



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

**Sistema inmune artificial multi-objetivo basado en
un esquema alternativo de selección**

Tesis que presenta

Cinthya Karina Duplán Avilés

para obtener el Grado de

Maestro en Ciencias

en Computación

Director de la Tesis

Dr. Carlos Artemio Coello Coello

México, D.F.

Septiembre 2012

Resumen

Existe un gran número de problemas en el mundo real que requieren de la optimización de varios objetivos de manera simultánea. En esta clase de problemas, los algoritmos evolutivos han tenido gran aceptación, lo que ha dado pie a que otras metaheurísticas sean extendidas también para ser usadas en problemas multi-objetivo. La mayoría de los algoritmos evolutivos basan su esquema de selección en la dominancia de Pareto; sin embargo, se ha mostrado que este método pierde efectividad conforme se incrementa el número de objetivos. En esta tesis se propone un nuevo algoritmo multi-objetivo basado en la metaheurística denominada sistema inmune artificial. Dicho algoritmo cuenta con un mecanismo de selección basado en las medidas de desempeño: distancia generacional y distancia generacional invertida, buscando así una mejor escalabilidad que los sistemas inmunes artificiales y los algoritmos evolutivos multi-objetivo basados en dominancia de Pareto.

Abstract

There are many real-world problems that need the simultaneous optimization of multiple objectives. In this kind of problems, the evolutionary algorithms have had wide acceptance, which has given rise to extending other meta-heuristics to be used in multi-objective optimization problems. The selection scheme of most evolutionary algorithms is based on Pareto dominance, but this criterion has shown to be less effective as the number of objectives is increased. In this thesis, we propose a new multi-objective algorithm, which is based on the meta-heuristic called artificial immune system. The selection mechanism of this algorithm is based on the generational distance and inverted generational distance quality indicators with the aim of achieving a better scalability than both multi-objective artificial immune systems and multi-objective evolutionary algorithms based on Pareto dominance.

Agradecimientos

Primeramente deseo agradecer a mi director de tesis el Dr. Carlos A. Coello por guiarme en el desarrollo de este trabajo. Deseo agradecer también al Dr. Luis Gerardo de la Fraga y al Dr. Gregorio Toscano Pulido por sus valiosas observaciones y correcciones a esta tesis.

Agradezco a mis compañeros en el CINVESTAV por compartir conmigo esta aventura, y de manera muy especial a Saúl Zapotecas Martínez y Cynthia Adriana Rodríguez Villalobos por todos sus consejos.

Al CINVESTAV y al CONACyT les agradezco por permitirme estudiar una maestría en una institución tan respetable. Este trabajo de tesis se derivó del proyecto de investigación titulado “Escalabilidad y Nuevos Esquemas Híbridos en Optimización Evolutiva Multiobjetivo” (Ref. 103570), cuyo responsable es el Dr. Carlos A. Coello Coello.

Sobre todo deseo agradecer a mi familia: a mis padres y hermanos por todo su apoyo y cariño y a Heriberto por su amor incondicional.

Índice general

Resumen	III
Abstract	V
Agradecimientos	VII
Índice de figuras	XII
Índice de tablas	XIII
1. Introducción	1
1.1. Descripción del problema	2
1.2. Estructura del documento	2
2. Sistemas inmunes artificiales	5
2.1. Fundamentos del sistema inmune	5
2.1.1. Inmunidad innata y adaptativa	8
2.1.2. Principales modelos de inmunología	10
2.2. Historia de los sistemas inmunes artificiales	15
2.2.1. Modelos de sistemas inmunes artificiales	17
2.2.2. Aplicaciones de sistemas inmunes artificiales	22
3. Optimización multi-objetivo	25
3.1. Fundamentos de optimización multi-objetivo	26
3.2. Fundamentos de computación evolutiva	30
3.3. Algoritmos de optimización multi-objetivo	34
3.3.1. Enfoques agregativos	35
3.3.2. Enfoques no agregativos	35
3.3.3. Algoritmos evolutivos multi-objetivo	37
3.3.4. Sistemas inmunes artificiales multi-objetivo	40
4. Modelo propuesto	45
4.1. Descripción del algoritmo general	45
4.2. Aportación de una solución a los indicadores DG y DGI en SIAMO-DGI	49
4.3. Reducción de la memoria inmunológica	50

4.3.1. Reducción mediante distancia de Mahalanobis	50
4.3.2. Algoritmo de agrupamiento	51
5. Validación	55
5.1. Conjunto de problemas ZDT (Zitzler-Deb-Thiele)	55
5.2. Conjunto de problemas DTLZ (Deb-Thiele-Laumanns-Zitzler)	58
5.3. Desempeño de nuestra propuesta con muchos objetivos	62
6. Conclusiones	65
6.1. Trabajo a futuro	66
A. Funciones de Prueba	67
A.1. Conjunto de problemas ZDT (Zitzler-Deb-Thiele)	67
A.2. Conjunto de problemas DTLZ (Deb-Thiele-Laumanns-Zitzler)	70
Bibliografía	77

Índice de figuras

2.1. Arquitectura multicapa del sistema inmune	6
2.2. Principales actores del sistema inmune	7
2.3. Anatomía del sistema inmune	7
2.4. Respuesta inmune	11
2.5. Selección clonal	12
2.6. Principales tipos de interacción entre un linfocito y un antígeno	13
2.7. Ilustración de la teoría del peligro	14
2.8. Teoría de la red inmune	15
2.9. Algoritmo general de selección negativa	18
2.10. Algoritmo general de redes inmunes artificiales	20
2.11. Algoritmo general de selección clonal	21
2.12. Algoritmo general de la teoría del peligro	23
3.1. Óptimo de Edgeworth-Pareto	27
3.2. Representación gráfica del calculo del hipervolumen	28
3.3. Ilustración del indicador I_c para un POM en dos dimensiones	29
3.4. Ilustración de I_{HD} para un POM en dos dimensiones	30
3.5. Esquema general del funcionamiento de un algoritmo evolutivo	31
3.6. Algoritmo general de la programación evolutiva	32
3.7. Algoritmo general de la estrategia evolutiva (1 + 1)	33
3.8. Esquema general de un algoritmo genético	34
3.9. Rejilla adaptativa en dos dimensiones usada por el algoritmo PAES	38
3.10. Ilustración de distancia de crowding	39
3.11. Algoritmo general de un sistema inmune artificial multi-objetivo	42
3.12. Afinidad antígeno-anticuerpo en MISA	43
4.1. Algoritmo propuesto	46
4.2. Calidad de una solución en SIAMO-DGI	50
4.3. Cúmulos estrictos vs difusos	51
4.4. Ejemplo de agrupamiento jerárquico	53
4.5. Ejemplo de particionamiento	54
5.1. Desempeño de HYPE en ZDT1	56
5.2. Grafica de tiempos de ejecución	64

A.1. Frente de Pareto verdadero de ZDT1	68
A.2. Frente de Pareto verdadero de ZDT2	68
A.3. Frente de Pareto verdadero de ZDT3	69
A.4. Frente de Pareto verdadero de ZDT4	69
A.5. Frente de Pareto verdadero de ZDT6	70
A.6. Frente de Pareto verdadero de DTLZ1	71
A.7. Frente de Pareto verdadero de DTLZ2	72
A.8. Frente de Pareto verdadero de DTLZ3	73
A.9. Frente de Pareto verdadero de DTLZ4	73
A.10. Frente de Pareto verdadero de DTLZ5	74
A.11. Frente de Pareto verdadero de DTLZ6	75
A.12. Frente de Pareto verdadero de DTLZ7	76

Índice de tablas

2.1. Características de las respuestas inmune innata y adaptativa	10
5.1. Configuración de parámetros	55
5.2. Resultados ZDT	58
5.3. Configuración de parámetros	59
5.4. Resultados DTLZ	62
5.5. Configuración de parámetros	62
5.6. Resultados de escalabilidad	64

Capítulo 1

Introducción

Una gran variedad de problemas del mundo real requieren de la optimización de varios objetivos (generalmente en conflicto) de manera simultánea. Imaginemos por ejemplo, el diseño de un avión: se desea que éste sea capaz de alcanzar altas velocidades, pero no es el único criterio a tomar en cuenta, pues debe considerarse también el ahorro de combustible. La velocidad y el ahorro son entonces dos objetivos en conflicto que desean optimizarse simultáneamente: un aumento de la velocidad traería consigo una disminución del ahorro. En este caso, será necesario encontrar una solución que produzca valores aceptables a los dos objetivos desde el punto de vista del diseñador, es decir que permita un buen compromiso entre velocidad y ahorro.

A diferencia de los problemas con un solo objetivo, el concepto de óptimo en los problemas de optimización multi-objetivo es relativo, por lo que es necesario decidir de alguna forma cual o cuales son las mejores soluciones al problema y la optimilidad de Pareto es el criterio que suele seguirse en estos casos [1]. Sin embargo, se ha demostrado que la optimilidad de Pareto pierde efectividad conforme se aumenta el número de objetivos [2] por lo que ha sido necesario pensar en criterios alternativos.

Los sistemas inmunes artificiales (SIAs), inspirados en el sistema inmune biológico, han sido utilizados para resolver de manera exitosa una gran diversidad de problemas, incluyendo los problemas de optimización [3].

En esta tesis se propone un nuevo sistema inmune artificial multi-objetivo inspirado en la selección clonal. En este algoritmo, el criterio de optimilidad de Pareto es sustituido por la aportación de las soluciones con respecto a los indicadores llamados distancia generacional y distancia generacional invertida.

El algoritmo propuesto (sistema inmune artificial multi-objetivo basado en distancia generacional invertida: SIAMO-DGI) es comparado con dos algoritmos evolutivos multi-objetivo el Nondominated Sorting Genetic Algorithm NSGA-II [4]: basado en la dominancia de Pareto y el Algorithm for Fast Hypervolume-Based Many-Objective Optimization HYPE [5]: basado en el indicador de hipervolumen y con un sistema

inmune artificial multi-objetivo el Vector Artificial Immune System VAIS [6]: basado en dominancia de Pareto del estado del arte. En las pruebas realizadas SIAMO-DGI mostró ser un buen compromiso entre calidad del frente encontrado y el tiempo de ejecución para problemas con muchos objetivos.

1.1. Descripción del problema

La mayoría de los algoritmos evolutivos multi-objetivo (AEMOs) utilizan esquemas de selección basados en la dominancia de Pareto. Sin embargo, se ha mostrado que este esquema no se comporta bien en presencia de muchos objetivos (más de 4) [2]. Una alternativa a la dominancia de Pareto es el uso de indicadores (medidas de desempeño), para el mecanismo de selección de un AEMO y ese es precisamente el enfoque adoptado en esta tesis.

Objetivos generales y específicos del proyecto

General

El objetivo general de esta tesis es proponer un nuevo SIA multi-objetivo cuyo esquema de selección esté basado en los indicadores llamados distancia generacional y distancia generacional invertida.

Particulares

- Respetar la metáfora del sistema inmune. El sistema propuesto estará inspirado por los principios, modelos y funciones del sistema inmune biológico.
- Probar el SIA propuesto con un conjunto amplio de problemas.
- El SIA implementado deberá ser competitivo con algoritmos multi-objetivo actuales.

1.2. Estructura del documento

En el capítulo 2 se proporcionan conceptos y modelos teóricos de inmunología básica en los que se inspiró el sistema inmune artificial multi-objetivo propuesto en este trabajo de tesis. También se da un breve repaso del estado actual del área de sistemas inmunes artificiales.

En el capítulo 3 se presentan algunos conceptos básicos de optimización y computación evolutiva, necesarios para entender el resto de esta tesis. Se da también una

visión general de los algoritmos de optimización multi-objetivo del estado del arte.

En el capítulo 4 se explica a detalle el sistema inmune artificial multi-objetivo propuesto.

El capítulo 5 muestra los resultados obtenidos en un amplio conjunto de problemas y las comparaciones con otros algoritmos multi-objetivo del estado del arte.

El capítulo 6 presenta las conclusiones, así como las mejoras y extensiones que podrían realizarse a este trabajo de tesis, con base en los resultados obtenidos.

Capítulo 2

Sistemas inmunes artificiales

Nunes de Castro y Timmis [7] nos proporcionan la siguiente definición de sistema inmune artificial:

“Los sistemas inmunes artificiales (SIAs) son sistemas adaptativos, inspirados por la inmunología teórica y las funciones, principios y modelos inmunes observados, los cuales son aplicados a resolver problemas”.

De la definición anterior resalta la importancia de hablar sobre el sistema inmune biológico. De tal forma, en la primera parte de este capítulo se proporcionan conceptos básicos de inmunología teórica, así como los principales modelos inmunes que existen. En la segunda parte de este capítulo se habla de la manera en que la metáfora del sistema inmune biológico ha inspirado el diseño de sistemas en diversas áreas, dando paso a los denominados sistemas inmunes artificiales.

2.1. Fundamentos del sistema inmune

Los seres vivos estamos expuestos constantemente a millones de agentes patógenos como bacterias, hongos, parásitos y virus, con potencial para dañarnos e incluso para matarnos. El sistema inmune es la respuesta de la evolución contra estas amenazas. El sistema inmune puede verse como una arquitectura de capas, con mecanismos de regulación y defensa actuando en diferentes niveles. La figura 2.1 muestra una descripción gráfica de dicha arquitectura. Las capas de protección pueden ser divididas de la siguiente manera [7]:

- **Barreras físicas:** nuestra piel sirve como un escudo contra invasores, tanto dañinos como no dañinos. El sistema respiratorio también nos ayuda a mantener infecciones potenciales fuera de nuestro organismo. Este mecanismo de defensa incluye la captura de invasores en las mucosas y vellos nasales, por ejemplo mediante estornudos. La piel y las membranas mucosas que recubren las vías

respiratorias y digestivas también contienen macrófagos y anticuerpos capaces de eliminar microorganismos dañinos.

- **Barreras bioquímicas:** fluidos como la saliva, el sudor y las lágrimas contienen enzimas destructivas. Los ácidos estomacales terminan con la mayoría de los microorganismos ingeridos al comer y beber. Los niveles de PH y la temperatura del cuerpo presentan condiciones de vida y reproducción desfavorables para algunos invasores.
- **Sistema inmune innato y sistema inmune adaptativo:** Constituyen las dos líneas de defensa principales del sistema inmune. Las diferencias entre estas dos líneas de defensa además del modo en que interactúan, se discute en más detalle en la sección 2.1.1.

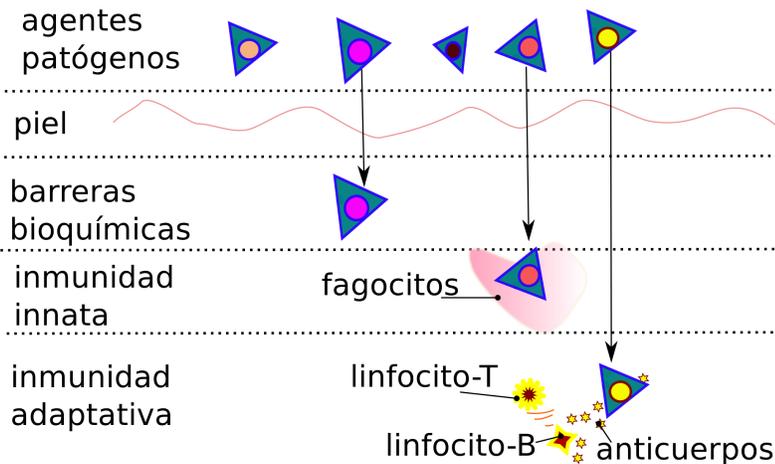


Figura 2.1: Arquitectura multicapa del sistema inmune

El sistema inmune está constituido por un complejo conjunto de células y moléculas encargadas de mediar la respuesta de nuestro cuerpo contra las infecciones. Tanto el sistema inmune innato como el adaptativo dependen en gran parte de los glóbulos blancos. La inmunidad innata es mediada principalmente por fagocitos capaces de ingerir y digerir una gran diversidad de microorganismos, mientras que la inmunidad adaptativa es mediada principalmente por linfocitos capaces de atacar a microorganismos concretos. Esto es mostrado en la figura 2.2.

Son varios los órganos distribuidos alrededor del cuerpo que permiten el funcionamiento del sistema inmune. Estos, órganos llamados órganos linfoides, son responsables de la producción, crecimiento y maduración de los linfocitos. En los órganos linfoides, los linfocitos interactúan con diferentes tipos de células, ya sea durante su producción o durante el inicio de una respuesta inmune. Los órganos linfoides pueden dividirse en primarios o centrales y en secundarios o periféricos. La figura 2.3 muestra

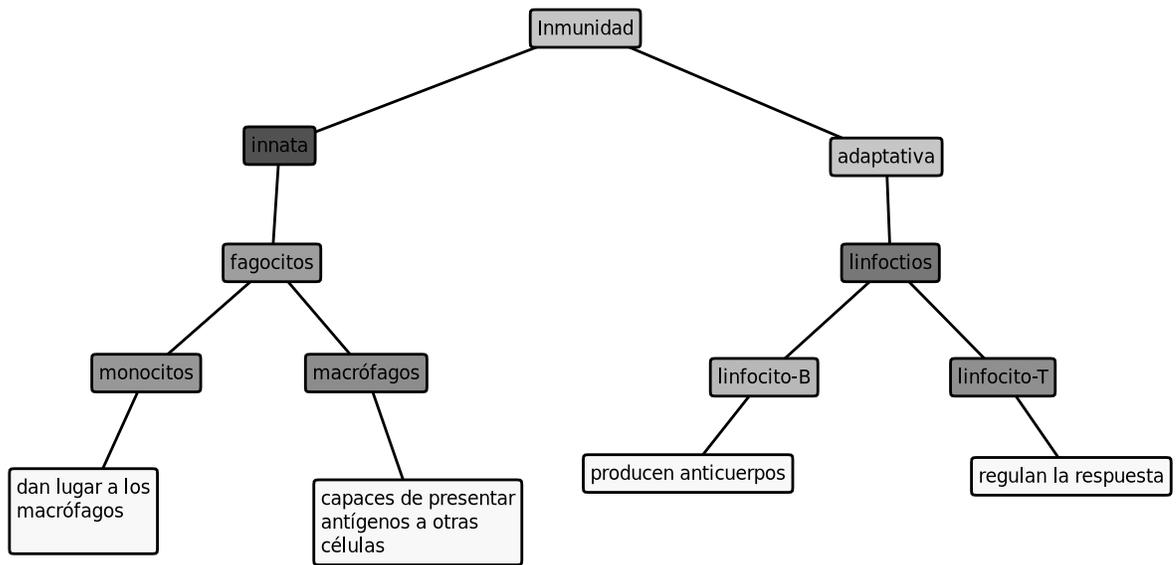


Figura 2.2: Principales actores del sistema inmune

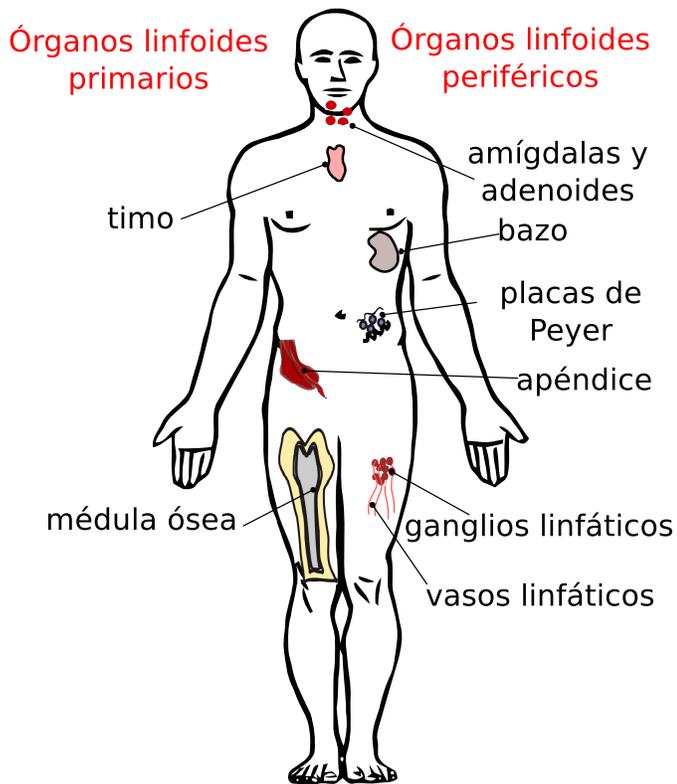


Figura 2.3: Anatomía del sistema inmune

los órganos linfoides localizados en el cuerpo.

Los principales órganos linfoides y sus funciones son los siguientes:

- **Primarios:** son los responsables de la producción y maduración de los linfocitos.
 - Médula ósea: es el sitio más importante de hematopoyesis (producción de células sanguíneas, incluyendo los glóbulos rojos, monocitos, granulocitos, linfocitos y plaquetas). Provee el ambiente necesario para la diferenciación de linfocitos-B.
 - La glándula timo: provee el ambiente necesario para la diferenciación de linfocitos-T, además de factores hormonales importantes para la maduración de estos linfocitos.
- **Secundarios:** en estos los linfocitos son estimulados por el agente patógeno, dando inicio a la respuesta inmune adaptativa.
 - Las glándulas Amígdalas y adenoides: esta colección de células linfoides son principalmente asociadas con la protección del sistema respiratorio.
 - Ganglios linfáticos: es en éstos donde ocurre la respuesta inmune adaptativa.
 - Apéndice y placas de Peyer: contienen células inmunes especializadas para proteger al sistema digestivo.
 - El órgano bazo: constituye el sitio donde los linfocitos pelean contra los microorganismos que invaden el flujo sanguíneo.
 - Vasos linfáticos: redes de canales alrededor del cuerpo, encargados de transportar el fluido celular a la sangre y a los órganos linfoides.

2.1.1. Inmunidad innata y adaptativa

El sistema inmune cuenta con dos respuestas para defender al organismo de infecciones: la respuesta inmune innata y la respuesta inmune adaptativa.

La inmunidad innata [8] se conforma por las primeras defensas con las que se enfrentan los agentes patógenos al intentar dañar al organismo y constituye una respuesta menos evolucionada y específica que la adaptativa. Esta respuesta es compensada por su rapidez y por el hecho de que no necesita un contacto previo con el agente como lo hace el sistema inmune adaptativo, quien actúa sólo si la primera línea de defensa es superada.

Una reacción inmune se divide en dos fases: (1) de reconocimiento y (2) efectora o de respuesta. De tal forma, para que los agentes sean eliminados, primero deben ser reconocidos. Esto se logra mediante la identificación de moléculas que le son propias.

Los sistemas de reconocimiento de la inmunidad innata son capaces de identificar moléculas comunes a multitud de patógenos. Mientras que la inmunidad adaptativa es capaz de diferenciar moléculas muy similares propias de agentes concretos, a los que se denomina antígenos.

En exposiciones repetidas al mismo agente, la inmunidad innata responderá de manera similar sin importar el número de veces que haya defendido al organismo de dicho agente; es decir carece de memoria inmunológica. No así la inmunidad adaptativa que, por lo general, responderá cada vez de mejor manera, tanto en rapidez como en intensidad.

Los responsables de reconocer y eliminar agentes patógenos en el sistema inmune adaptativo de los vertebrados son los linfocitos, principalmente los tipos B (B-cell) y T (T-cell). Los linfocitos T no pueden aniquilar al agente. Su papel es el de regular la defensa del sistema inmune adaptativo, al liberar, una vez que son activados, citoquinas y otras señales que provocan un aumento en la actividad de diferentes células del sistema inmune adaptativo capaces de eliminar al antígeno. Por ejemplo, los linfocitos-B, que generan anticuerpos al ser activados y suelen sufrir hiper-mutaciones durante la proliferación lo que puede provocar un incremento o decremento en la calidad de los anticuerpos que generan. La capacidad de un linfocito-B para combatir a un antígeno es conocida como afinidad anticuerpo-antígeno, mientras que la similitud entre dos linfocitos-B es conocida como afinidad anticuerpo-anticuerpo.

Aunque el objetivo tanto de la inmunidad innata como de la adaptativa consiste en defender al organismo frente a elementos considerados potencialmente agresivos, cada una posee características particulares. En la tabla 2.1 se listan las diferencias entre la respuesta innata y la respuesta adaptativa.

Inmunidad innata (primera línea de defensa)	Inmunidad adaptativa (segunda línea de defensa)
Innata. Se nace con las células y moléculas que conforman esta respuesta.	Adaptativa. Aunque se nace con el repertorio de células capaces de responder al antígeno, éstas no alcanzan un número suficientemente elevado hasta que no se enfrentan a él.
Inespecífica. Los receptores que reconocen a muchas de las moléculas están sobre una misma célula.	Específica. Cada célula posee sólo receptores que reconocen a un antígeno concreto.
Independiente del antígeno. Las células y moléculas están preparadas para la defensa desde antes de enfrentarse con el agente patógeno.	Dependiente del antígeno. Los linfocitos con un receptor concreto proliferan cuando su receptor reacciona con el antígeno, adquiriendo sólo entonces funciones efectoras.

Inespecífica del antígeno. No discrimina entre agentes próximos o incluso alejados.	Específica del antígeno. Ante la llegada del antígeno sólo proliferan los linfocitos que poseen el receptor específico para el mismo.
Respuesta máxima inmediata. Como los elementos y mecanismos están siempre presentes, se ponen en marcha de forma inmediata.	Respuesta máxima retardada o con fase de latencia. Los efectos de esta inmunidad no se notan hasta que las células no han proliferado y alcanzado un número elevado que garantice su éxito.
Sin memoria inmunológica. La respuesta es similar independientemente del número de veces que se haya respondido a un agente concreto previamente.	Con memoria inmunológica. La respuesta es por lo general muy superior (en rapidez e intensidad) en exposiciones sucesivas al mismo antígeno.

Tabla 2.1: Características de las respuestas inmune innata y adaptativa

La manera en que el sistema inmune responde a la amenaza de un agente patógeno es ilustrada en la figura 2.4. En la imagen, diferentes virus (representados por triángulos) son detenidos en su intento por dañar al organismo, en primera instancia por la inmunidad innata. Cuando esta primera línea de defensa es superada es el sistema inmune adaptativo quien responde. Los linfocitos-T que reconocen al antígeno (representado por un círculo) propio del virus que entró al organismo, comienzan a liberar señales que aumentan la actividad de los linfocitos-B afines a dicho antígeno. Estos a su vez generan anticuerpos (representados por estrellas) que aniquilarán al virus, sanando al organismo.

2.1.2. Principales modelos de inmunología

Los modelos de inmunología tratan de explicar los mecanismos fisiológicos de defensa de la integridad biológica de un organismo. En esta sección se exponen algunos de los modelos más importantes de inmunología:

- Selección clonal.
- Selección negativa.
- Teoría del peligro.
- Teoría de la red inmune.

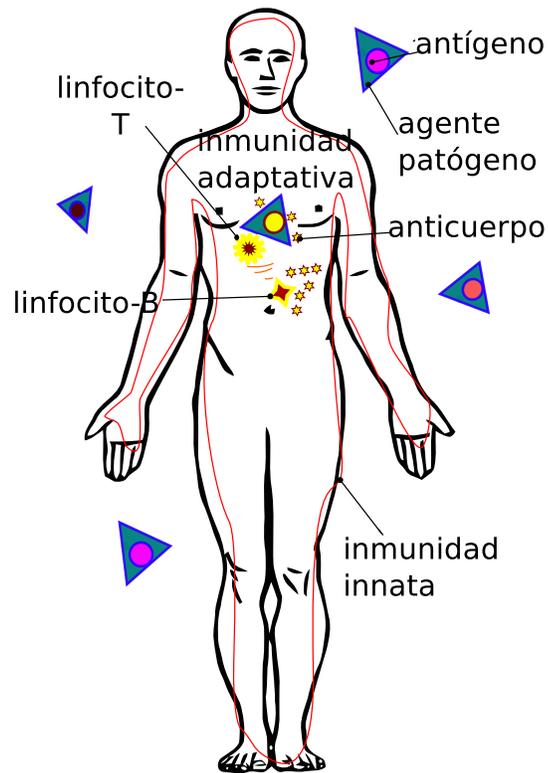


Figura 2.4: Respuesta inmune

Selección clonal

La selección clonal [9] explica la manera en que el sistema inmune adaptativo responde al estímulo de un antígeno. Establece la idea de que sólo las células que reconocen a dicho antígeno como extraño pueden proliferarse y tomar parte en la etapa efectora, produciendo grandes volúmenes de células que se denominan clones. Así mismo, los antígenos reconocidos como propios serán tolerados.

En la proliferación, la reproducción se da de manera asexual. Un antígeno provoca la activación de varias células; sin embargo, el número de descendientes de cada célula es proporcional a su afinidad con respecto al antígeno. Las células que mayor afinidad tienen con el antígeno tendrán un mayor número de clones.

Los clones sufren una hipermutación somática; estos cambios genéticos son expresados en los anticuerpos que producen. La mutación sufrida es inversamente proporcional a la afinidad que presenta la célula con el antígeno: los clones con mayor afinidad, sufren un menor grado de mutación. Este proceso es responsable de generar y mantener un repertorio diverso de anticuerpos.

Los clones serán sometidos a una fuerte presión de selección lo que dará como

resultado un incremento en la afinidad hacia el antígeno. Este proceso de mutación y selección es conocido como maduración de la afinidad de la respuesta inmune.

La memoria inmunológica está constituida por aquellas células que mostraron mayor efectividad contra el antígeno. Con el objetivo de desarrollar una respuesta más efectiva contra el agente patógeno en caso de que éste llegara a atacar nuevamente, las células de memoria tienen un mayor tiempo de vida.

El principio de selección clonal es ilustrado en la figura 2.5. Ante la presencia de un antígeno-extraño, son seleccionados para su proliferación los linfocitos que mejor reconocen a dicho antígeno. Después de la proliferación, se almacenan en la memoria inmunológica aquellas células que mostraron mayor afinidad con él, previniendo futuras invasiones.

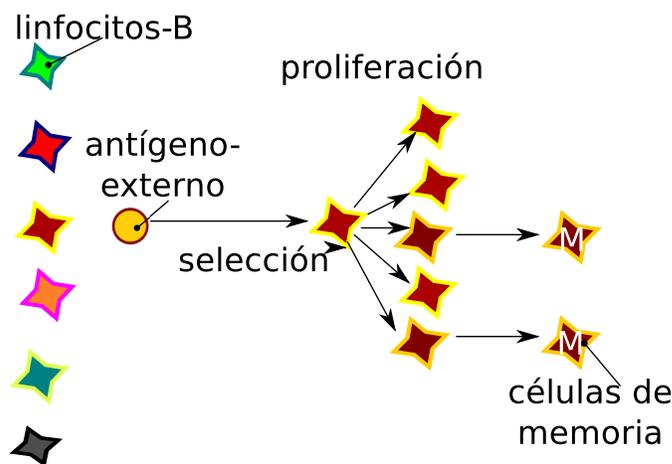


Figura 2.5: Selección clonal

Selección negativa

Un proceso importante dentro del sistema inmune es poder reconocer a sus propias células de las células extrañas. Este proceso permite construir una defensa contra ataques externos en vez de auto destruirse.

La selección negativa [10] es responsable de la inhibición o supresión de los linfocitos que son sobreactivados (producen anticuerpos contra antígenos propios), y que eventualmente podrían causar enfermedades auto inmunes.

Los linfocitos atraviesan un proceso de selección para asegurar el reconocimiento de antígenos no propios. Este proceso cuenta con dos fases: la selección negativa y la selección positiva. Durante la selección positiva, se descartan aquellos linfocitos que no son capaces de reconocer a los antígenos externos. El propósito de la selección negativa, es por tanto, evaluar la tolerancia de las células propias: se descartan

los linfocitos que reconocen antígenos propios. Así, la selección negativa puede verse como un filtro donde sólo se mantienen aquellos linfocitos que no atacan a las células propias.

Algunos de los resultados del encuentro entre un linfocito y un antígeno son ilustrados en la figura 2.6 y listados a continuación:

- **Expansión clonal:** reconocimiento de antígenos externos en la presencia de señales estimuladoras.
- **Selección positiva:** reconocimiento de antígenos externos por linfocitos maduros.
- **Selección negativa:** reconocimiento de antígenos propios en los órganos linfoides. Reconocimiento de antígenos propios en la ausencia de señales estimuladoras.
- **Tolerancia:** todas las circunstancias en las cuales una célula potencialmente activa, falla al reaccionar contra un antígeno; por ejemplo si la concentración del antígeno o la afinidad del linfocito con el antígeno es muy baja.
- **Supresión clonal:** inhibición de la reacción inmunológica en marcha
- **Anergia:** proceso en el que las células autorreactivas no proliferan en respuesta a la presentación del antígeno.

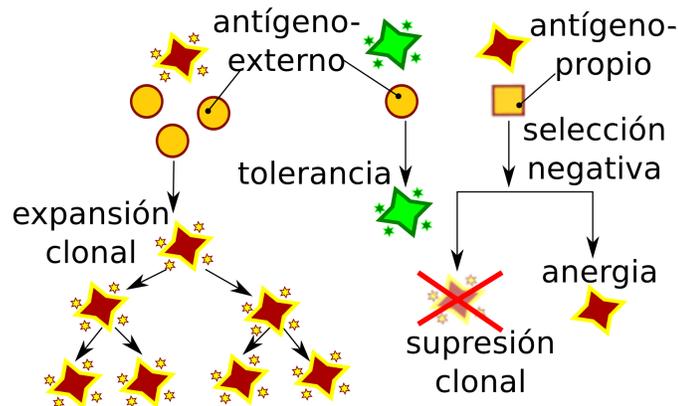


Figura 2.6: Principales tipos de interacción entre un linfocito y un antígeno

Teoría del peligro

La discriminación entre lo propio de lo extraño (selección clonal), ha sido sustituida en la teoría del peligro [11], la cual plantea que el sistema inmune responde a señales de alarma o peligro independientemente del origen del antígeno. Las células dendríticas captan estas señales y adecuan su respuesta en función del tipo de

señal que reciben. Si la señal es de peligro o alarma, la respuesta será de resistencia; si la señal no es de peligro, la respuesta será de tolerancia.

Cuando una amenaza ataca al organismo, las células y tejidos no dañados son los responsables de transmitir señales que producen la captura de antígenos por parte de las células presentadoras de antígenos. Luego éstas viajan al ganglio linfático local y presentan el antígeno a los linfocitos.

La zona de peligro existe alrededor de cada señal de peligro (en la figura 2.7 se ejemplifica gráficamente esta teoría). Por lo tanto, sólo los linfocitos-B cuyos anticuerpos coincidan con los antígenos de la zona de peligro, serán estimulados y proliferados.

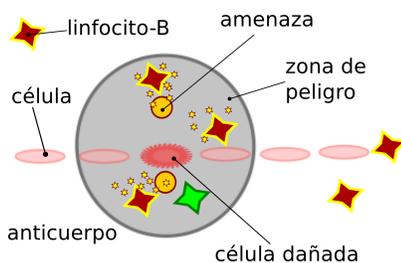


Figura 2.7: Ilustración de la teoría del peligro

Teoría de la red inmune

La teoría de la red inmune [12] fue propuesta para intentar explicar cómo se forma la memoria inmunológica. Esta teoría nos dice que el sistema inmune trabaja como una red regulada de anticuerpos y anti-anticuerpos llamada red idiotípica. Los integrantes de dicha red se reconocen unos con otros, aún en la ausencia de antígenos, en vez de ser un conjunto aislado de clones que responden sólo cuando son estimulados por los antígenos.

Según esta teoría, los linfocitos T y B forman un circuito complejo de células que interactúan y cuya función puede ser estimular o suprimir la respuesta inmune.

La porción sobre la superficie de un antígeno que es capaz de ser reconocida por un anticuerpo se denomina epítipo. Un antígeno puede contener varios epítipos. La parte de un anticuerpo usada para reconocer al antígeno se llama parátipo. El epítipo de un anticuerpo se denomina idiotipo.

La teoría de la red inmune es ilustrada en la figura 2.8. Los linfocitos-B, al ser estimulados por un antígeno determinado, producen anticuerpos. El anticuerpo contiene un sitio de combinación antigénico que a su vez induce la formación de anticuerpos anti-idiotípicos similares al antígeno (imagen espejo), lo que posibilita la obtención

de vacunas en ausencia del antígeno original. El sistema inmune es dinámico y continúa produciendo anti-anticuerpos anti-idiotípicos hasta transformarse en una red de interacciones idiotípicas que persisten mucho tiempo después de la metabolización del antígeno. Para el funcionamiento normal del sistema inmune se debe llegar a un equilibrio entre estímulos y supresiones, lo que provoca que la red de anticuerpos idiotípicos sea eventualmente neutralizada y suprimida.

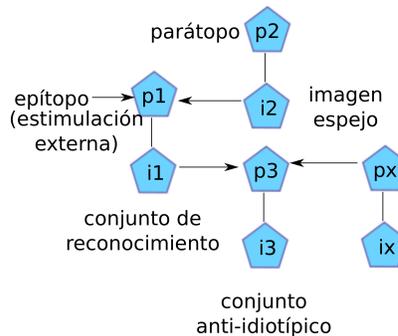


Figura 2.8: Teoría de la red inmune

2.2. Historia de los sistemas inmunes artificiales

Los sistemas inmunes artificiales (SIAs) son un área de investigación que engloba las disciplinas de inmunología, ciencias de la computación e ingeniería. Las investigaciones sobre los SIAs pueden ser divididas en modelos de inmunología, SIAs teóricos y SIAs aplicados [13]. Los primeros comprenden trabajos de investigación sobre modelos detallados y simulaciones de sistemas inmunes naturales y artificiales. Mientras que los trabajos teóricos sobre SIAs [14] incluyen modelos matemáticos de algoritmos, y análisis de convergencia, desempeño y complejidad de dichos algoritmos. Por último, los trabajos en SIAs aplicados [3], van desde el desarrollo e implementación de algoritmos inmuno-inspirados, hasta la aplicación de los SIAs a diversos problemas del mundo real.

En la última década, el sistema inmune se ha perfilado como una importante fuente de inspiración potencial para el desarrollo de nuevos enfoques en la resolución de problemas computacionales complejos. Su arquitectura altamente distribuida, aprendizaje adaptativo y auto-organización natural, aunado a su capacidad de memoria, extracción de características y reconocimiento de patrones, nos ofrece una rica metáfora para el diseño de sistemas artificiales.

La comunidad de SIAs ha producido un fecundo conjunto de algoritmos inmuno-inspirados gracias a la extracción de mecanismos útiles de las teorías, procesos y elementos del sistema inmune. Esto ha dado pie a la producción de un vasto cuerpo de literatura en esta área, incluyendo libros de texto de SIAs y una colección importante

de artículos sobre aplicaciones exitosas [15], además de diversas conferencias internacionales.

El libro *Sistemas Inmunes Artificiales: Un Nuevo Enfoque de la Inteligencia Computacional* [7] provee una introducción accesible a quienes se inician en el estudio de esta área. Da una definición clara de los SIAs y sus algoritmos básicos, y analiza la manera en la que el sistema inmune se relaciona a sistemas y procesos biológicos. No es necesario contar con conocimientos previos de inmunología para leerlo, pues los fundamentos esenciales de dicha área son cubiertos en los primeros capítulos.

Computación inmunológica: Principios y Aplicaciones [16] introduce el término computación inmunológica (CI) como un nuevo enfoque computacional. Establece una rigurosa base matemática de los descubrimientos recientes de inmunología y también presenta una serie de aplicaciones de CI utilizadas exitosamente para resolver problemas de la vida real. Analiza y discute los aspectos de la aplicación en hardware del concepto de CI en una inmuno-computadora, como un nuevo medio informático y sus posibles conexiones con modernos *microchips biológicos* y futuras computadoras biomoleculares. Este libro está dirigido a expertos en ciencias de la computación, inteligencia artificial y computación biomolecular.

El libro *Sistemas inmuno-inspirados* [17] presenta un marco para la comunicación de agentes autónomos inspirados en el sistema inmunológico. Explora las redes auto-organizadas basadas en el modelo de la red inmune. Este libro muestra también varias aplicaciones y simulaciones de SIAs.

Más recientemente, el libro *Computación inmunológica: Teoría y Aplicaciones* [18] provee una visión general de los conceptos fundamentales de inmunología y algunos modelos teóricos de los procesos inmunes. Presenta las técnicas computacionales inmuno-inspiradas que están siendo desarrolladas por grupos de investigación de SIAs actuales. Da una revisión de representaciones de datos, medidas de afinidad, reglas de correspondencia, y abstracciones generales de algunos elementos y procesos que son usados en la mayoría de los SIAs.

La Conferencia internacional de sistemas inmunes artificiales (ICARIS por las siglas en inglés: International Conference on Artificial Immune Systems), constituye la conferencia especializada en SIAs más importante. Es celebrada anualmente desde el 2002 y provee una gran oportunidad para la presentación y difusión de los últimos trabajos en SIAs. También suelen aparecer trabajos de SIAs en otras conferencias internacionales, como la Conferencia de computación genética y evolutiva (GECCO por las siglas en inglés: Genetic and Evolutionary Computation Conference), el Congreso mundial IEEE de inteligencia computacional (IEEE WCCI por las siglas en inglés: IEEE World Congress on Computational Intelligence) y la Conferencia internacional de inteligencia artificial y cómputo suave (ICAISC por las siglas en inglés: Interna-

tional Conference on Artificial Intelligence and Soft Computing).

2.2.1. Modelos de sistemas inmunes artificiales

Los algoritmos inmuno-inspirados que han sido constantemente desarrollados, obteniendo popularidad son:

- Algoritmos de selección negativa (ASN)
- Redes inmunes artificiales (RIA)
- Algoritmos de selección clonal (ASC)
- Algoritmos de la teoría del peligro (ATP)

Estos algoritmos son tratados en esta sección.

Algoritmos de selección negativa

En [19] se propuso un método llamado “algoritmo de selección negativa” para distinguir lo propio de lo externo, basado en la generación de linfocitos-T en el sistema inmune (ver sección 2.1.2). Este método fue aplicado primeramente a la detección de virus informáticos. Desde la publicación de dicho artículo, los algoritmos de selección negativa han atraído la atención de numerosas investigaciones y han sido implementados en diversas aplicaciones del mundo real.

La figura 2.9 describe los pasos principales del algoritmo:

- **Etapas de generación:** los detectores son inicializados por un proceso aleatorio y eliminados cuando tienen una alta afinidad con las muestras propias. Los candidatos restantes se mantienen como detectores.
- **Etapas de detección:** el conjunto de detectores generado en la etapa anterior es utilizado para comprobar si una instancia de datos de entrada es propia o no. Si ésta coincide con cualquiera de los detectores, se identifica como una anomalía.

Con los años, una gran cantidad de variaciones de este algoritmo han sido propuestas; sin embargo, las características principales de la concepción original todavía permanecen [20]. Estas características incluyen:

- **Representación negativa de la información:** la salida en este tipo de sistemas suele ser el concepto complementario del objetivo real. Aunque es obviamente distinta de cualquier mecanismo de aprendizaje en el espacio positivo, su fuerza y aplicación están aún en exploración y sujetos a controversia.

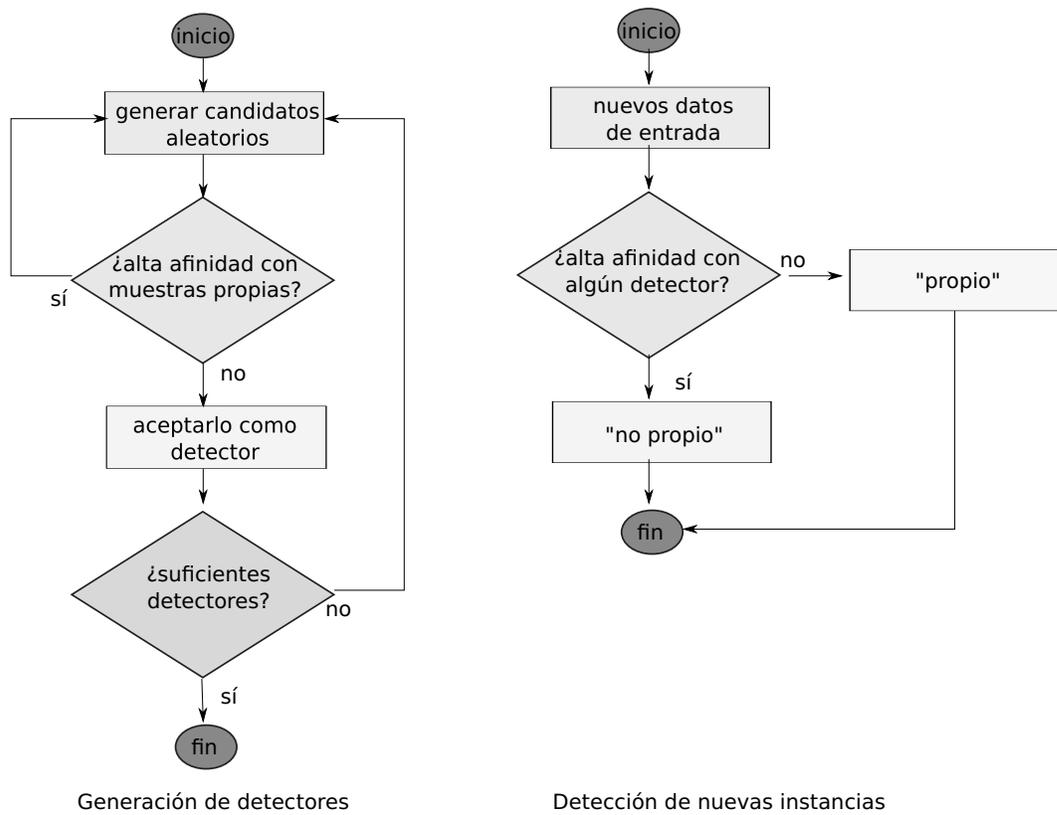


Figura 2.9: Algoritmo general de selección negativa

- **Uso de algún conjunto de detectores como mecanismo de detección:** esta característica ofrece posibilidades de extender este método a un entorno distribuido.
- **Clasificación de una sola clase:** aunque el objetivo es la discriminación entre dos clases, solamente están disponibles para entrenar al sistema las muestras de una clase (propia). Hay algunos trabajos que tratan de generalizar este método para la clasificación de dos clases [21] o de clases múltiples [22].

La representación de datos es uno de los principales aspectos de los algoritmos de selección negativa; por lo general, se utilizan vectores de cadenas o representación real. Por último, un algoritmo de SN se distingue por una determinada regla de correspondencia, que normalmente se basa en una distancia o medida de “similitud”.

Redes inmunes artificiales

Las redes inmunes artificiales (RIAs) son otro modelo exitoso dentro de los SIAs. Se inspiran en el modelo de red inmune (ver sección 2.1.2). Este modelo establece la premisa de que los anticuerpos pueden ser reconocidos por otros anticuerpos. Esta capacidad de reconocer y ser reconocidos por otros elementos del sistema inmune dota

al sistema de un comportamiento intrínsecamente dinámico.

El primer algoritmo de este tipo fue propuesto en [23] e implementado en [24]. El sistema se compone de un objeto que representa la médula ósea, una red de objetos que representan a los linfocitos-B y una población de objetos que representan a los antígenos. Los linfocitos-B de la población son generados aleatoriamente por la médula ósea. La población de antígenos es cargada en el sistema. Cada linfocito-B es elegido aleatoriamente e insertado en la red. Si un linfocito-B presenta una alta afinidad con la población de antígenos, una gran cantidad de clones es generada. Después de pasar por un proceso de mutación, los clones con mayor afinidad a las células que ya están en la red son añadidas a ésta.

Las RIAs consideran las interacciones anticuerpos-antígenos a través de una métrica para determinar cuán afines son. Si el grado de estimulación no rebasa un umbral entonces el anticuerpo no reacciona ante la presencia del antígeno. La conexión entre un anticuerpo y un antígeno depende de qué tan bien se complementan el paratopo del anticuerpo con el epítipo del antígeno: cuanto mayor sea la afinidad más fuerte será la conexión. También será necesario medir el grado de afinidad entre los anticuerpos. Por último, se debe definir el número de epítopes que pueden tener los antígenos y los anticuerpos, y el número de parátopes que puede tener un anticuerpo.

Las aplicaciones de RIAs suelen comenzar con un conjunto de datos de entrada que corresponde a un conjunto de antígenos. Este conjunto estimulará a una red inmune que atravesará procesos dinámicos hasta alcanzar estabilidad, por ejemplo, hasta que se alcance cierta concentración de anticuerpos de diferentes tipos.

Una gran cantidad de variaciones del modelo inicial han sido propuestas; sin embargo, en [25] se presenta un modelo general que reúne las características comunes de estas variaciones. La figura 2.10 muestra los pasos principales del algoritmo:

- **Inicialización:** se genera una población de linfocitos-B, y una estructura de red.
- **Presentación de antígenos:** se calcula la afinidad y estimulación entre los antígenos y los linfocitos-B.
- **Interacción de los linfocitos-B:** se calcula la estimulación/supresión entre los linfocitos.
- **Maduración de la afinidad:** se crean clones de cada linfocito-B y se mutan. Se calcula la estimulación/supresión de los nuevos linfocitos-B.
- **Actulización de la población de linfocitos-B y de la estructura de la red:** se crean y eliminan las conexiones necesarias, con base en los demás pasos.

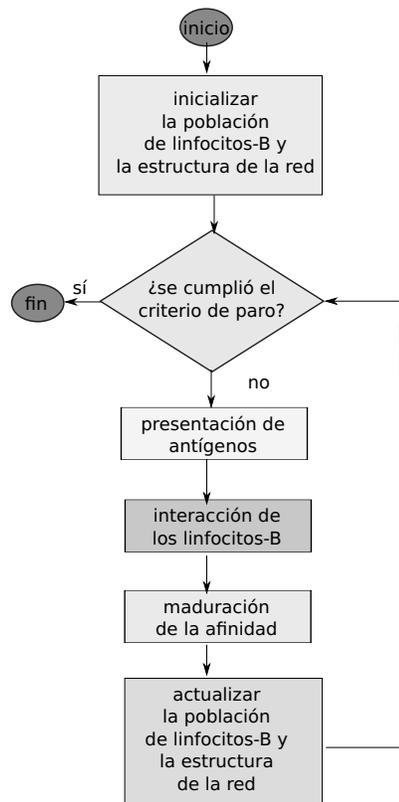


Figura 2.10: Algoritmo general de redes inmunes artificiales

Algoritmos de selección clonal

En [26] se propuso un algoritmo, basado en los principios de selección clonal y maduración de la afinidad (ver sección 2.1.2). Este modelo llamado “algoritmo de selección clonal” (ASC) incluye la inicialización de soluciones candidatas, mecanismos de selección, clonación y mutación, y una política de reemplazo, lo que lo hace de alguna manera similar a un algoritmo genético (explicado en la sección 3.2 en la pág 33).

Inicialmente este algoritmo fue desarrollado para tareas de reconocimiento de patrones y aprendizaje de máquina. Luego, fue modificado para poder resolver problemas de optimización multimodales. La comparación realizada en [26] entre un ASC y un algoritmo genético (AG), mostró que el primero puede llegar a un conjunto diverso de óptimos locales. En contraste, todas las soluciones candidatas en el AG convergen a la mejor solución perdiendo diversidad conforme aumenta el número de iteraciones realizadas.

A través de la validación del algoritmo en un problema de reconocimiento de caracteres, la optimización de tres funciones multimodales y una instancia de treinta ciudades para el problema del agente viajero, el artículo concluye que el algoritmo tiene una alta capacidad de aprendizaje y de almacenamiento de conocimiento.

La figura 2.12 muestra los pasos principales del ASC:

- **Inicialización:** se crea una población aleatoria de anticuerpos.
- **Cálculo de la afinidad:** se mide la afinidad de cada anticuerpo con respecto a la función objetivo.
- **Clonación:** el número de clones para cada anticuerpo, estará dado de manera proporcional a su afinidad.
- **Mutación:** el grado de mutación en un clon estará dado de manera inversamente proporcional a su afinidad.
- **Actualización de la población:** la nueva población estará formada por los mejores anticuerpos.
- **Diversificación:** se reemplazan a los peores anticuerpos de la población por anticuerpos generados aleatoriamente.

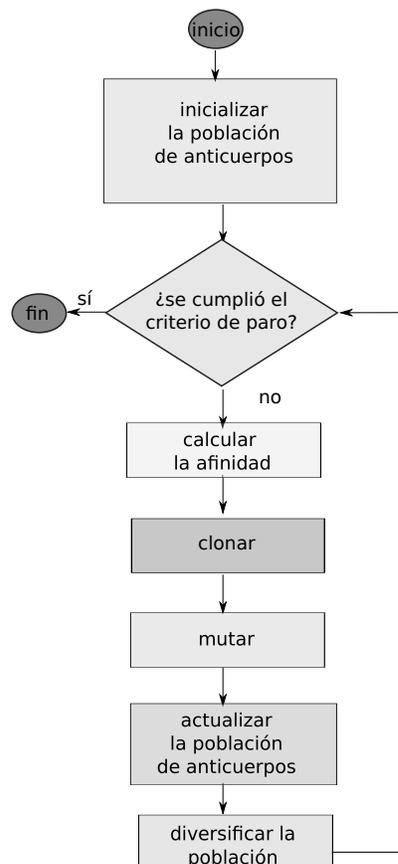


Figura 2.11: Algoritmo general de selección clonal

Algoritmos de la teoría del peligro

Otras áreas de la inmunología reportadas recientemente en la literatura han inspirado el desarrollo de algoritmos y herramientas computacionales, un ejemplo de ello es la teoría del peligro [27].

El concepto central de esta teoría es que el sistema inmune reacciona al peligro. Una amenaza es medida por el daño que provoca a las células, e indicada por señales de estrés que son enviadas cuando las células mueren de manera no natural (ver sección 2.1.2).

Existen tres leyes que rigen a los algoritmos basados en la teoría del peligro (ATP) [28]:

1. Un linfocito necesita dos señales para ser activado. El linfocito morirá si recibe la señal uno sin la coestimulación de la señal dos. En ausencia de la señal uno, la señal dos será ignorada.
2. La señal uno puede venir de cualquier célula. Pero la señal dos sólo proviene de una célula presentadora de antígenos. La señal dos para la activación de los linfocitos-B proviene de los linfocitos-T.
3. Los linfocitos activados no necesitan de la señal dos, éstos regresarán a su estado normal después de un corto tiempo. Los linfocitos inmaduros son capaces de reconocer señales dos de cualquier parte.

La figura 2.12 muestra los pasos principales del ATP:

- **Inicialización:** para este algoritmo es necesaria una población de antígenos.
- **Señal uno:** para cada antígeno se emite una señal de precaución. Los linfocitos que reciben esta señal estarán preparados para activarse en caso de recibir la señal dos.
- **Señal dos:** esta señal activa la zona de peligro.
- **Activación de linfocitos:** generación de anticuerpos.

2.2.2. Aplicaciones de sistemas inmunes artificiales

Los SIAs han sido aplicados a diferentes áreas: optimización, minería de datos, redes de computadoras, seguridad informática, automatización y diseño, detección de anomalías, bioinformática, procesamiento de texto, reconocimiento de patrones, algoritmos de agrupamiento, etc. Siendo la optimización el tema de interés en esta tesis, el siguiente capítulo se refiere a dicha área; sin embargo, a continuación se dan

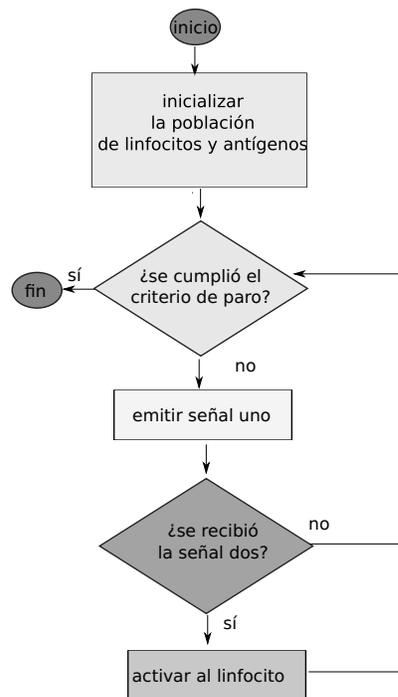


Figura 2.12: Algoritmo general de la teoría del peligro

algunos ejemplos de SIAs aplicados a otras áreas.

En [29] se propone un sistema inmune artificial de reconocimiento (SIAR) para llevar a cabo la clasificación automática de género, de música tradicional malaya. El sistema propuesto consta de tres etapas: extracción de características, selección y, finalmente, clasificación. Basándose en los resultados de los experimentos realizados, la precisión de la clasificación obtenida del sistema propuesto es del 88.6%. Para realizar los experimentos se utilizaron 191 archivos de música de 10 diferentes géneros, proporcionados por la academia de arte de Malasia.

En [30] se presenta un nuevo enfoque de SIAs para el diagnóstico de fallas en una red de procesadores. Se investiga cómo los SIAs pueden ser utilizados para el diagnóstico de fallas a nivel de sistema. El análisis teórico y los resultados experimentales en este trabajo demuestran la eficacia del enfoque para grandes y pequeñas redes. El trabajo concluye que el enfoque propuesto es una alternativa viable a los métodos tradicionales de diagnóstico de fallos.

En [31] se describe el modelado de un agente móvil basado en un sistema robótico que funciona bajo los principios de un SIA. En este trabajo los agentes móviles son vistos de forma analógica a los linfocitos-B de un sistema inmune que migran de un nodo (órgano) a otro y que mediante la generación de conjuntos de reglas (anticuerpos) ayudan al robot a superar un problema (antígeno).

En [32] se aplicaron SIAs para la detección de fraudes de tarjetas de crédito. Un AG fue utilizado para optimizar el conjunto de parámetros del SIA, minimizando el costo de fraude de una base de datos de tarjetas de crédito proporcionada por un emisor brasileño. El SIA propuesto fue comparado con otros métodos: redes neuronales [33], redes bayesianas [34], clasificador bayesiano ingenuo [35] y árboles de decisión [36]; mostrando un mejor rendimiento.

El desarrollo de la tecnología de microarrays (dispositivos utilizados para determinar cómo interactúan los genes unos con otros y cómo una célula es regulada mediante la expresión de un vasto número de genes) ha proporcionado un gran volumen de datos para la predicción y el diagnóstico de cáncer. Muchas técnicas populares de aprendizaje de máquina se han utilizado en el análisis de los datos proporcionados por estos dispositivos. En [37], se propone un SIAR para realizar la clasificación de estos datos. Tres clasificadores tradicionales también son empleados en los experimentos para comparar el rendimiento del SIAR propuesto. Los resultados demuestran el potencial de los SIAR en el análisis de datos de microarrays.

En [38] se propone una nueva solución para la detección de spam inspirado en el SI. Se presentan las pruebas realizadas del algoritmo propuesto con seis tipos de e-mail. Los resultados obtenidos son comparados con los obtenidos por un clasificador bayesiano y otro método de clasificación binaria desarrollado anteriormente para aplicaciones de minería de textos biomédicos. Se muestra que el modelo propuesto es un algoritmo competitivo para la detección de spam, en particular, y para realizar clasificación binaria en general.

Capítulo 3

Optimización multi-objetivo

La optimización multi-objetivo es un tema importante de investigación no sólo por el gran número de problemas del mundo real que son de este tipo, sino también por la cantidad de preguntas que se encuentran sin resolver dentro de esta área [1].

Un problema de optimización multi-objetivo (POM) se define como [39]: Encontrar un vector de variables de decisión que optimice un vector de funciones, cuyos elementos representan funciones objetivo normalmente en conflicto. De aquí que, el término “optimizar” signifique encontrar una solución que dé valores aceptables para el usuario encargado de tomar la decisión, de todas las funciones objetivo.

En esta clase de problemas, los algoritmos evolutivos multi-objetivo (AEMOs) inspirados en la teoría de Darwin sobre la evolución [40], han sido utilizados de manera exitosas al presentar una serie de ventajas [41] con respecto a las técnicas de programación matemática:

Las técnicas de programación matemática tienen ciertas limitantes; por ejemplo, algunas técnicas son susceptibles a la forma y continuidad del frente de Pareto (no trabajan bien con frentes cóncavos o discontinuos), otras requieren la primera (o incluso la segunda) derivada de las funciones objetivo. Adicionalmente, por lo general los métodos de programación matemática sólo generan una solución por cada ejecución y ésta depende del punto de inicio. Por lo tanto, muchas de ellas necesitan varias ejecuciones desde diferentes puntos de inicio a fin de poder generar varias soluciones no dominadas. En contraste los AEMOs permiten encontrar un conjunto de soluciones óptimas de Pareto en una sola ejecución, son menos susceptibles a la forma o continuidad del frente de Pareto (pueden lidiar con frentes de Pareto discontinuos y cóncavos) y no requieren mucha información sobre el problema (por ejemplo, no requieren las derivadas de las funciones objetivo).

En años recientes, una nueva metaheurística basada en el sistema inmune (capítulo 2) ha sido empleada para desarrollar algoritmos de optimización, mono y multi-objetivo, efectivos y eficientes [42].

En las primeras dos secciones de este capítulo, se presentan algunos conceptos básicos de optimización y computación evolutiva, necesarios para entender el resto de esta tesis. En la siguiente sección se da una visión general de los algoritmos de optimización multi-objetivo, enfocándose principalmente a los AEMOs y los SIAs aplicados a resolver POM (a los que llamamos sistemas inmunes artificiales multi-objetivo o SIAMOs).

3.1. Fundamentos de optimización multi-objetivo

Un problema de optimización (PO) puede plantearse como [43]:

minimizar (o maximizar) $f(x) = [f_1(x), \dots, f_k(x)]$; $x \in \Omega$

Una solución a PO minimiza (o maximiza) los componentes de un vector $f(x)$ donde x es un vector de variables de decisión n -dimensional $x = [x_1, \dots, x_n]$ de algún universo Ω .

Ω contiene todas las posibles x que pueden ser usadas para satisfacer una evaluación de $f(x)$. Cuando $k = 1$ se trata de un PO mono-objetivo; si $k \geq 2$ nos referimos a un PO multi-objetivo.

En un POM (en esta tesis los PO serán tratados como problemas de minimización) el vector ideal se define como: $f_{\text{ideal}} = [\text{mín } f_1(x), \dots, \text{mín } f_k(x)]$; si f_{ideal} representa una solución posible al POM, entonces los objetivos no están en conflicto y el POM tiene una única solución. De manera similar el punto de nadir se define como: $f_{\text{nadir}} = [\text{máx } f_1(x), \dots, \text{máx } f_k(x)]$.

La mayoría de los POMs no tienen una única solución, sino un conjunto de soluciones, que permiten buenos compromisos entre los diferentes objetivos. En estos casos suele adoptarse la noción de óptimo propuesta por Francis Ysidro Edgeworth y generalizada por Vilfredo Pareto llamada óptimo de Edgeworth-Pareto (ver figura 3.1). Decimos que una solución es un óptimo de Pareto o no dominada si no existe otra solución factible que decrezca algún criterio (en el caso de minimización de todos los objetivos) sin causar un incremento simultáneo en al menos algún otro criterio. Un conjunto de soluciones no dominadas de un problema multi-objetivo es llamado conjunto de óptimos de Pareto y los valores correspondientes a dicho conjunto en el espacio de las funciones objetivo se denomina frente de Pareto [43].

Un vector $u = [u_1, \dots, u_k]$ domina a otro vector $v = [v_1, \dots, v_k]$ (denotado por $u \preceq v$) si y sólo si u es parcialmente menor que v , i.e., $\forall i \in 1, \dots, k, u_i \leq v_i, i \in \{1, \dots, k\} : u_i < v_i$.

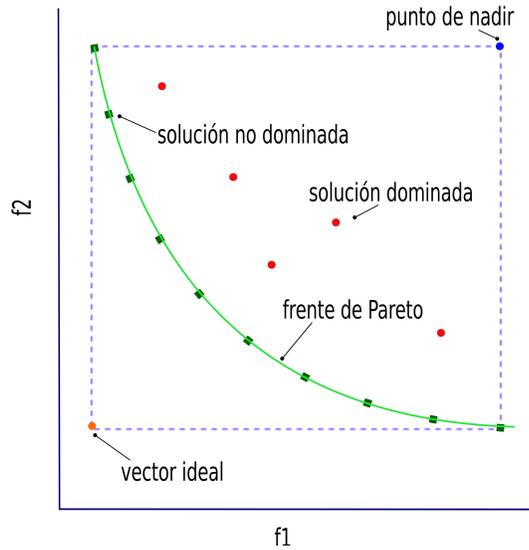


Figura 3.1: Óptimo de Edgeworth-Pareto

Medidas de desempeño en optimización multi-objetivo

Un indicador en general es una función que mapea k frentes de Pareto a un número real. Los más comunes son los indicadores unarios ($k = 1$). Los indicadores binarios pueden ser usados para comparar la calidad de dos aproximaciones del frente de Pareto.

Algunos ejemplos de indicadores son [43]:

- **Tasa de error (TE):** reporta el número de vectores en el frente conocido que no son miembros del verdadero frente de Pareto.

$$TE = \frac{\sum_{i=1}^n e_i}{n} \text{ donde:}$$

n es el número de elementos en el frente conocido y e_i toma el valor de cero si el i -ésimo vector del frente conocido es miembro del frente verdadero o uno si no lo es.

- **Hipervolumen (HV):** define el espacio cubierto por un frente conocido con respecto a un punto. Para un POM con dos objetivos, esto equivale a la suma de todas las áreas rectangulares, encerradas por algún punto de referencia y $[f_1(x), f_2(x)]$.

$$HV = \{\cup w_i | v_i \in F_{\text{conocido}}\} \text{ donde:}$$

v_i es un vector no dominado en el frente conocido (F_{conocido}) y

w_i es el volumen entre el origen y v_i .

El hipervolumen puede ser maximizado o minimizado, pero el punto de referencia debe ser elegido adecuadamente. Para minimizar el hipervolumen, el punto de referencia debería ser menor o igual al vector ideal. En contraste, para maximizar el hipervolumen, el punto de referencia debería ser mayor o igual al punto

de Nadir (ver figura 3.2). Este indicador no es invariante a la escala (es sensible al punto de referencia elegido).

El hipervolumen es el único indicador unario conocido que garantiza monotonicidad estricta con respecto a la dominancia de Pareto: un conjunto F alcanza el valor máximo de hipervolumen si y sólo si todos los puntos $f \in F$ son óptimos de Pareto.

El tiempo para calcular el hipervolumen, crece de manera exponencial conforme crece el número de funciones objetivo (éste es un problema NP-completo).

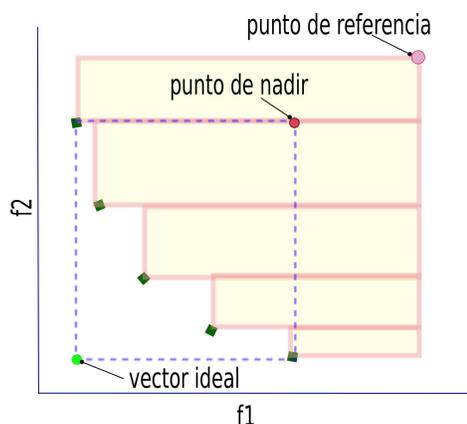


Figura 3.2: Representación gráfica del cálculo del hipervolumen

- **Distancia generacional (DG):** representa qué tan lejos está un frente conocido del verdadero frente de Pareto. $DG = \frac{(\sum_{i=1}^n d_i^p)^{1/p}}{n}$ donde: n es el número de elementos en el frente conocido, $p = 2$ y d_i es la distancia euclidiana del i -ésimo vector del frente conocido y el vector más cercano a él en el frente verdadero.
- **Distancia generacional invertida (DGI):** representa qué tan lejos está el verdadero frente de Pareto de un frente conocido. $DGI = \frac{(\sum_{i=1}^N d_i^p)^{1/p}}{N}$ donde, N es el número de elementos en el frente verdadero, $p = 2$ y d_i es la distancia euclidiana del i -ésimo vector del frente verdadero y el vector más cercano a él en el frente conocido.
- **Espaciado:** describe la extensión de los vectores en el frente. $S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i^n)^2}$, y $d_i = \min_j (|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|)$ $i, j = 1, \dots, n$ donde, n es el número de elementos en el frente conocido, k es el número de objetivos y \bar{d} es el promedio de todos los d_i
- En [44] se propone un indicador alternativo para medir la distancia al frente de Pareto $\Delta_p = \max(DG_p, DGI_p)$:

- DG_p : versión alternativa a la distancia generacional.
 $DG_p = (\sum_{i=1}^n d_i^p)^{1/p}$ donde, n es el número de elementos en el frente conocido, Δ_p es una semi-métrica para $1 \leq p < \infty$ y una métrica para $p = \infty$, y d_i es la distancia euclidiana del i -ésimo vector del frente conocido y el vector más cercano a él en el frente verdadero.
- DGI_p : indicador análogo a DG_p .
 $DGI_p = (\sum_{i=1}^N d_i^p)^{1/p}$ donde, N es el número de elementos en el frente verdadero, Δ_p es una semi-métrica para $1 \leq p < \infty$ y una métrica para $p = \infty$, y d_i es la distancia euclidiana del i -ésimo vector del frente verdadero y el vector más cercano a él en el frente conocido.

■ En [45] se presentan los indicadores siguientes:

- I_e : indica la distancia mínima necesaria que debe trasladarse un frente para dominar débilmente a otro.
 $I_e(A, B) = \max[f_i(A) - f_i(B)]$
 Un vector $u = [u_1, \dots, u_k]$ domina débilmente a otro vector $v = [v_1, \dots, v_k]$ (denotado por $u \preceq v$) si y sólo si $\exists i \in 1, \dots, k, u_i < v_i, i \in \{1, \dots, k\} \therefore$

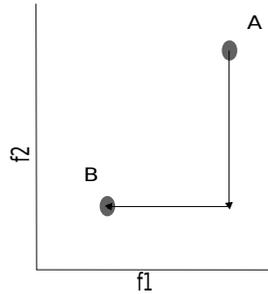


Figura 3.3: Ilustración del indicador I_e para un POM en dos dimensiones

En la figura 3.3 el indicador es aplicado a dos soluciones A y B , note que:
 $I_e(A, B) > 0$ mientras que $I_e(B, A) < 0$

- I_{HD} : $I_{HD}(A, B)$ representa el volumen del espacio que es dominado por B pero no por A .

$$I_{HD}(A, B) = \begin{cases} H(B) - H(A) & \text{si } B \succ A \\ H(A + B) - H(A) & \text{de otro modo} \end{cases}$$
 donde: $H(X)$ representa el volumen que es dominado por X

En la figura 3.4 el indicador es aplicado a dos soluciones A y B , note que:
 $I_{HD}(A, B) = -I_{HD}(B, A) < 0$.

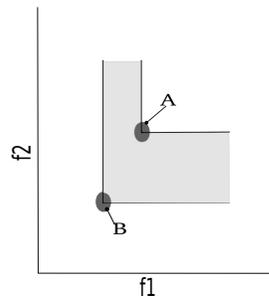


Figura 3.4: Ilustración de I_{HD} para un POM en dos dimensiones

3.2. Fundamentos de computación evolutiva

El término computación evolutiva o algoritmos evolutivos (AEs) se refiere a una serie de técnicas inspiradas en la teoría de Darwin sobre la evolución natural. Aunque existen diferentes variaciones de AEs, la idea general de todas estas técnicas es la misma: dada una población de individuos, la presión del medio ambiente provoca la denominada selección natural (supervivencia del más apto), lo que causa un incremento en la aptitud de la población.

La figura 3.5 describe los pasos principales de un AE [46]:

- **Inicialización:** un conjunto de soluciones candidatas (individuos) es generado (por lo general de manera aleatoria).
- **Evaluación:** las soluciones candidatas son evaluadas mediante una función de calidad a ser maximizada (aptitud).
- **Selección de padres:** basándose en la aptitud, algunas de las mejores soluciones son elegidas para generar nuevas soluciones mediante la recombinación y/o mutación.
- **Recombinación:** operador aplicado a dos o más individuos seleccionados (padres) dando como resultado uno o más nuevos individuos (hijos). Los nuevos individuos se forman con segmentos de sus padres.
- **Mutación:** operador aplicado a un individuo dando como resultado uno nuevo, tras realizarle pequeñas perturbaciones a su cromosoma.
- **Actualización:** los mejores individuos resultantes de los pasos anteriores son elegidos para formar la nueva población.

Principales paradigmas de la computación evolutiva

Son tres los paradigmas principales en los AEs:

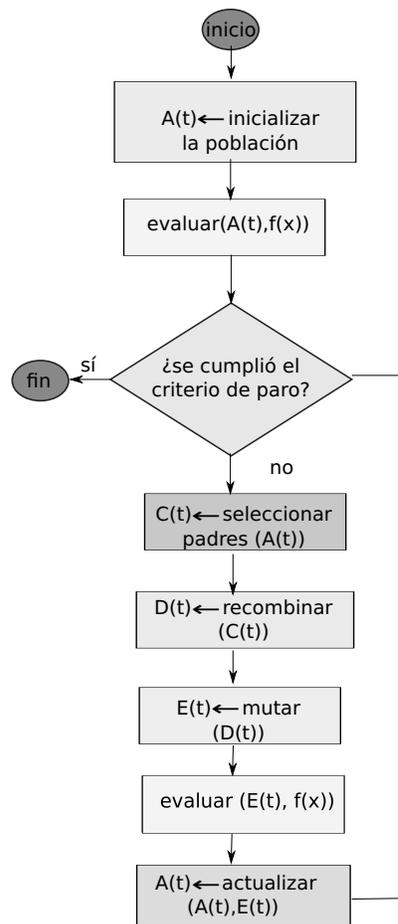


Figura 3.5: Esquema general del funcionamiento de un algoritmo evolutivo

- **Programación evolutiva (PE):** esta técnica propuesta en [47] representa una abstracción de la evolución al nivel de las especies; por esta razón se omite el operador de recombinación: dos especies diferentes no pueden procrear entre sí. La figura 3.6 describe los pasos principales de la PE:

- **Inicialización:** la población inicial se genera normalmente de manera aleatoria. En esta técnica las soluciones candidatas contienen vectores con las variables del problema en su representación original.
- **Mutación:** es el único operador utilizado para generar nuevas soluciones, por lo cual suelen utilizarse diversas variantes del mismo en una misma implementación.
- **Evaluación:** la aptitud de cada solución está dada por una versión normalizada de la función objetivo.
- **Actualización:** mediante torneo se determina cuáles serán las soluciones que se retendrán. El proceso de selección es probabilístico, pero extinti-

vo (es decir, los individuos menos aptos tienen una probabilidad cero de sobrevivir).

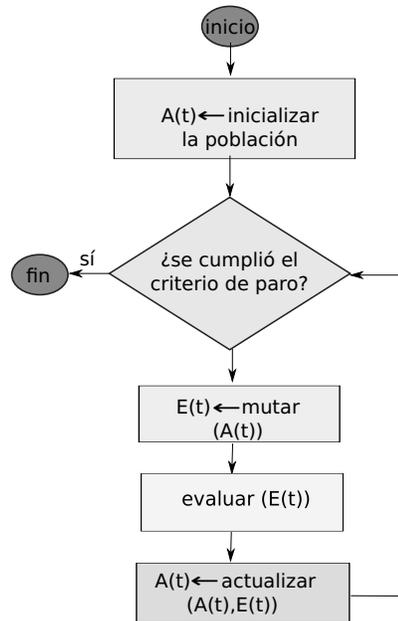


Figura 3.6: Algoritmo general de la programación evolutiva

- **Estrategias evolutivas (EEs):** esta técnica fue desarrollada originalmente para resolver problemas hidrodinámicos complejos [48]. En contraste con la PE, las EEs representan una abstracción de la evolución al nivel de un individuo, por lo que el operador de recombinación está permitido aunque se utiliza como un operador secundario. El operador principal es la mutación. La figura 3.7 describe los pasos principales del algoritmo original de la EE-(1 + 1):

- **Generación del individuo inicial:** en esta técnica las soluciones candidatas típicamente constan de vectores de números reales. En la versión original se utiliza un solo padre para generar un solo hijo (por eso se denomina (1+1)).
- **Evaluación:** la aptitud de cada solución está dada por la función objetivo.
- **Generación de un nuevo individuo:** un nuevo individuo es generado mediante la siguiente fórmula:

$$x^{t+1} = x + N(0, \sigma)$$
donde:
 - t se refiere al número de iteración (generación) y $N(0, \sigma)$ es un vector de números Gaussianos independientes con una media cero y desviación estándar σ .
- **Actualización:** si el hijo generado es mejor que su padre, lo sustituye; de lo contrario, es eliminado. El proceso de selección es extintivo.

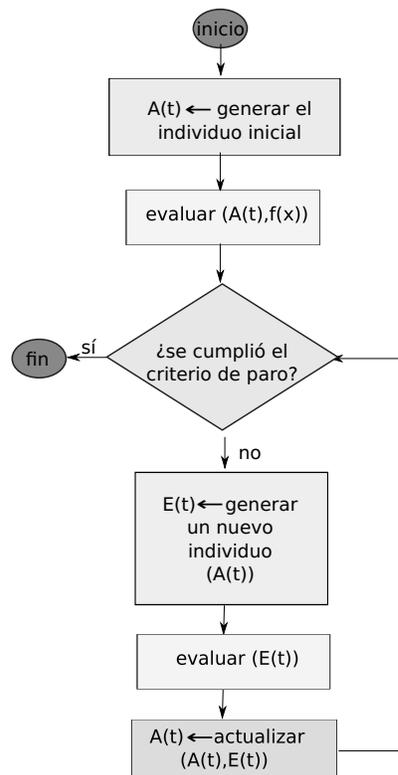


Figura 3.7: Algoritmo general de la estrategia evolutiva (1 + 1)

En [49] se propuso la $EE-(\mu + 1)$: en ella, μ padres generan un solo hijo; éste sólo puede reemplazar al peor padre de la población. En las $EE-(\mu + \lambda)$ y las $EE-(\mu - \lambda)$ se propuso la idea de múltiples hijos [50, 51]: en la primera de ellas, las μ mejores soluciones obtenidas de la unión de padres e hijos sobreviven; mientras que, en la segunda técnica sólo los μ mejores hijos son retenidos.

- **Algoritmos genéticos:** esta técnica, propuesta en [52] enfatiza la importancia de la cruce (recombinación) sobre la mutación. La figura 3.8 describe los pasos principales de un AG:
 - **Inicialización:** en esta técnica se suele adoptar una codificación binaria para representar las soluciones potenciales al problema que deseamos resolver. A la cadena binaria que codifica todas la variables del problema se le denomina cromosoma; cada variable de la cadena es denominada gen y a cada bit de un gen se le llama alelo.
 - **Evaluación:** la aptitud de cada solución está dada por una versión normalizada de la función objetivo.
 - **Selección de padres:** con base en la aptitud, algunas de las soluciones son elegidas para generar, mediante cruce y mutación, a los individuos de

la siguiente población. El proceso de selección es probabilístico.

- **Cruza:** es el operador principal.
- **Mutación:** es un operador secundario.
- **Actualización:** los individuos resultantes de la cruce y mutación forman la nueva población. Un aspecto importante en un AG es el elitismo: el mejor individuo de cada generación pasa intacto a la siguiente generación (ni se cruza, ni se muta).

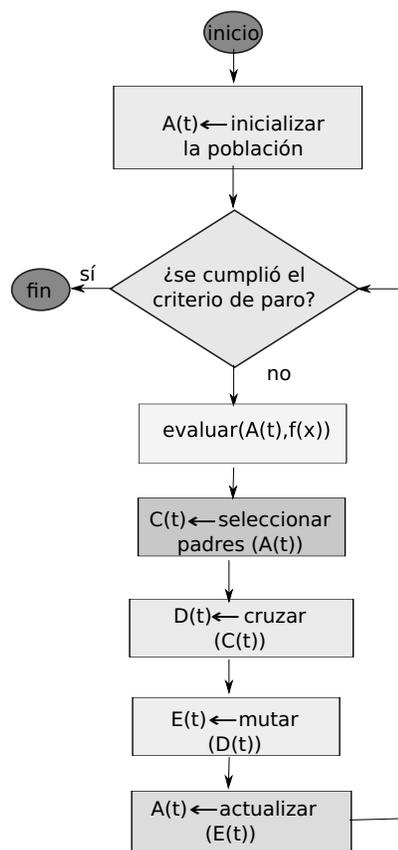


Figura 3.8: Esquema general de un algoritmo genético

3.3. Algoritmos de optimización multi-objetivo

La idea más simple para resolver un POM consiste en combinar todos los objetivos en un valor escalar (p.ej., sumando los valores de las funciones objetivo). A este tipo de enfoque se le denomina agregativo y se puede aplicar en diversas variantes, como se indica a continuación.

3.3.1. Enfoques agregativos

Algunas de las técnicas más populares de los enfoques agregativos son [1]:

- **Suma ponderada:** este método consiste en sumar todas las funciones objetivo usando diferentes coeficientes de peso para cada una. El POM es transformado en un problema de optimización escalar de la forma: $\sum_{i=1}^k w_i f_i(x)$ donde, $w_i \geq 0$ son los coeficientes de peso. Usualmente se presupone que $\sum_{i=1}^k w_i = 1$.

Se hace notar que, todas las funciones deben ser normalizadas a fin de que los pesos adoptados puedan expresar su importancia relativa. El problema queda entonces transformado a la forma: $\sum_{i=1}^k w_i f_i(x) c_i$ donde, c_i representa a las constantes multiplicadoras que escalaran a los objetivos. Los mejores resultados son usualmente obtenidos con $c_i = \frac{1}{\min f_i(x)}$. Ejemplos de aplicaciones que utilizan esta técnica son: [53, 54, 55, 56].

- **Programación por metas:** en este método el usuario encargado de tomar la decisión asigna metas que desea alcanzar para cada objetivo. La función objetivo tratará entonces de minimizar la diferencia entre los valores actuales y las metas deseadas. La formulación más simple de un POM en esta técnica es: $\min \sum_{i=1}^k w_i |f_i(x) - M_i|$, donde M_i denota la meta puesta por el usuario encargado de tomar la decisión a la función objetivo $f_i(x)$. Ejemplos de aplicaciones que utilizan esta técnica son: [57, 58].
- **Método de la restricción ϵ :** esta técnica minimiza una (la principal) función objetivo, y considera a los otros objetivos como restricciones adicionales. El método puede ser formulado como sigue:

1. Minimizar el objetivo r :
 $\min f_r(x)$ sujeto a restricciones adicionales de la forma:
 $f_i(x) \leq \epsilon_i$ para $i = 1, \dots, k$ e $i \neq r$, donde ϵ_i representa el valor de f_i que no se desea exceder.
2. Repetir el paso anterior para diferentes valores de ϵ_i . La información derivada de la elección de valores apropiados para ϵ_i puede ser útil en la toma de decisiones. La búsqueda es detenida cuando el usuario encargado de tomar la decisión encuentra una solución satisfactoria.

Ejemplos de aplicaciones que utilizan esta técnica son: [59, 60].

3.3.2. Enfoques no agregativos

Algunas de las técnicas más populares de los enfoques no agregativos que no se basan en la dominancia de Pareto son [1]:

- **Algoritmo genético evaluado con vectores (VEGA por sus siglas en inglés: Vector Evaluated Genetic Algorithm):** este algoritmo sólo difiere

del AG original en la manera en que la selección es llevada a cabo: en cada generación, se genera un cierto número de subpoblaciones y en ellas se aplica selección proporcional de acuerdo a cada función objetivo. Para un problema con k objetivos, k subpoblaciones de tamaño N/k son generadas (suponiendo que N es el tamaño total de la población). Estas subpoblaciones forman una nueva población de tamaño N , sobre la que actuarán los operadores de cruce y mutación de la manera tradicional. Esta propuesta presenta un problema conocido en genética como especiación: la evolución de una especie dentro de una población la cual destaca de las demás en algún aspecto. Este problema ocurre porque la técnica de selección elige individuos que destacan en un objetivo, sin tomar en cuenta a los demás. Esto se contrapone de alguna forma a la optimalidad de Pareto, pues una solución que represente un buen compromiso, pero que no sea mejor que las demás en ningún objetivo, no será seleccionada. Esto podría descartar a las soluciones que no producen los mejores valores para ningún objetivo en específico, pero cuyo comportamiento es moderadamente bueno en todas las dimensiones. Ejemplos de aplicaciones que utilizan esta técnica están en [61, 62].

- **Ordenamiento lexicográfico:** en este método, se ordenan los objetivos de acuerdo a su importancia. La solución óptima $f(x^*)$ es entonces el resultado de minimizar las funciones objetivo iniciando con la más importante y siguiendo de acuerdo al orden de importancia asignado. En esta técnica $f_1(x)$ y $f_k(x)$ representan la más y la menos importante de las funciones objetivo respectivamente.

El primer problema es formulado como: $\min f_1(x)$, con lo que se obtiene f_1^* . Posteriormente el segundo problema se define como: $\min f_2(x)$ sujeto a: $f_1(x) = f_1^*$ con lo que f_2^* se obtiene. Este procedimiento es repetido hasta que los k objetivos hayan sido considerados. Ejemplos de aplicaciones que utilizan esta técnica son: [63, 64].

- **Teorema de contacto para detectar soluciones óptimas de Pareto:** Existe un algoritmo basado en el teorema del contacto [65], que puede usarse para determinar la distancia relativa de una solución al conjunto de óptimos de Pareto. A este método se le conoce también como el método de la distancia. Una solución es inicialmente generada aleatoriamente y es considerada óptima de Pareto. Su aptitud d_1 , la cual es elegida arbitrariamente, es llamada distancia inicial. Posteriormente las demás soluciones son generadas y la “distancia” para cada una de ellas es calculada usando la fórmula $z_l(x) = \sqrt{\sum_{i=1}^k \left(\frac{f_{il} - \Phi_i(x)}{f_{il}}\right)^2}$ para $l = 1, \dots, l_p$ donde, l_p es el número de soluciones óptimas de Pareto encontradas hasta el momento, $\Phi_i(x)$ es el valor del i -ésimo objetivo de la solución y f_{il} es el valor del i -ésimo objetivo de la l -ésima solución óptima de Pareto. En el siguiente paso, se obtienen el valor mínimo del conjunto $z_l(x)$ y su índice correspondiente l^* . Este valor es llamado $z_{l^*}(x)$. El proceso identifica la solución más cercana a la nueva solución generada. Si la solución generada es un óptimo de Pareto,

su aptitud es asignada de acuerdo a la siguiente fórmula $\text{aptitud} = d_l^* + z_l^*$. Después de la primera generación, d_l es definida usando el máximo valor de las distancias de todas las soluciones óptimas de Pareto. Si la nueva solución generada no es un óptimo de Pareto, su aptitud es calculada usando la fórmula $\text{aptitud} = d_l^* - z_l^*$ y $\text{aptitud} = 0$ en caso de que el resultado sea un valor negativo. Ejemplos de aplicaciones que utilizan esta técnica son: [66, 67].

- **Pesos generados aleatoriamente y elitismo:** este algoritmo utiliza pesos generados aleatoriamente para transformar el POM a un problema escalar que usará como aptitud de una solución a la expresión siguiente: $\text{aptitud} = \sum_{i=1}^k w_i f_i(x)$, donde k es el número de objetivos $w_i = \frac{r_i}{(r_1 + \dots + r_k)}$ para $i = 1, \dots, k$ y r_1, \dots, r_k son k números aleatorios en el intervalos $[0, 1]$. En este caso se utiliza una memoria secundaria $B(t)$ para almacenar las soluciones no dominadas (elitismo). En [68] se utiliza esta técnica para resolver problemas de planeación de tareas.

3.3.3. Algoritmos evolutivos multi-objetivo

En [69] se plantea por primera vez, la incorporación del concepto de óptimo de Pareto en la selección de un algoritmo evolutivo multi-objetivo (AEMO). La idea básica es identificar a las soluciones que no son dominadas por las demás soluciones de la población, asignarles la más alta calificación e ignorarlas en futuras competencias. El siguiente paso consiste en encontrar un nuevo conjunto de soluciones no dominadas de la población restante y asignarle la siguiente calificación más alta. Este proceso debe repetirse hasta que todas las soluciones hayan sido calificadas. Se sugirió también el uso de alguna técnica de nichos, a fin de mantener diversidad en la población. Un nicho en términos biológicos es un conjunto de condiciones a la que un grupo específico está especialmente bien adaptado. En AEMOs un nicho es tratado como un subconjunto de soluciones similares en algún espacio (el de las variables o el de los objetivos). Para proteger la diversidad y evitar que el AEMO converja a un solo punto del frente se desea que una solución recién generada sustituya a otra de su propio nicho. No se provee una implementación de estas ideas, pero, prácticamente todos los AEMOs desarrollados después de la publicación de este libro fueron influenciados por las ideas en él contenidas.

Otro principio importante en los AEMOs es el elitismo. El elitismo en los AEMOs se refiere a retener a todas las soluciones no dominadas con respecto a todos los individuos que han sido generados hasta el momento. Por razones prácticas, sin embargo, suele limitarse la cantidad de soluciones retenidas.

Algunos de los primeros AEMOs en incorporar esta idea fueron: [70, 71]. En la actualidad, la mayoría de los AEMOs incorporan poblaciones externas (o secundarias) $B(t)$ donde se almacenan las mejores soluciones encontradas o alguna otra técnica alternativa que permita elitismo. Al igual que en los AEs mono-objetivo, el uso de

elitismo en los AEMOs es un requerimiento teórico para garantizar convergencia [72].

El Nondominated Sorting Genetic Algorithm (NSGA) [73] fue el primer AEMO publicado en una revista especializada en computación evolutiva. Este algoritmo clasifica a las soluciones de acuerdo al esquema propuesto en [69]. Las soluciones con la aptitud más alta; es decir, las que no son dominadas por ninguna otra solución son las que reciben un mayor número de copias en el proceso de selección. Éste es un AEMO elitista.

El Strength Pareto Evolutionary Algorithm (SPEA) [74] utiliza un archivo donde almacena las soluciones no dominadas encontradas anteriormente al que llama: “conjunto externo de soluciones no dominadas” $B(t)$. Se calcula para cada una de estas soluciones un valor de fuerza (proporcional a la cantidad de soluciones que dominan a esta solución). Un punto crítico en este AEMO es el tamaño de $B(t)$, pues si éste es demasiado grande, podría reducir la presión de selección lo que podría provocar un decremento en la velocidad de búsqueda.

La Pareto Archived Evolution Strategy (PAES) [75] es una EE-(1+1) combinada con un archivo externo que almacena las soluciones no dominadas generadas durante el proceso evolutivo. Un aspecto importante en este AEMO es el procedimiento que usa para mantener la diversidad (ver figura 3.9) : el espacio de las funciones objetivo es dividido de manera recursiva (formando una rejilla). Cada una de las soluciones se sitúa en uno de los hipercubos resultantes basándose en el valor de sus objetivos (usados como coordenadas). La cuenta del número de soluciones en cada hipercubo es utilizada para dar preferencia a las soluciones que se encuentren en los hipercubos menos poblados. Dado que este mecanismo es adaptativo el único parámetro extra requerido es el número de divisiones en el espacio de las funciones objetivo. PAES es in AEMO elitista.

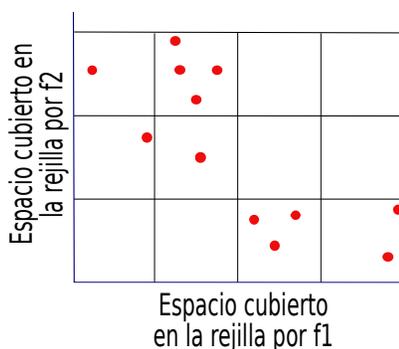


Figura 3.9: Rejilla adaptativa en dos dimensiones usada por el algoritmo PAES

El Nondominated Sorting Genetic Algorithm-II (NSGA-II) [4] fue propuesto como una versión mejorada del NSGA aunque es bastante diferente a su predecesor. En este

AEMO se determina para cada solución, el número de soluciones que la dominan y el número de soluciones a las que domina. Estima la densidad de cada solución mediante la distancia de “crowding”; que es el promedio de la longitud de los lados del cuboide formado por los vectores más cercanos a una solución en cada uno de los objetivos (ver figura 3.10). El mecanismo de selección toma en cuenta ambos aspectos: las soluciones no dominadas son preferidas a las dominadas; en caso de empate en este aspecto, las soluciones con mayor distancia de “crowding” son preferidas (por estar más aisladas). El NSGA-II no utiliza una población secundaria para almacenar las mejores soluciones. La técnica de elitismo de este AEMO consiste en combinar a la población de padres con la de hijos, reteniéndose la mejor mitad de los individuos resultantes.

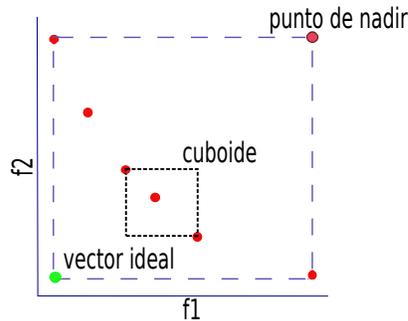


Figura 3.10: Ilustración de distancia de crowding

La mayoría de los AEMOs utilizan esquemas de selección basados en la dominancia de Pareto. Sin embargo, se ha mostrado que este esquema no se comporta bien en presencia de muchos objetivos (más de 4) [2]. Una alternativa a la dominancia de Pareto es el uso de indicadores (medidas de desempeño), para el mecanismo de selección de un AEMO. La idea general es calificar a las soluciones de acuerdo a su aportación al indicador: las soluciones que más aportan son preferidas. En [76, 77, 78, 79] se utiliza la medida de desempeño conocida como hipervolumen. En [45] se utilizan los indicadores I_ϵ y el indicador basado en hipervolumen I_{HD} . Tanto I_ϵ como I_{HD} mostraron un mejor desempeño en [80], al ser utilizados como esquemas de selección, que los indicadores basados en dominancia de Pareto I_{Ben} , I_{Sri} , I_{Fon} , presentados en el mismo artículo.

En [81] se propone un AEMO que minimiza al indicador Δ_p . Este algoritmo se basa en el AE denominado evolución diferencial [82]. La aptitud de cada individuo es determinada de acuerdo a su contribución al indicador Δ_p . Dado que el indicador Δ_p necesita un conjunto de referencia y el frente verdadero es generalmente desconocido, se construye una aproximación. En la DG, la aportación de un individuo I_{GD} está dada por la distancia entre él y el vector más cercano a él en el frente de referencia. En la DGI la aportación del individuo a_i está dada por:

$$I_{DGI} = \begin{cases} \sqrt[p]{\sum_{i=1}^{q \in Q} dis(q, a_i)^p} & \text{si } Q \neq 0 \\ -1 & \text{si } Q = 0 \end{cases}$$

donde Q es el conjunto de soluciones del frente de referencia para los que a_i es la solución más cercana en la población.

Los individuos que contribuyen más a la DGI son preferidos. En caso de empate, se utiliza la DG: los individuos más cercanos ganan la competencia. Esta propuesta obtiene resultados competitivos para problemas con 2 y 3 objetivos y presenta resultados prometedores para problemas con muchos objetivos.

3.3.4. Sistemas inmunes artificiales multi-objetivo

El reconocimiento de un antígeno por el sistema inmune puede ser visto como un problema de búsqueda [42], pues se necesitan encontrar aquellos anticuerpos que mejor combatan al antígeno. En este caso, un PO puede ser visto como el antígeno o el antígeno polivalente en el caso de un POM, y las soluciones candidatas serán entonces conocidas como anticuerpos.

La mayoría de los SIAs multi-objetivo (SIAMOs) están inspirados en la selección clonal y utilizan esquemas de selección basados en la dominancia de Pareto. Algunos aspectos importantes que deben considerarse en el diseño de un SIAMO son [83]:

- **Elitismo:** es introducido para preservar buenas soluciones y no perderlas durante las respuestas inmunológicas.
- **Población adaptativa:** la mayoría de los SIAMO trabajan con un tamaño de población fijo. Sin embargo, el adaptar el tamaño de la población (por ejemplo mediante la supresión de los linfocitos con mayor afinidad anticuerpo-anticuerpo) permite que este parámetro sea no crucial y disminuye el número de evaluaciones a las funciones objetivo.
- **Mutación proporcional a la calidad:** la mutación sufrida por un linfocito puede estar en función de su calidad.
- **Memoria inmunológica:** guarda las mejores soluciones encontradas durante las iteraciones (población secundaria o externa) y es considerado un elemento clave para el éxito de un AEMO.

La figura 3.11 describe los pasos principales del algoritmo canónico propuesto en [42] para los SIAMOs:

- **Definición del problema:** definir el espacio de búsqueda X , el tamaño de la población N , las funciones objetivo $f(\cdot)$ y las restricciones $g(\cdot)$.
- **Inicializar la población:** las soluciones de este conjunto representan a las células que combaten al antígeno polivalente (amenaza).

- **Inicializar la población secundaria:** las soluciones de este conjunto representan a las células de memoria; es decir, las mejores soluciones encontradas a lo largo de las iteraciones.
- **Evaluación:** las soluciones de la población son evaluadas usando $f(\cdot)$ y $g(\cdot)$.
- **Evaluación de calidad:** la afinidad antígeno-anticuerpo y anticuerpo-anticuerpo de cada célula es asignada de acuerdo al paso anterior.
- **Selección:** en este paso se eligen las células que podrán tomar parte en la respuesta inmune.
- **Variación:** las células que participan en la respuesta inmunológica pueden sufrir mutaciones.
- **Actualización:** el exceso de células provocado por una respuesta inmune debe ser suprimido. Las células que mostraron un mejor desempeño con respecto al antígeno son retenidas con el fin de mejorar la siguiente respuesta inmune.

En [84] se implementó el primer algoritmo multi-objetivo que adopta conceptos del sistema inmune biológico, el cual en realidad consiste en un algoritmo genético [85] con representación binaria, que utiliza operadores comunes de selección, cruza y mutación. Los principios de inmunología son empleados para modificar la aptitud de la población $A(t)$. En este algoritmo, la población secundaria $B(t)$ contiene las soluciones no dominadas y es llamada población de antígenos. La población principal es evaluada inicialmente, combinando todos los objetivos en un valor escalar usando pesos aleatorios. Un antígeno es seleccionado de manera aleatoria de $B(t)$ y S anticuerpos son seleccionados también de manera aleatoria de $A(t)$. La afinidad (ψ) entre un antígeno y un anticuerpo se calcula de tal forma que un anticuerpo con afinidad alta, tiene un incremento elevado en su aptitud.

El Multiobjective immune system algorithm (MISA) [86] es considerado como el primer SIA multi-objetivo basado en dominancia de Pareto. Este algoritmo se basa en el principio de selección clonal y usa una memoria secundaria donde se almacenan las soluciones no dominadas generadas. Para lograr una distribución uniforme se utiliza, al igual que en PAES [75], una rejilla adaptativa (en el espacio de las funciones objetivo); con lo que se define un nuevo criterio: densidad-región, dando prioridad (incrementando el número de clones o facilitando la entrada a la memoria cuando el tamaño máximo ha sido alcanzado) a los individuos que pertenecen a las regiones menos pobladas del espacio de búsqueda. La población consiste de cadenas binarias y se le divide en antígenos $B(t)$ y anticuerpos $A(t)$ utilizando la dominancia de Pareto. Los individuos no dominados son los antígenos y los dominados son los anticuerpos. En problemas con restricciones, los individuos factibles también son antígenos; los antígenos son clasificados en “muy buenos” o simplemente “buenos”, de acuerdo a las siguientes reglas: $w = 4$ antígenos factibles no dominados, $w = 3$ antígenos infactibles

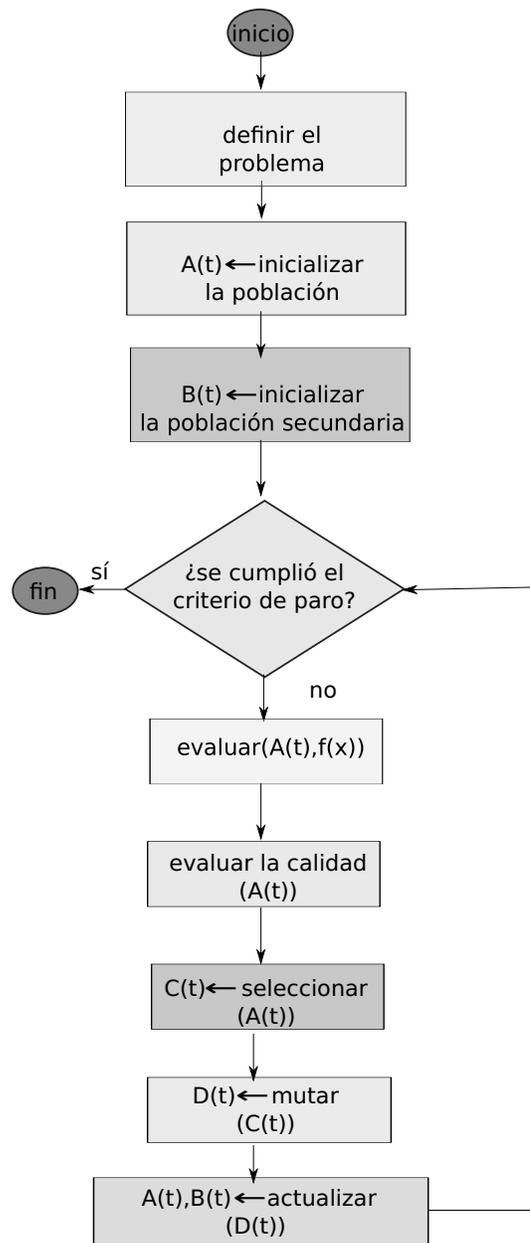


Figura 3.11: Algoritmo general de un sistema inmune artificial multi-objetivo

no dominados, $w = 2$ antígenos factibles dominados. La afinidad antígeno-anticuerpo es medida por ψ (ver figura 3.12) propuesta en [87] tomando en cuenta el número de coincidencias consecutivas entre el antígeno (seleccionado al azar) y el anticuerpo y la calidad del antígeno. Con base en ψ se seleccionan los anticuerpos que serán clonados y se establece la frecuencia de mutación (mutación proporcional a la calidad).

El Vector Artificial Immune System (VAIS) [6] es un SIA multi-objetivo basado en la selección clonal. En este algoritmo se tienen dos poblaciones: una de antígenos

Antígeno: 0 1 1 1 1 0 0 1 0
 Anticuerpo: 0 1 1 0 0 1 1 1 0
 Coincidencias: 5
 Longitud: 3 2

$$\psi = 5 + 3^w + 2^w$$

Figura 3.12: Afinidad antígeno-anticuerpo en MISA

$B(t)$ y una de anticuerpos $A(t)$. Después del proceso de clonación y mutación (proporcional a la calidad) aplicado a la población $A(t)$, se realiza una búsqueda local sobre cada uno de los clones. Las soluciones no dominadas encontradas son almacenadas en $B(t)$. Las soluciones almacenadas, cuya distancia euclidiana a la solución más cercana no supere el umbral $\frac{\sqrt{k}}{M}$ (donde: k es el número de objetivos y M el tamaño de $B(t)$) son eliminadas. A partir de $B(t)$ se genera la nueva población. Cada cierto número de iteraciones se generan nuevos anticuerpos de manera aleatoria.

A Clonal Selection Algorithm for Multiobjective Optimization (ACSAMO) [88] es otro SIA multi-objetivo basado en el principio de selección clonal, el cual utiliza representación real, y mutación proporcional a la calidad. Maneja dos poblaciones: una de antígenos $B(t)$ y otra de anticuerpos $A(t)$. En la primera se almacenan las soluciones no dominadas obtenidas a lo largo del proceso de búsqueda. En cada generación se retienen las dos “mejores” soluciones encontradas (“la mejor en las iteraciones anteriores” b_{mejor} y la “mejor en esta generación” a_{mejor}) de acuerdo a un enfoque de suma de pesos aleatorios: $WF_i = \sum_{j=1}^m w_j f_j$ donde $\sum_{j=1}^m w_j = 1$ y m es el número de funciones objetivo. La calidad de cada anticuerpo se calcula como: $\psi = \text{dis}(a, b_{mejor}) + \text{dis}(a, a_{mejor})$ donde dis se refiere a la distancia euclidiana. Después del proceso de mutación y clonación, los anticuerpos que constituirán a la siguiente población son elegidos dando prioridad: 1) a los no dominados y 2) a aquellos con una baja afinidad anticuerpo-anticuerpo. Una vez que la población de antígenos llega a su tamaño máximo, las soluciones que pertenecen a las regiones más pobladas del frente son eliminadas.

El Bayesian Artificial Immune System for Multi-Objective Optimization (MO-BAIS) [89] sustituye los operadores tradicionales de clonación y mutación de los SIAs por un modelo de probabilidades con el afán de evitar la destrucción de los bloques constructores (soluciones parciales de alta calidad codificadas en el anticuerpo). El modelo consiste de una red bayesiana que representa la distribución conjunta de soluciones prometedoras y, posteriormente, utiliza este modelo para el muestreo de nuevas soluciones. El operador de selección es inspirado por el usado en NSGA-II basado en la dominancia de Pareto. Los anticuerpos $a \in A(t)$ que no son dominados por ningún

individuo tienen la posición 1. A los anticuerpos que solamente son dominados por individuos con posición 1 se les asigna la posición 2. Los anticuerpos que son dominados únicamente por los individuos con posición 1 y 2 obtienen la posición 3, y así sucesivamente. Después de seleccionar los anticuerpos con posiciones bajas, se eliminan los anticuerpos más parecidos y se generan nuevos de manera aleatoria

El Weight-Based Multiojective Artificial Immune System (WBMOAIS) [90] es un SIA multi-objetivo que sigue la estructura del SIA mono-objetivo opt-aiNET [91], el cual está basado en la Teoría de la red Inmune. Al igual que ACSAMO, utiliza una función agregativa de pesos aleatorios para definir la calidad de una solución. En este algoritmo se presupone que la población $B(t)$ es un conjunto de células y que cada una de ellas representa una solución al problema de optimización. Después del proceso de clonación y mutación (proporcional a la calidad), se realiza una búsqueda local sobre cada uno de los clones. La dirección de la búsqueda está dada por el valor de los pesos del clon. Las soluciones no dominadas encontradas son almacenadas (cuando se alcanza el tamaño máximo de la memoria, las soluciones más parecidas son eliminadas) y a partir de éstas se genera la nueva población. Cada cierto número de iteraciones, se generan nuevas células de manera aleatoria.

El Immunological Clonal Selection Algorithm (I-CSA) [92] es un sistema inmune artificial mono-objetivo basado en selección clonal probado por un amplio conjunto de funciones numéricas [93] para demostrar su capacidad de escapar de óptimos locales. Fue comparado con diversas variantes de evolución diferencial y algunos otros algoritmos bio-inspirados, mostrando un comportamiento competitivo. En I-CSA la intensidad de la hipermutación que sufren los linfocitos-B depende de la afinidad ψ que presentan con el antígeno. Un linfocito con alta afinidad anticuerpo-antígeno tiene una baja probabilidad de mutación. Para mantener la biodiversidad y evitar caer en óptimos locales, se integra el concepto de edad de una célula. Así los linfocitos viejos (que superan la edad máxima permitida, la cual es un parámetro definido por el usuario), son suprimidos de la población $B(t)$ y sustituidos por nuevos linfocitos generados aleatoriamente. La edad de cada célula es incrementada en cada iteración del algoritmo a excepción de la mejor solución (elitismo).

En la literatura revisada para el desarrollo de esta tesis, no se ha encontrado registro de SIAMOs que basen su esquema de selección en medidas de desempeño. Partiendo de la idea de que el uso de una medida de desempeño en el esquema de selección de un SIAMO en lugar de la dominancia de Pareto, permitiría al algoritmo lidiar con problemas de muchos objetivos, en esta tesis se propone lo que parece ser el primer SIAMO basado en un indicador. En el capítulo siguiente se explican los detalles del modelo propuesto.

Capítulo 4

Modelo propuesto

El sistema inmune artificial multi-objetivo que proponemos (SIAMO-DGI) está inspirado en el principio de selección clonal (abordado en la sección 2.1.2 en la pág 20) y emplea representación real y mutación proporcional a la calidad. Los indicadores DG y DGI son utilizados en el esquema de selección en lugar de la dominancia de Pareto, con lo que se intenta que SIAMO-DGI sea capaz de lidiar con problemas de muchos objetivos. En este capítulo se explica a detalle el algoritmo propuesto.

4.1. Descripción del algoritmo general

La figura 4.1 muestra los pasos principales del algoritmo propuesto, los cuales son explicados a continuación.

Definición del problema y los parámetros de entrada. Definir el espacio de búsqueda y las funciones objetivo $f(\cdot)$ (antígeno polivalente), el tamaño de la memoria inmunológica Nm , el número de clones de un linfocito al proliferarse N_c , el tamaño de la población N , el número máximo de respuestas del sistema inmune (iteraciones) T , cada cuántas iteraciones k se desea regenerar la población, el porcentaje p de soluciones que se desea generar aleatoriamente cada k iteraciones y los parámetros de mutación γ y n_m .

Inicializar la población. Generar una población inicial de linfocitos $A(0)$ de manera aleatoria. Cada uno de estos linfocitos representa una posible solución al POM.

Inicializar la población secundaria. La población secundaria $B(t)$ representa a la memoria inmunológica. Constituye el mecanismo de elitismo del algoritmo al almacenar las mejores soluciones encontradas a lo largo de las respuestas inmunes. $B(0)$ es inicializada con las soluciones no dominadas de $A(0)$

Evaluación en las funciones objetivo. La calidad de cada linfocito es proporcional a la calidad de los anticuerpos que produce. La forma de un anticuerpo producido

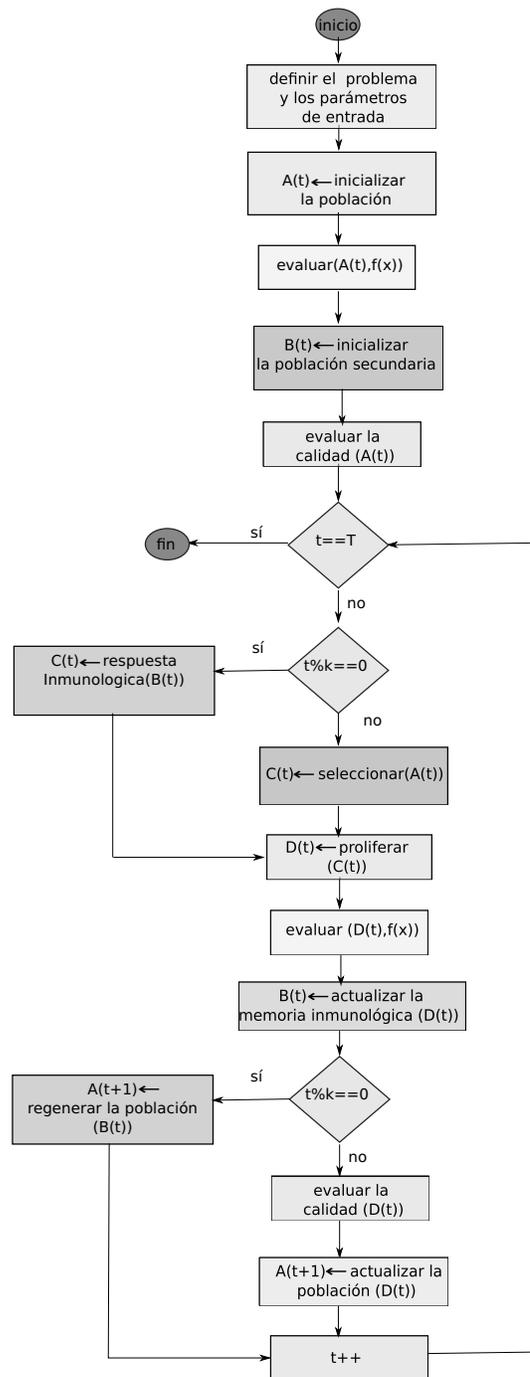


Figura 4.1: Algoritmo propuesto

por un linfocito está dada por su evaluación en las funciones objetivo $f(x)$.

Evaluación de calidad. La calidad de un linfocito, en nuestra propuesta está dada por su aportación a la respuesta inmunológica (aportación de una solución al

indicador). Se utilizan los indicadores DG y DGI para medir la afinidad antígeno-anticuerpo. El frente utilizado como referencia está compuesto por las soluciones almacenadas en $B(t)$. Se utiliza también la distancia de “crowding” como estimador de densidad afinidad anticuerpo-anticuerpo. La manera en que se define la calidad de una solución en nuestra propuesta se explica a detalle en la sección 4.2.

Selección. Se selecciona a la mitad de la población $A(t)$ mediante torneos binarios, con base en el siguiente algoritmo:

- Si $I_{DGI_x} > I_{DGI_y}$ entonces gana x :
 - Si no entonces Si $I_{DGI_y} > I_{DGI_x}$ entonces gana y :
 - Si no entonces Si $\text{crowding}_x > \text{crowding}_y$ entonces gana x :
 - ◊ Si no entonces gana y

donde:

I_{DGI_z} representa la aportación de $z \in A(t)$ al indicador DGI y crowding_z representa la distancia de “crowding” de $z \in A(t)$.

Respuesta Inmunológica. Cada k generaciones el mecanismo de selección es reemplazado por el siguiente algoritmo:

1. Agrupar las soluciones $b \in B(t)$ en $\frac{n}{4}$ cúmulos: algoritmo jerárquico ligado-completo (ver sección 4.3.2)
donde: n es igual al número de soluciones existentes en $B(t)$.
2. Seleccionar mediante el algoritmo de la ruleta [85] $\frac{N}{2}$ soluciones $b \in B(t)$ para ser proliferadas. La probabilidad de una solución b de ser elegida, es inversamente proporcional al número de soluciones que existen en el “cúmulo” al que pertenecen (las soluciones que pertenecen a los cúmulos con menos elementos, tienen una mayor probabilidad de ser elegidas para la proliferación).

Proliferación. La magnitud de la hipermutación que sufrirán los N_c clones de las soluciones elegidas para la proliferación depende de la afinidad que presentan con el antígeno polivalente. Así la probabilidad de mutación para un clon está dada por: $\alpha = e^{-f(x)}$ donde:

$f(x)$ es igual a I_{DG_x} normalizada $[0, 1]$,

siendo $f(x) = 1$ la distancia de la solución más cercana al frente de referencia.

Son dos las mutaciones utilizadas en el algoritmo propuesto:

- M_1 [92]: la variable x_i con $i \in [1, \dots, l]$ (donde l es la longitud del clon) escogida aleatoriamente para ser mutada de acuerdo a α es reemplazada por:
 $x_i^{(t+1)} = ((1 - \beta) \times x_i^{(t)}) + (\beta \times x_{random}^{(t)})$ donde:
 $x_{random}^{(t)} \neq x_i^{(t)}$ y $\beta \in [0, 1]$ son escogidos aleatoriamente.
- M_2 [94]: la variable x_i con $i \in [1, \dots, l]$ (donde l es la longitud del clon) escogida aleatoriamente para ser mutada de acuerdo a α es reemplazada por:
 $x_i^{(t+1)} = x_i^{(t)} + (\limSuperior_i - \limInferior_i) \times \delta_i$ donde:
 $x_i \in [\limSuperior_i, \limInferior_i]$,

$$\delta_i = \begin{cases} (2r_i)^{\frac{1}{n_m+1}} - 1 & \text{si } r_i < 0.5 \\ 1 - |2(1 - r_i)|^{\frac{1}{n_m+1}} & \text{de otro modo} \end{cases} \quad \text{donde:}$$

r_i representa un número aleatorio entre 0 y 1,

Estas dos mutaciones se aplican de acuerdo al siguiente algoritmo:

1. $b := \text{flip}(\gamma)$
2. Para cada $i \in [1, \dots, l]$
 - a) Si $\text{flip}(\alpha)$ entonces:
 - 1) Si b entonces $M_1(x_i)$ de lo contrario:
 $a' M_2(x_i)$

donde:

l es la longitud de x ,

$\gamma \in [0, 1]$,

$\text{flip}(\lambda) = \text{verdadero}$ con una probabilidad de $\lambda \in [0, 1]$.

Actualización de la población principal. La solución n y la solución $n + 1$ en la población $A(t)$, son sustituidas en la población $A(t + 1)$ por las dos mejores soluciones resultantes de la variación de la solución n (en la competencia participan tanto las soluciones mutadas como la solución original).

Actualización de la memoria inmunológica. La población secundaria $B(t)$ es actualizada de acuerdo al siguiente algoritmo:

1. Eliminar de $B(t)$ las soluciones que son dominadas por alguna solución resultante de la proliferación.
2. **Si** el número de soluciones en $B(t)$ es menor a N_{mreal} **entonces** almacenar las soluciones que no son dominadas por ninguna solución de $B(t)$ **si no**:
 - a) Reducir la población secundaria (ver sección 4.3):

- 1) Reducir $B(t)$ mediante distancia de Mahalanobis.
- 2) Agrupar las soluciones $b \in B(t)$, en N_m cúmulos (algoritmo jerárquico ligado-completo).
- 3) Conservar las N_m soluciones de $B(t)$ más cercanas a los centros de los cúmulos.

Regeneración de la población. Cada k iteraciones la población es regenerada como sigue:

1. Generar pN soluciones aleatorias.
2. Seleccionar las $(N - pN)$ soluciones más representativas (usando algoritmo de agrupamiento de ligado-completo) de $B(t)$.
3. Formar $A(t + 1)$ con las soluciones generadas en 1 y las seleccionadas en 2.

4.2. Aportación de una solución a los indicadores DG y DGI en SIAMO-DGI

La figura 4.2 ilustra la manera en que se define la calidad de una solución en nuestra propuesta:

- En el caso del indicador DG la aportación de una solución x , está dada por la distancia euclidiana entre x y su vector más cercano en la población $B(t)$ (en el espacio de los objetivos). Esta aportación se denota por I_{DGx} y es un aspecto que desea minimizarse. En el ejemplo planteado en la figura 4.2 $I_{DGx} = c$ porque x se encuentra a una distancia c de su vector más cercano en $B(t)$.
- En el caso del indicador DGI la aportación de una solución x , está dada por el número de vectores de la población $B(t)$ para los que x representa el vector más cercano (en el espacio de los objetivos). Esta aportación se denota por I_{DGIx} y es un aspecto que desea maximizarse. En el ejemplo planteado en la figura 4.2 $I_{DGIx} = 2$ porque x representa el vector más cercano para dos vectores en $B(t)$.
- La distancia de crowding para una solución x se denota por crowding_x y es un aspecto que desea maximizarse.
En el ejemplo planteado en la figura 4.2 $\text{crowding}_x = (a + b)/2$.

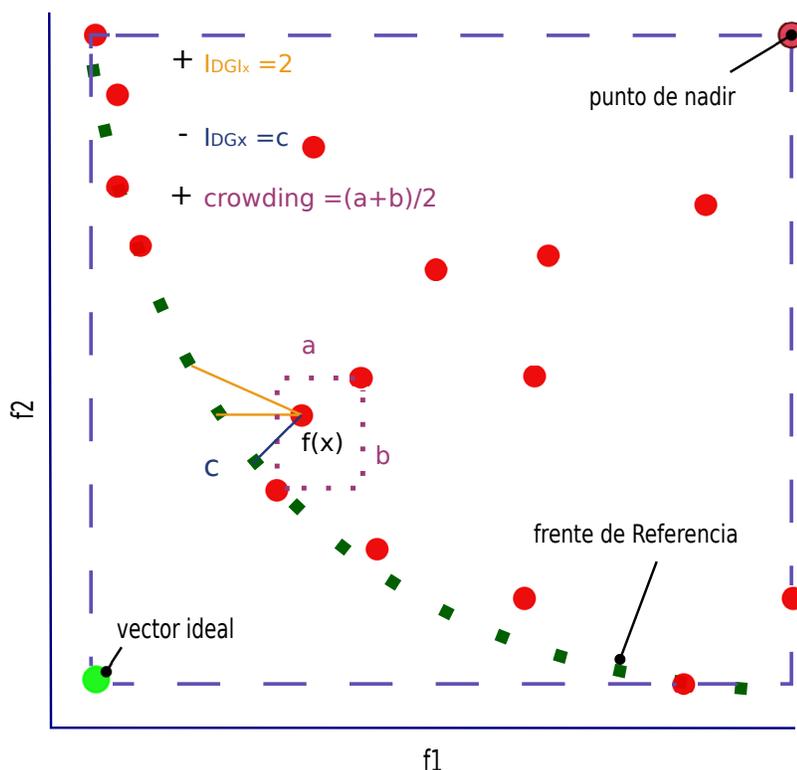


Figura 4.2: Calidad de una solución en SIAMO-DGI

4.3. Reducción de la memoria inmunológica

Debido a que un POM puede tener soluciones infinitas pero la población secundaria $B(t)$ tiene un tamaño fijo, es necesario contar con un mecanismo de reducción una vez que el tamaño máximo de $B(t)$ ha sido alcanzado.

En el algoritmo son utilizados dos tamaños para la población secundaria $B(t)$. El primero N_m representa el número de soluciones no dominadas que el usuario desea obtener al terminar la ejecución. El segundo $N_{m_{\text{real}}} = 1.25N_m$ es en realidad el tamaño de memoria que se maneja en el algoritmo y representa el tamaño máximo que puede alcanzar $B(t)$ antes de ser reducido a N_m . El método empleado en este sistema para llevar a cabo la reducción de $B(t)$ combina la distancia de Mahalanobis con un algoritmo de agrupamiento.

4.3.1. Reducción mediante distancia de Mahalanobis

Para cada $b \in [B(t)]$, ψ_b es calculado de acuerdo a la distancia de Mahalanobis entre b y el vector ideal:

$$\psi_b = \sqrt{(f(b) - f_{\text{ideal}})^T S^{-1} (f(b) - f_{\text{ideal}})}$$

donde:

$$S = \frac{1}{n} X^T X^T$$

donde n = tamaño actual de $B(t)$ y $X_j = f_j(b) - f_j(\bar{x})$

Eliminar un máximo de $0.25N_m$ (se desea reducir el tamaño de la memoria de $N_{m_{\text{real}}}$ a N_m , donde $N_{m_{\text{real}}} = 1.25N_m$) soluciones de $B(t)$ para las cuales $\psi_b - \bar{\psi} > u\sigma_\psi$ donde $\bar{\psi}$ y σ_ψ representan la media y la desviación estándar, respectivamente y u es un parámetro establecido por el usuario. Esta primer reducción es utilizado con el fin de eliminar soluciones difícilmente dominables (soluciones resistentes a ser dominadas). Estas soluciones serían privilegiadas por el algoritmo de agrupamiento al encontrarse alejadas de las demás soluciones en $B(t)$, pero podrían afectar la búsqueda al permanecer en $B(t + 1)$ en lugar de soluciones que representan mejores compromisos.

4.3.2. Algoritmo de agrupamiento

Se conoce como algoritmo de agrupamiento a la clasificación no supervisada de patrones, dentro de cúmulos [95]. Un algoritmo de agrupamiento puede definirse como: “El proceso de organizar objetos en grupos cuyos miembros son similares de alguna manera”. Un cúmulo sería entonces una colección de objetos, los cuales son “similares” entre ellos y “no similares” a los objetos pertenecientes a otros cúmulos.

Los cúmulos pueden dividirse en estrictos y difusos (ver figura 4.3). En los estrictos cada patrón pertenece a un solo cúmulo. En los difusos se asigna grados de membresía en varios cúmulos a cada patrón.

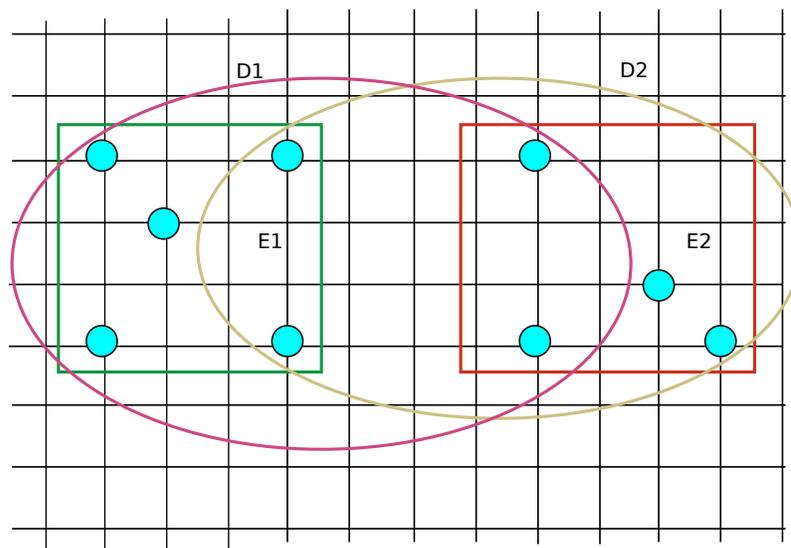


Figura 4.3: Cúmulos estrictos vs difusos

Los algoritmos de agrupamiento pueden dividirse en:

- **Jerárquicos:** dan como resultado el dendograma que representa la agrupación de los patrones y la similitud entre los niveles. Son costosos, lo que los vuelven difíciles de usar con grupos grandes de datos. A continuación se muestra el algoritmo general de agrupamiento jerárquico:
 1. Considerar cada patrón como un cúmulo.
 2. Construir una matriz de “similitud” entre los diferentes cúmulos.
 3. **Mientras** (\neg criterio de paro) **hacer:**
 - a) Combinar los dos cúmulos más cercanos.
 - b) Actualizar la matriz de “similitud”.

Las tres versiones más populares de este tipo de algoritmos son: ligado-simple, ligado-completo y ligado-promedio. La diferencia entre estas versiones reside en la manera en que se mide la “similitud” entre dos cúmulos:

- **Ligado-simple:** la “similitud” entre dos cúmulos está dada por la distancia entre sus elementos más cercanos.
- **Ligado-completo:** la “similitud” entre dos cúmulos está dada por la distancia entre sus elementos más lejanos.
- **Ligado-promedio:** la “similitud” entre dos cúmulos está dada por la distancia entre sus centros.

En la figura 4.4 se ilustra el agrupamiento jerárquico.

- **Particionamiento:** obtienen una partición simple de los datos. Es necesario definir el número de cúmulos iniciales. Son sensibles a los estados iniciales. Son poco costosos, lo que los vuelve una opción en grandes grupos de datos. A continuación se muestra el algoritmo general de particionamiento:
 1. Seleccionar una partición inicial.
 2. **Mientras** (\neg criterio de paro) **hacer:**
 - a) Asignar cada patrón a su centro de cúmulo más cercano.
 - b) Calcular los nuevos centros de cúmulos.
 3. Combinar y dividir los cúmulos, basándose en algún umbral.

En la figura 4.5 se ilustra el agrupamiento de particionamiento.

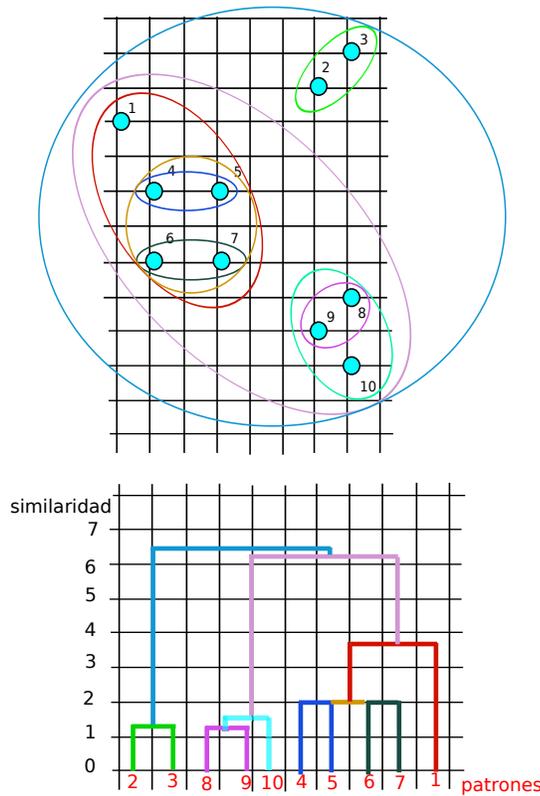


Figura 4.4: Ejemplo de agrupamiento jerárquico

Una vez que R datos han sido agrupados en r cúmulos, el número de datos puede ser reducido conservando solo los datos más representativos de cada “cúmulo” (por ejemplo: para cada “cúmulo” elegir el elemento más cercano a su centro).

Los algoritmo jerárquicos son más versátiles que los algoritmos de particionamiento, pero por su alto costo no pueden aplicarse a grandes conjuntos de datos. Entre los algoritmos jerárquicos los algoritmos de ligado-completo y ligado-promedio han mostrado un mejor desempeño que los algoritmos de ligado-simple [95].

En esta tesis se utiliza un algoritmo de agrupamiento jerárquico de ligado completo. Se eligió un algoritmo jerárquico pues no es necesario trabajar con un gran número de datos. Aún cuando las versiones de ligado-completo y ligado-promedio han mostrado un desempeño similar, el ligado-completo no requiere del calculo del centro de los clústers, por lo que se ha elegido está versión.

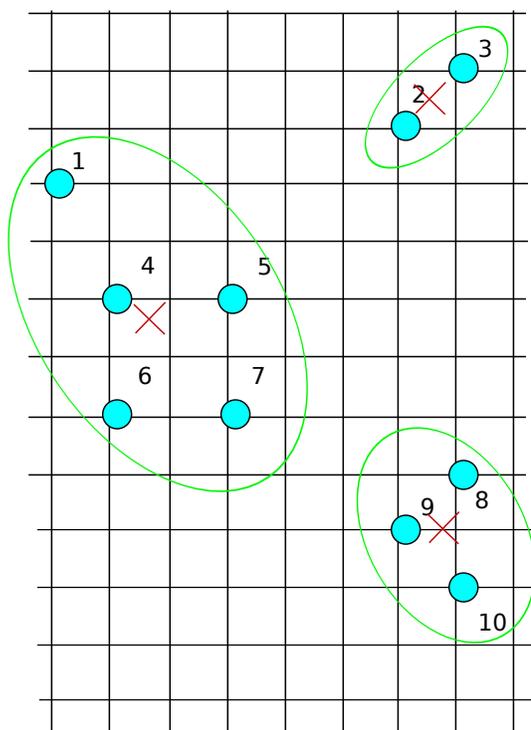


Figura 4.5: Ejemplo de particionamiento

Capítulo 5

Validación

En este capítulo el algoritmo propuesto es comparado con tres algoritmos del estado del arte: NSGA-II [4], HYPE [5] y VAIS [6].

5.1. Conjunto de problemas ZDT (Zitzler-Deb-Thiele)

Para validar el funcionamiento del algoritmo propuesto con problemas de dos objetivos, hemos seleccionado el conjunto de problemas ZDT (ver anexo A en la página 67). Se hicieron 20 ejecuciones, permitiendo un máximo de 300,000 evaluaciones de la función objetivo. La configuración de los parámetros para cada algoritmo es mostrada en la tabla 5.1. La tabla 5.2 muestra las medias obtenidas por cada algoritmo para los indicadores espaciado, distancia generacional invertida (DGI) e hipervolumen (HV). El punto de referencia para el calculo del HV ha sido puesto como (1.1, 1.1) para todos los problemas.

SIAMO-DGI		VAIS		NSGA-II		HYPE	
N	60	N	100	N	100	N	100
N_m	100	N_m	100				
N_c	5	N_c	10				
p	10	p	20				
k	10	ciclos _{int}	5				
γ	0.3						
n_m	10	β	0.1	prob _{mut}	$\frac{1}{ \vec{x} }$	prob _{mut}	$\frac{1}{ \vec{x} }$
				n_m	10	n_m	15
				prob _{rec}	0.9	prob _{rec}	1.0
				n_r	10	n_r	20
u	3.0σ					bounds	2000
						M	1000

Tabla 5.1: Configuración de parámetros

Resultados

NSGA-II muestra los mejores resultados para el conjunto de problemas ZDT. Nuestra propuesta es el algoritmo que más se acerca a estos resultados, mejorando el desempeño de VAIS y HYPE.

Cabe destacar que en este conjunto de problemas HYPE a diferencia de los otros algoritmos, no aumenta la calidad del frente encontrado conforme se aumentan el número de generaciones. Esto es ilustrado en la figura 5.1. En la generación 50, el HV para el frente encontrado por HYPE para el problema ZDT1 es de 0.836199; en la generación 200, es de 0.777087, en la generación 1000 es de 0.646381 y en la generación 3000 es de 0.735056. Esto indica un claro deterioro de las soluciones conforme se aumenta el número de iteraciones.

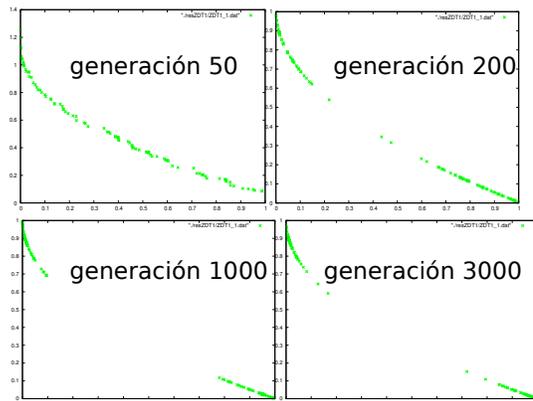


Figura 5.1: Desempeño de HYPE en ZDT1

El problema más complicado de este conjunto para SIAMO-DGI es ZDT4. La dificultad de ZDT4 es alta al ser un problema multifrontal. En nuestra propuesta la población secundaria $B(t)$ es utilizada como frente de referencia. Para que $B(t)$ sea capaz de guiar la búsqueda de manera exitosa, es necesario que represente una aproximación al frente verdadero y que cuente con soluciones a lo largo de todo el frente. Tomando en cuenta estas consideraciones se implementó el mecanismo de respuesta inmunológica (ver capítulo 4 en la página 45) que busca generar soluciones en las partes más despobladas del frente.

Desafortunadamente en el caso de ZDT4 unos cuantos individuos (e incluso a veces un solo individuo) suelen dominar a todas las soluciones de las poblaciones $A(t)$ y $B(t)$, por lo que $B(t)$ queda formada por muy pocos individuos, los cuales la mayoría de las veces se encuentran ubicados en una pequeña porción del frente. Cuando esto ocurre, $B(t)$ no es capaz de guiar la búsqueda y el algoritmo necesita un gran número de iteraciones para generar soluciones a lo largo de todo el frente.

En la tabla se muestran los resultados de las pruebas descritas en la sección ante-

rior. Para cada algoritmo se muestra la media (arriba) y la varianza (abajo), de las 20 ejecuciones.

Num. evaluaciones	Espaciado	DGI	HV
ZDT1			
SIAMO-DGI			
20,000	0.009126 0.000472	0.002036 0.000389	0.851932 0.008299
300,000	0.009358 0.000290	0.000341 0.000018	0.867896 0.000411
NSGA-II			
20,000	0.007211 0.000365	0.00256 0.000006	0.860579 0.000150
300,000	0.007027 0.000564	0.000257 0.000014	0.870581 0.000207
HYPE			
20,000	0.015146 0.006643	0.001247 0.000176	0.848047 0.003424
300,000	0.017005 0.005057	0.008045 0.001436	0.696739 0.041192
VAIS			
20,000	0.022547 0.001122	0.026250 0.002179	0.135354 0.046447
300,000	0.012537 0.000846	0.002024 0.000084	0.800560 0.002437
ZDT2			
SIAMO-DGI			
20,000	0.008914 0.001361	0.001535 0.000754	0.499372 0.015663
300,000	0.010595 0.000759	0.000308 0.000021	0.534650 0.001746
NSGA-II			
20,000	0.007428 0.000332	0.00261 0.000010	0.527422 0.000165
300,000	0.007435 0.000590	0.000264 0.000010	0.537457 0.000214
HYPE			
20,000	0.008329 0.005474	0.013849 0.015391	0.351142 0.197319
300,000	0.004170 0.005108	0.024029 0.010627	0.197894 0.110695
VAIS			
20,000	0.021729 0.003829	0.037035 0.002899	- -
300,000	0.018129 0.003426	0.004657 0.002306	0.377709 0.060713
ZDT3			
SIAMO-DGI			
20,000	0.008995 0.001555	0.004657 0.000745	0.870163 0.013073
300,000	0.009763 0.002596	0.000812 0.000062	0.947321 0.001743
NSGA-II			
20,000	0.007703 0.000746	0.000571 0.000031	0.953995 0.000108
300,000	0.007342 0.000926	0.000573 0.000019	0.954026 0.000160
HYPE			
20,000	0.071923 0.041882	0.047460 0.000582	0.198213 0.008411
300,000	0.042067 0.026925	0.058110 0.002963	0.073855 0.024306
VAIS			
20,000	0.023493	0.050713	0.122053

	0.003011	0.002017	0.054690
300,000	0.018175	0.009938	0.781017
	0.002108	0.002878	0.058124
ZDT4			
SIAMO-DGI			
20,000	0.187360	0.027548	0.144586
	0.271947	0.009944	0.017360
300,000	0.009013	0.002661	0.637150
	0.002555	0.002839	0.021107
NSGA-II			
20,000	0.00028507	0.000898	0.622293
	0.002650	0.001359	0.054217
300,000	0.008120	0.000195	0.653514
	0.000817	0.000013	0.000362
HYPE			
20,000	0.357648	0.022541	0.169025
	0.493816	0.019517	0.089616
300,000	0.017348	0.023972	0.177519
	0.018919	0.016933	0.159917
VAIS			
20,000	0.205177	0.022910	-
	0.078622	0.001735	-
300,000	0.121414	0.022910	-
	0.067308	0.000000	-
ZDT6			
SIAMO-DGI			
20,000	0.005561	0.000578	0.490608
	0.000516	0.000059	0.001936
300,000	0.007969	0.000314	0.502240
	0.000561	0.000022	0.000278
NSGA-II			
20,000	0.005786	0.000319	0.498921
	0.000380	0.000018	0.000810
300,000	0.007696	0.000276	0.502716
	0.000438	0.000018	0.000274
HYPE			
20,000	0.008676	0.000949	0.480004
	0.000563	0.000041	0.001490
300,000	0.019027	0.013055	0.317950
	0.021190	0.003271	0.047552
VAIS			
	0.170224	0.036939	0.138893
	0.037236	0.006167	0.024164
300,000	0.007079	0.001799	0.458520
	0.004049	0.000393	0.010156

Tabla 5.2: Resultados ZDT

5.2. Conjunto de problemas DTLZ (Deb-Thiele-Laumanns-Zitzler)

Para validar el funcionamiento del algoritmo propuesto con problemas de tres objetivos, hemos seleccionado el conjunto de problemas DTLZ (ver anexo A). Se hicieron 20 ejecuciones, permitiendo un máximo de 120,000 evaluaciones. La configuración de los parámetros para cada algoritmo es mostrada en la tabla 5.3. La tabla 5.4 muestra las medias obtenidas por cada algoritmo para los indicadores espaciado, DGI y HV. El punto de referencia para el calculo del HV ha sido puesto como (1.1, 1.1, 1.1) para DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5 y DTLZ6 y como (1.1, 1.1, 7.0) para

DTLZ7.

SIAMO-DGI		VAIS		NSGA-II		HYPE	
N	60	N	100	N	100	N	100
N_m	100	N_m	100				
N_c	5	N_c	10				
p	10	p	20				
k	10	ciclos _{int}	5				
γ	0.7						
n_m	5	β	0.1	prob _{mut}	$\frac{1}{ \bar{x} }$	prob _{mut}	$\frac{1}{ \bar{x} }$
				n_m	10	n_m	15
				prob _{rec}	0.9	prob _{rec}	1.0
				n_r	10	n_r	20
u	DTLZ2,DTLZ6= σ					bounds	2000
	DTLZ3,DTLZ5= 1.5σ					M	1000
DTLZ7= 6.0σ	DTLZ1,DTLZ4= 2.5σ						

Tabla 5.3: Configuración de parámetros

Resultados

Al igual que en el conjunto de problemas ZDT, en DTLZ NSGA-II muestra los mejores resultados, superando el desempeño de los demás algoritmos en 5 de los 7 problemas usados (en DTLZ1 HYPE tiene el mejor desempeño y en DTLZ4 SIAMO-DGI tiene el mejor desempeño).

Nuestra propuesta supera el desempeño de VAIS en todos los problemas de este conjunto y a HYPE en 5 de los 7 problemas. Cabe destacar que en la literatura consultada para el desarrollo de esta tesis no se encontraron resultados publicados de SIAMOs puros para este conjunto de problemas. Los únicos SIAMOs que reportan buenos resultados para DTLZ usan el operador de recombinación.

El problema DTLZ3 es donde SIAMO-DGI muestra el peor desempeño. En este problema (al igual que en ZDT4) la memoria inmunológica $B(t)$ no proporciona la suficiente información para guiar la búsqueda.

En la tabla se muestran los resultados de las pruebas descritas en la sección anterior.

	Espaciado	DGI	HV
DTLZ1			
	SIAMO-DGI		
20,000	6.339024	0.018045	0.338744
	7.229140	0.012890	0.211225
120,000	0.175952	0.001898	1.239065

		0.158659	0.000269	0.016563
	NSGA2			
20,000	11.336744	0.229139	-	
	6.731222	0.095627	-	
120,000	0.021405	0.001008	1.283021	
	0.002584	0.000299	0.008172	
	HYPE			
20,000	3.808263	0.011744	0.773731	
	3.889371	0.011883	0.617895	
120,000	5.059955	0.001606	1.287054	
	5.377280	0.000280	0.003612	
	VAIS			
20,000	7.893521	0.008365	-	
	0.423279	0.000002	-	
120,000	5.627976	0.358280	-	
	0.927964	0.537620	-	
DTLZ2				
	SIAMO-DGI			
20,000	0.052516	0.002949	0.659790	
	0.006405	0.000356	0.009516	
120,000	0.041630	0.003952	0.687622	
	0.005029	0.000195	0.007967	
	NSGA-II			
20,000	0.060428	0.031644	0.684011	
	0.004524	0.000003	0.004527	
120,000	0.056897	0.002467	0.688329	
	0.005081	0.000172	0.007712	
	HYPE			
20,000	0.057213	0.031686	0.631123	
	0.015255	0.000026	0.009258	
120,000	0.047148	0.031628	0.657290	
	0.010514	0.000001	0.009036	
	VAIS			
20,000	0.066306	0.034110	0.401734	
	0.005240	0.000918	0.051189	
120,000	0.070395	0.033752	0.419405	
	0.006762	0.000488	0.031144	
DTLZ3				
	SIAMO-DGI			
20,000	48.670612	0.279007	-	
	21.466793	0.418145	-	
120,000	2.116720	0.132744	0.142128	
	1.367775	0.097518	0.020459	
	NSGA-II			
20,000	16.594915	0.419757	-	
	12.229084	0.141463	-	
120,000	0.942519	0.001799	0.685249	
	3.623992	0.000145	0.013269	
	HYPE			
20,000	27.641482	0.405120	0.000000	
	13.277907	0.130077	0.000000	
120,000	1.159489	0.006239	0.497665	
	0.446552	0.001871	0.098881	
	VAIS			
20,000	33.301401	0.022361	-	
	0.909121	0.000012	-	
120,000	24.865371	0.022361	-	
	2.655899	0.000301	-	
DTLZ4				
	SIAMO-DGI			
20,000	0.047131	0.005883	0.634585	
	0.008528	0.003707	0.036778	
120,000	0.051907	0.002563	0.704908	
	0.003403	0.000077	0.006899	

	NSGA-II		
20,000	0.057959	0.002382	0.693428
	0.002432	0.000097	0.003425
120,000	0.053833	0.003927	0.669067
	0.013089	0.006804	0.125876
	HYPE		
20,000	0.027529	0.015323	0.536457
	0.004166	0.007047	0.130911
120,000	0.037180	0.014097	0.538911
	0.026628	0.008414	0.122519
	VAIS		
20,000	0.037180	0.014097	0.538911
	0.026628	0.008414	0.122519
120,000	0.069126	0.008433	0.523907
	0.016718	0.000956	0.025295
DTLZ5			
	SIAMO-DGI		
20,000	0.024321	0.000127	0.431653
	0.008904	0.000018	0.001460
120,000	0.019335	0.000118	0.431694
	0.004129	0.000020	0.003991
	NSGA-II		
20,000	0.009670	0.000071	0.437314
	0.000775	0.000002	0.000054
120,000	0.009506	0.000068	0.437688
	0.001169	0.000004	0.000227
	HYPE		
20,000	0.018804	0.000286	0.420235
	0.006389	0.000062	0.002743
120,000	0.020420	0.000854	0.388012
	0.015403	0.000438	0.023425
	VAIS		
20,000	0.024435	0.000650	0.363991
	0.004920	0.000190	0.017778
120,000	0.027265	0.000676	0.356416
	0.007289	0.000107	0.017000
DTLZ6			
	SIAMO-DGI		
20,000	0.337663	0.049111	-
	0.029676	0.003445	-
120,000	0.039749	0.000949	0.313003
	0.020156	0.000334	0.036743
	NSGA-II		
20,000	0.076758	0.006152	0.007483
	0.017774	0.000504	0.006595
120,000	0.025941	0.000741	0.346891
	0.020336	0.000273	0.031123
	VAIS		
20,000	0.536358	0.046849	-
	0.059745	0.018852	-
120,000	0.537734	0.055373	-
	0.063338	0.006030	-
	HYPE		
20,000	0.249988	0.020844	-
	0.052300	0.004782	-
120,000	0.028240	0.002037	0.270736
	0.012307	0.000838	0.050936
DTLZ7			
	SIAMO-DGI		
20,000	0.319314	0.011319	1.820516
	0.245733	0.002060	0.121808
120,000	0.136623	0.001643	2.746512
	0.094129	0.000239	0.035826
	NSGA-II		

20,000	0.042335	0.005548	2.568203
	0.013890	0.003151	0.280277
120,000	0.059407	0.002236	2.864371
	0.015931	0.001993	0.187715
HYPE			
20,000	0.150170	0.003745	2.558153
	0.107624	0.000842	0.136141
120,000	0.129397	0.005552	2.236407
	0.098280	0.000749	0.117804
VAIS			
20,000	0.086620	0.007141	1.225421
	0.016613	0.001377	0.436853
120,000	0.081384	0.007216	1.346144
	0.013522	0.003141	0.454770

Tabla 5.4: Resultados DTLZ

5.3. Desempeño de nuestra propuesta con muchos objetivos

Para validar el funcionamiento del algoritmo propuesto con problemas de muchos objetivos, hemos seleccionado la función de prueba DTLZ2. Se hicieron 20 ejecuciones (permitiendo un máximo de 120,000 evaluaciones a la función objetivo) para cada algoritmo desde 3 hasta 10 objetivos. La configuración de los parámetros para cada algoritmo es mostrada en la tabla 5.5. La tabla 5.6 muestra las medias obtenidas por cada algoritmo para los indicadores de HV y espaciado, así como el tiempo de ejecución promedio. El punto de referencia para el cálculo del HV ha sido puesto como (1.1).

SIAMO-DGI		VAIS		NSGA-II		HYPE	
N	60	N	100	N	100	N	100
N_m	100	N_m	100				
N_c	5	N_c	10				
p	10	p	20				
k	20	ciclos _{int}	5				
γ	0.8						
n_m	5	β	0.1	prob _{mut}	$\frac{1}{ \bar{x} }$	prob _{mut}	$\frac{1}{ \bar{x} }$
				n_m	10	n_m	15
				prob _{rec}	0.9	prob _{rec}	0.5
				n_r	10	n_r	20
u	σ					bounds	2000
						M	1000

Tabla 5.5: Configuración de parámetros

Resultados

NSGA-II es el algoritmo más veloz en esta comparación, sin embargo, al utilizar un esquema de selección basado por completo en la dominancia de Pareto, su desempeño se ve gravemente afectado al aumentar el número de objetivos.

VAIS el segundo algoritmo más veloz, pero no logra tener resultados competitivos para el problema DTLZ2. Al igual que NSGA-II, VAIS está basado en la dominancia de Pareto, por lo que tampoco es capaz de lidiar con problemas con muchos objetivos.

HYPE es el algoritmo que requiere un mayor tiempo de ejecución en esta comparación, pues está basado en HV (como se menciona en el capítulo 3, el cálculo del HV es NP-Completo). Sin embargo, al utilizar la aportación de una solución al indicador de HV en lugar de la dominancia de Pareto como criterio de selección, es capaz de lidiar con problemas de muchos objetivos.

Nuestra propuesta obtiene resultados competitivos con HYPE en problemas de muchos objetivos (es capaz de superar el valor del indicador de HV obtenido por HYPE en la mayoría de las pruebas realizadas). Sin embargo, al estar basada en el indicador DGI y DG en lugar del indicador de HV, nuestra propuesta es en promedio 10 veces más rápida que HYPE.

En la tabla 5.6 se muestran los resultados de las pruebas descritas en la sección anterior. Los valores de HV y tiempo de ejecución han sido normalizados, para permitir ver de manera más clara el compromiso que ofrece cada algoritmo entre estos dos criterios. En el caso del HV el valor de 1.0 corresponde al algoritmo que obtuvo el mejor HV promedio, mientras que en el caso de tiempo de ejecución el valor de 1.0 corresponde al algoritmo más lento. La gráfica 5.2 muestra los tiempos de ejecución para los resultados reportados en la tabla 5.6.

	Espaciado	HV	Tiempo de ejecución
DTLZ2 3 objetivos			
SIAMO-DGI	0.041630	0.998	0.136
NSGA-II	0.056897	1.0	0.009
HYPE	0.041363	0.812	1.0
VAIS	0.070499	0.597	0.021
DTLZ2 4 objetivos			
SIAMO-DGI	0.090478	1.0	0.10911
NSGA-II	0.112374	0.998	0.005
HYPE	0.085520	0.916	1.0
VAIS	0.210365	0.129	0.011
DTLZ2 5 objetivos			
SIAMO-DGI	0.149456	0.974	0.071246
NSGA-II	0.192616	0.896	0.004
HYPE	0.126667	1.0	1.0
VAIS	0.321399	0.107	0.015
DTLZ2 6 objetivos			
SIAMO-DGI	0.175671	1.0	0.095612
NSGA-II	0.402108	0.238	0.004
HYPE	0.243933	0.909	1.0
VAIS	0.409821	0.111	0.0157831

DTLZ2 7 objetivos			
SIAMO-DGI	0.205632	1.0	0.097
NSGA-II	0.632541	0.151	0.004
HYPE	0.172016	0.952	1.0
VAIS	0.492278	0.083	0.027
DTLZ2 8 objetivos			
SIAMO-DGI	0.215435	1.0	0.108
NSGA-II	0.767139	0.18281	0.0028
HYPE	0.103517	0.955	1.0
VAIS	0.494311	0.091	0.020
DTLZ2 9 objetivos			
SIAMO-DGI	0.233273	1.0	0.111463
NSGA-II	0.919554	0.2299	0.00357
HYPE	0.152512	0.957	1.0
VAIS	0.563987	0.119972	0.02202
DTLZ2 10 objetivos			
SIAMO-DGI	0.233916	1.0	0.11118
NSGA-II	1.042842	0.22816	0.0034729
HYPE	0.132134	0.99748	1.0
VAIS	0.697885	0.13286	0.02378

Tabla 5.6: Resultados de escalabilidad

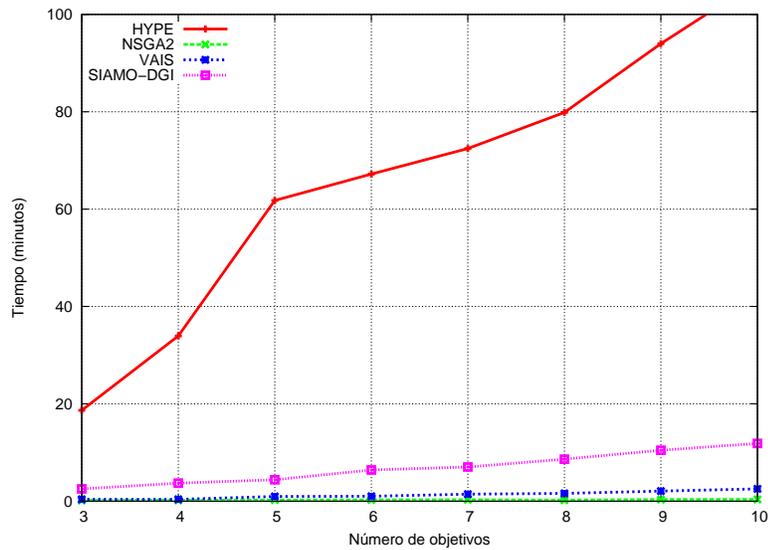


Figura 5.2: Grafica de tiempos de ejecución

Capítulo 6

Conclusiones

En esta tesis se propuso un nuevo sistema inmune artificial multi-objetivo (llamado SIAMO-DGI) el cual está basado en la medida de desempeño denominada distancia generacional invertida (DGI). El sistema propuesto respeta la metáfora del sistema inmune al ser inspirado por el modelo de selección clonal.

El funcionamiento de SIAMO-DGI fue validado con los conjuntos de problemas ZDT y DTLZ y comparado con los algoritmos basados en la dominancia de Pareto: NSGA-II y VAIS, así como un algoritmo basado en el hipervolumen: HYPE.

De acuerdo a las pruebas realizadas en este trabajo de tesis, se concluye que SIAMO-DGI es capaz de obtener buenos resultados en problemas con pocos objetivos (2 y 3) al mostrar un desempeño ligeramente inferior a NSGA-II (el cual obtuvo los mejores resultados para este tipo de problemas). Además el algoritmo propuesto constituye un buen compromiso entre tiempo de ejecución y calidad del frente encontrado, para problemas de muchos objetivos, al obtener en la mayoría de la veces mejores valores de HV que NSGA-II, VAIS y HYPE en un tiempo razonable, para más de tres objetivos.

Dado que tanto HYPE como SIAMO-DGI mostraron un mejor desempeño que NSGA-II y VAIS en los problemas con muchos objetivos, podemos concluir también, que el uso de indicadores en el mecanismo de selección de un algoritmo multi-objetivo, resulta más exitoso en este tipo de problemas, que el criterio de dominancia de Pareto.

Los indicadores utilizados en este trabajo de tesis (distancia generacional y distancia generacional invertida) no son compatibles con la optimilidad de Pareto. Sin embargo, a pesar de ello, mostraron ser capaces de guiar la búsqueda adecuadamente en problemas con muchos objetivos.

6.1. Trabajo a futuro

En esta propuesta la población secundaria $B(t)$ es utilizada como frente de referencia para medir la aportación de las soluciones en la población $A(t)$ a los indicadores DG y DGI, con el fin de seleccionar los linfocitos que serán proliferados. Dado que $B(t)$ juega un papel muy importante en nuestra propuesta, se podría probar con algoritmos diferentes de archivado y búsquedas locales que permitan generar soluciones en las partes más despobladas de $B(t)$. La población secundaria no siempre es capaz de guiar la búsqueda pues algunas veces está conformada por muy pocas soluciones. En este caso podría ser útil construir un frente de referencia a partir de $A(t)$, que aún cuando no asegure respetar la forma del frente verdadero, sirva para continuar la búsqueda hasta que $B(t)$ represente una aproximación a éste.

La mutación es el operador principal en los sistemas inmunes artificiales puros (el operador de cruza no está permitido). En este trabajo de tesis se combinan dos tipos de mutaciones: la propuesta en [92] y la propuesta en [94]; sin embargo, sería conveniente el probar con otros tipos de operadores de mutación.

Algunos de los sistemas inmunes artificiales actuales, sacrifican la metáfora del sistema inmune al incluir el operador de recombinación, con el fin de lidiar con conjuntos de problemas como ZDT y DTLZ. Podría resultar interesante comparar el desempeño del algoritmo propuesto con una variante que incluya el uso del operador de recombinación.

La escalabilidad del sistema propuesto sólo ha sido probada con DTLZ2, por lo que es necesario hacer un mayor número de pruebas de escalabilidad con diferentes problemas.

Apéndice A

Funciones de Prueba

A.1. Conjunto de problemas ZDT (Zitzler-Deb-Thiele)

Estas funciones están descritas en [96].

ZDT1 tiene un frente de Pareto convexo:

$$\begin{aligned}f_1(x) &= x_1 \\f_2(x, g) &= g(x) \left(1 - \sqrt{\frac{f_1}{g(x)}} \right) \\g(x) &= 1 + \frac{9}{n-1} \sum_{i=2}^n x_i\end{aligned}$$

donde $n = 30$, y $x_i \in [0, 1]$. El $F_{\text{verdadero}}$ es formado con $g(x) = 1$ y es mostrado en la figura A.1.

ZDT2 tiene un frente de Pareto cóncavo:

$$\begin{aligned}f_1(x) &= x_1 \\f_2(x, g) &= g(x) \left(1 - \left(\frac{f_1}{g(x)} \right)^2 \right) \\g(x) &= 1 + \frac{9}{n-1} \sum_{i=2}^n x_i\end{aligned}$$

donde $n = 30$, y $x_i \in [0, 1]$. El $F_{\text{verdadero}}$ es formado con $g(x) = 1$ y es mostrado en la figura A.2..

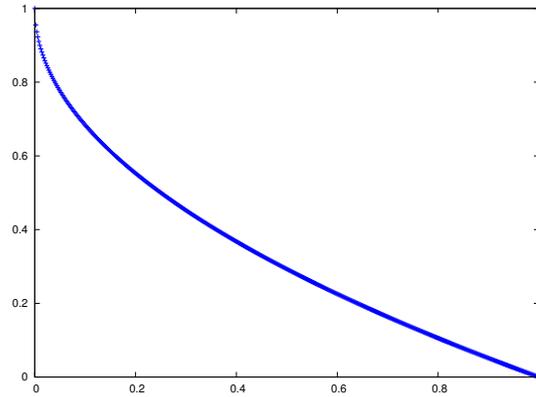


Figura A.1: Frente de Pareto verdadero de ZDT1

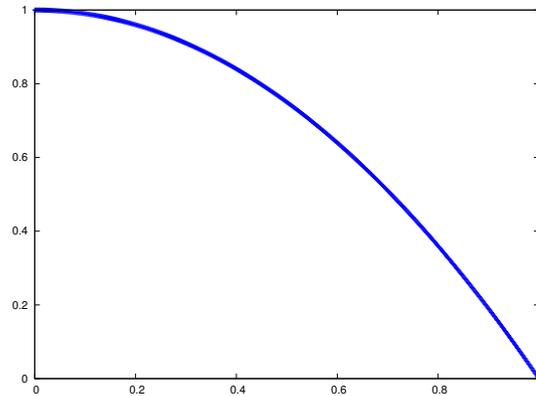


Figura A.2: Frente de Pareto verdadero de ZDT2

ZDT3 tiene un frente de Pareto discontinuo, convexo:

$$\begin{aligned}
 f_1(x) &= x_1, \\
 f_2(x, g) &= g(x) \left(1 - \sqrt{\frac{f_1}{g(x)}} - \frac{f_1}{g(x)} \sin(10\pi f_1) \right) \\
 g(x) &= 1 + \frac{9}{n-1} \sum_{i=2}^n x_i
 \end{aligned}$$

donde $n = 30$, y $x_i \in [0, 1]$. El $F_{\text{verdadero}}$ es formado con $g(x) = 1$ y es mostrado en la figura [A.3](#).

ZDT4 tiene 21^9 frentes locales, lo que pone a prueba la habilidad de los algoritmos evolutivos multi-objetivo (AEMOs) para lidiar con problemas multifrontales:

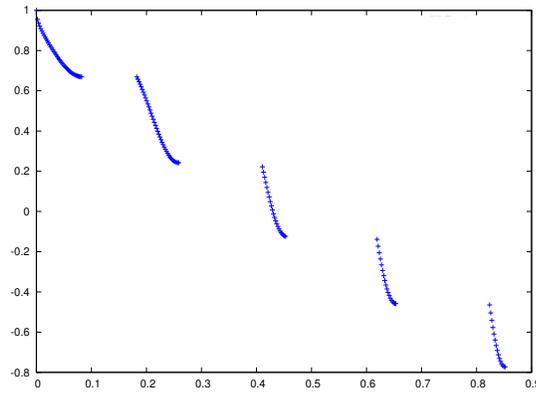


Figura A.3: Frente de Pareto verdadero de ZDT3

$$f_1(x) = x_1,$$

$$f_2(x, g) = g(x) \left(1 - \sqrt{\frac{f_1}{g(x)}} \right),$$

$$g(x) = 1 + 10(n - 1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$$

donde $n = 10$, y $x_1 \in [0, 1]$ y $x_2, \dots, x_n \in [-5, 5]$. El $F_{\text{verdadero}}$ es formado con $g(x) = 1$ y es mostrado en la figura A.4.

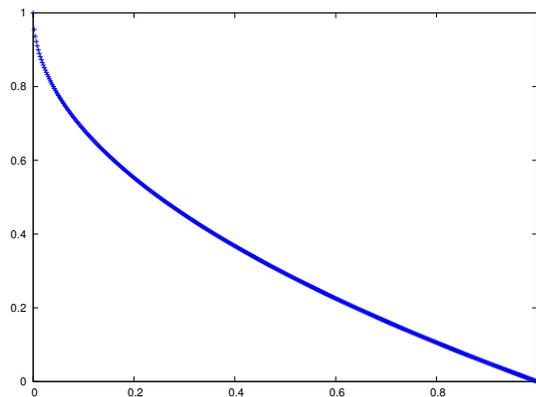


Figura A.4: Frente de Pareto verdadero de ZDT4

ZDT6 el espacio de búsqueda en este problema es no uniforme, la densidad de las soluciones es baja cerca del $F_{\text{verdadero}}$ y alta lejos del frente.

$$\begin{aligned}f_1(x) &= 1 - \exp(-4x_1) \sin^6(6\pi x_1), \\f_2(x, g) &= g(x) \left(1 - \left(\frac{f_1}{g(x)} \right)^2 \right), \\g(x) &= 1 + 9 \left[\frac{\sum_{i=2}^n}{9} \right]^{0.25}\end{aligned}$$

donde $n = 10$, y $x_1 \in [0, 1]$ y $x_2, \dots, x_n \in [-5, 5]$. El $F_{\text{verdadero}}$ es formado con $g(x) = 1$ y es mostrado en la figura A.5.

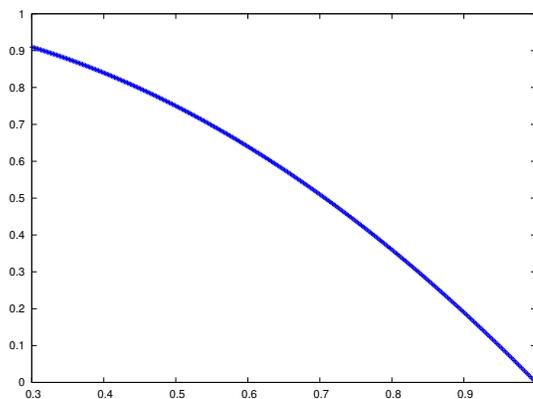


Figura A.5: Frente de Pareto verdadero de ZDT6

A.2. Conjunto de problemas DTLZ (Deb-Thiele-Laumanns-Zitzler)

Estas funciones están descritas en [97].

DTLZ1 tiene un frente de Pareto lineal, separable, multimodal.

$$\begin{aligned}
 f_1(x) &= \frac{1}{2}x_1x_2 \dots x_{M-1}(1 + g(x)) \\
 f_2(x) &= \frac{1}{2}x_1x_2 \dots (1 - x_{M-1})(1 + g(x)) \\
 &\vdots \\
 f_M(x) &= \frac{1}{2}(1 - x_1)(1 + g(x)) \\
 g(x) &= 100 \left[k + \sum_{i=M}^n (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right]
 \end{aligned}$$

donde $n = M + k - 1$ (un valor de $k = 5$ es sugerido) y $x_i \in [0, 1]$. El espacio de búsqueda contiene $11^k - 1$ frentes locales. El $F_{\text{verdadero}}$ para $M = 3$ es mostrado en la figura A.6.

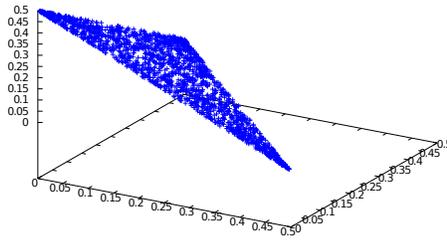


Figura A.6: Frente de Pareto verdadero de DTLZ1

DTLZ2 tiene un frente de Pareto cóncavo. Esta función puede ser usada para investigar la habilidad de un AEMO lidiar con problemas que tengan muchos objetivos.

$$\begin{aligned}
 f_1(x) &= \cos\left(x_1 \frac{\pi}{2}\right) \cos\left(x_2 \frac{\pi}{2}\right) \dots \cos\left(x_{M-1} \frac{\pi}{2}\right) (1 + g(x)) \\
 f_2(x) &= \cos\left(x_1 \frac{\pi}{2}\right) \cos\left(x_2 \frac{\pi}{2}\right) \dots \sin\left(x_{M-1} \frac{\pi}{2}\right) (1 + g(x)) \\
 f_3(x) &= \cos\left(x_1 \frac{\pi}{2}\right) \cos\left(x_2 \frac{\pi}{2}\right) \dots \sin\left(x_{M-2} \frac{\pi}{2}\right) (1 + g(x)) \\
 &\vdots \\
 f_{M-1}(x) &= \cos\left(x_1 \frac{\pi}{2}\right) \sin\left(x_2 \frac{\pi}{2}\right) (1 + g(x)) \\
 f_M(x) &= \sin\left(x_1 \frac{\pi}{2}\right) (1 + g(x)) \\
 g(x) &= \sum_i (x_i - 0.5)^2
 \end{aligned}$$

donde $n = M + k - 1$ (un valor de $k = 10$ es sugerido) y $x_i \in [0, 1]$. El $F_{\text{verdadero}}$ para $M = 3$ es mostrado en la figura A.7.

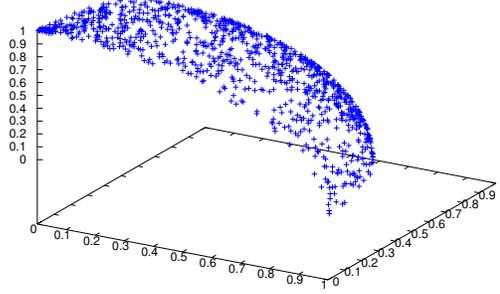


Figura A.7: Frente de Pareto verdadero de DTLZ2

DTLZ3 es el mismo que DTLZ2 excepto por una nueva función $g(x)$. Tiene un frente de Pareto cóncavo y multimodal. Prueba la habilidad de un AEMO de converger al $F_{\text{verdadero}}$.

$$\begin{aligned}
 f_1(x) &= \cos\left(x_1 \frac{\pi}{2}\right) \cos\left(x_2 \frac{\pi}{2}\right) \dots \cos\left(x_{M-1} \frac{\pi}{2}\right) (1 + g(x)) \\
 f_2(x) &= \cos\left(x_1 \frac{\pi}{2}\right) \cos\left(x_2 \frac{\pi}{2}\right) \dots \sin\left(x_{M-1} \frac{\pi}{2}\right) (1 + g(x)) \\
 f_3(x) &= \cos\left(x_1 \frac{\pi}{2}\right) \cos\left(x_2 \frac{\pi}{2}\right) \dots \sin\left(x_{M-2} \frac{\pi}{2}\right) (1 + g(x)) \\
 &\vdots \\
 f_{M-1}(x) &= \cos\left(x_1 \frac{\pi}{2}\right) \sin\left(x_2 \frac{\pi}{2}\right) (1 + g(x)) \\
 f_M(x) &= \sin\left(x_1 \frac{\pi}{2}\right) (1 + g(x)) \\
 g(x) &= 100\left[k + \sum_{i=M}^n (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right]
 \end{aligned}$$

donde $n = M + k - 1$ (un valor de $k = 10$ es sugerido) y $x_i \in [0, 1]$. La función $g(x)$ introduce $3k - 1$ frentes locales paralelos al frente global. El $F_{\text{verdadero}}$ para $M = 3$ es mostrado en la figura A.8.

DTLZ4 tiene un frente de Pareto cóncavo, separable, multimodal. Esta función prueba la habilidad de los AEMOs para mantener una buena distribución de las soluciones no dominadas.

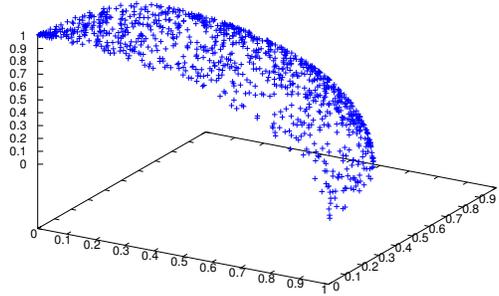


Figura A.8: Frente de Pareto verdadero de DTLZ3

$$\begin{aligned}
 f_1(x) &= \cos\left(x_1^\alpha \frac{\pi}{2}\right) \cos\left(x_2^\alpha \frac{\pi}{2}\right) \dots \cos\left(x_{M-1}^\alpha \frac{\pi}{2}\right) (1 + g(x)) \\
 f_2(x) &= \cos\left(x_1^\alpha \frac{\pi}{2}\right) \cos\left(x_2^\alpha \frac{\pi}{2}\right) \dots \sin\left(x_{M-1}^\alpha \frac{\pi}{2}\right) (1 + g(x)) \\
 f_3(x) &= \cos\left(x_1^\alpha \frac{\pi}{2}\right) \cos\left(x_2^\alpha \frac{\pi}{2}\right) \dots \sin\left(x_{M-2}^\alpha \frac{\pi}{2}\right) (1 + g(x)) \\
 &\vdots \\
 f_{M-1}(x) &= \cos\left(x_1^\alpha \frac{\pi}{2}\right) \sin\left(x_2^\alpha \frac{\pi}{2}\right) (1 + g(x)) \\
 f_M(x) &= \sin\left(x_1^\alpha \frac{\pi}{2}\right) (1 + g(x)) \\
 g(x) &= \sum_i (x_i - 0.5)^2
 \end{aligned}$$

donde $n = M + k - 1$ ($k = 10$ y $\alpha = 100$ son sugeridos) y $x_i \in [0, 1]$. El $F_{\text{verdadero}}$ para $M = 3$ es mostrado en la figura A.9.

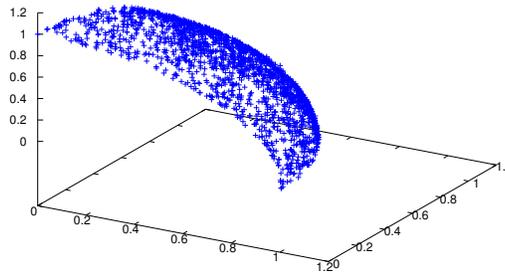


Figura A.9: Frente de Pareto verdadero de DTLZ4

DTLZ25 tiene un frente de Pareto curvo.

$$\begin{aligned}
f_1(x) &= \cos\left(\theta_1 \frac{\pi}{2}\right) \cos\left(\theta_2 \frac{\pi}{2}\right) \dots \cos\left(\theta_{M-1} \frac{\pi}{2}\right) (1 + g(x)) \\
f_2(x) &= \cos\left(\theta_1 \frac{\pi}{2}\right) \cos\left(\theta_2 \frac{\pi}{2}\right) \dots \sin\left(\theta_{M-1} \frac{\pi}{2}\right) (1 + g(x)) \\
f_3(x) &= \cos\left(\theta_1 \frac{\pi}{2}\right) \cos\left(\theta_2 \frac{\pi}{2}\right) \dots \sin\left(\theta_{M-2} \frac{\pi}{2}\right) (1 + g(x)) \\
&\vdots \\
f_{M-1}(x) &= \cos\left(\theta_1 \frac{\pi}{2}\right) \sin\left(\theta_2 \frac{\pi}{2}\right) (1 + g(x)) \\
f_M(x) &= \sin\left(\theta_1 \frac{\pi}{2}\right) (1 + g(x)) \\
\theta_1 &= \frac{\pi}{2} x_1 \\
\theta_i &= \frac{\pi}{4(1 + g(x))} (1 + 2g(x)x_i), \quad \text{para } i = 2, 3, \dots, (M-1) \\
g(x) &= \sum_i (x_i - 0.5)^2
\end{aligned}$$

donde $n = M + k - 1$ (un valor de $k = 10$ es sugerido) y $x_i \in [0, 1]$. El $F_{\text{verdadero}}$ para $M = 3$ es mostrado en la figura A.10.

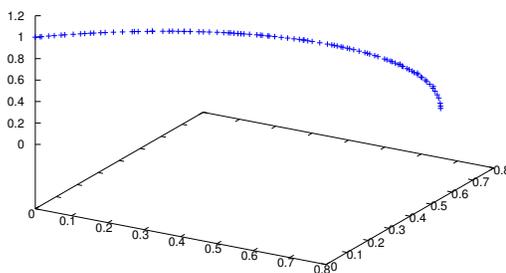


Figura A.10: Frente de Pareto verdadero de DTLZ5

DTLZ6 modifica la función $g(x)$ de DTLZ5 volviéndolo un problema más difícil. Tiene un frente de Pareto curvo.

$$\begin{aligned}
 f_1(x) &= \cos\left(\theta_1 \frac{\pi}{2}\right) \cos\left(\theta_2 \frac{\pi}{2}\right) \dots \cos\left(\theta_{M-1} \frac{\pi}{2}\right) (1 + g(x)) \\
 f_2(x) &= \cos\left(\theta_1 \frac{\pi}{2}\right) \cos\left(\theta_2 \frac{\pi}{2}\right) \dots \sin\left(\theta_{M-1} \frac{\pi}{2}\right) (1 + g(x)) \\
 f_3(x) &= \cos\left(\theta_1 \frac{\pi}{2}\right) \cos\left(\theta_2 \frac{\pi}{2}\right) \dots \sin\left(\theta_{M-2} \frac{\pi}{2}\right) (1 + g(x)) \\
 &\vdots \\
 f_{M-1}(x) &= \cos\left(\theta_1 \frac{\pi}{2}\right) \sin\left(\theta_2 \frac{\pi}{2}\right) (1 + g(x)) \\
 f_M(x) &= \sin\left(\theta_1 \frac{\pi}{2}\right) (1 + g(x)) \\
 \theta_1 &= \frac{\pi}{2} x_1 \\
 \theta_i &= \frac{\pi}{4(1 + g(x))} (1 + 2g(x)x_i) \quad \text{para } i = 2, 3, \dots, (M - 1) \\
 g(x) &= \sum_i = (x_i - 0.5)^0.1
 \end{aligned}$$

donde $n = M + k - 1$ (se sugiere usar un valor de $k = 10$) y $x_i \in [0, 1]$. El $F_{\text{verdadero}}$ para $M = 3$ es mostrado en la figura A.11.

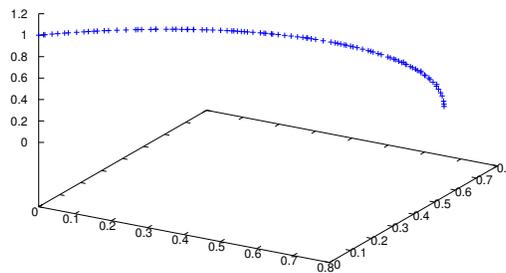


Figura A.11: Frente de Pareto verdadero de DTLZ6

DTLZ7 tiene un frente de Pareto discontinuo. Este problema prueba la habilidad de un AEMO de mantener individuos en las diferentes regiones del frente de Pareto.

$$\begin{aligned}
f_1(x) &= x_1 f_2(x) && = x_2 \\
&\vdots \\
f_{M-1}(x) &= x_{M-1} \\
f_M(x) &= (1 + g(x_M)) h(f_1, f_2, \dots, f_{M-1} g(x)) \\
g(x) &= 1 + \frac{9}{k} \sum_{i=2}^n x_i \\
h(f_1, f_2, \dots, f_{M-1} g(x)) &= M - \sum_{i=1}^{M-1} \left(\frac{f_i}{1 + g(x)} (1 + \sin 3\pi f_i) \right)
\end{aligned}$$

donde $n = M + k - 1$ (un valor de $k = 20$ es sugerido) y $x_i \in [0, 1]$. El $F_{\text{verdadero}}$ para $M = 3$ es mostrado en la figura A.12.

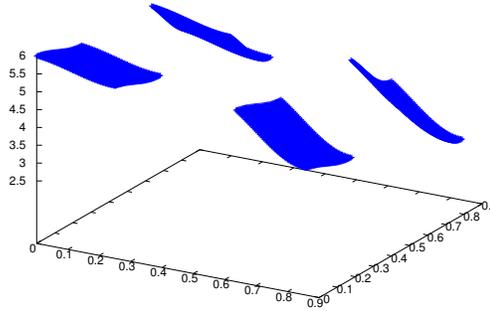


Figura A.12: Frente de Pareto verdadero de DTLZ7

Bibliografía

- [1] C. A. Coello Coello. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2):109–143, June 2000.
- [2] R.C. Purshouse and P.J. Fleming. On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 11(6):770–784, December 2007.
- [3] J. Timmis, T. Knight, L.N.D. Castro, and E. Hart. An overview of artificial immune systems. In R. Paton, H. Bolouri, M. Holcombe, J.H Parish, and R. Tateson, editors, *Computation in Cells and Tissues: Perspectives and Tools for Thought. Natural Computation*, chapter 3, pages 51–86. Springer, 2004.
- [4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
- [5] J. Bader and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. In *Proceedings of the Congress on Evolutionary Computation*, pages 45–76, New Orleans, USA, June 5-8 2011. IEEE.
- [6] F. Freschi and M. Repetto. Multiobjective optimization by a modified artificial immune system algorithm. In *Proceedings of the International Conference on Artificial Immune Systems*, volume 3627 of *Lecture Notes in Computer Science*, pages 248–261, Alberta, CAN, August 14–17 2005. Springer.
- [7] L.N.D. Castro and J. Timmis. *Artificial immune systems: a new computational intelligence approach*. Springer, London, UK, 1st edition, 2002.
- [8] V.M. Collado, R. Porrás, M.T. Cutuli de Simón, and L.E. Gómez. El sistema inmune innato I: sus mecanismos. *Revista complutense de ciencias veterinarias*, 2(1):1–16, January 2008.
- [9] F.R. Fekety. The clonal selection theory of acquired immunity. *The Yale Journal of Biology and Medicine*, 32(6):480, September 1960.
- [10] T.K. Starr, S.C. Jameson, and K.A. Hogquist. Positive and negative selection of t cells. *Annual review of immunology*, 21(1):139, April 2003.

- [11] J. Corado and I. de Cortesía. Células dendríticas, respuesta inmunitaria y señales de peligro. *Gac Méd Caracas*, 113(4):474–484, December 2005.
- [12] J. FASEB. Towards a network theory of the immune system. *Ann Immunol*, 125(1-2):373–89, January 1974.
- [13] D. Dasgupta, S. Yu, and L.F. Niño. Recent advances in artificial immune systems: Models and applications. *Applied Soft Computing*, 11(2):1574 – 1587, March 2011.
- [14] J. Timmis, A. Hone, T. Stibor, and E. Clark. Theoretical advances in artificial immune systems. *Theoretical Computer Science*, 403(1):11 – 32, August 2008.
- [15] D. Dasgupta, N. Majumdar, and F. Nino. Artificial immune systems: A bibliography. *Computer Science Division, University of Memphis, Technical Report, (CS-07-004)*, December 2007.
- [16] A.O. Tarakanov, V.A. Skormin, S.P. Sokolova, and S.S. Sokolova. *Immunocomputing: principles and applications*. Springer, New York, USA, 1st edition, 2003.
- [17] Y. Ishida. *Immunity-based systems: a design perspective*. Springer, Berlin, GER, 1st edition, 2004.
- [18] D. Dasgupta and L.F. Niño. *Immunological computation: theory and applications*. Auerbach Publications, Florida, USA, 1st edition, 2008.
- [19] S. Forrest, A.S. Perelson, L. Allen, and R. Cherukuri. Self-nonsel self discrimination in a computer. In *Proceedings of Computer Society Symposium on Research in Security and Privacy*, pages 202–212, California, USA, May 25–27 1994. IEEE.
- [20] Z. Ji and D. Dasgupta. Revisiting negative selection algorithms. *Evolutionary Computation*, 15(2):223–251, May 2007.
- [21] F. Gonzalez, D. Dasgupta, and R. Kozma. Combining negative selection and classification techniques for anomaly detection. In *Proceedings of the Congress on Evolutionary Computation*, pages 705–710, Hawaii, USA, May 12–17 2002. IEEE.
- [22] D.W. Lee and K.B. Sim. Negative selection for DNA sequence classification. In *Proceedings of International Conference on Soft Computing and Intelligent Systems and International Symposium on Advanced Intelligent Systems*, Yokoham, JPN, September 21–24 2004. SCIS & ISIS.
- [23] Y. Ishida. Fully distributed diagnosis by PDP learning algorithm: towards immune network PDP model. In *Proceedings of International Joint Conference on Neural Networks*, pages 777–782, California, USA, January 15–19 1990. IEEE.

- [24] J. Timmis and M. Neal. A resource limited artificial immune system for data analysis. *Knowledge-Based Systems*, 14(3-4):121–130, June 2001.
- [25] J.C. Galeano, S.A. Veloza, and F.A. González. A comparative analysis of artificial immune network models. In *Proceedings of the Conference on Genetic and Evolutionary Computation*, pages 361–368, Washington, USA, June 25 – 29 2005. ACM.
- [26] L.N.D. Castro and F.J. Von Zuben. The clonal selection algorithm with engineering applications. In *Proceedings of the Conference on Genetic and Evolutionary Computation*, pages 36–39, Nevada, USA, 8–12 July 2000. Morgan Kaufmann.
- [27] U. Aickelin and S. Cayzer. The danger theory and its application to artificial immune systems. In *Proceedings of the International Conference on Artificial Immune Systems*, pages 141–148, London, UK, 9–11 Sep 2002. Morgan Kaufmann.
- [28] P. Matzinger. Tolerance, danger, and the extended family. *Annual review of immunology*, 12(1):991–1045, April 1994.
- [29] S. Golzari, S. Doraisamy, Md Sulaiman, N. Udzir, and N. Norowi. Artificial immune recognition system with nonlinear resource allocation method and application to traditional malay music genre classification. In *Proceedings of the International Conference on Artificial Immune Systems*, volume 5132 of *Lecture Notes in Computer Science*, pages 132–141, Phuket, THA, August 10–13 2008. Springer.
- [30] Y. Hui, M. Elhadeif, A. Nayak, and Y. Xiaofan. Network fault diagnosis: An artificial immune system approach. In *Proceedings of IEEE International Conference on Parallel and Distributed Systems*, pages 463 –469, Victoria, AUS, December 8–10 2008.
- [31] W. Godfrey and S. Nair. An immune system based multi-robot mobile agent network. In *Proceedings of the International Conference on Artificial Immune Systems*, volume 5132 of *Lecture Notes in Computer Science*, pages 424–433, Phuket, THA, August 10–13 2008. Springer.
- [32] M. Gadi, X. Wang, and D.A Lago. Credit card fraud detection with artificial immune system. In *Proceedings of the International Conference on Artificial Immune Systems*, volume 5132 of *Lecture Notes in Computer Science*, pages 119–131, Phuket, THA, August 10–13 2008. Springer.
- [33] B. Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd, Chennai, IND, 1st edition, 2004.
- [34] F.V. Jensen. *An introduction to Bayesian networks*. UCL, London, UK, 1st edition, 1996.

- [35] D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of European Conference on Machine Learning*, volume 1398 of *Lecture Notes in Computer Science*, pages 4–15, Chemnitz, GER, April 21–24 1998. Springer.
- [36] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, March 1986.
- [37] C. Chuanliang, X. Chuan, B. Rongfang, and X.Z. Gao. Artificial immune recognition system for DNA microarray data analysis. In *Proceedings of International Conference on Natural Computation*, pages 633–637, Shandong, CHN, October 18–20 2008. IEEE.
- [38] A.H. Abi and L. Rocha. Adaptive spam detection inspired by a cross-regulation model of immune dynamics: A study of concept drift. In *Proceedings of the International Conference on Artificial Immune Systems*, volume 5132 of *Lecture Notes in Computer Science*, pages 36–47, Phuket, THA, August 10–13 2008. Springer.
- [39] A. Osyczka. Multicriteria optimization for engineering design. *Design optimization*, 1(1):193–227, January 1985.
- [40] C. Darwin, J.W. Burrow, et al. *The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life*. AL Burt, New York, USA, 1st edition, 2009.
- [41] C.A. Coello Coello. Evolutionary multiobjective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1):28–36, February 2006.
- [42] F. Campelo, F. Guimaraes, and H. Igarashi. Overview of artificial immune systems for multi-objective optimization. In *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, volume 4403 of *Lecture Notes in Computer Science*, pages 937–951, Matshushima, JPN, March 5–8 2007. Springer.
- [43] C.A. Coello Coello, G.B. Lamont, and D.A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, USA, 2nd edition, 2007.
- [44] O. Schuetze, X. Equivel, A. Lara, and C.A. Coello Coello. Some comments on GD and IGD and relations to the Hausdorff distance. In *Proceedings of the Conference on Genetic and Evolutionary Computation*, pages 1971–1974, Oregon, USA, July 7–11 2010. ACM.
- [45] E. Zitzler and S. Künzli. Indicator-based Selection in Multiobjective Search. In *Proceedings of the Workshop on Parallel Problem Solving From Nature*, volume

- 3242 of *Lecture Notes in Computer Science*, pages 832–842, Birmingham, UK, September 18–22 2004. Springer.
- [46] A.E. Eiben and J.E. Smith. *Introduction to evolutionary computing*. Springer, Berlin, GER, 1st edition, 2003.
- [47] L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial intelligence through simulated evolution*. Wiley, Michigan, USA, 1st edition, 1966.
- [48] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York, USA, 1st edition, 1996.
- [49] I. Rechenberg. *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, Stuttgart, GER, 1st edition, 1973.
- [50] H.P. Schwefel. *Numerische Optimierung Von Computer-Modellen Mittels Der Evolutionsstrategie: Mit Einer Vergleichenden Einführung in Die Hill-Climbing- und Zufallsstrategie*. Birkhäuser, Basel, GER, 1st edition, 1977.
- [51] H.P. Schwefel. *Numerical optimization of computer models*. John Wiley & Sons, Inc., West Sussex, UK, 1st edition, 1981.
- [52] John H. Holland. Outline for a logical theory of adaptive systems. *J. ACM*, 9(3):297–314, July 1962.
- [53] G. Syswerda and J. Palmucci. The application of genetic algorithms to resource scheduling. In *Proceedings of the International Conference on Genetic Algorithms*, page 502â508, California, USA, July 13–16 1991. ACM.
- [54] W. Jakob, M. Gorges-Schleuter, and C. Blume. Application of genetic algorithms to task planning and learning. In *Proceedings of the Workshop on Parallel Problem Solving From Nature*, pages 293–302, Brussels,BEL, September 28–30 1992. Elsevier.
- [55] G. Jones, R.D. Brown, D.E. Clark, P. Willett, and R.C. Glen. Searching databases of two-dimensional and three-dimensional chemical structures using genetic algorithms. In *Proceedings of the International Conference on Genetic Algorithms*, pages 597–602, California, USA, 17–21 July 1993. Morgan Kaufmann Publishers Inc.
- [56] L. Xiaojian, D. W. Begg, and R. J. Fishwick. Genetic approach to optimal topology/controller design of adaptive structures. *International Journal for Numerical Methods in Engineering*, 41(5):815–830, March 1998.

- [57] D. Wienke, C. Lucasius, and Kateman G. Multicriteria target vector optimization of analytical procedures using a genetic algorithm: Part i. theory, numerical simulations and application to atomic emission spectroscopy. *Analytica Chimica Acta*, 265(2):211 – 225, January 1992.
- [58] E. Sandgren. Multicriteria design optimization by goal programming. In *Advances in Design Optimization*, chapter 7, pages 225–265. Chapman and Hall, New York, USA, 1994.
- [59] D. Quagliarella and A. Vicini. Coupling genetic algorithms and gradient based optimization techniques. In D. Quagliarella, J. Périaux, Poloni C., and G. Winter, editors, *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science—Recent advances and industrial applications*, chapter 9, pages 289–309. John Wiley and Sons Ltd, 1997.
- [60] D.H. Loughlin and S. Ranjithan. The neighborhood constraint method: a genetic algorithm-based multiobjective optimization technique. In *Proceedings of the International Conference on Genetic Algorithms*, volume 673, California,USA, July 19–23 1997. Morgan Kaufmann Publishers Inc.
- [61] P. Surry, N. Radcliffe, and I. Boyd. A multi-objective approach to constrained optimisation of gas supply networks: The comoga method. In Terence Fogarty, editor, *Proceedings of the Workshop on Evolutionary Computing*, volume 993 of *Lecture Notes in Computer Science*, pages 166–180, Brighton, UK, April 1–2 1995. Springer.
- [62] H. Tamaki, H. Kita, and S. Kobayashi. Multi-objective optimization by genetic algorithms: a review. In *Proceedings of the International Conference on Evolutionary Computation*, Nagoya,JPN, may 20–22 1996.
- [63] M.P. Fourman. Compaction of symbolic layout using genetic algorithms. In *Proceedings of the International Conference on Genetic Algorithms*, pages 141–153, New Jersey, USA, July 24–26 1985. L. Erlbaum Associates Inc.
- [64] K. Kursawe. A variant of evolution strategies for vector optimization. In *Proceedings of the Workshop on Parallel Problem Solving from Nature*, volume 496 of *Lecture Notes in Computer Science*, pages 193–197, Dortmund, GER, October 1–3 1991. Springer.
- [65] J.G. Lin. Proper equality constraints and maximization of index vectors. *Journal of Optimization Theory and Applications*, 20(2):215–244, January 1976.
- [66] S. Kundu. A multicriteria genetic algorithm to solve optimization problems in structural engineering design. *Information Processing in Civil and Structural Engineering Design*, 8(4):225–233, May 1996.

- [67] S. Kundu and A. Osyczka. The effect of genetic algorithm selection mechanisms on multicriteria optimization using the distance method. In *Proceedings of the International Conference on Intelligent Systems*, pages 164–168, Nevada, USA, June 19–21 1996. ISCA.
- [68] H. Ishibuchi and T. Murata. Multi-objective genetic local search algorithm. In *Proceedings of International Conference on Evolutionary Computation*, pages 119–124, Nagoya, JPN, may 20–22 1996.
- [69] D.E Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Massachusetts, USA, 1st edition, 1989.
- [70] P. Husbands. Distributed coevolutionary genetic algorithms for multi-criteria and multi-constraint optimisation. In *Proceedings of the Workshop on Evolutionary Computing*, volume 865 of *Lecture Notes in Computer Science*, pages 150–165, Leeds, UK, April 11–13 1994. Springer.
- [71] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.
- [72] G. Rudolph and A. Agapie. Convergence properties of some multi-objective evolutionary algorithms. In *Proceedings of the Congress on Evolutionary Computation*, pages 1010–1016, New Jersey, USA, July 16–19 2000. IEEE.
- [73] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(4):221–248, October 1994.
- [74] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *Proceedings of the Conference on Parallel Problem Solving From Nature*, pages 292–304, Amsterdam, NED, September 27–30 1998.
- [75] J.D. Knowles and D.W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172, June 2000.
- [76] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 15(1):1–28, December 2007.
- [77] B. Naujoks, N. Beume, and M.T.M. Emmerich. Multi-objective optimisation using S-metric selection: application to three-dimensional solution spaces. In *Proceedings of the Congress on Evolutionary Computation*, pages 1282–1289, Edinburgh, UK, September 2–4 2005. IEEE.

- [78] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima. Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions. In *Proceedings of the Conference on Genetic and Evolutionary Computation*, pages 527–534, Orego, USA, July 7–11 2010. ACM.
- [79] T. Wagner and H. Trautmann. Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *Trans. Evol. Comp.*, 14(5):688–701, October 2010.
- [80] M. Basseur and E. K. Burke. Indicator-Based Multi-Objective Local Search. In *Proceedings of the Congress on Evolutionary Computation*, pages 3100–3107, Singapore, September 25–28 2007. IEEE.
- [81] Cynthia Adriana Rodríguez Villalobos. Un nuevo algoritmo evolutivo multiobjetivo basado en una medida de desempeño. Master’s thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2011.
- [82] K. Price, R.M. Storn, and J.A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin, GER, 1st edition, 2005.
- [83] F. Freschi, C.A. Coello Coello, and M. Repetto. Multiobjective Optimization and Artificial Immune Systems: A Review. In Hongwei Mo, editor, *Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies*, chapter 1, pages 1–21. Medical Information Science Reference, New York, USA, 2009.
- [84] J. Yoo and P. Hajela. Immune network simulations in multicriterion design. *Structural and Multidisciplinary Optimization*, 18(2-3):85–94, October–November 1999.
- [85] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Massachusetts, USA, 1st edition, 1989.
- [86] C.A. Coello Coello and N. Cortés Cruz. Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6(2):163–190, June 2005.
- [87] J.D. Farmer, N.H. Packard, and A.S. Perelson. The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, 22(1-3):187–204, October–November 1986.
- [88] X.L. Wang and M. Mahfouf. ACSAMO: an adaptive multiobjective optimization algorithm using the clonal selection principle. In *Proceedings of the European Symposium on Nature-inspired Smart Information Systems*, Tenerife, ESP, November 26–28 2006.

- [89] P. Castro and F. Von Zuben. MOBAIS: A bayesian artificial immune system for multi-objective optimization. In *Proceedings of the International Conference on Artificial Immune Systems*, volume 5132 of *Lecture Notes in Computer Science*, pages 48–59, Phuket, THA, August 10–13 2008. Springer.
- [90] J. Gao and J. Wang. WBMOAIS: A novel artificial immune system for multi-objective optimization. *Computers Operations Research*, 37(1):50 – 61, January 2010.
- [91] L.N.D. Castro and J. Timmis. An artificial immune network for multimodal function optimization. In *Proceedings of the Congress on Evolutionary Computation*, pages 699 –704, Hawaii, USA, May 12-17 2002. IEEE.
- [92] V. Cutello, G. Nicosia, M. Pavone, and G. Stracquadanio. An information-theoretic approach for clonal selection algorithms. In *Proceedings of the International Conference on Artificial Immune Systems*, volume 6209 of *Lecture Notes in Computer Science*, pages 144–157, Edinburgh, UK, July 26–29 2010. Springer.
- [93] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, July 1999.
- [94] K. Deb and R.B. Agrawal. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.
- [95] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, September 1999.
- [96] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, June 2000.
- [97] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.