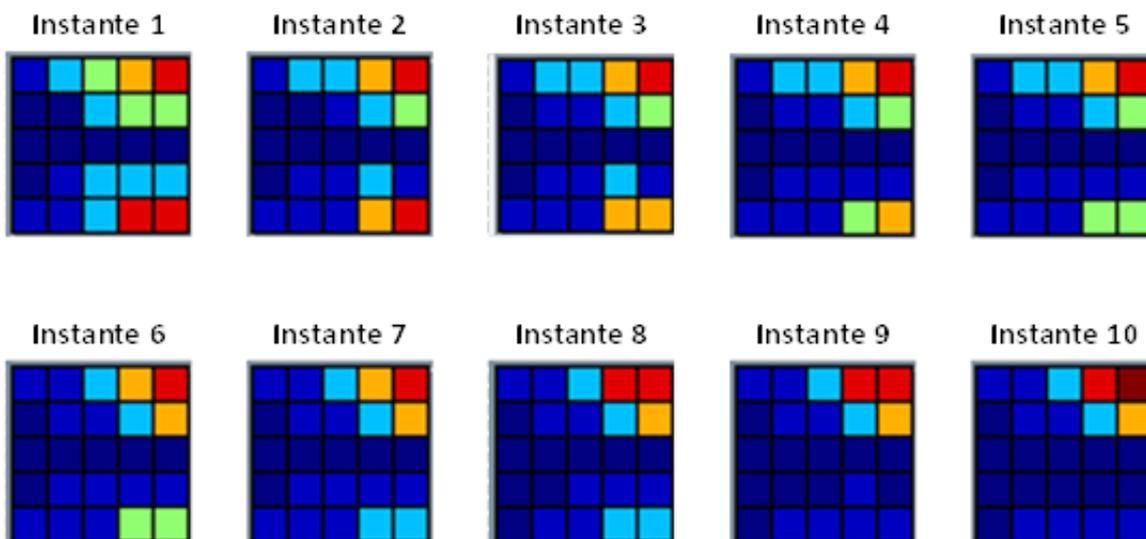


Figura 64. Simulación completa de la variable "comida"

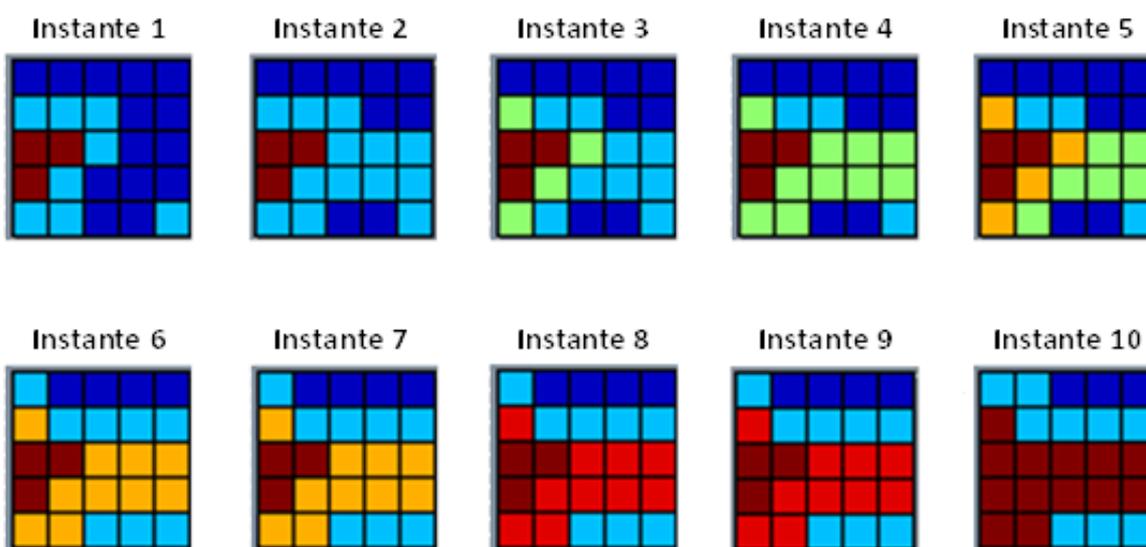


Figura 65. Simulación completa de la variable "agua"

5.4.2 Módulo de simulación: expansión de la población

Para este experimento se ha partido de los datos de presencia que se aportan a continuación:

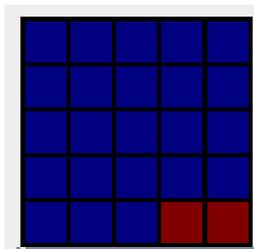


Figura 66. Distribución inicial de la población

La simulación de los cambios en la distribución de la población se ha realizado mediante un autómata celular estocástico definido en el Alcance del Sistema (**3.1.2 Modelado del desplazamiento**). Éste tiene como elemento central el atractivo del terreno, que marca la probabilidad de que la población se extienda hacia celdas adyacentes o incluso mantengan su presencia o no en la celda de partida.

El modelo de atractivo es una función especial que prioriza la presencia de agua, ésta permite que pueda existir vida con algunos niveles de agua, pero si no hay agua o hay demasiada (inundación), la especie no podrá sobrevivir. Esta fórmula se ha ajustado de la siguiente forma:

$$\text{Atractivo} = \begin{cases} \text{Comida} & \Leftrightarrow 0.6 \geq \text{agua} \geq 0.1 \\ 0 & \Leftrightarrow \text{En otro caso} \end{cases}$$

Figura 67. Función utilizada para calcular el Atractivo del experimento “Inundación”

A continuación se muestra el resultado del atractivo a lo largo de los distintos instantes, aplicando en el modelo de atractivo los valores de los factores relevantes descritos anteriormente:

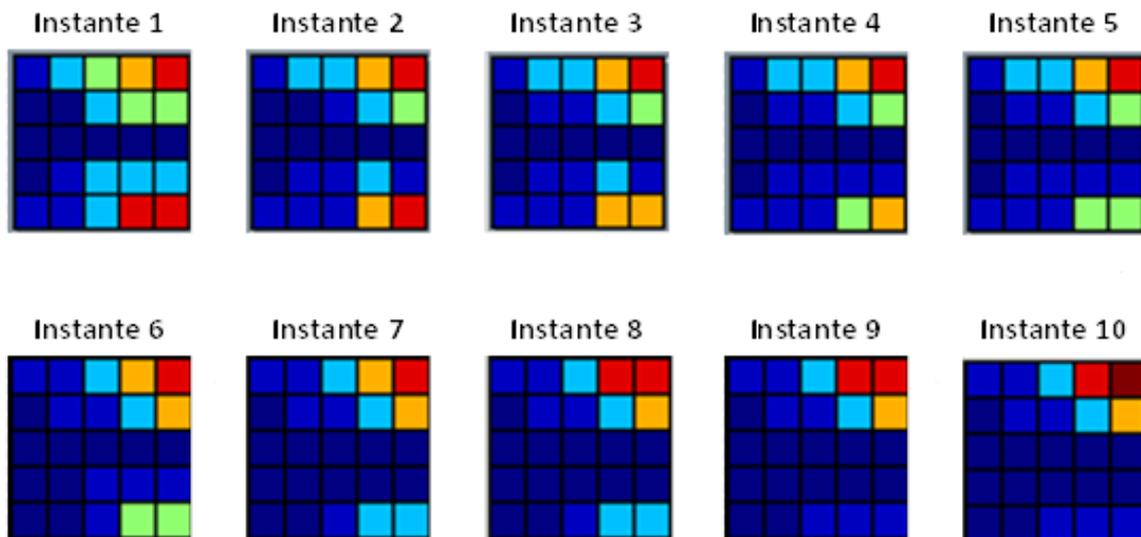


Figura 68. Atractivo simulado

Como se puede observar, hay dos zonas con un atractivo alto en los primeros instantes, en el norte y en el sur (en ésta última está la población en el primer instante). En el último instante prácticamente sólo hay celdas con atractivo alto en la zona norte del escenario (coordenadas [5:5]). Se puede comprobar que existe una franja que no tiene un gran atractivo para la población en los primeros instantes, tiempo después la franja es mayor, haciendo imposible que la población pueda extenderse hacia el objetivo debido a la expansión de la masa de agua (y que las intervenciones que proponemos solo afectan al factor relevante comida).

Dada la naturaleza estocástica de la propuesta de dispersión y de la propia simulación, se realizan **100 evaluaciones**, el resultado es, por tanto, una media de todas las ejecuciones. Se muestra la distribución de población en los distintos instantes:

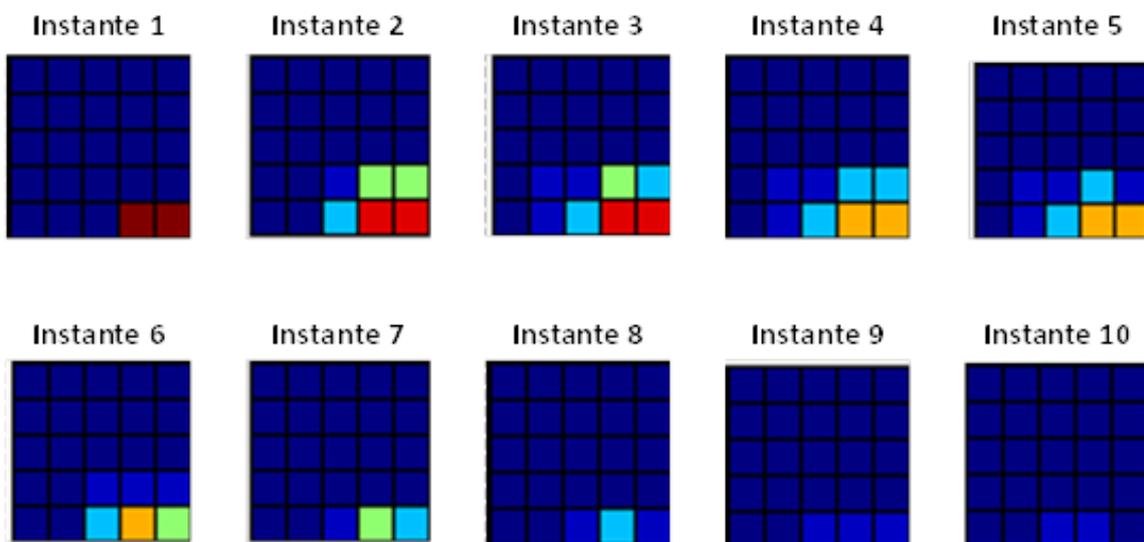


Figura 69. Simulación de población del experimento "Inundación"

5.4.3 Generador de escenarios

Para este experimento se ha decidido utilizar intervenciones en el primer factor relevante (comida), estas intervenciones se ha considerado que tenga la característica de mantenerse en el tiempo. Una intervención propuesta podría ser la siguiente:

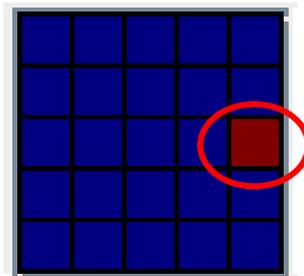


Figura 70. Cambio en valor del factor relevante "Comida"

Por tanto, los valores de los factores relevantes quedarían de la siguiente forma (comparar con las variables iniciales).

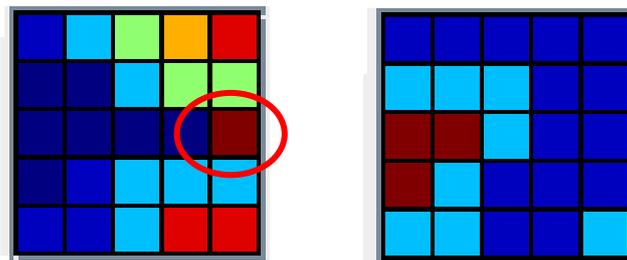


Figura 71. Cambio en la variable Comida (izquierda), la variable Agua (derecha) sigue igual

Como la intervención se mantiene en el tiempo, el instante final variará para Comida. A continuación se muestran los escenarios del último instante.

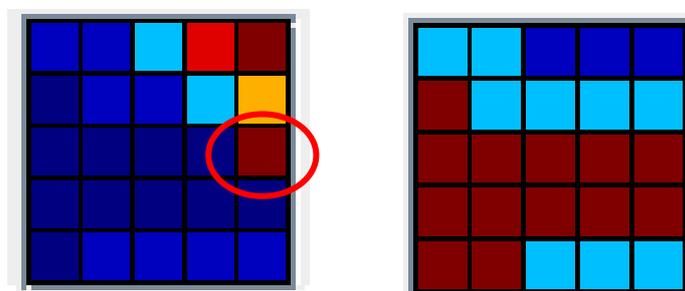


Figura 72. Variables finales: comida (izquierda) modificada por la intervención, agua (derecha) sigue igual

Por tanto, el atractivo de los terrenos utilizando nuestro modelo de atractivo y los nuevos valores resultantes de la intervención, es el que se muestra a continuación:

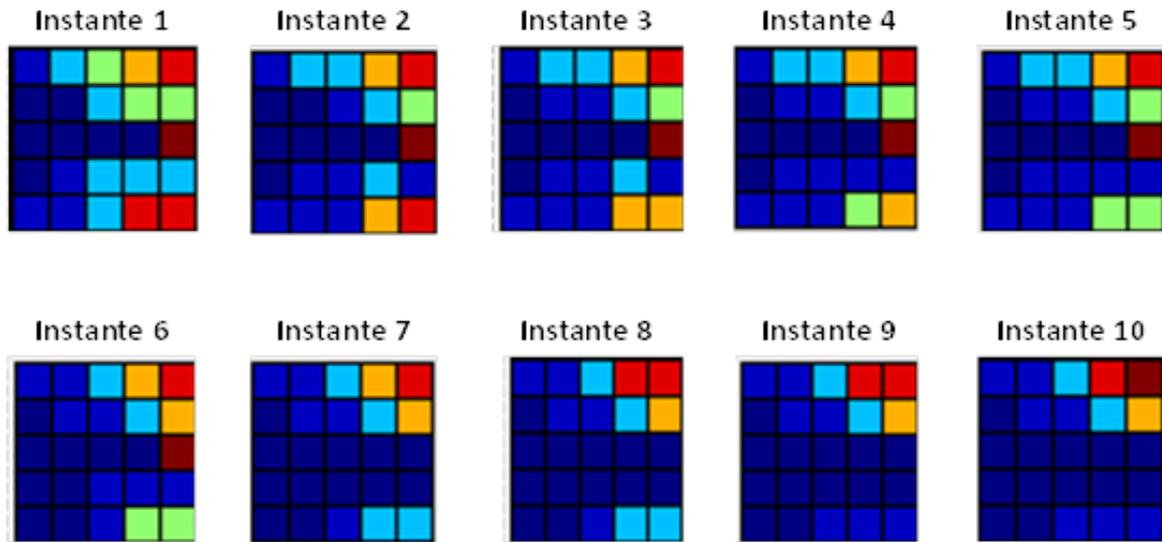


Figura 73. Atractivo simulado después de aplicar la intervención

Como se puede apreciar, hasta el instante 7 la zona intervenida es muy atractiva debido a la gran presencia de agua y comida, no obstante en ese instante la zona se inunda haciéndola poco atractiva.

Gracias a esta intervención, en la que subir una casilla de valor 0% a 100%, obtenemos la siguiente distribución de población:

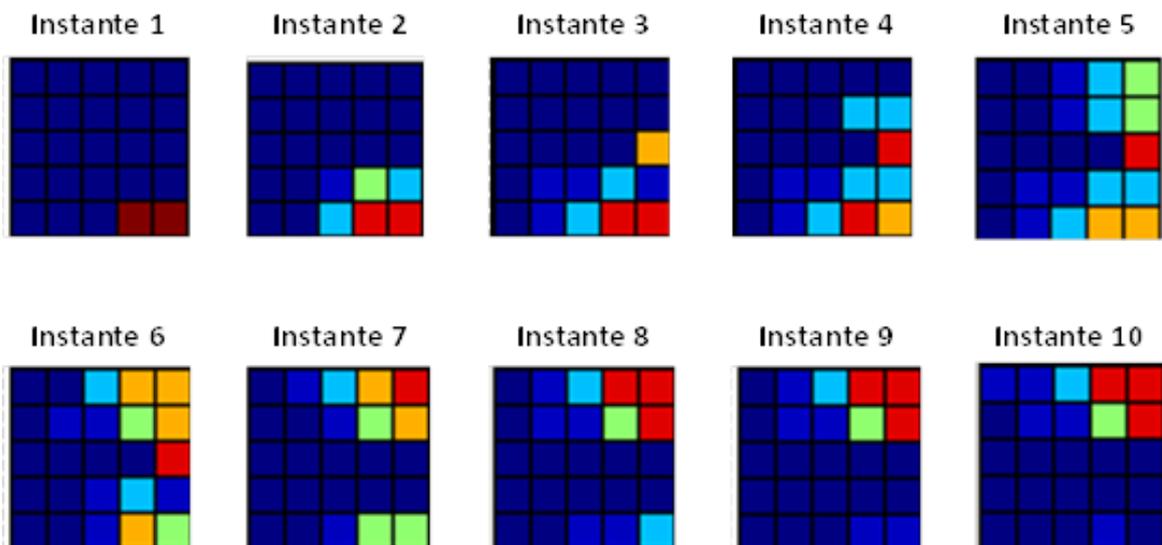


Figura 74. Simulación de la población después de la intervención

El coste asociado a la intervención es la diferencia, en valor absoluto, entre el valor anterior a la intervención y el introducido en la intervención y dividido por 100, por lo que el coste de la intervención es 1.

En esta nueva distribución de la población se puede ver como ésta ha logrado pasar a la mitad superior del tablero y ha conseguido asentarse allí con una probabilidad de presencia de un 82%.

5.4.4 Optimización

La optimización automática se produce en base al tablero inicial visto anteriormente y las fórmulas de cálculo de atractivo exclusivo de este problema, además se utilizan los siguientes parámetros configurables:

Se debe tener en cuenta que para esta prueba de concepto, se intentará maximizar el objetivo presencia, de modo que la optimización será buscar el mínimo coste con la mayor probabilidad de presencia de población posible.

5.4.4.1 Resultados con 100 simulaciones

5.4.4.1.1 Resultados globales

Al igual que en el caso de la plaga, comenzamos aportando los resultados obtenidos en términos de spread e hipervolumen.

	Media	Mediana	Var.	Max	Min
SP_10_2000	0.713	0.711	0.034	1.119	0.300
SP_10_20000	0.785	0.772	0.037	1.153	0.282
SP_10_40000	0.801	0.775	0.026	1.156	0.473
SP_50_2000	0.763	0.745	0.031	1.186	0.402
SP_50_20000	0.833	0.804	0.022	1.170	0.515
SP_50_40000	0.848	0.908	0.058	1.191	0.254
SP_100_2000	0.717	0.750	0.042	1.173	0.342
SP_100_20000	0.809	0.791	0.016	1.041	0.591
SP_100_40000	0.839	0.850	0.022	1.118	0.500

Tabla 85: métrica SP (Experimento "Inundación" con 100 simulaciones)

Tal y como se aprecia en la tabla anterior, hay una relación positiva entre el spread y el mínimo de generaciones.

	SP_10_2000	SP_10_20000	SP_10_40000	SP_50_2000	SP_50_20000	SP_50_40000	SP_100_2000	SP_100_20000
SP_10_20000	=							
SP_10_40000	+	=						
SP_50_2000	=	=	=					
SP_50_20000	++	=	=	=				
SP_50_40000	++	+	+	+	=			
SP_100_2000	=	=	-	=	--	--		
SP_100_20000	+	=	=	=	=	-	+	
SP_100_40000	++	=	=	+	=	=	++	=

Tabla 86: métrica SP (Experimento "Inundación" con 100 simulaciones)

La tabla anterior pone de manifiesto que las diferencias entre la mayor parte de combinaciones de tamaño de población y generaciones no es lo suficientemente grande para descartar la hipótesis de que sean iguales a los niveles propuestos.

A continuación se muestran los resultados de la métrica Hipervolumen:

	Media	Mediana	Var.	Max	Min
HP_10_2000	0.537	0.548	0.023	0.848	0.293
HP_10_20000	0.548	0.563	0.009	0.703	0.382
HP_10_40000	0.530	0.532	0.009	0.791	0.382
HP_50_2000	0.499	0.490	0.021	0.868	0.264
HP_50_20000	0.509	0.471	0.015	0.806	0.365
HP_50_40000	0.534	0.507	0.012	0.754	0.397
HP_100_2000	0.508	0.501	0.026	0.791	0.283
HP_100_20000	0.501	0.482	0.011	0.826	0.377
HP_100_40000	0.578	0.591	0.016	0.779	0.387

Tabla 87: métrica HP (Experimento "Inundación" con 100 simulaciones)

Los hipervolumenes son extremadamente parecidos. Sólo hay un caso en el que el hipervolumen parece superior al resto con relativa claridad, y es el correspondiente a una población de tamaño 100 y 40.000 generaciones. Esto mismo se aprecia en la tabla dedicada a las cantidades donde, salvo por este caso, apenas hay una diferencia significativa.

	HP_10_2000	HP_10_20000	HP_10_40000	HP_50_2000	HP_50_20000	HP_50_40000	HP_100_2000	HP_100_20000
HP_10_20000	=							
HP_10_40000	=	=						
HP_50_2000	=	=	=					
HP_50_20000	=	=	=	=				
HP_50_40000	=	=	=	=	=			
HP_100_2000	=	=	=	=	=	=		
HP_100_20000	=	-	=	=	=	=	=	
HP_100_40000	=	=	=	+	+	=	+	++

Tabla 88: métrica HP (Experimento "Inundación" con 100 simulaciones)

El frente que combina las mejores soluciones de toda la experimentación con 100 simulaciones por evaluación se muestra en la siguiente figura:

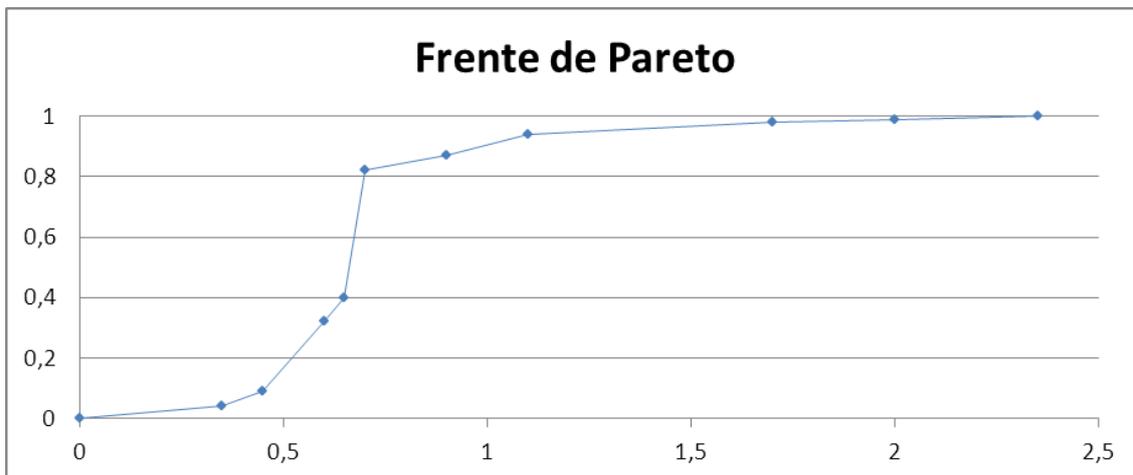


Figura 75. Soluciones “No Dominadas” (frente de Pareto) (Experimento “Inundación” con 100 simulaciones)

5.4.4.1.2 Resultados agrupados por generaciones

A continuación se mostrará la comparativa entre las distintas poblaciones para 2.000 generaciones:

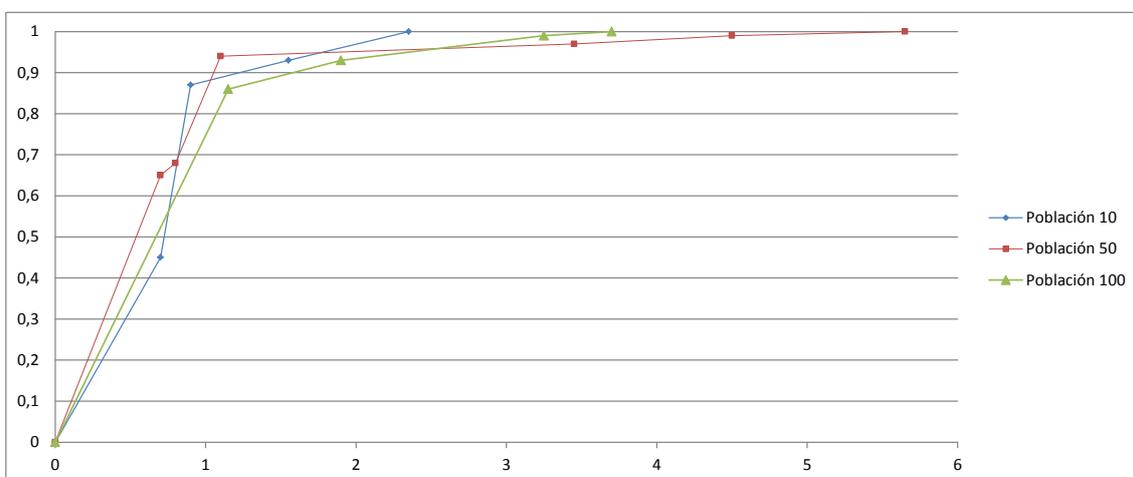


Figura 76. Comparativa de soluciones “No Dominadas” para 2.000 generaciones (Experimento “Inundación” con 100 simulaciones)

En el anterior gráfico se puede apreciar que para todas las experimentaciones de las distintas poblaciones se obtiene un conjunto de soluciones “No Dominadas” similar en número. En este caso, también se puede concluir que con población igual a 50, se tiende a obtener (excepto por un punto de población 10) mejores resultados que con 10 y 100 de población para probabilidades de presencia de población menores que 95%.

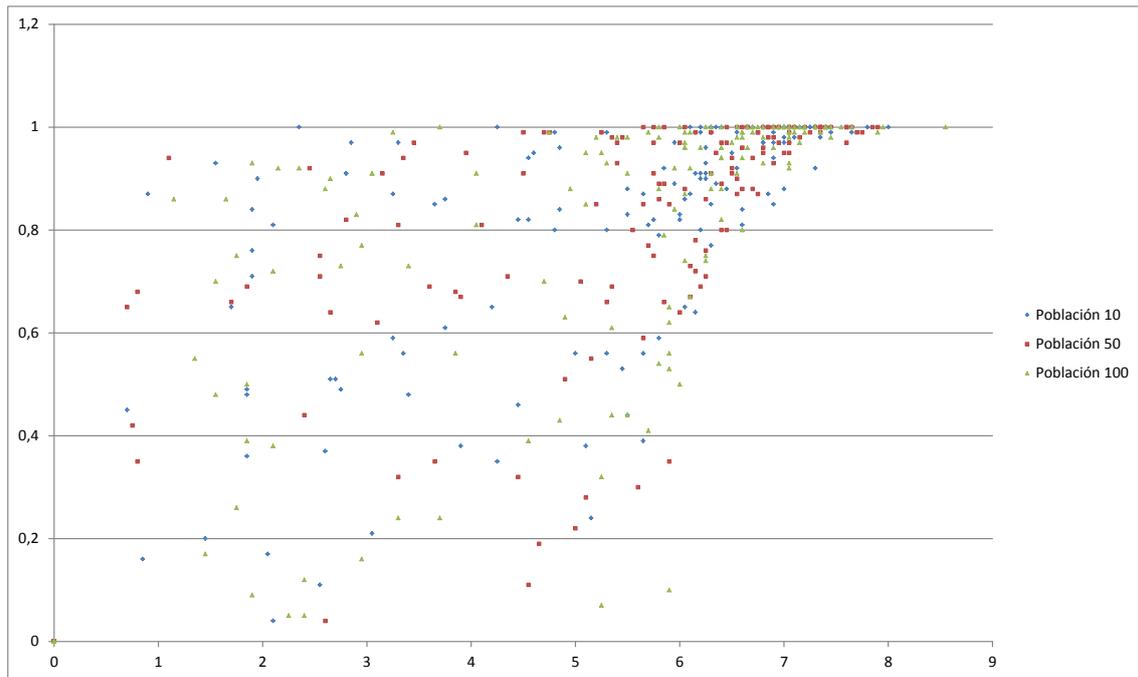


Figura 77. Total de soluciones para 2.000 generaciones (Experimento “Inundación” con 100 simulaciones)

Se presenta a continuación la comparativa con 20.000 generaciones:

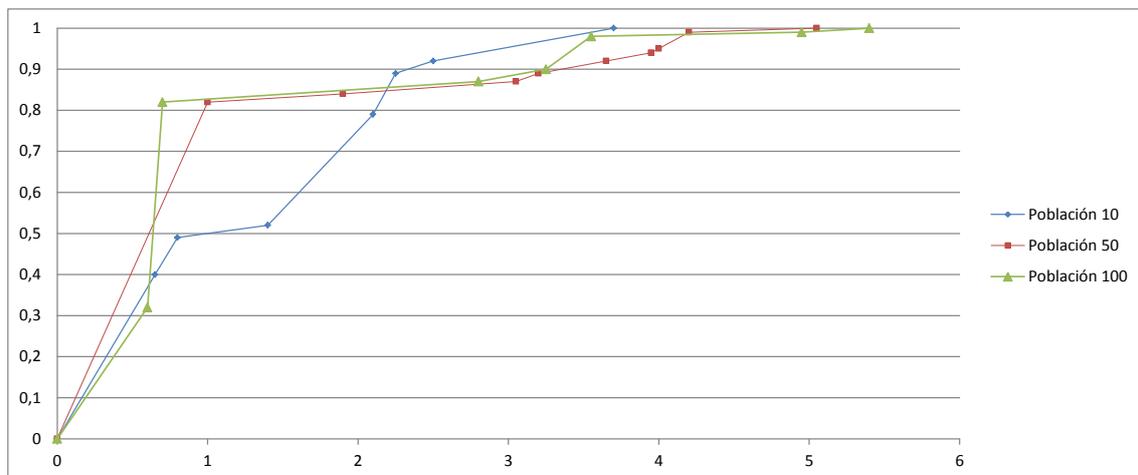


Figura 78. Comparativa de soluciones “No Dominadas” para 20.000 generaciones (Experimento “Inundación” con 100 simulaciones)

En el gráfico anterior se puede observar que para 20.000 generaciones, las soluciones con probabilidad de presencia de población inferior al 80% son mejores si en la experimentación se utiliza un tamaño de población de 50 o 100, ofreciendo este último mejores soluciones. Para las soluciones con probabilidad de población cercanas al 100% se obtiene que las soluciones con menor coste son producidas utilizando 10 individuos como tamaño de población en cada generación.

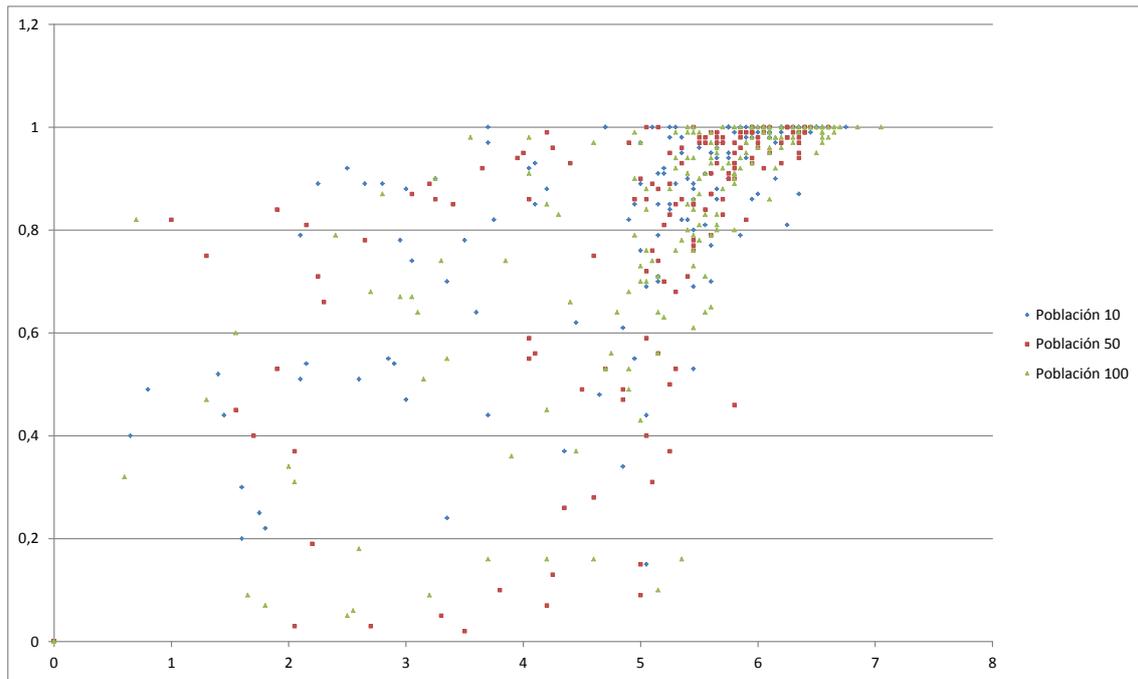


Figura 79. Total de soluciones para 20.000 generaciones (Experimento "Inundación" con 100 simulaciones)

En la figura anterior, no se aprecia ningún tipo de evolución al modificar el tamaño de la población.

Por último, se muestra la comparativa para 40.000 generaciones:

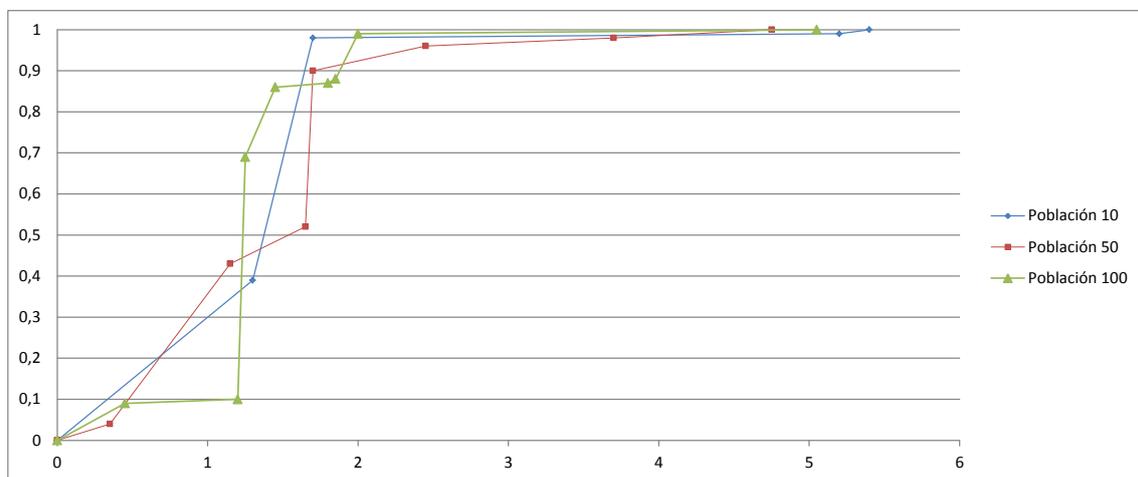


Figura 80. Comparativa de soluciones "No Dominadas" para 40.000 generaciones (Experimento "Inundación" con 100 simulaciones)

En la figura anterior se puede encontrar un gran número de soluciones para todas las poblaciones. Cabe destacar que no hay ninguna población con un dominio claro, no obstante cada tipo de población tiene un protagonismo para una probabilidad de presencia determinada.

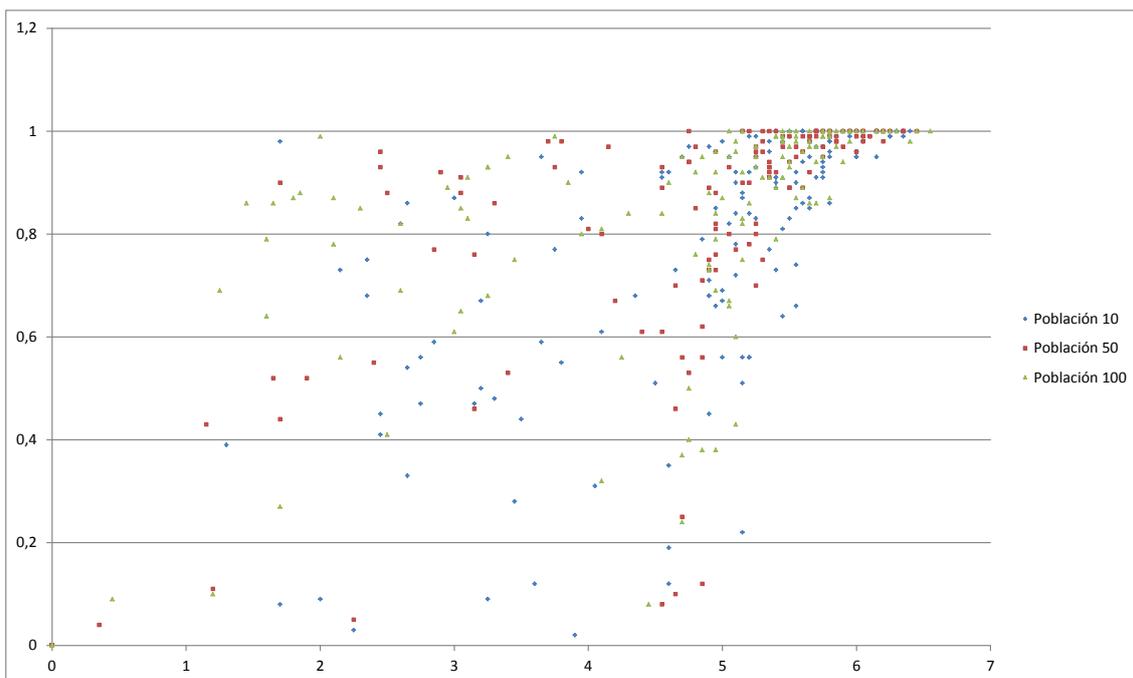


Figura 81. Total de soluciones para 40.000 generaciones (Experimento “Inundación” con 100 simulaciones)

Estudiando esta dispersión, se establece como una de las posibles causas de los resultados la variabilidad que introduce el automáta celular estocástico. Al simular dos veces el mismo escenario con una misma entrada, se obtienen dos salidas distintas (en una casilla determinada hay presencia de población y en otra no). Por esta razón se realiza la misma experimentación con 1.000 simulaciones.

5.4.4.1.3 Resultados agrupados por tamaño de población

A continuación se estudian los resultados para los distintos tamaños de población en función de las generaciones que estuvo evolucionando el algoritmo genético. La siguiente figura muestra los resultados para la población de tamaño 10.

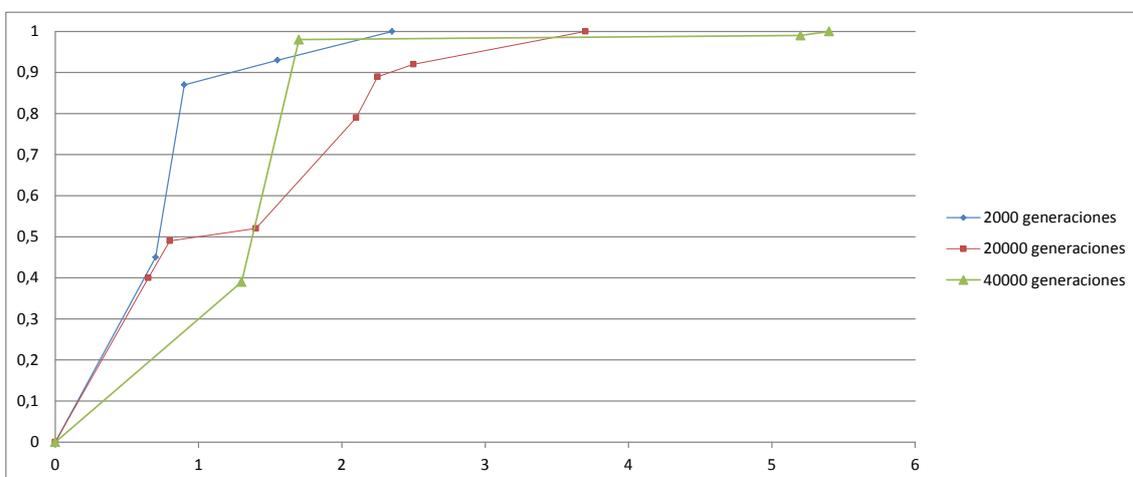


Figura 82. Comparativa de soluciones “No Dominadas” para población igual a 10 (Experimento “Inundación” con 100 simulaciones)

Como se puede apreciar en el anterior gráfico, los resultados obtenidos no corresponden a un resultado esperable, que habría arrojado soluciones mejores aumentando el número de generaciones. No obstante, aunque las soluciones con 2.000 generaciones dominan al resto, se puede encontrar una solución cercana con probabilidad de presencia de población cercana al 100% con coste 1,5 con 40.000 generaciones y población de 10. Las peores soluciones son el resultado de la experimentación con 20.000 generaciones, excepto en el punto que se alcanza un porcentaje de población del 40% con coste 0,7, mientras que con 40.000 generaciones se obtiene ese mismo porcentaje con coste 1,3.

En vista de los resultados se ha estimado interesante estudiar todas las soluciones .

Se estudia cada solución independientemente, en lugar de centrar la atención en las no dominadas. En el siguiente gráfico se muestra esta comparativa con población igual a 10 y las distintas generaciones de nuestro experimento:

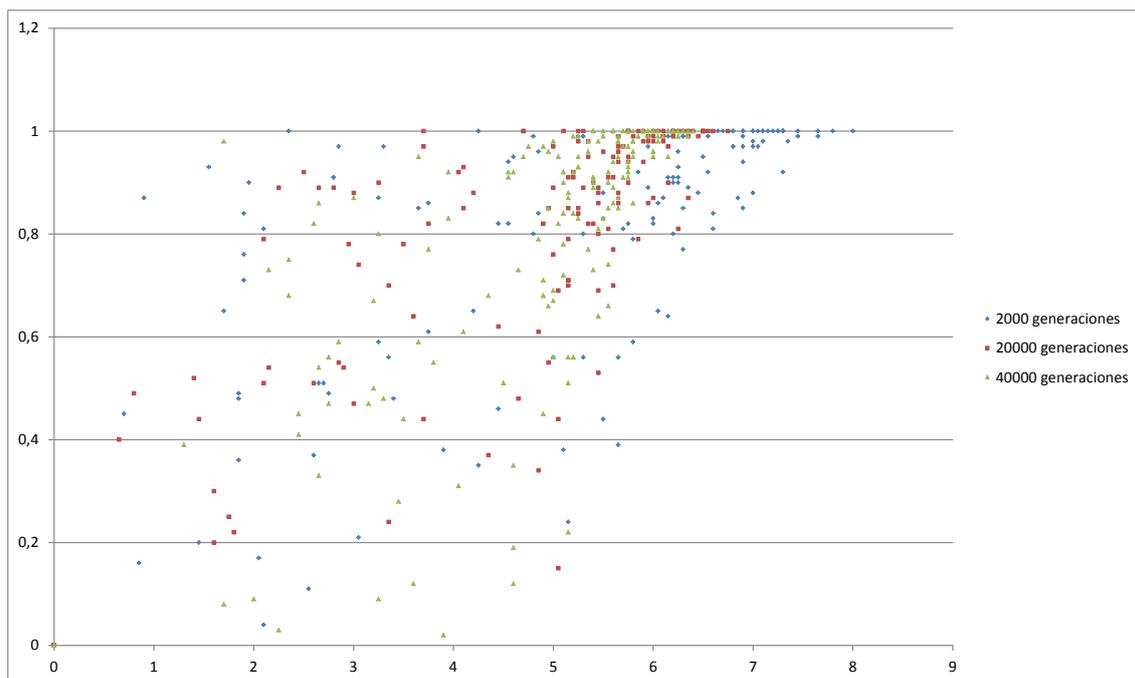


Figura 83. Total de soluciones para población 10 (Experimento “Inundación” con 100 simulaciones)

En el gráfico anterior se puede observar que se obtiene un gran número de soluciones con coste 7 (o mayor) entre las soluciones con 2.000 generaciones, mientras que con 20.000 y 40.000 generaciones generalmente se encuentran soluciones con menor coste.

También se puede observar una gran dispersión de las soluciones conforme nos alejamos de una probabilidad de presencia de 100%. Aunque generalmente se

obtingan mejores soluciones con experimentaciones con muchas generaciones, en ocasiones, por azar, se han obtenido mejores soluciones con una cantidad de generaciones baja, lo que provoca que en los frentes aparezcan como mejores soluciones algunas con generaciones bajar (2.000).

A continuación, se muestra a continuación la comparativa con población 50:

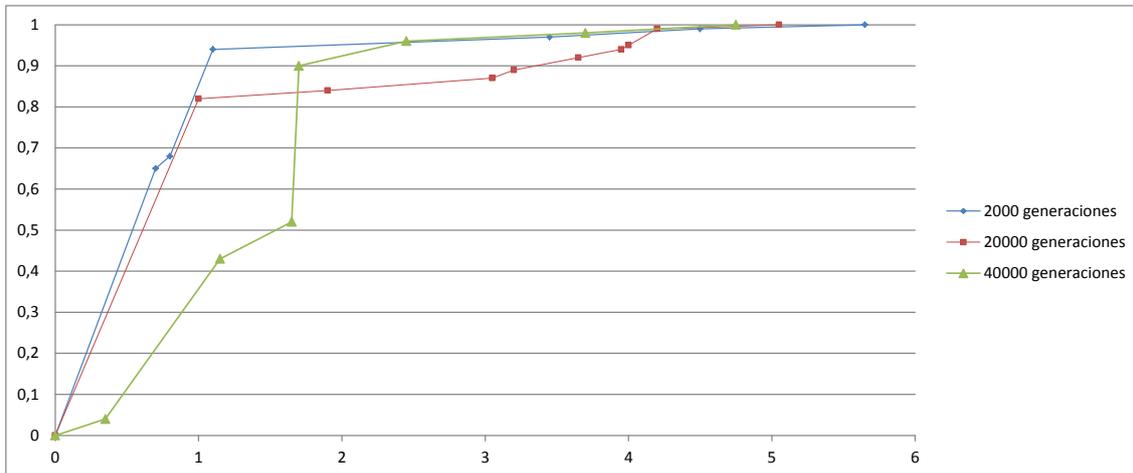


Figura 84. Comparativa de soluciones “No Dominadas” para población igual a 50 (Experimento “Inundación” con 100 simulaciones)

En la figura anterior se observa que con 2.000 generaciones se encuentra la peor solución con una probabilidad de presencia de población del 100%, todas las demás soluciones con 2.000 generaciones dominan a la experimentación con 20.000 o 40.000 generaciones.

En esta comparativa, también se puede apreciar que se encuentran más cantidad de soluciones no dominadas con la experimentación de 20.000 y 40.000 generaciones, aunque generalmente éstas sean peores se puede concluir que la exploración de soluciones es mayor cuantas más generaciones se realicen.

Por último, se vuelve a encontrar que generalmente, con 2.000 generaciones se obtienen mejores soluciones que con 40.000, las peores soluciones son arrojadas por la experimentación con 20.000 generaciones (excepto con probabilidad de presencia muy cercana al 100%).

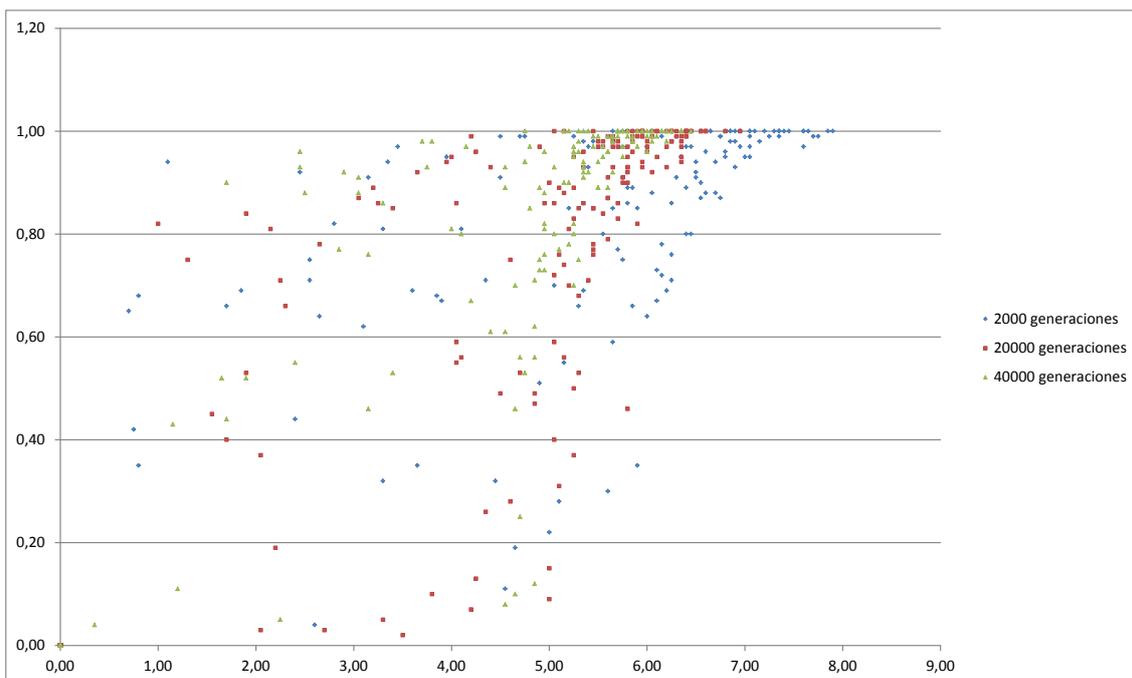


Figura 85. Total de soluciones para población 50 (Experimento “Inundación” con 100 simulaciones)

Las conclusiones para la anterior figura son las mismas que para población 10, existe una evolución de las soluciones conforme aumentan las generaciones, aunque esto no es tan evidente al pasar de 20.000 a 40.000 generaciones.

A continuación se muestra la comparativa con población igual a 100:

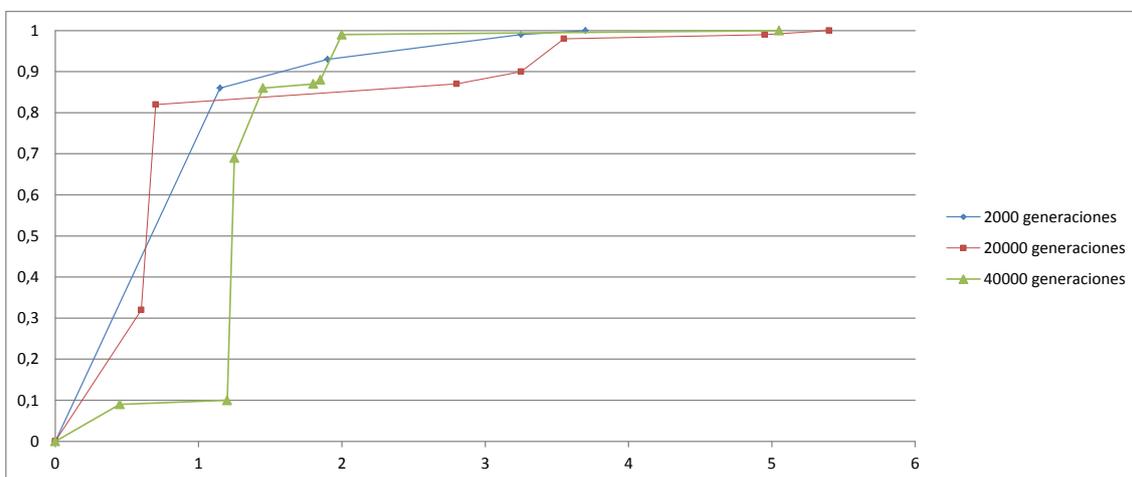


Figura 86. Comparativa de soluciones “No Dominadas” para población igual a 100 (Experimento “Inundación” con 100 simulaciones)

En la figura anterior se puede destacar que cada solución aporta una solución que domina al resto para probabilidades de presencia distintas. En el caso de 2.000 generaciones dominan con una solución con probabilidades de presencia cercano al 90%, con 40.000 generaciones domina las solución cercana al 100% de probabilidad de presencia, en el resto domina las soluciones con 2.000 generaciones.

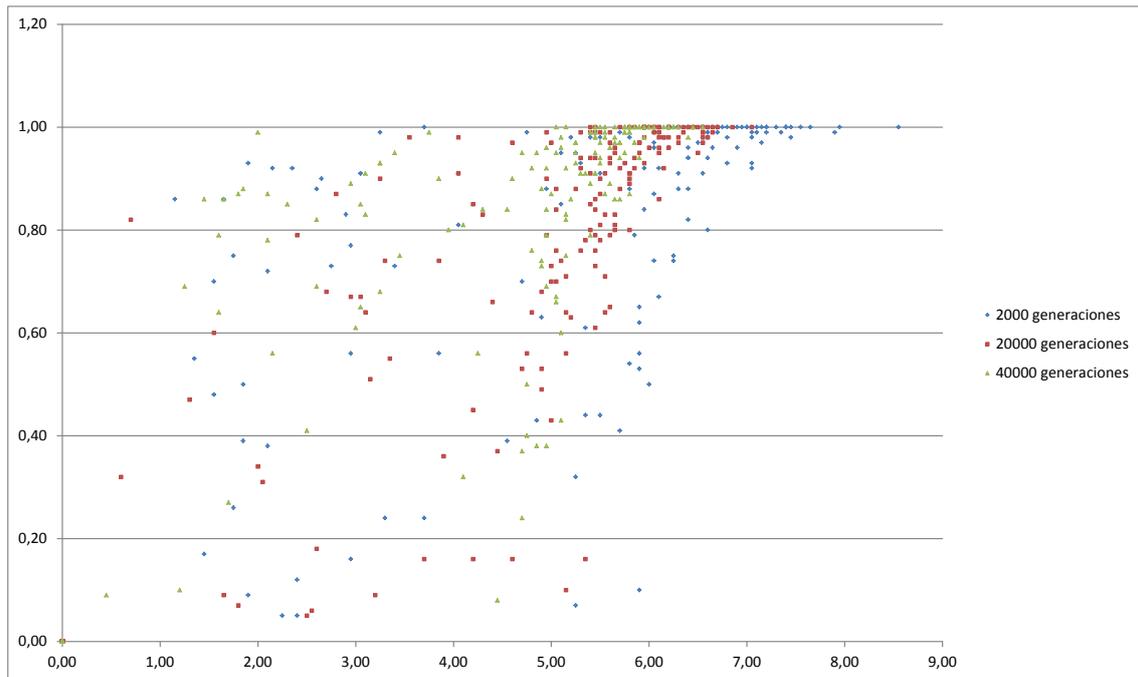


Figura 87. Total de soluciones para población 100 (Experimento “Inundación” con 100 simulaciones)

A la vista de los resultados, se establece un nuevo conjunto de experimentos con 1.000 simulaciones para determinar las causas de la variabilidad de soluciones ante una misma entrada y la razón por la que, aun ocurriendo una evolución en las soluciones, aparecen soluciones mejores con un número de generaciones bajo.

5.4.4.2 Resultados con 1000 simulaciones

5.4.4.2.1 Resultados globales

Se obtienen los siguientes resultados al utilizar este Frente de Pareto en las distintas métricas que se están utilizando para este proyecto:

	Media	Mediana	Var.	Max	Min
SP_10_2000	0.786	0.779	0.037	1.155	0.437
SP_10_20000	0.828	0.815	0.026	1.077	0.384
SP_10_40000	0.818	0.780	0.027	1.179	0.557
SP_50_2000	0.764	0.735	0.028	1.170	0.502
SP_50_20000	0.838	0.818	0.020	1.213	0.575
SP_50_40000	0.873	0.903	0.022	1.253	0.587
SP_100_2000	0.819	0.823	0.025	1.166	0.456
SP_100_20000	0.786	0.781	0.016	1.017	0.540
SP_100_40000	0.834	0.865	0.023	1.131	0.544

Tabla 89: métrica SP (Experimento “Inundación” con 1000 simulaciones)

Tal y como se aprecia en la tabla anterior, hay una relación positiva entre el spread y el mínimo de generaciones. Esta relación no ocurre para tamaño de población 100 y las soluciones de 20.000 generaciones y 2.000 generaciones.

	SP_10_2000	SP_10_20000	SP_10_40000	SP_50_2000	SP_50_20000	SP_50_40000	SP_100_2000	SP_100_20000
SP_10_20000	=							
SP_10_40000	=	=						
SP_50_2000	=	=	=					
SP_50_20000	=	=	=	+				
SP_50_40000	+	=	=	++	=			
SP_100_2000	=	=	=	=	=	=		
SP_100_20000	=	=	=	=	=	--	=	
SP_100_40000	=	=	=	+	=	=	=	=

Tabla 90: métrica SP (Experimento "Inundación" con 1000 simulaciones)

Prácticamente ningún resultado tiene diferencias significativas con otros excepto el conjunto de pruebas con población 50 y 2.000 generaciones que tiene un nivel de confianza con los demás conjuntos de pruebas con población 50, y con población 100 y 40.000 generaciones. El contraste indica, para los tamaño de población 50, una diferencia significativa al 1% entre los valores con 20.000 y 2.000 generaciones y una diferencia significativa al 5% entre los valores con 40.000 y 2.000 generaciones.

A continuación se muestran los resultados de la métrica Hipervolumen:

	Media	Mediana	Var.	Max	Min
HP_10_2000	0.478	0.451	0.016	0.749	0.313
HP_10_20000	0.563	0.581	0.014	0.833	0.356
HP_10_40000	0.532	0.507	0.009	0.754	0.408
HP_50_2000	0.474	0.458	0.015	0.667	0.289
HP_50_20000	0.530	0.487	0.013	0.793	0.388
HP_50_40000	0.561	0.552	0.013	0.824	0.383
HP_100_2000	0.475	0.491	0.013	0.714	0.279
HP_100_20000	0.536	0.519	0.011	0.767	0.36
HP_100_40000	0.544	0.526	0.011	0.801	0.386

Tabla 91: métrica HP (Experimento "Inundación" con 1000 simulaciones)

Observando la tabla anterior, podemos ver que los resultados mejoran al aumentar las iteraciones, puesto que los valores de hipervolumen indican que las soluciones abarcan un área mayor.

	HP_10_2000	HP_10_20000	HP_10_40000	HP_50_2000	HP_50_20000	HP_50_40000	HP_100_2000	HP_100_20000
HP_10_20000	++							
HP_10_40000	+	=						
HP_50_2000	=	--	-					
HP_50_20000	+	=	=	+				
HP_50_40000	++	=	=	++	=			
HP_100_2000	=	--	-	=	=	--		
HP_100_20000	+	=	=	+	=	=	+	
HP_100_40000	+	=	=	+	=	=	++	=

Tabla 92: métrica HP (Experimento "Inundación" con 1000 simulaciones)

El contraste indica una diferencia significativa al 5% entre los valores con 20.000 y 2.000 generaciones y los de 40.000 generaciones y 2.000 generaciones. El área que abarca las soluciones con 40.000 y 20.000 generaciones es mayor y por tanto es mejor, que las de 2.000 generaciones. El contraste indica que la diferencia entre 20.000 y 40.000 generaciones no son significativas.

A continuación se muestra el Frente de Pareto para el conjunto experimental con 1.000 simulaciones:

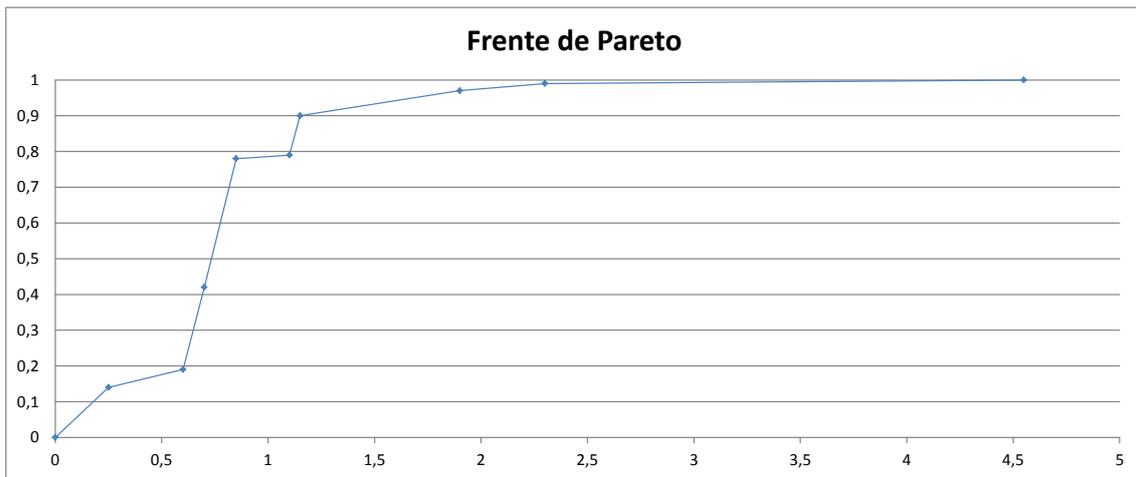


Figura 88. Soluciones "No Dominadas" (frente de Pareto) (Experimento "Inundación" con 1000 simulaciones)

Al comparar la figura anterior con el frente de mejores soluciones obtenido con 100 simulaciones permite observar que se han obtenido peores resultados con 1.000 simulaciones.

5.4.4.2.2 Resultados agrupados por generaciones

Se comparan los resultados agrupados por generaciones. Se comienza por 2.000 generaciones:

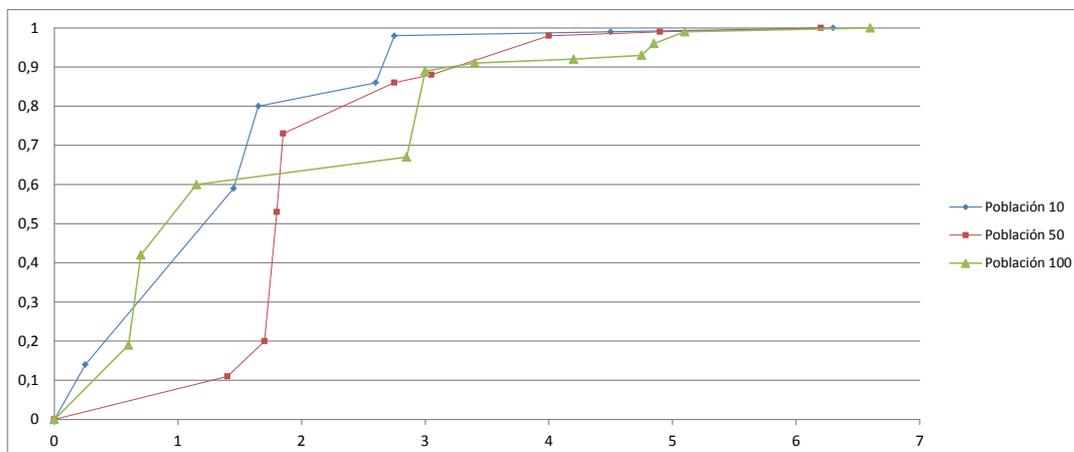


Figura 89. Comparativa de soluciones “No Dominadas” para 2.000 generaciones (Experimento “Inundación” con 1000 simulaciones)

Los resultados muestran que para la probabilidad de presencia de población mayor que 80% representadas en el gráfico anterior, con población igual a 10 se obtienen soluciones mejores con relación probabilidad – coste, también son mejores las soluciones con 15% de probabilidad de presencia. En el resto de soluciones (dos puntos con probabilidad de presencia de 40% y 60%) existe un claro dominio de las soluciones generadas con población igual a 100. Se destaca también el gran número de soluciones para todas las poblaciones.

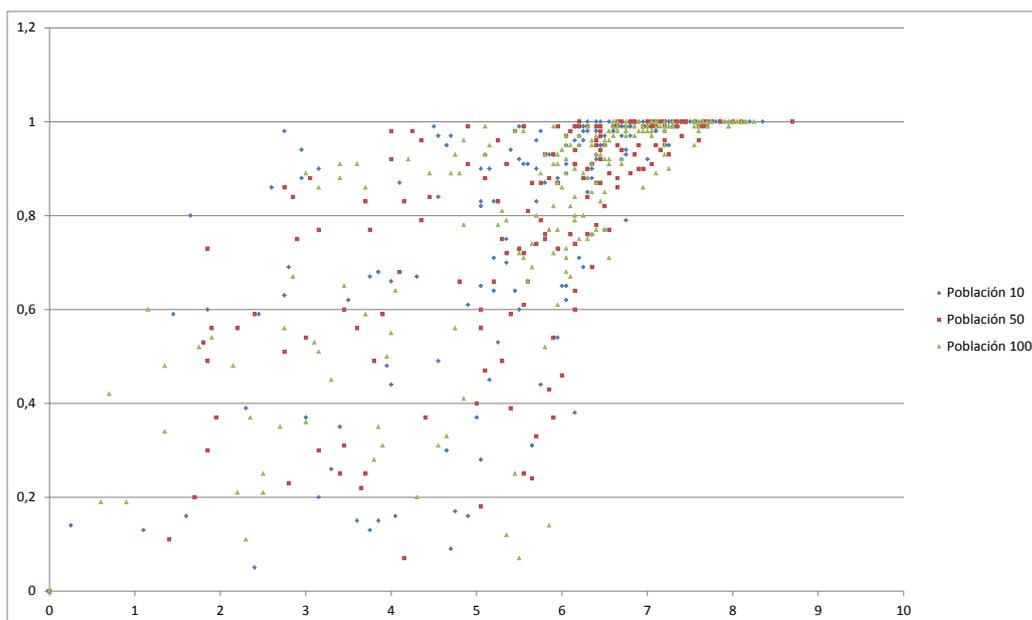


Figura 90. Total de soluciones para 2.000 generaciones (Experimento “Inundación” con 1000 simulaciones)

En la figura anterior no se aprecia que las soluciones sean mejores dependiendo de su tamaño de población.

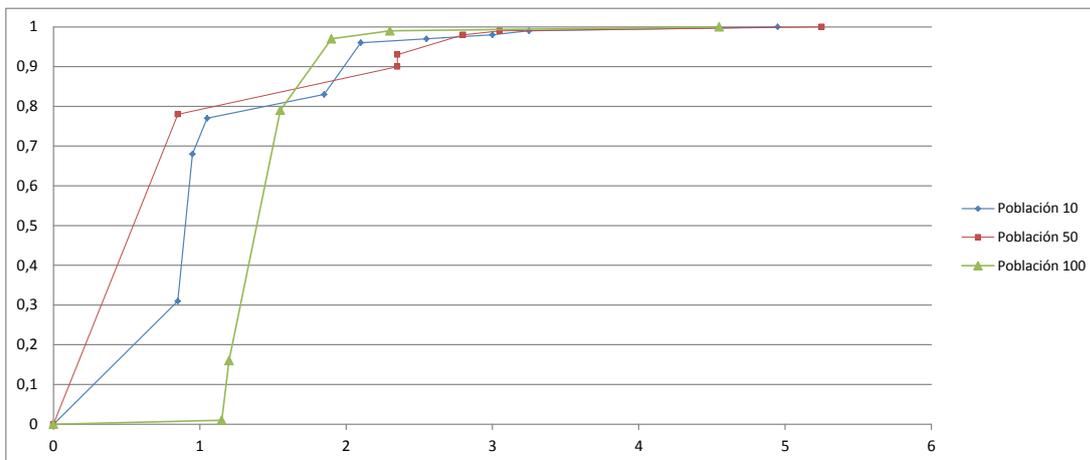


Figura 91. Comparativa de soluciones “No Dominadas” para 20.000 generaciones (Experimento “Inundación” con 1000 simulaciones)

En el gráfico anterior se puede observar que, con población 50 una solución mejor para probabilidad de presencia 80%, aunque posteriormente se suceden mejores soluciones con población igual a 100 para probabilidades de presencia cercanos al 100%. También se puede destacar que se exploran las soluciones con probabilidades de presencia altas para los tres tamaños de población.

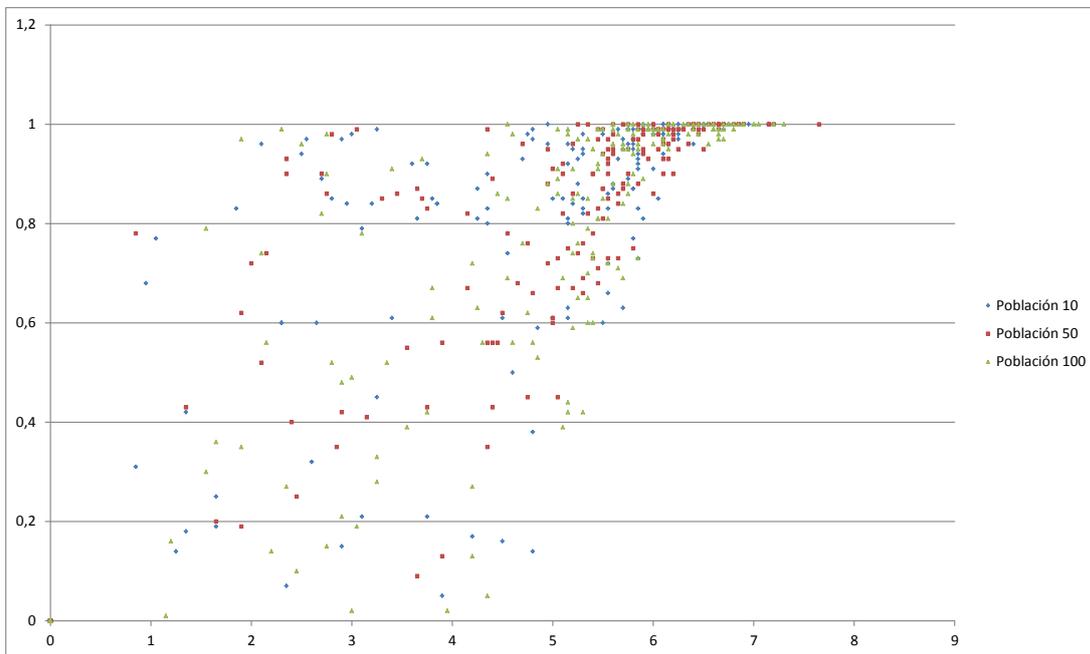


Figura 92. Total de soluciones para 20.000 generaciones (Experimento “Inundación” con 1000 simulaciones)

Para la figura anterior se obtienen las mismas conclusiones que para el total de soluciones con 2.000 generaciones.

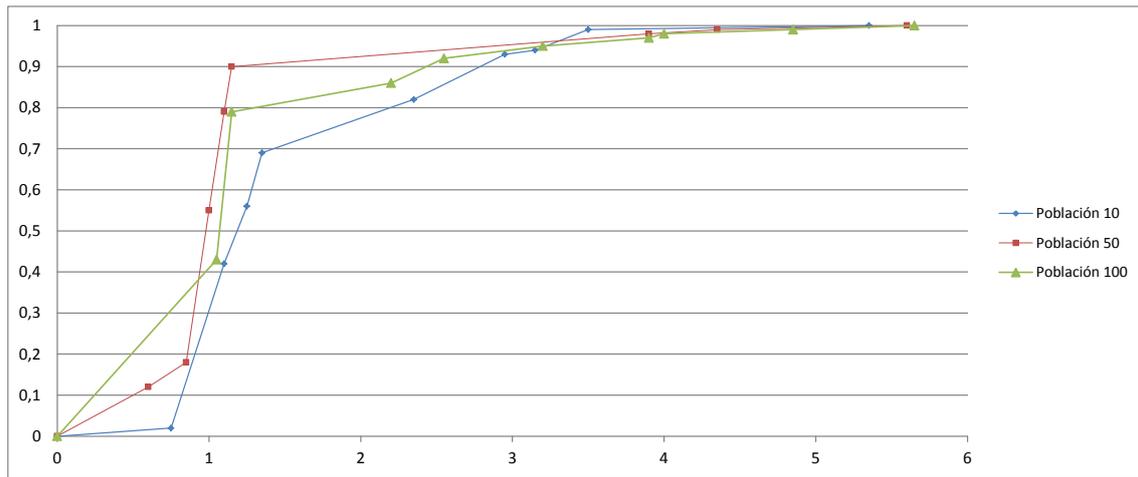


Figura 93. Comparativa de soluciones “No Dominadas” para 40.000 generaciones (Experimento “Inundación” con 1000 simulaciones)

En el gráfico anterior puede observarse que las peores soluciones las ofrecen los experimentos con población igual a 10 excepto con probabilidad de presencia cercano al 100%, las mejores son ofrecidas por los experimentos con población igual a 50 excepto en el punto mencionado para población igual a 10.

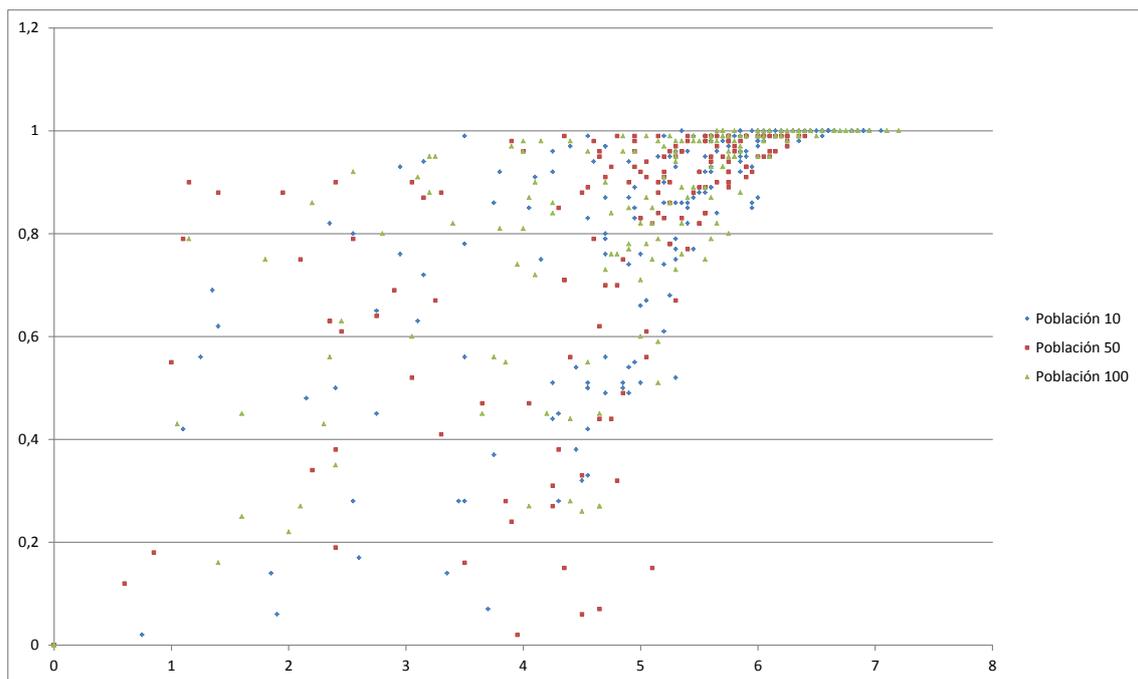


Figura 94. Total de soluciones para 40.000 generaciones (Experimento “Inundación” con 1000 simulaciones)

No se aprecia una evolución en las soluciones al aumentar la población, por lo que no puede concluirse que los resultados explicados anteriormente sean genéricos pudiéndose haber producido por azar.

5.4.4.2.3 Resultados agrupados por tamaño de población

A continuación se muestra gráficamente los resultados agrupados por tamaño de población 10:

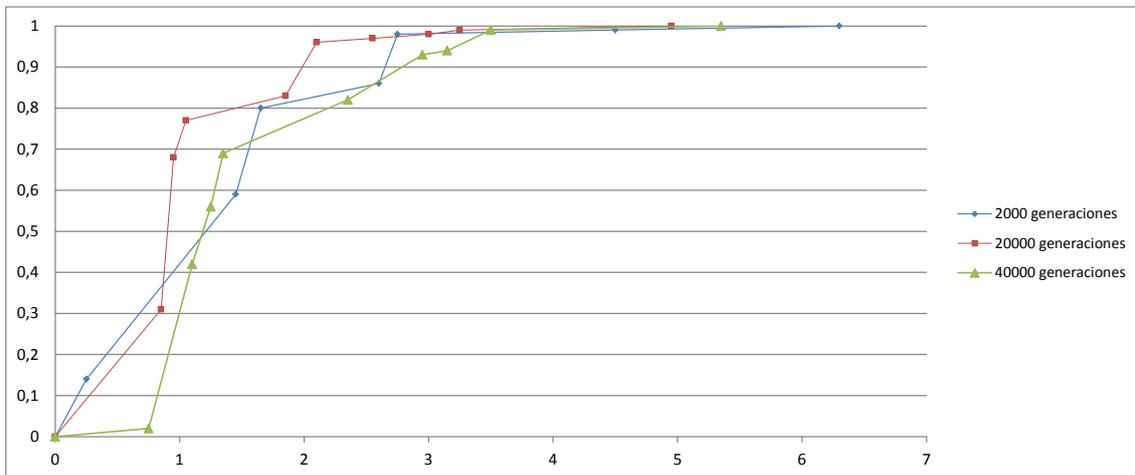


Figura 95. Comparativa de soluciones “No Dominadas” para población 10 (Experimento “Inundación” con 1000 simulaciones)

Se puede observar que se producen más soluciones con mejor relación probabilidad de ausencia y coste con 20.000 generaciones, solo existe un punto con 15% de probabilidad de población donde se supera con 2.000 generaciones, en cuanto a las ejecuciones con 40.000 generaciones, todas las soluciones reflejan ser peores que las ejecuciones con 20.000 generaciones.

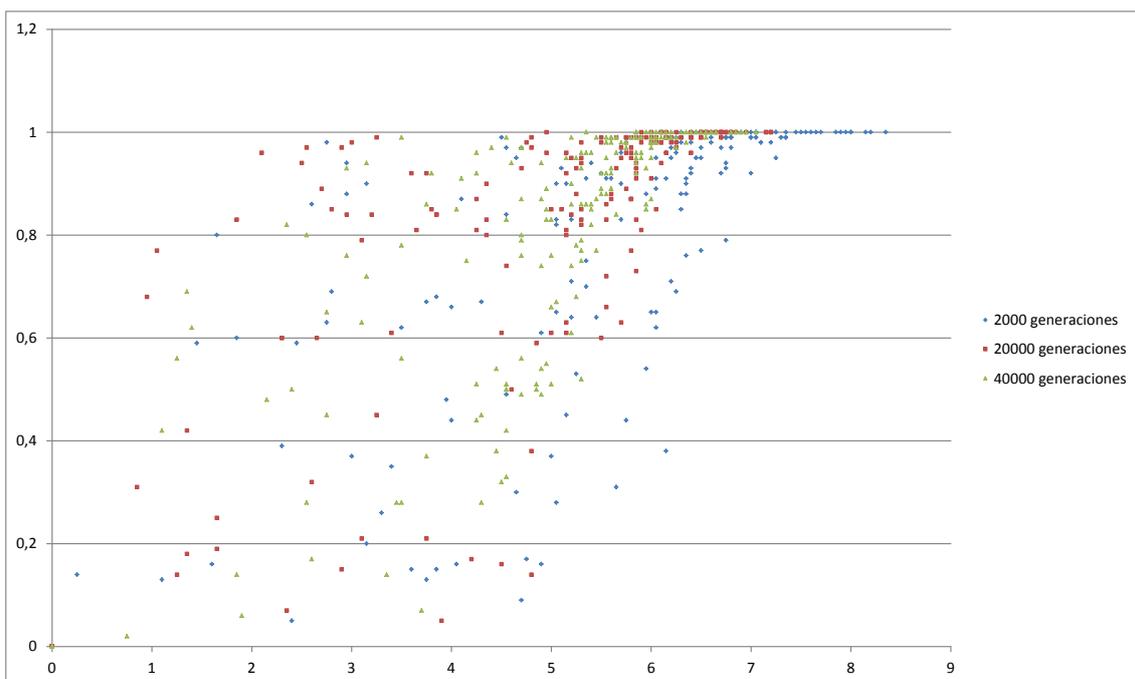


Figura 96. Total de soluciones para población 10 (Experimento “Inundación” con 1000 simulaciones)

Se puede observar que, pese a la dispersión, existe una acumulación de puntos con coste cercanos a 8 para las ejecuciones con 2.000 generaciones. Mostrándose la tendencia de evolución en el eje de coste al aumentar el número de generaciones.

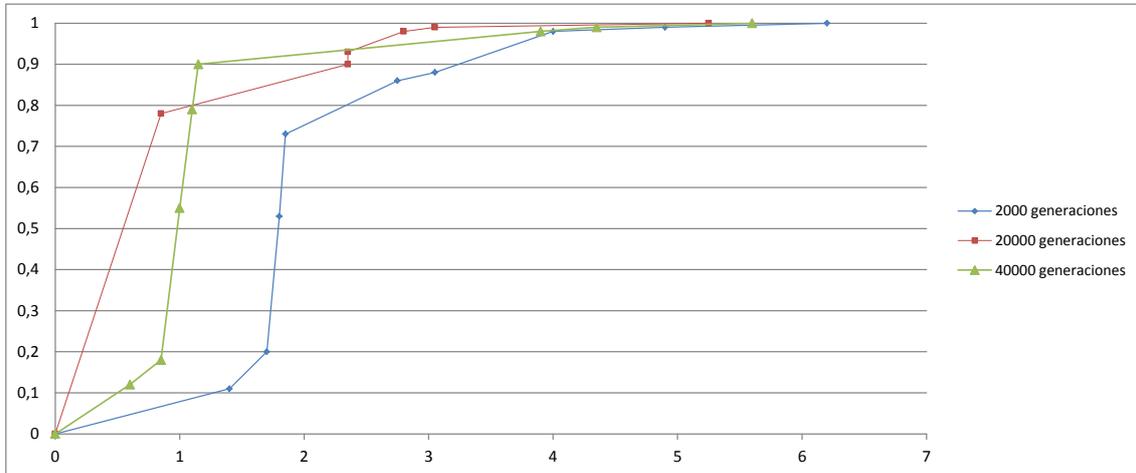


Figura 97. Comparativa de soluciones “No Dominadas” para población 50 (Experimento “Inundación” con 1000 simulaciones)

En esta ocasión se puede apreciar que las soluciones con 40.000 generaciones dominan en número y en soluciones con probabilidad de presencia baja, aunque una ejecución con 20.000 generaciones domine sobre el resto con probabilidad de presencia cercana al 80% y con probabilidades altas.

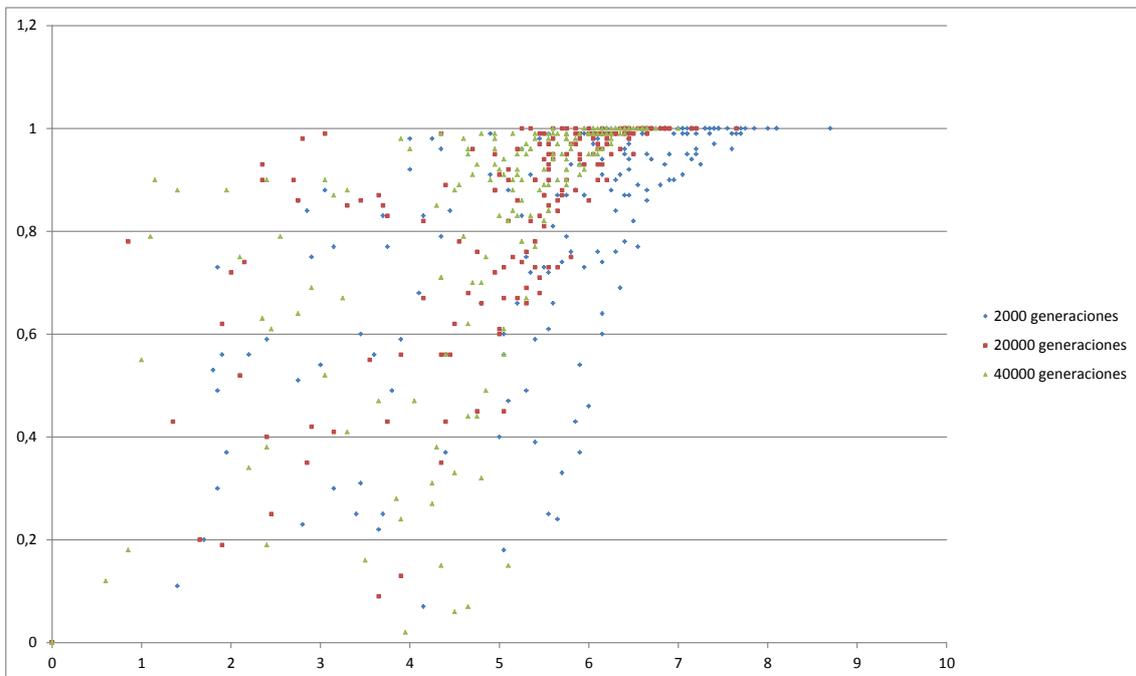


Figura 98. Total de soluciones para población 50 (Experimento “Inundación” con 1000 simulaciones)

Se puede ver que al aumentar las generaciones existe una evolución en las soluciones a costes menores, sin embargo esto no es extensible entre las soluciones con 20.000 y 40.000 individuos.

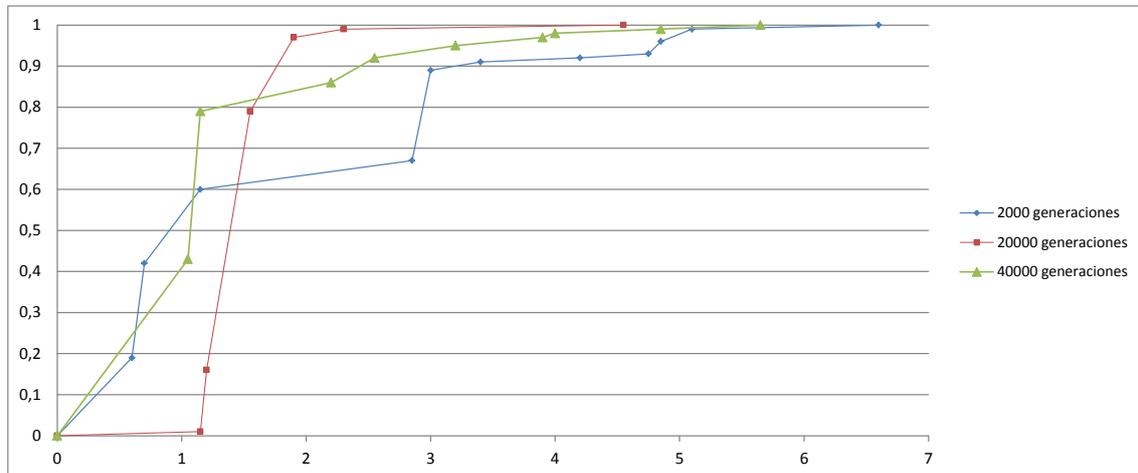


Figura 99. Comparativa de soluciones “No Dominadas” para población 100 (Experimento “Inundación” con 1000 simulaciones)

Según se puede observar, los frentes se cruzan varias veces, cuando la probabilidad de presencia es alta (aproximadamente 90%) dominan las soluciones de 20.000 generaciones, con 80% de probabilidad de presencia domina una solución con 40.000 generaciones, para el resto dominan los puntos con probabilidad de presencia de 20% y 40% para 2.000 generaciones.

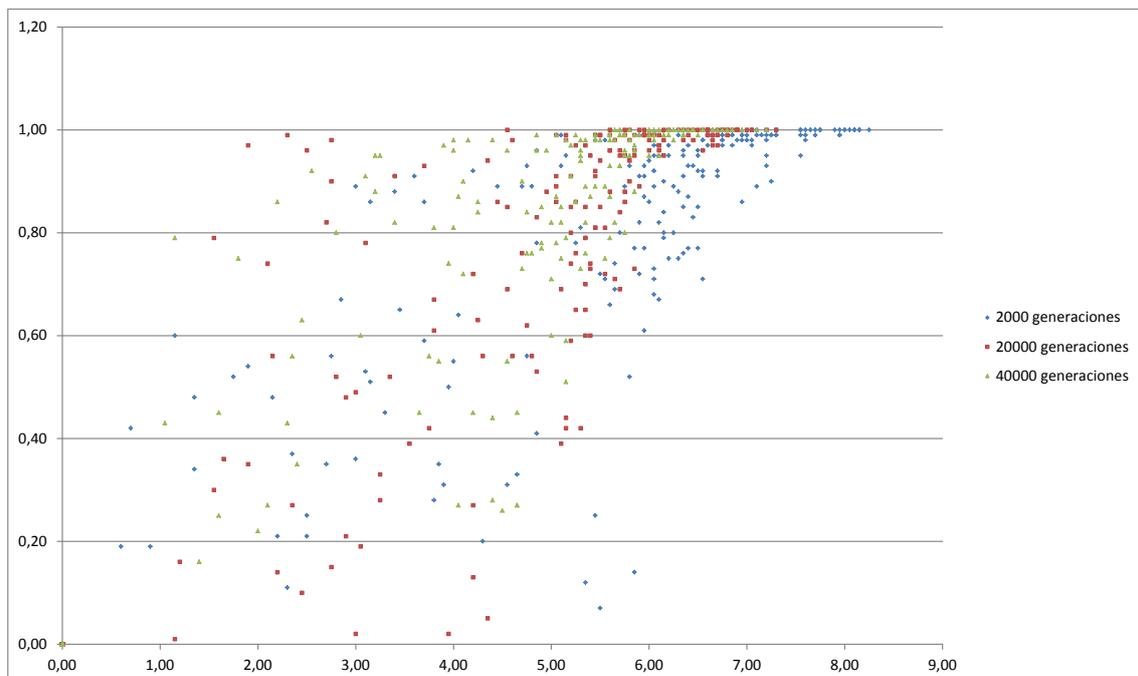


Figura 100. Total de soluciones para población 100 (Experimento “Inundación” con 1000 simulaciones)

Se puede observar que existe una mejor dispersión en las soluciones, lo que provoca que se evite que por azar, unas soluciones con menos generaciones obtenga mejores resultados.

Aunque se hayan obtenido resultados más positivos con 1.000 simulaciones, todavía se produce cierta variabilidad introducida por el autómata celular estocástico, por lo que sería conveniente establecer un número de repeticiones lo suficientemente alto y explorar otros modos de mejorar dicha dispersión.

CAPÍTULO 6: CONCLUSIONES

Una vez concluido el desarrollo y el trabajo experimental, se analizarán los resultados en función de los objetivos establecidos. En este capítulo se comentan las principales conclusiones extraídas. Así mismo, se comentan las posibles mejoras que podrían incluirse en esta aplicación o trabajos futuros o investigaciones que podrían desarrollarse a partir del mismo.

6.1 Objetivos alcanzados

El proyecto desarrollado cumple con los objetivos planteados inicialmente:

- Se ha definido una forma de representar un entorno geográfico que permite simular el movimiento de poblaciones a lo largo del tiempo. La herramienta cuenta con la posibilidad de hacer esta simulación a partir de un autómata celular estocástico que opera en base a un modelo de atractivo. Este modelo se construye a partir de un grupo de factores relevantes para la población.
- Se ha podido crear un entorno para modificar los factores relevantes donde se puede evaluar el impacto de las intervenciones en el movimiento de la población, y se permite calcular el coste asociado a las mismas.
- Se permite optimizar de forma automática con varios objetivos.
- Se permite opcionalidad en todas las opciones relacionadas con la aplicación de intervenciones.
- Se ofrece modularidad en la aplicación.

6.2 Conclusiones finales

El modelado de sistemas ecológicos es un escenario adecuado para el uso de autómatas celulares estocásticos. Tal y como dejan de manifiesto los trabajos en el apartado dedicado a la revisión de la literatura, esta técnica ofrece muy buenos resultados en cuanto a la fidelidad con la realidad, mostrando pautas y comportamientos que pueden verse en sistemas dinámicos reales. El trabajo realizado muestra una posible vía para su aplicación en tareas de planificación de desplazamiento de poblaciones a través de modelos de atractivo construidos a partir factores relevantes.

La aplicación desarrollada ilustra cómo se podría hacer predicción sobre presencia o ausencia de población y cómo aprovechar este modelo para predecir las consecuencias de los cambios sobre el medio. Una vez esto es factible, se cuenta con la posibilidad de estimar las consecuencias de alterar el medio a propósito, lo que aporta nuevas posibilidades para la gestión. La aplicación permite simular escenarios intervenidos y hacer optimización automatizada de intervenciones, sirviendo como

herramienta de soporte a la toma de decisiones. En relación con esto último, hemos visto como los algoritmos genéticos multiobjetivo, por la capacidad y flexibilidad que presentan, parecen una técnica adecuada para determinar las combinaciones de alteraciones sobre los factores relevantes más eficientes económicamente.

La construcción de la aplicación tenía como objetivo establecer un marco de trabajo estable y completo sobre el que se pudiese seguir avanzando en el futuro. Esto se ha podido conseguir diseñando una aplicación modular que ya incorpora capacidad suficiente para poder modelar y experimentar con distintos problemas, pero que es fácilmente extensible.

La experimentación con la aplicación tenía como objetivo mostrar que el comportamiento era adecuado y servir de guía a posibles formas de utilizarla. Entre los aspectos a validar pretendía que las simulaciones base mostrasen un comportamiento adecuado, que las intervenciones sobre los factores definidas por los usuarios a través de la interfaz gráfica fuesen sencillas y llevasen a resultados esperables, y que el módulo de optimización encontrase soluciones cercanas o idénticas a los óptimos (relación coste / probabilidad de presencia).

Para valorar esto último, se han hecho pruebas con dos problemas sintéticos, uno que modelaba una plaga y otro una inundación. Cada uno requirió 540 ejecuciones (1080 pruebas en total) en las que se variaron los tamaños de población y generaciones en el algoritmo genético (30 ejecuciones independientes por combinación). También se estudió la influencia de aumentar el número de simulaciones realizadas cada vez que se evaluaba una solución, lo que llevó el número de experimentos a 2160. Los resultados se evaluaron a partir de los resultados obtenidos en las métricas de Hipervolumen y Spread.

A pesar de que los resultados indican que se produce una optimización de las soluciones, los resultados no fueron tan claros como se esperaba. Al estudiar detalladamente el conjunto de soluciones resultantes se puede observar que existe un progreso en el que aparecen mejores soluciones si se incrementa el número de generaciones, sin que suceda lo mismo al aumentar la población. Analizando los resultados se ve como hay ejecuciones en las que se encuentran soluciones excelentes muy puntuales que distorsionan los frentes elaborados agrupando resultados parciales por categorías.

La primera batería de experimentos se realizó con 100 simulaciones por evaluación de la función objetivo de presencia. Al estudiar los resultados del escenario plaga gráficamente, se observó que la misma solución, evaluada varias veces, llevaba valores de la función muy diferentes (en algunos casos un 10%). Ante esta fuente de incertidumbre, se consideró necesario realizar pruebas con mil simulaciones por si la

reducción de la variabilidad en los resultados del autómatas celular modificase los resultados observados.

En este segundo grupo de pruebas se puede observar la misma pauta de mejora en el conjunto cuando se estudian los resultados en conjunto independientemente, aunque se sigue observando una gran variabilidad. Cuando se estudian los frentes de Pareto y las métricas, podemos observar que existe una cierta mejora con respecto a los resultados anteriores. Se producen menos fluctuaciones y es menos frecuente ver anomalías como encontrar soluciones mejores con menor número de generaciones.

Es muy posible que la codificación trivial escogida para la optimización para las pruebas de concepto también esté siendo fuente de problemas. En el caso de la inundación, la búsqueda de la solución pasa por modificar una celda crítica. Hasta que no se modifica esta, las aportaciones a la probabilidad de presencia de modificar otras son extremadamente limitadas.

En definitiva, podemos concluir que:

- El trabajo realizado muestra una nueva vía prometedora para desarrollar herramientas de apoyo a la toma de decisiones relacionadas con el desplazamiento de poblaciones
- Los algoritmos genéticos multiobjetivo se adaptan al problema, ofreciendo de forma automática un abanico de soluciones que maximizan o minimizan la población para distintos niveles de probabilidad.
- Con la codificación y operadores propuestos, se encuentran soluciones mejores al aumentar el número de generaciones, pero no se ve un patrón claro en relación al tamaño de la población.
- Según los resultados obtenidos, la componente estocástica del autómatas celular propuesto para modelar el movimiento introduce mucho ruido en el proceso de optimización. Por esta razón, el número de repeticiones del proceso de simulación es un parámetro crítico.
- Existen fuertes indicios de que la codificación simple elegida podría ser manifiestamente mejorable.
- El enfoque propuesto es computacionalmente muy intensivo, como se demuestra por las 24 horas aproximadamente que llevó en un procesador, cada uno de los 1.080 experimentos con 1.000 simulaciones por evaluación en los ejemplos propuestos.

6.3 Líneas futuras de investigación

Como trabajo de investigación a partir de este proyecto, se propone:

- Se propone el análisis, diseño e implementación de un modo de paralelizar la simulación y el proceso de optimización entre varias CPU, minimizando así el tiempo desde que el usuario pretende realizar la simulación y el que se informa al usuario de la solución.
- Analizar con detenimiento la influencia del número de repeticiones de la simulación sobre la dispersión de los resultados obtenidos en la experimentación. Esta investigación puede ser complementaria a otras y requerirá gran capacidad de CPU y memoria.
- Estudiar la escalabilidad, aumentando el espacio de experimentación, aumentando el número de celdas, el número de factores relevantes y/o los instantes.
- Examinar otras alternativas con otras técnicas de optimización, por ejemplo con otro optimizador multiobjetivo evolutivo como SPEA2, comparando los resultados con la experimentación desarrollada en el marco de este proyecto y evaluando las mejoras.
- Hacer un estudio exhaustivo de la sensibilidad de los resultados a los parámetros de mutación, tamaño de torneo, número de generaciones etc., comparando los resultados con la experimentación desarrollada en el marco de este proyecto y evaluando las mejoras.
- Creación de nuevos módulos o modelos de simulación y atractivo, incorporando si fuera necesario nuevas propiedades a los factores relevantes e intervenciones, aumentando las capacidades de la aplicación.
- Aplicar la solución a otros entornos, como puede ser la propagación de incendios forestales, expansión de empresas o franquicias, etcétera.
- Estudio de alternativas en la forma de codificar las soluciones en el algoritmo genético que faciliten el proceso de búsqueda.
- Explorar las posibilidades de las técnicas de optimización evolutiva para cambiar la incertidumbre en el objetivo de probabilidad de presencia.

6.4 Problemas encontrados

El primer problema encontrado se localiza en el desconocimiento del dominio del problema, y del desconocimiento sobre la utilización y capacidades de todos los framework utilizados, aunque algunos si habían sido utilizados previamente, otros eran escogidos para facilitar el desarrollo y por reutilización de código.

Aunque conocía los fundamentos de la optimización evolutiva, también existía un gran desconocimiento sobre la optimización multiobjetivo, aplicaciones y alcance. Fueron necesarias frecuentes revisiones con el tutor, y una revisión de la documentación.

Los requisitos de la aplicación eran numerosos pero sencillos, por lo que no hubo un problema en este sentido, aunque algunos de los requisitos no eran concretos (por ejemplo, modularidad), dejando varias vías por las que desarrollar la aplicación.

Uno de los objetivos personales planteados para este proyecto era la mejora en mi capacidad para el desarrollo de interfaces de usuario, este proyecto me ha dado la oportunidad de crear una sencilla e intuitiva, no obstante ha requerido una mejora de técnica y destreza importante y una evolución en mi capacidad de diseño orientado a interfaces y programación.

También se han encontrado problemas con una aproximación inicial a los programas estadísticos (R), y a la experimentación científica que requería este proyecto.

La fase de experimentación ha sido muy laboriosa, teniendo que reaccionar ante la falta de capacidad de computación para hacer frente a la gran cantidad de pruebas y complejidad de estas.

Por último, el mayor problema que se ha percibido a lo largo del proyecto y detectado como un riesgo de no finalización del mismo, es el tiempo. Con otras actividades extraacadémicas no se ha podido tener todo el tiempo necesario para desarrollarlo en un tiempo óptimo, alargando el tiempo de implementación.

6.5 Opiniones personales

El desarrollo de este proyecto ha supuesto un gran esfuerzo y dedicación, no obstante me ha aportado grandes conocimientos técnicos, de diseño, de formalismos y otros menos medibles de organización y planificación.

De hecho, este proyecto es uno de los más extensos a los que he tenido que enfrentarme sin poder contar con una ayuda en las fases determinantes del proyecto, como parte negativa, esto ha hecho reducir el tiempo, pero como parte positiva he podido aprender mucho más en detalle cada parte del proyecto.

Un detalle que agradezco enormemente a este proyecto, es la introducción en investigaciones científicas basada en una experimentación rigurosa, con la aplicación de métricas.

El punto anterior, junto a mi admiración personal sobre las técnicas de computación evolutivas hace que un trabajo tan dificultoso haya tenido siempre un motivo de satisfacción por el trabajo realizado y de sensación de progreso.

Anexo A: PLANIFICACIÓN

En este anexo se incluye la planificación del proyecto. En primer lugar se muestra la planificación establecida inicialmente y, posteriormente, la planificación final.

A.1 Planificación

En este apartado final se incluye la realizada después de comenzar a desarrollar el proyecto, y muestra el resultado final y el resumen de la actividad total. Como se puede observar, el proyecto comenzó a principios de septiembre de 2009, y termina el 14 de octubre de 2011. Lo que hace un total de 24 meses y medio de trabajo.

Id.	Nombre de tarea	Comienzo	Fin	Duración	2009				2010								2011								
					sep	oct	nov	dic	ene	feb	mar	abr	may	jun	jul	ago	sep	oct	nov	dic	ene	feb	mar	abr	may
1	Definición y Alcance del Sistema	01/09/2009	12/01/2010	96d	█																				
2	Fase de Análisis	01/01/2010	16/04/2010	76d					█																
3	Fase de Diseño	05/04/2010	20/09/2010	121d					█																
4	Fase de Implementación	15/09/2010	12/07/2011	215d									█												
5	Fase de experimentación	12/07/2011	07/10/2011	64d													█								
6	Fase de documentación	15/09/2010	14/10/2011	283d									█												

Figura 101. Planificación final

Aunque el proyecto comenzó despacio por el último curso de carrera 2009-2010, en el año 2010 se hizo un avance muy significativo tanto en análisis/diseño como en implementación, no obstante, actividades laborales durante el año 2011 han desechado varias planificaciones y la consecución se ha alargado a mi pesar más de lo pretendido inicialmente.

Aunque la fase de documentación está presente en toda la fase de desarrollo del proyecto, se ha reflejado el trabajo de la memoria, por ser más representativo y más cuatificable.

También se puede observar que la fase más duradera ha sido la de implementación-pruebas y experimentación.

Anexo B: PRESUPUESTO

En el siguiente anexo se mostrarán los distintos gastos necesarios para llevar a cabo el desarrollo del proyecto. En cada uno de los apartados se especificarán, de forma desglosada, todos y cada uno de los gastos.

B.1 Cálculo de costes

En los costes finales del proyecto se presupuestan los gastos finales en concepto de hardware, software, gastos de personal y derivados.

En gastos personales se recoge el trabajo de 425 días laborables netos, a 8 horas de trabajo/día y 20€ la hora tenemos un total de 68.000€

En el apartado de gastos de hardware y software, se recogen los gastos de tres estaciones de trabajo con Windows XP (595€ cada estación y 199€ cada licencia) y el paquete Microsoft office 2010 (licencia de 699€), mientras que los gastos fungibles recogen el gasto en material de oficina.

Descripción	Importe
Gastos de personal	68.000 €
Gastos hardware	1.785 €
Gastos software	1.296 €
Gastos fungibles	100 €
Total costes directos	71.181 €
Riesgos (10%)	7.118,1 €
Beneficios (10%)	7.118,1 €
Total	85.417,2 €

Tabla 93: presupuesto final del proyecto

Por tanto, el presupuesto final es de 85.417,2 € (ochenta y cinco mil cuatrocientas diecisiete coma dos).

Anexo C: DEFINICIONES Y REFERENCIAS

C.1 Referencias

- [1] **Dictionary of Bioscience**. McGRAW-HILL Companies, Inc. Second Edition.
- [2] **Viability of populations in a landscape**. P.J. Darwen, D.G. Green. *Ecological Modelling* 85, 1996, 165-171.
- [3] **Definición de autómata celular según Wikipedia**.
http://es.wikipedia.org/wiki/Automata_celular
- [4] **Cellular Automata as a Paradigm for Ecological Modeling**. P. Hogeweg. *Bioinformática*, 1988, Volume 27 Issue 1
- [5] **Biocomplexity of deforestation in the Caparo tropical forest reserve in Venezuela: An integrated multi-agent and cellular automata model**. Niandry Moreno, Raquel Quintero, Magdiel Ablan, Rodrigo Barros, Jacinto Dávila, Hirma Ramírez, Giorgio Tonella, Miguel F. Acevedo. *Environmental Modelling & Software* 22, 2007, 664-673
- [6] **Applying cellular automata to complex environmental problems: The simulation of the bioremediation of contaminated soils**. S. Di Gregorio, R. SerrabT, M. Villani. *Theoretical Computer Science* 217, 1999, 131-156
- [7] **A Cellular Automaton Framework for Infectious Disease Spread Simulation**. Bernhard Pfeifer, Karl Kugler, Maria M. Tejada, Christian Baumgartner, Michael Seger, Melanie Osl, Michael Netzer, Michael Handler, Andreas Dander, Manfred Wurz, Armin Graber and Bernhard Tilg. *The Open Medical Informatics Journal*, 2008, 2, 70-81
- [8] **Cellular Automata in Ecological and Ecohydraulics Modelling**. Arthur Mynett, Qiuwen Chen. 2004, Volume 3305/2004, 502-512
- [9] **A model for predicting forest fire spreading using cellular Automata**. Ioannis Karafyllidis, Adonios Thanailakis. *Ecological Modelling* 99, 1997, 87-97.
- [10] **A Cellular Automata Model for Population Dynamics Simulation of Two Plant Species with Different Life Strategies**. Jie Xu, Baojing Gu, Yanting Guo, Jie Chang, Ying Ge, Yong Min, Xiaogang Jin. *Proceedings of 2010 IEEE International Conference of Intelligent Systems and Knowledge Engineering*, Nov. 15-16, 2010, Hangzhou, China, pp. 517-523

- [11] ***Evolutionary Methods for Multi-Objective Portfolio Optimization***. I. Radziukynienė, A. Žilinskas. Proceedings of the World Congress on Engineering 2008 Vol II
- [12] ***Definición de Interpolación Lineal en Wikipedia***. http://es.wikipedia.org/wiki/Interpolacion_lineal
- [13] ***Maximum Likelihood Estimation With Java and Ruby***. Peter von Tessin. November 6, 2005.
- [14] ***Optimización robusta de carteras de inversión mediante SPEA2***. Carmen Lozano Masdemont. 2010
- [15] ***Aplicación de NSGAI y SPEA2 para la optimización multiobjetivo de redes multicast***. <http://redalyc.uaemex.mx/pdf/852/85201702.pdf>
- [16] ***Introducción a la optimización Multiobjetivo***. Transparencias de Dr. A Coello Coello. Julio 2005.
- [17] ***Definición de Eficiencia de Pareto en Wikipedia***. http://es.wikipedia.org/wiki/Eficiencia_de_Pareto
- [18] ***Definición de Algoritmo Evolutivo en Wikipedia***. http://es.wikipedia.org/wiki/Algoritmo_evolutivo
- [19] ***Definición de Algoritmo Genético en Wikipedia***. http://es.wikipedia.org/wiki/Algoritmo_gen%C3%A9tico
- [20] ***Definición de caso de uso en Wikipedia***. http://es.wikipedia.org/wiki/Caso_de_uso
- [21] ***A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II***. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, T. Meyarivan. IEEE Transactions on evolutionary computation, Abril 2002, Vol. 6, No. 2
- [22] ***Multi-objective optimization using genetic algorithms: A tutorial***. Abdullah Konak, David W. Coit, Alice E. Smith. Reliability Engineering and System Safety 91, 2006, 992–1007
- [23] ***jMetal 3.1 User Manual***. Antonio J. Nebro, Juan J. Durillo. 2010
- [24] ***Stochastic Cellular Automata Models In Disease Spreading and Ecology***. Klaus Lichtenege. 2005

- [25] ***Una Versión Paralela del Algoritmo Evolutivo para Optimización Multiobjetivo NSGA-II y su Aplicación al Diseño de Redes de Comunicaciones Confiables.*** Sergio Nasmachnow. Uruguay, 2004.

- [26] ***Disposición óptima de antenas usando CHC multiobjetivo.*** Enrique Alba, Guillermo Molina y Antonio J. Nebro. Artículo, España 2005.

Anexo D: MANUAL DE USUARIO

En este primer anexo se incluye el manual de usuario que acompañará la aplicación. También se incluye el formato de entrada y salida de los ficheros.

D.1 Ficheros de entrada

Fichero inicial

El fichero de entrada tendrá el siguiente formato Microsoft Excel2003:

1. Pestaña "Detalles experimento". Se podrá ver en la segunda columna los siguientes valores:
 - Fila 1: número de instantes. Número de instantes de la simulación.
 - Fila 2: número de mundos. Número de mundos que se ofrecen en el fichero inicial (será necesario al menos dos, uno inicial y uno final).
 - Fila 3: número de variables. Número de tableros o variables que se instancian.
 - Fila 4: tamaño del tablero. Número de filas y columnas del tablero.
2. Pestaña de tableros o variables. Se podrá ver en la segunda columna el número de instante de ese tablero. A continuación, en la siguiente fila se muestra el tablero según el tamaño del tablero, siguiéndose de una fila en blanco posteriormente.

Los decimales de los números estarán indicados con un punto (ejemplo de valor: 0.2)

Se mostrarán tantos tableros como "número de mundos" se detallan en la primera pestaña. Existirán tantas pestañas de tablero o variables como "número de variables"

3. Pestaña de población. Se detallará un tablero con tamaño indicado en la primera pestaña en el que se indique la población inicial (1= existe población en la celda; 0=no existe población en la celda)

Fichero de intervenciones

El fichero de intervenciones tendrá el formato Microsoft Excel2003 y se detallarán todos los instantes para todas las variables con sus respectivas intervenciones. Si en una celda no se desea realizar intervención se podrá poner el mismo valor que en la celda original o un -1 que indica que no se realizará intervención para esa celda e instante.

Fichero de costes

El fichero de costes tendrá el formato Microsoft Excel2003 y se detallarán todos los instantes para todas sus variables con sus respectivos costes.

Fichero de salida VAR y FUN

La optimización automática realizada con JMetal ofrecerá el resultado por archivo, que será el detallado también mediante la vista de gráfico, esta salida contemplará cada solución en una línea distinta. En el fichero FUN se detalla los resultados en los siguientes términos: población (espacio) coste. Para el fichero VAR se detalla las intervenciones realizadas en la optimización automática.

D.2 Manual de Usuario

En este documento se muestra la funcionalidad básica de la aplicación así como la navegación básica entre ventanas.

1.1 Funcionalidad

Esta aplicación ha sido diseñada con el objetivo de encontrar intervenciones en características de terreno que permitan aumentar o disminuir la densidad de población en un terreno determinado. Para ello se ofrecen los siguientes módulos:

- Módulo de simulación, que permite establecer relaciones entre las características de un terreno y la población.
- Módulo de cambios que ofrece soporte para realizar intervenciones y volver a simular con dichas intervenciones.
- Módulo de optimización para realizar una optimización automática multi-objetivo. Permite mostrar el frente de Pareto resultante.

1.2 Requisitos mínimos

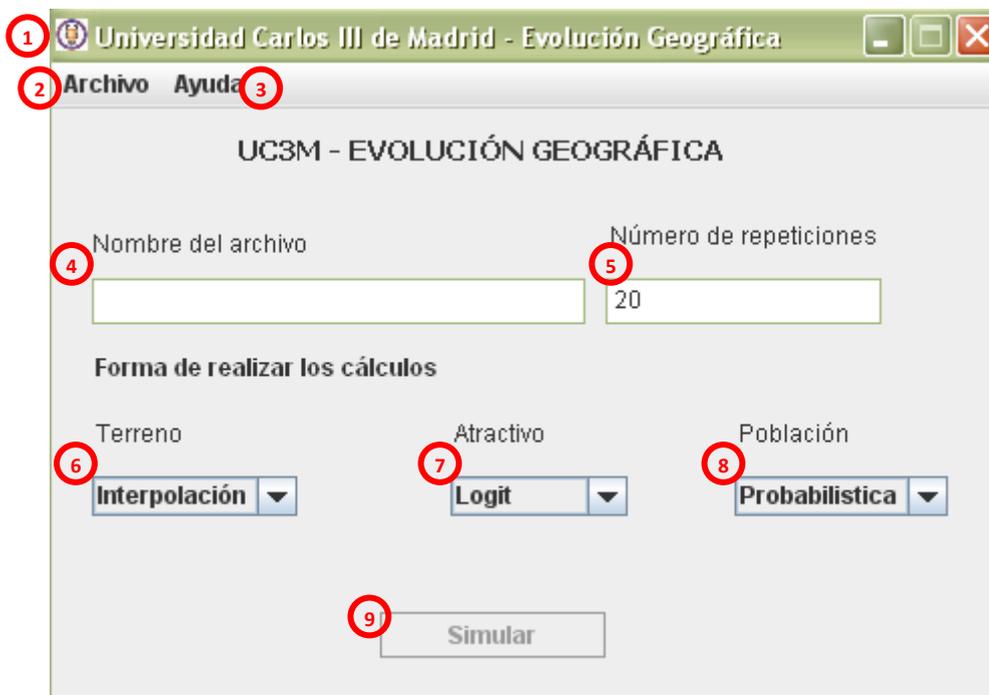
- Para ejecutar la aplicación se requiere tener instalado el entorno de ejecución java (JRE, Java Runtime Environment) 1.6 o superior¹. Se puede ejecutar la aplicación en cualquier sistema compatible con dicho entorno de ejecución, independientemente del Sistema Operativo.
- Se recomienda un entorno de desarrollo gráfico como Netbeans o Eclipse para la ejecución de la aplicación.
- Para realizar los contrastes estadísticos de las métricas de desempeño, se requiere tener instalado el entorno de programación R². Puede ejecutarse sobre cualquier Sistema Operativo, y no es necesario conocer este lenguaje para poder realizar los contrastes, ya que se proporciona el script necesario.

¹ Se puede descargar en <http://java.sun.com>

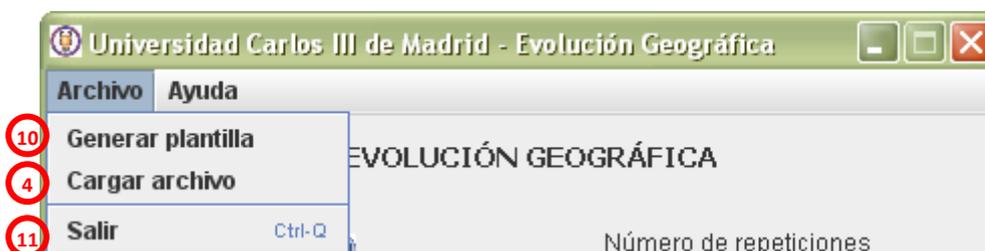
² Se puede descargar en <http://cran.r-project.org/>

1.3 Pantalla Principal

Al abrir la aplicación se mostrará la pantalla principal, los elementos que la componen y las acciones que se pueden realizar desde aquí son las siguientes:



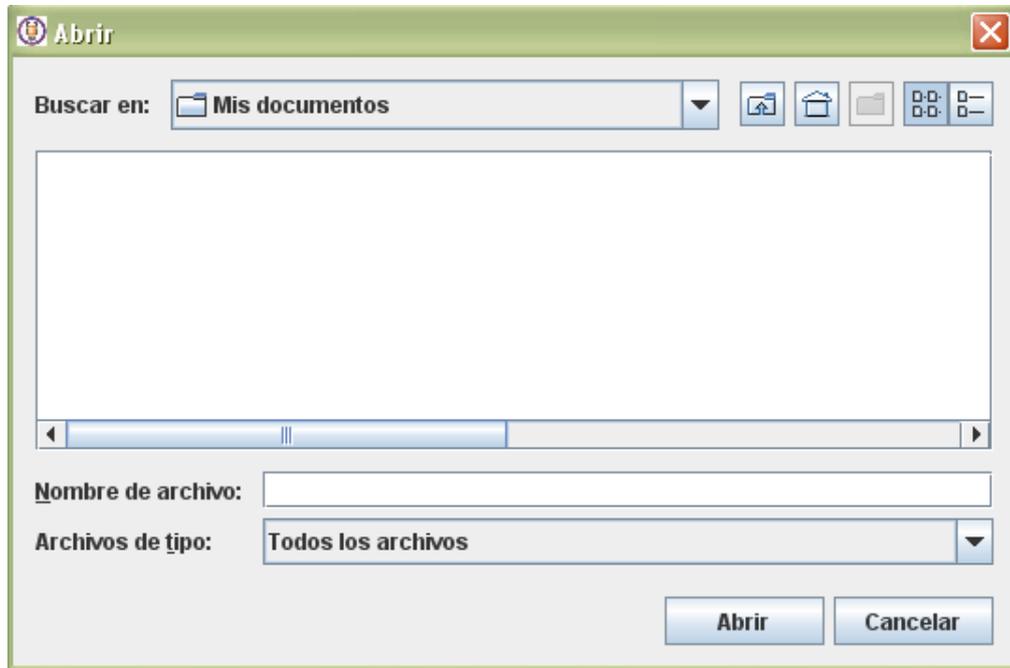
1. Barra de título. Muestra el nombre de la aplicación y de la universidad. En esta barra se podrá identificar la ventana en la que nos encontremos. Se podrá minimizar y cerrar la ventana (esta acción cerrará la aplicación y liberará sus recursos).
2. Menú Archivo. Puede desplegarse mostrando los siguientes elementos:



3. Menú Ayuda. Puede desplegarse de forma que aparezca este manual e información del programa.
4. Cargar Archivo inicial. Existen dos formas de cargar un archivo en la aplicación: desde el menú de Archivo > Cargar archivo, en este caso se podrá elegir el archivo a cargar seleccionándolo, o directamente en la barra de texto de la pantalla principal, rotulada como "Nombre del archivo" (en este último caso deberá referirse al archivo con su ruta

absoluta). De cualquiera de las dos formas, si la carga es correcta por tener un formato adecuado, se cargará en memoria el fichero y se habilitará el botón “Simular”

En el caso de cargar archivo mediante el Menú Archivo > Cargar Archivo se interactuará con la aplicación mediante la siguiente ventana:



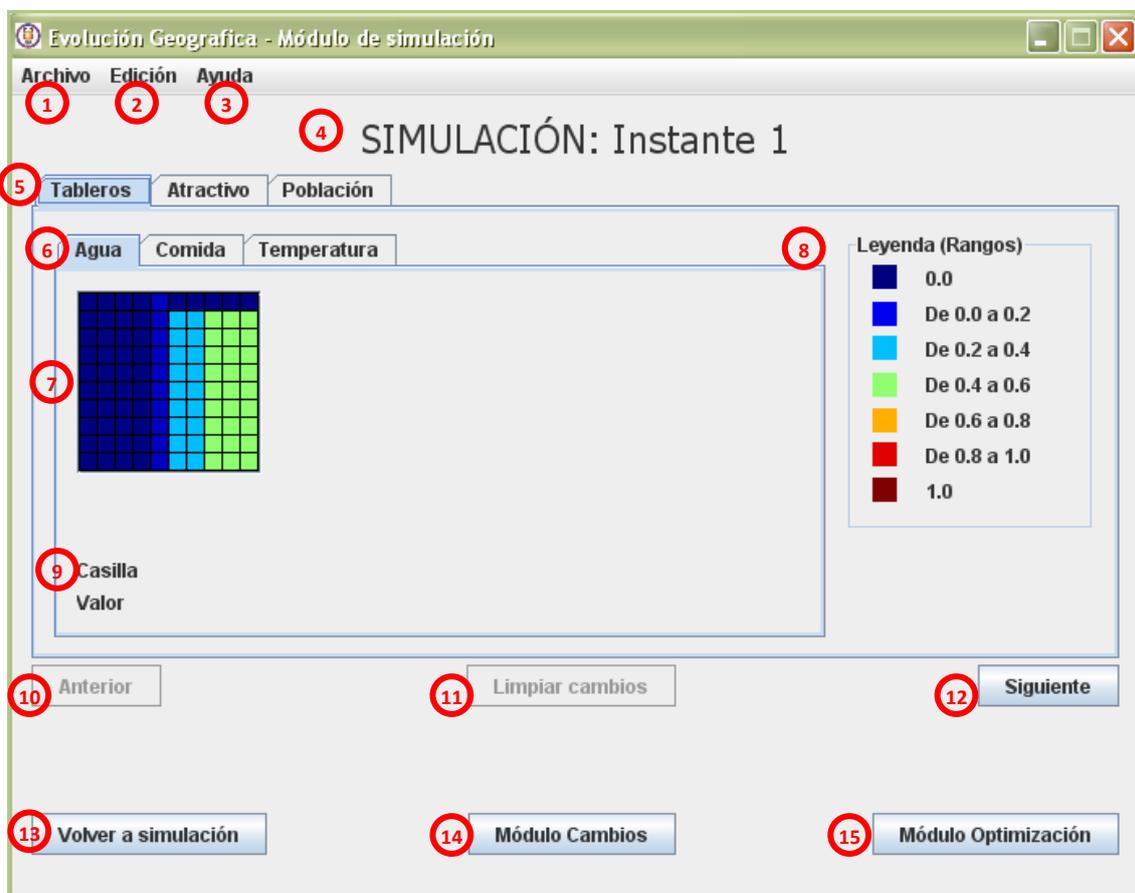
5. Número de repeticiones. Este valor representa el número de veces que se realizará la simulación. Establecer un número alto de repeticiones provocará que no se introduzca tanta aleatoriedad en el movimiento de población entre simulaciones independientes, pero se tardará más tiempo cuanto mayor sea este valor.
6. Simulador de Terreno. Será el simulador que se utilizará para asignar valores a los terrenos en los distintos instantes de tiempo.
 - 6.1. Interpolación. Este algoritmo para simular terrenos ofrece una asignación de valores de modo que se interpola los valores entre el instante inicial y final. La representación de valores sería una progresión lineal entre los valores iniciales hasta los valores finales.
7. Simulador de Atractivo. Será el simulador que se utilizará para asignar valores al atractivo en base a los valores de los terrenos de ese instante.
 - 7.1. Logit. Algoritmo que realiza una estimación del interés de cada celda analizando las relaciones entre las características del terreno inicial y su ocupación. Una vez establecida la relación y asignados unos pesos a cada variable, solo es necesario conocer el valor de las variables del terreno para establecer el atractivo.
8. Simulador de Población. Será el simulador que se utilizará para asegurar la presencia en una casilla e instante determinado, dependiendo de las presencias en las celdas propias y adyacentes en el instante anterior, y del atractivo de la celda propia en ese instante.

- 8.1. Probabilístico. Algoritmo que implementa un autómata celular estocástico. Este autómata, se moverá de una celda a otra adyacente (o se quedará en esa celda) de modo probabilístico dependiendo del atractivo de la celda destino.
9. Simular. Botón que permite iniciar la simulación con los parámetros escogidos (tipos de simuladores y número de repeticiones).
10. Generación de plantilla. Se puede generar una plantilla de forma que se tenga una referencia de cómo realizar un archivo de carga inicial. Al usar esta opción se creará una plantilla en formato Excel 2003 en el directorio que se elija, se generará un fichero con formato de tablero inicial.
11. Salir de la aplicación. Cierra la aplicación y libera los recursos.

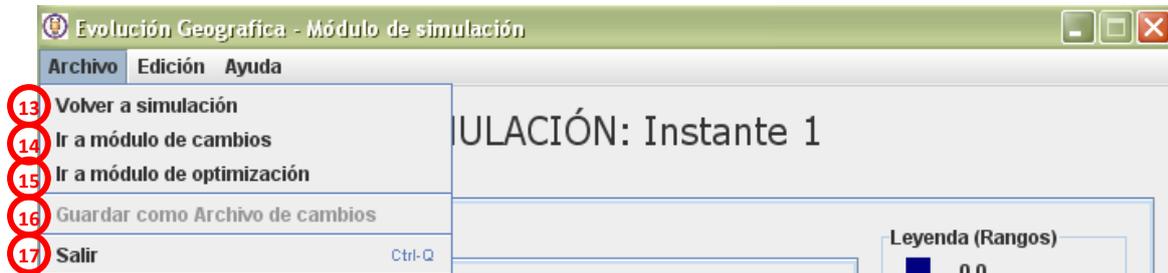
Una vez cargado un mundo inicial y aceptados los parámetros por defectos (o modificados según las preferencias) será posible realizar la simulación (para ello, se habilitará automáticamente el botón “Simular”). Cuando se hayan concluido las simulaciones (definidas por el parámetro “Número de repeticiones”) se mostrará la ventana de simulación.

1.4 Ventana de simulación

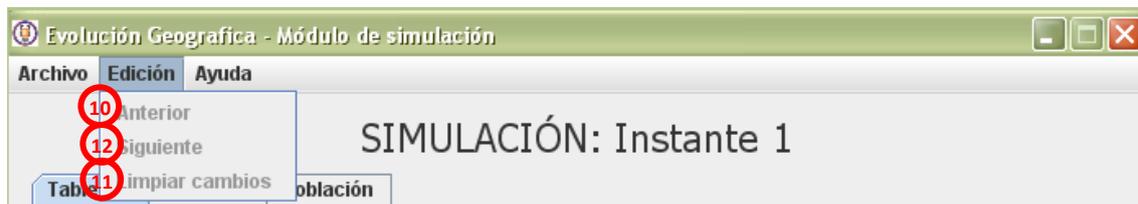
La ventana que se mostrará una vez realizada las simulaciones será la siguiente:



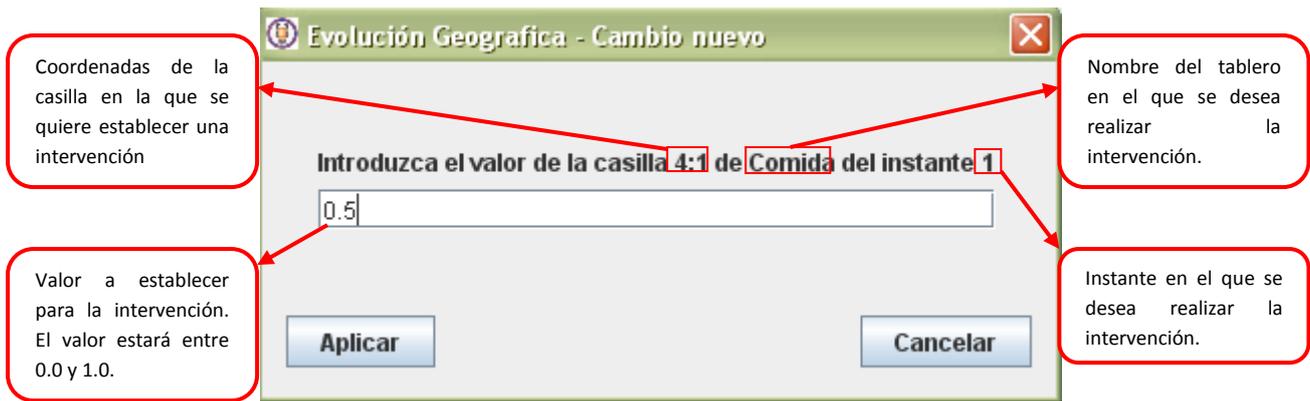
1. Menú Archivo. El menú Archivo tiene las siguientes opciones:



2. Menú Edición.



3. Menú Ayuda. Puede desplegarse de forma que aparezca este manual e información del programa.
4. Título de la ventana de simulación. Indica el instante que se está visualizando, se modificará automáticamente al presionar los botones anterior o siguiente (ver 10 y 12 respectivamente).
5. Pestañas de resultados de los simuladores. Se podrá cambiar la vista en cualquier momento cambiando de pestaña. Las posibilidades son: mostrar los mapas de terrenos, mostrar el mapa de atractivo, mostrar el mapa de población.
6. Pestañas de terrenos. Se podrá cambiar la vista en cualquier momento cambiando de pestaña. El número de pestañas viene determinado por las variables definidas en el fichero de entrada.
7. Tablero. Resultado de la simulación, en la que los valores han sido transformados a códigos de colores para una fácil identificación, este código de colores se puede ver en la leyenda (ver 8). Al pasar el puntero del ratón por encima del tablero podremos identificar su valor y su posición (ver 9). Además, es posible introducir una modificación haciendo 'click' en cualquier celda del tablero. En ese caso se mostrará la siguiente ventana:

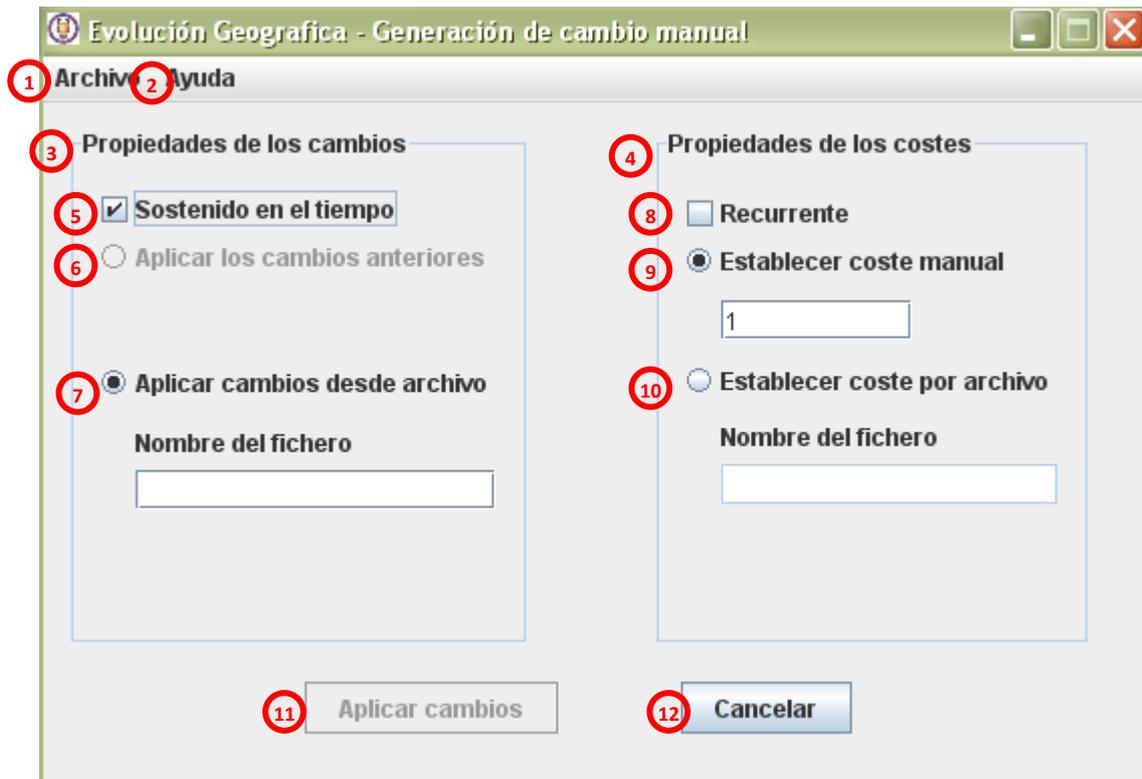


8. Leyenda. Identificación del color de las celdas como valores numéricos.
9. Valores de 'Casilla' y 'Valor'. Valores que se actualizan automáticamente al pasar por encima del Tablero (ver 7) con el ratón. "Casilla" indica las coordenadas de ubicación de la casilla (filas : columnas) y "Valor" indica el valor de ésta casilla.
10. Anterior. Botón que se activa si estamos visualizando un instante posterior al primer instante. Si se realiza una acción sobre éste se visualizará el instante anterior al visualizado actualmente.
11. Limpiar cambios. Botón que se activa cuando se ha introducido algún cambio manualmente. Si se realiza una acción sobre éste se borrarán todos los cambios introducidos.
12. Siguiente. Botón que se activa si estamos visualizando un instante anterior del último instante definido. Si se realiza una acción sobre éste se visualizará el instante posterior al visualizado actualmente.
13. Volver a simulación. Vuelve a la pantalla inicial, borrando de memoria todas las entidades creadas y permitiendo comenzar con nuevas simulaciones.
14. Módulo de cambios.
15. Módulo de optimización.
16. Guardar como Archivo de Cambios. Guarda los cambios realizados en el Tablero en un archivo de cambios con el formato Excel2003.
17. Salir. Finaliza la aplicación de modo que la cierra y libera los recursos.

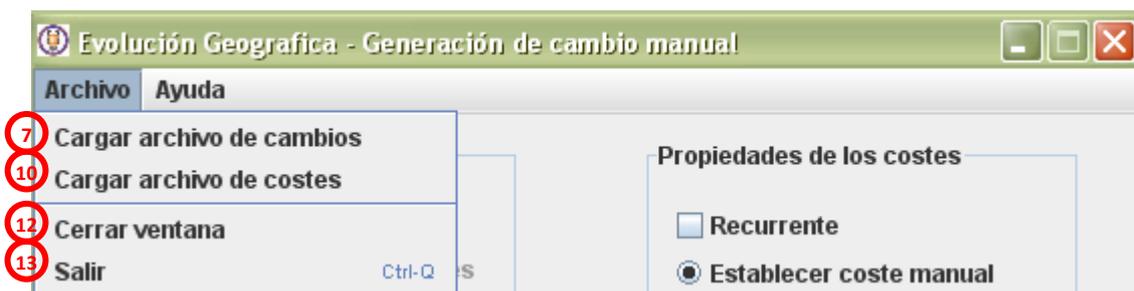
Cuando se ha realizado la primera simulación, será posible realizar cambios y/o abrir la ventana de opciones para insertar intervenciones manuales, o abrir la ventana de opciones del optimización.

1.5 Ventana de opciones de intervenciones

La ventana de opciones para las intervenciones introducidas manualmente por el usuario es la siguiente:



1. Menú Archivo. El menú Archivo tiene las siguientes opciones:



2. Menú Ayuda. Puede desplegarse de forma que aparezca este manual e información del programa.

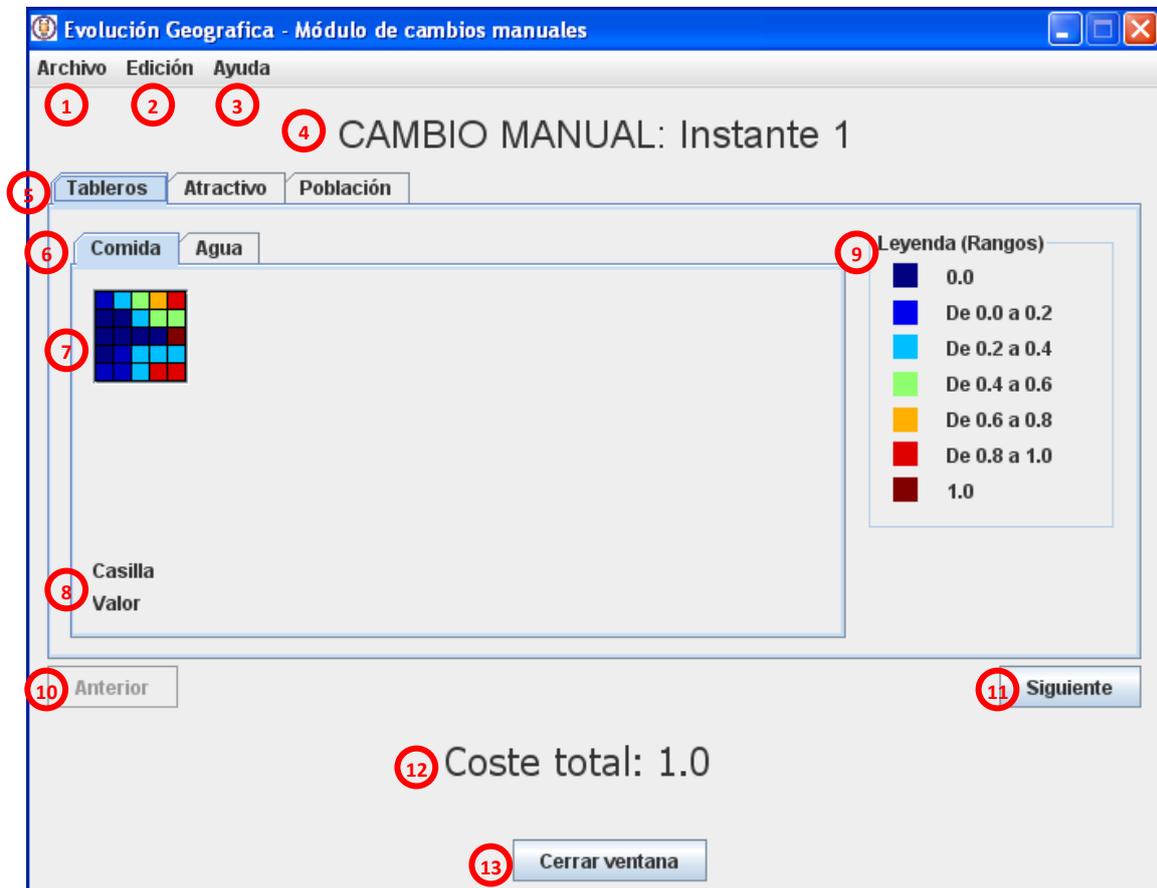
3. Propiedades de los cambios. Sección de la ventana de opciones de intervención que se centra en las diversas propiedades que se deben aplicar al realizar la simulación utilizando las intervenciones introducidas por el usuario.

4. Propiedades de los costes. Sección de la ventana de opciones de intervención que se centra en las propiedades que se deben aplicar al realizar la simulación utilizando las intervenciones introducidas por el usuario con respecto al cálculo de costes asociados a estos.
5. Opción "Sostenido en tiempo". Si se selecciona, el valor del campo será el de la intervención y se mantendrá constante desde el instante inicial hasta el instante final.
6. Opción aplicar intervenciones introducidas en el tablero. Opción que puede marcarse si se han definido cambios en el propio tablero mostrado en la ventana de simulación, esta opción es excluyente con la opción de carga de intervenciones desde archivo.
7. Opción aplicar intervenciones desde archivo. Opción que se marcará por defecto si no se han definido cambios en el propio tablero mostrado en la ventana de simulación. La opción es excluyente con la opción aplicar intervenciones introducidas en el tablero.
8. Opción "Recurrente". Opción que se marcará si se desea que la intervención tenga coste desde el instante que se realiza hasta el último (recurrente), en caso contrario el coste se aplicará en el instante en el que se realiza la intervención.
9. Opción establecer coste manualmente. Opción que permite establecer el coste manualmente. El formato será un número decimal. Opción excluyente con la opción de establecer el coste por archivo.
10. Opción establecer coste por archivo. Opción que permite establecer el coste por archivo (el coste podrá determinarse por cada celda de los tableros y por cada instante. Esta opción es excluyente con la opción de establecer coste manualmente.
11. Aplicar intervenciones. Botón que permite simular de nuevo con las intervenciones introducidas manualmente o por archivo.
12. Cancelar. Cancela la acción y cierra la ventana.
13. Salir de la aplicación. Cierra la aplicación.

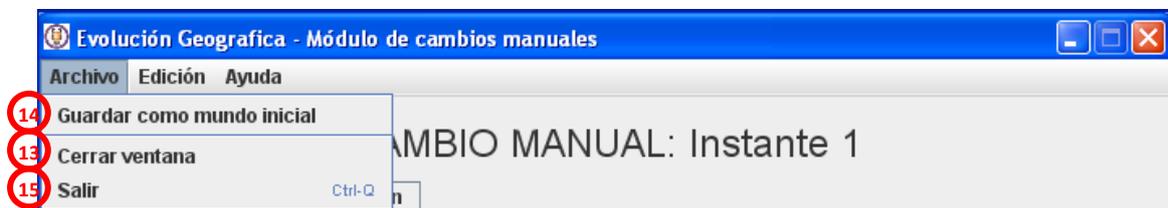
Una vez realizadas las intervenciones y la simulación se mostrará la ventana de simulación de intervenciones de usuario.

1.6 Ventana de simulación de intervenciones de usuario

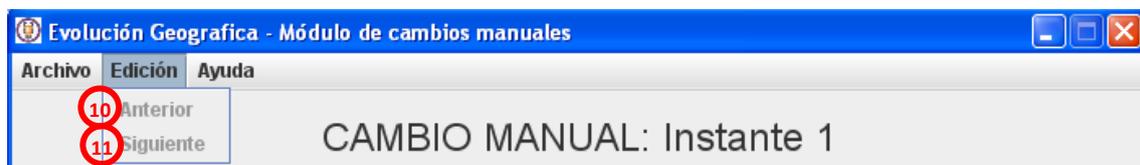
Esta ventana de simulación tiene en cuenta las intervenciones de usuario es la siguiente:



1. Menú Archivo. El menú Archivo tiene las siguientes opciones:



2. Menú Edición. El menú Edición tiene las siguientes opciones:

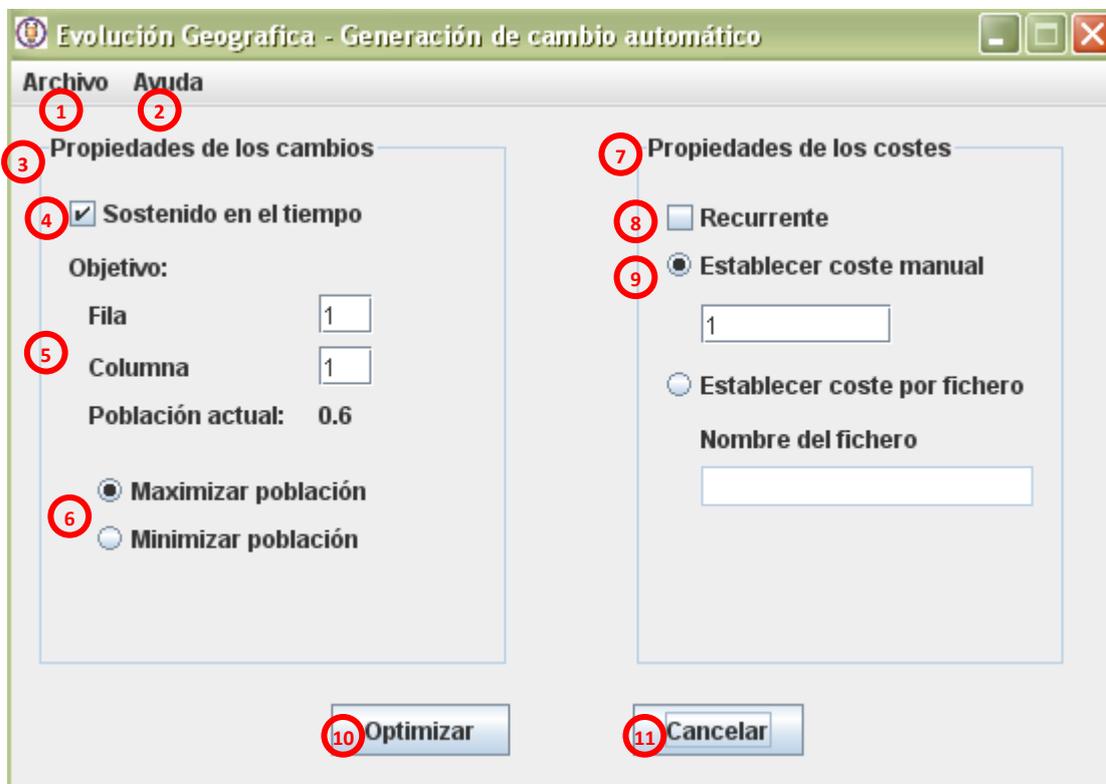


3. Menú Ayuda. Puede desplegarse de forma que aparezca este manual e información del programa.
4. Título de la ventana de simulación de intervenciones de usuario. Indica el instante que se está visualizando, se modificará automáticamente al presionar los botones anterior o siguiente (ver 10 y 11 respectivamente).
5. Pestañas de resultados de los simuladores. Se podrá cambiar la vista en cualquier momento cambiando de pestaña. Las posibilidades son: mostrar los mapas de terrenos, mostrar el mapa de atractivo, mostrar el mapa de población.
6. Pestañas de terrenos. Se podrá cambiar la vista en cualquier momento cambiando de pestaña. El número de pestañas viene determinado por las variables definidas en el fichero de entrada.
7. Tablero. Resultado de la simulación con intervenciones introducidas por usuario, en la que los valores han sido transformados a códigos de colores para una fácil identificación, este código de colores se puede ver en la leyenda (ver 9). Al pasar el puntero del ratón por encima del tablero podremos identificar su valor y su posición (ver 8). No es posible modificar el valor como ocurre en el caso de la ventana de simulación.
8. Valores de 'Casilla' y 'Valor'. Valores que se actualizan automáticamente al pasar por encima del Tablero (ver 7) con el ratón. "Casilla" indica las coordenadas de ubicación de la casilla (filas : columnas) y "Valor" indica el valor de ésta casilla.
9. Leyenda. Identificación del color de las celdas como valores numéricos.
10. Anterior. Botón que se activa si estamos visualizando un instante posterior al primer instante. Si se realiza una acción sobre éste se visualizará el instante anterior al visualizado actualmente.
11. Siguiente. Botón que se activa si estamos visualizando un instante anterior del último instante definido. Si se realiza una acción sobre éste se visualizará el instante posterior al visualizado actualmente.
12. Coste total. Se muestra el coste total de las intervenciones según las opciones indicadas.
13. Cerrar ventana. Cierra la ventana de simulación de intervenciones de usuario.
14. Guardar como Archivo inicial. Guarda el tablero en un archivo en formato Excel2003.
15. Salir. Finaliza la aplicación de modo que la cierra y libera los recursos.

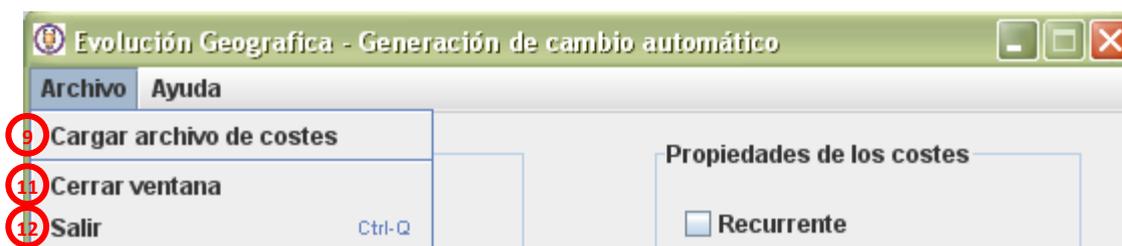
Cuando se ha realizado la simulación con las intervenciones introducidas por el usuario se podrá comparar con la ventana de simulación, donde se podrá apreciar los resultados que arrojan esas intervenciones y el coste asociado a estas.

1.7 Ventana de opciones de optimización

Esta ventana permite cambiar las opciones para la búsqueda de soluciones automáticas que permite la aplicación.



1. Menú Archivo. El menú Archivo tiene las siguientes opciones:



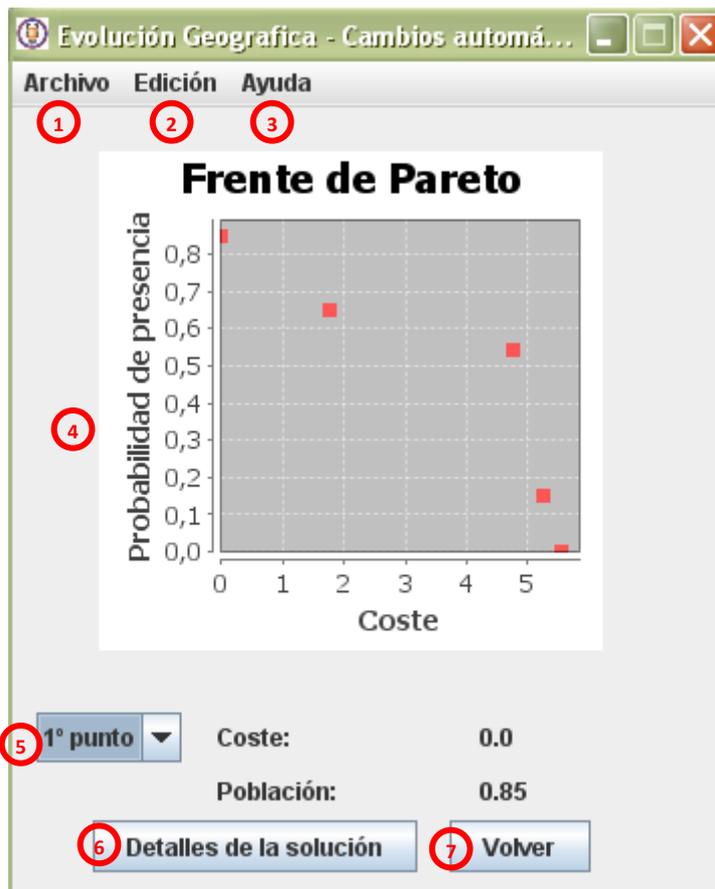
2. Menú Ayuda. Puede desplegarse de forma que aparezca este manual e información del programa.
3. Sección de propiedades de los cambios. Sección en la que se muestran las opciones modificables referentes a la optimización desde la perspectiva de las intervenciones.
4. Opción "Sostenido en el tiempo". Esta opción permite que una intervención dada en un instante determinado se aplique también para los instantes posteriores.

5. Objetivo. Se permite elegir la celda donde se pretende realizar la optimización. La población se muestra en el campo "Población actual" una vez introducida la fila y columna.
6. Opción de Maximizar / Minimizar. Se permite maximizar o minimizar en la optimización.
7. Sección propiedades de los costes. Sección en la que se muestra las opciones modificables referentes a la optimización desde la perspectiva de los costes.
8. Opción "Recurrente". Opción que determina si el coste se incrementa por cada instante en el que se mantenga el cambio. Esta opción solo puede seleccionarse si está activa la opción "el cambio se mantiene en el tiempo".
9. Establecer coste. Permite establecer el coste manualmente o mediante archivo cargando con el formato establecido (Excel 2003).
10. Optimizar. Acepta las opciones y realiza la optimización.
11. Cerrar ventana. Cierra la ventana de simulación de intervenciones de usuario.
12. Salir. Finaliza la aplicación de modo que la cierra y libera los recursos.

Cuando se han elegido las opciones para optimizar será posible realizar la optimización teniendo en cuenta los valores introducidos por el usuario.

1.8 Ventana de resultados de la optimización

Esta ventana presente los resultados de la optimización. Se presenta una gráfica en la que se presentan las soluciones dependiendo de dos valores: coste (eje X) y probabilidad de presencia de población (eje Y).



1. Menú Archivo. El menú Archivo tiene las siguientes opciones:



2. Menú Edición. El menú Edición tiene las siguientes opciones:



3. Menú Ayuda. Puede desplegarse de forma que aparezca este manual e información del programa.
4. Gráfico de soluciones. Representación de las soluciones teniendo en cuenta el coste de aplicar las intervenciones y el resultado que arrojan las intervenciones en términos de población final.
5. Solución destacada. Muestra la solución numéricamente de una solución elegida por el usuario.
6. Detalles de la solución. Indica todo el detalle de la solución, es decir, las intervenciones del tablero en el formato de simulación mediante una ventana nueva.
7. Cerrar ventana. Cierra la ventana de simulación de intervenciones de usuario.
8. Salir. Finaliza la aplicación de modo que la cierra y libera los recursos.

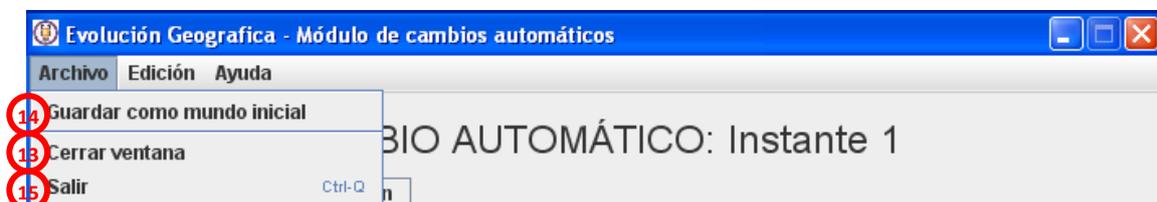
Cuando se ha realizado la simulación mediante el módulo de optimización automático se podrá elegir que opción necesita el usuario teniendo en cuenta las dos variables: coste de las intervenciones que está dispuesto a aceptar y probabilidad de población final.

1.9 Ventana de simulación de intervenciones de optimizador

Esta ventana de simulación tiene en cuenta las intervenciones automáticas encontradas mediante la aplicación:



1. Menú Archivo. El menú Archivo tiene las siguientes opciones:



2. Menú Edición. El menú Edición tiene las siguientes opciones:



3. Menú Ayuda. Puede desplegarse de forma que aparezca este manual e información del programa.
4. Título de la ventana de simulación de intervenciones automáticas. Indica el instante que se está visualizando, se modificará automáticamente al presionar los botones anterior o siguiente (ver 10 y 11 respectivamente).
5. Pestañas de resultados de los simuladores. Se podrá cambiar la vista en cualquier momento cambiando de pestaña. Las posibilidades son: mostrar los mapas de terrenos, mostrar el mapa de atractivo, mostrar el mapa de población.
6. Pestañas de terrenos. Se podrá cambiar la vista en cualquier momento cambiando de pestaña. El número de pestañas viene determinado por las variables definidas en el fichero de entrada.
7. Tablero. Resultado de la simulación con intervenciones introducidas por usuario, en la que los valores han sido transformados a códigos de colores para una fácil identificación, este código de colores se puede ver en la leyenda (ver 9). Al pasar el puntero del ratón por encima del tablero podremos identificar su valor y su posición (ver 8). No es posible modificar el valor como ocurre en el caso de la ventana de simulación.
8. Valores de 'Casilla' y 'Valor'. Valores que se actualizan automáticamente al pasar por encima del Tablero (ver 7) con el ratón. "Casilla" indica las coordenadas de ubicación de la casilla (filas : columnas) y "Valor" indica el valor de ésta casilla.
9. Leyenda. Identificación del color de las celdas como valores numéricos.
10. Anterior. Botón que se activa si estamos visualizando un instante posterior al primer instante. Si se realiza una acción sobre éste se visualizará el instante anterior al visualizado actualmente.
11. Siguiente. Botón que se activa si estamos visualizando un instante anterior del último instante definido. Si se realiza una acción sobre éste se visualizará el instante posterior al visualizado actualmente.
12. Coste total. Se muestra el coste total de las intervenciones según las opciones indicadas.
13. Cerrar ventana. Cierra la ventana de simulación de intervenciones automáticas.
14. Guardar como Mundo inicial. Guarda el tablero en un archivo en formato Excel2003.
15. Salir. Finaliza la aplicación de modo que la cierra y libera los recursos.

Cuando se ha realizado la simulación mediante el módulo de optimización automático se podrá comparar con la ventana de simulación, donde se podrá apreciar los resultados que arrojan esas intervenciones y el coste asociado a estas.