



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
SECCIÓN DE COMPUTACIÓN

Un estudio de las Estrategias Evolutivas para problemas Multiobjetivo.

Tesis que presenta
Adriana Lara López

para obtener el Grado de
Maestra en Ciencias

en la especialidad de
Ingeniería Eléctrica,
Opción Computación.

Codirectores de la Tesis
Dr. Carlos A. Coello Coello
Dr. Alin Carsteanu

Mexico, D.F.

Julio de 2003

Resumen

En esta tesis se establece la teoría para el estudio de Estrategias Evolutivas en el contexto de la optimización multi-objetivo. Se analiza el trabajo previo en convergencia hacia el Frente de Pareto de Algoritmos Evolutivos realizado por Günter Rudolph, Alexandru Agapie, David Van Veldhuizen y Thomas Hanne, para crear pruebas de convergencia asintótica propias de las Estrategias multiobjetivo $(1 + 1)$, $(\mu + \lambda)$ y (μ, λ) .

Se hace un análisis de los operadores principales de las Estrategias Evolutivas, para comprender mejor el funcionamiento de algunas de sus variantes en relación a la convergencia de los algoritmos.

Se propone, por último, un mecanismo de auto-adaptación para la Estrategia Evolutiva multimiembro y se estudia su velocidad de convergencia, así como varios aspectos teóricos propios de la implementación y comportamiento del algoritmo.

Abstract

In this thesis we establish the theory for the study of Evolution Strategies in the context of the multi-objective optimization. We analyze the previous work on the convergence toward the Pareto Front performed by Günter Rudolph, Alexandru Agapie, David Van Veldhuizen and Thomas Hanne, to make the specific asymptotic proofs for the convergence of Evolution Strategies $(1 + 1)$, $(\mu + \lambda)$ and (μ, λ) .

We show an study of the main operators of Evolution Strategies to better understand the behavior of the variants of these algorithms mainly in terms of convergence.

Finally, we propose a self-adaptation mechanism for the multi-member Evolution Strategy and we study its velocity of convergence and several theoretical aspects of its implementation and performance.

Agradecimientos

A mi amado esposo Jesús, por ser mi apoyo constante.

A mis compañeros y amigos del CINVESTAV: Eloy, Ricardo Landa, Joselito, Julio, Lorena, Ricardo Quiroz, Juan Manuel, Mireya, Omar y Miguel, por acompañarme durante todo el viaje. A Margarita, Nareli, Goyo, Lalo y Efrén, por darme su amistad.

A mis asesores Dr. Alin Carsteanu y Dr. Carlos Coello por hacer posible esta tesis.

A todos quienes como Gerardo de la Fraga, Guillermo Morales y Oscar Olmedo me brindaron consejos académicos y personales que atesoraré por siempre.

A mis dos familias, por su amor y por extrañarme en algunas fiestas.

Se agradece la beca terminal de maestría recibida a través del proyecto CONACYT titulado “Nuevos Paradigmas en Optimización Evolutiva Multiobjetivo” (Ref. 34201-A) cuyo responsable es el Dr. Carlos A. Coello Coello.

Índice General

1	Optimización y problemas multi-objetivo	3
1.1	Técnicas clásicas de optimización	3
1.1.1	Optimización lineal	3
1.1.2	Complejidad computacional	7
1.1.3	Heurísticas	8
1.1.4	Principales complicaciones	9
1.2	Problemas multi-objetivo	12
1.2.1	Conjuntos parcialmente ordenados	14
1.3	Sumario	18
2	Computación evolutiva	19
2.1	Antecedentes y usos	20
2.2	Paradigmas principales	22
2.2.1	Estrategias evolutivas	23
2.2.2	Programación evolutiva	25
2.2.3	Algoritmos genéticos	25
2.3	Abordaje de problemas multi-objetivo con técnicas evolutivas	26
2.3.1	La importancia del trabajo en convergencia	27
2.3.2	Estudios actuales sobre convergencia de AE's en problemas multi-objetivo	28
2.4	Jerarquización de Pareto	28
2.4.1	Esquema de dominación	29
2.4.2	Frente de Pareto	29
2.5	Sumario	31
3	Herramientas básicas	33
3.1	Espacios de probabilidad	33
3.1.1	Conjuntos medibles y σ -álgebras	33

3.1.2	Funciones aditivas y espacio de probabilidad	36
3.1.3	Variables aleatorias	37
3.2	Funciones de distribución	38
3.3	Esperanza condicional	39
3.4	Modelos estocásticos	41
3.4.1	Cadenas de Markov finitas	42
3.4.2	Análisis en mono-objetivo de algoritmos genéticos con cade- nas de Markov	43
3.5	Definiciones de convergencia	45
3.5.1	Convergencia de un algoritmo evolutivo multi-objetivo . . .	46
3.6	Sumario	48
4	Estrategias Evolutivas Multi-objetivo	49
4.1	La estrategia $(\mu \uparrow \lambda)$ multi-objetivo	49
4.2	Operadores	52
4.2.1	Mutación	52
4.2.2	Selección	53
4.2.3	Recombinación	55
4.3	Kernel de variación	57
4.4	Auto-adaptación de EEs mono-objetivo.	59
4.5	Sumario	61
5	Convergencia	63
5.1	Aspectos globales para la convergencia de EEsM	63
5.1.1	Convergencia de la $(1 + 1)$ -EE multiobjetivo	64
5.2	Convergencia de la $(\mu + \lambda)$ -EE multi-objetivo	70
5.2.1	Restringiendo el tamaño de la población de padres	72
5.3	Convergencia de la (μ, λ) -EE multi-objetivo	74
5.3.1	Restringiendo el tamaño de la población secundaria	75
5.4	Sumario	77
6	Aspectos locales en la convergencia de EEsM	79
6.1	Problema de prueba	80
6.2	Poblaciones de tamaño variable	81
6.2.1	Correlación entre las funciones objetivo	82
6.3	Auto-adaptación	84
6.3.1	Algoritmo propuesto	85
6.3.2	Restringiendo a la población de padres	88

6.4	Velocidad de Convergencia	89
6.4.1	Distancia genotípica vs. distancia fenotípica	91
6.4.2	Ventajas poblacionales	95
6.4.3	Probabilidad de acercamiento	96
6.5	Algoritmo para la EE Multi-objetivo	101
6.6	Resultados	102
6.6.1	Resultados sin auto-adaptación	102
6.6.2	Resultados con auto-adaptación	104
6.7	Sumario	105

Notación y Acrónimos:

AA Auto-adaptación.

AE Algoritmo Evolutivo.

AG Algoritmo Genético.

CE Computación Evolutiva.

EE Estrategia Evolutiva.

EEsM Estrategias Evolutivas Multiobjetivo.

MOP Problema de Optimización Multiobjetivo.

Introducción

Existen áreas del conocimiento y el quehacer humano dentro de las cuales es común el surgimiento de problemas que consisten en la mejora de ciertas soluciones, procedimientos o proporciones de las que se obtiene algún beneficio ya sea particular o común. La disciplina que se enfoca a estudiar este tipo de problemas y sus respectivas alternativas es conocida como *optimización*.

La computación evolutiva es la aplicación, mediante la implementación en una computadora, de algoritmos inspirados en el principio, que rige en la naturaleza, de la sobrevivencia del individuo más apto. Los algoritmos evolutivos tienen como objetivo principal “evolucionar” individuos, los cuales normalmente representan soluciones a un cierto problema de optimización; los individuos se irán renovando generación tras generación bajo el principio del más apto, por lo que cada nueva generación debe poseer características mejores que las anteriores, acercándonos con cada iteración a la solución óptima del problema.

Originalmente las técnicas evolutivas no fueron concebidas como técnicas de optimización sino de aproximación, esto debido principalmente al vacío teórico que, al igual que otras heurísticas, tuvieron en sus inicios. Sin embargo, con el paso del tiempo y las mejoras, han mostrado dar buenos resultados aproximándose a muy buenas soluciones en tiempos razonables, logrando resolver los problemas en cuestión de manera satisfactoria. Actualmente ya se cuenta con pruebas de convergencia para algunos algoritmos evolutivos y también se han ido fundamentando y estandarizando los conceptos básicos de estas técnicas motivando cada vez más incursiones en materia teórica.

El primer objetivo de este trabajo es describir en una versión multi-objetivo los algoritmos correspondientes a las “estrategias evolutivas”. Los principales esfuerzos son para desarrollar una teoría de convergencia de la estrategia evolutiva multi-objetivo en términos generales, es decir, con respecto a su comportamiento al límite. En segundo término, se presenta un estudio más detallado sobre su comportamiento, para un problema multi-objetivo en particular, resaltando algunos

aspectos teóricos referentes a los espacios de búsqueda, velocidad de convergencia y auto-adaptación.

Basados en las técnicas presentadas en [52], [48], y [60] para algoritmos evolutivos multi-objetivo y los estudios de [8] referentes al comportamiento asintótico de las estrategias evolutivas mono-objetivo, se escriben la teoría y resultados correspondientes al estudio de la convergencia para estrategias evolutivas multi-objetivo. Con esto se resumen los resultados de varios artículos y las herramientas matemáticas necesarias para la comprensión y desarrollo del estudio de la convergencia en técnicas evolutivas y su análisis visto como un sistema dinámico en el tiempo.

El primer capítulo se divide en dos secciones: la primera introduce conceptos de optimización y heurísticas, resaltando las complicaciones que motivan al uso técnicas evolutivas en vez de técnicas clásicas de optimización; la segunda sección se dedica a optimización multi-objetivo.

En el Capítulo 2 se describen las principales técnicas evolutivas, motivando su uso en problemas multi-objetivo, a manera de introducción para el Capítulo 4 donde se particulariza en el paradigma de la Estrategia Evolutiva.

El Capítulo 3 presenta conceptos y resultados básicos de teoría de la probabilidad y procesos estocásticos necesarios para explicar los detalles de las técnicas matemáticas aquí utilizadas. El lector familiarizado con estas áreas puede omitirlo.

Dentro del Capítulo 5 se proponen algoritmos para las estrategias evolutivas $(1+1)$ -EE multi-objetivo, $(\mu+\lambda)$ -EE multi-objetivo y (μ, λ) -EE multi-objetivo, presentando sus respectivas pruebas de convergencia hacia el frente de Pareto.

Finalmente, en el Capítulo 6, se estudian varios aspectos de la estrategia evolutiva multi-objetivo sobre un caso de estudio particular para el cual se determina su velocidad de convergencia al frente de Pareto. También se propone un mecanismo de auto-adaptación en el operador de mutación y se comprueba experimentalmente que dicho mecanismo mejora la velocidad de convergencia del algoritmo.

Capítulo 1

Optimización y problemas multi-objetivo

1.1 Técnicas clásicas de optimización

El término optimización es usado comúnmente para describir el proceso de hallar la “mejor” solución de entre un conjunto de opciones llamadas “espacio de búsqueda”. La mejor solución es dicha en el sentido de óptimo global (el cual será definido a continuación) referente a una cierta función objetivo f del espacio de búsqueda.

Definición 1.1.1 *Dado un problema de minimización de la función*

$$f : A \subseteq S = \mathbb{R}^n \rightarrow \mathbb{R}, \quad A \neq \emptyset$$

El valor $-\infty < f^ := f(\vec{x}^*)$ es llamado el óptimo global (o mínimo) si y sólo si para todos los valores*

$$\vec{x} \in A \quad \text{se cumple que} \quad f(\vec{x}^*) \leq f(x).$$

A la función f se le conoce como función objetivo.

1.1.1 Optimización lineal

Definición 1.1.2 Se llama una igualdad (ecuación) lineal *no homogénea* en las indeterminadas x_1, x_2, \dots, x_n a una expresión de la forma:

$$a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_nx_n = b$$

con los escalares $a_i, b \in Q$ donde Q puede ser \mathbb{R}, \mathbb{Q} o \mathbb{Z} .

Definición 1.1.3 Al conjunto de m ecuaciones lineales en n indeterminadas se le llama *sistema* de ecuaciones lineales de tamaño $m \times n$.

$$\begin{array}{cccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m \end{array}$$

Definición 1.1.4 Una *desigualdad* lineal o *inecuación* lineal es una expresión que puede tomar una de las siguientes cuatro formas:

$$\begin{array}{l} a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n < b \\ a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n \leq b \\ a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n > b \\ a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n \geq b. \end{array}$$

Análogamente a la definición 1.1.3 puede definirse un sistema de desigualdades lineales.

Definición 1.1.5 El conjunto de valores ordenados lexicográficamente

$$x_1 = c_1, \quad x_2 = c_2, \quad \cdots, \quad x_n = c_n$$

con

$$c_i \in Q \text{ donde } Q \text{ puede ser } \mathbb{R}, \mathbb{Q} \text{ o } \mathbb{Z}$$

se dice ser una *solución* a un sistema S si y sólo si la sustitución de los valores x_1, x_2, \dots, x_n en S mantiene la consistencia en el sistema. Por ejemplo para un sistema de igualdades lineales tener

$$a_{i1}c_1 + a_{i2}c_2 + \cdots + a_{in}c_n = b_i$$

para todo $i = 1, 2, \dots, m$.

La solución a un sistema de ecuaciones (o desigualdades) lineales puede verse como la solución a la ecuación matricial:

$$Ax = b \quad (Ax \leq b)$$

donde

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

y

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

representan coeficientes e incógnitas, y cuyos términos independientes están dados por:

$$b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

La producción industrial, el flujo de recursos en la economía y el manejo de finanzas entre otros ejemplos, requieren de la coordinación de actividades las cuales están interrelacionadas. Si estos sistemas pueden modelarse (o aproximarse de manera razonable) a través de desigualdades y/o igualdades lineales, su estudio puede entonces catalogarse dentro del área de la *programación lineal*. No basta con hallar las proporciones para que un sistema de los anteriores pueda funcionar, sino que hay que hallar la distribución correcta entre sí para que den un rendimiento mayor, es decir hallar la solución óptima.

La programación lineal se basa en el estudio de la convexidad de las regiones factibles (también llamadas espacio solución). Debido a que las restricciones en un programa lineal son hiper-planos contenidos en algún espacio de \mathbb{R}^n se prueba que la intersección de estas restricciones es una región convexa; aplicando entonces algunos resultados de convexidad podemos demostrar que los óptimos de funciones lineales sobre este tipo de regiones se hallan en los *vértices* (también conocidos como *puntos extremos*) de éstas.

De esta forma se han desarrollado métodos matriciales y algebraicos como el *simplex* que de manera algorítmica, para un problema lineal, nos lleva al óptimo. En [14] y [23] se da una extensa metodología para la resolución de problemas lineales.

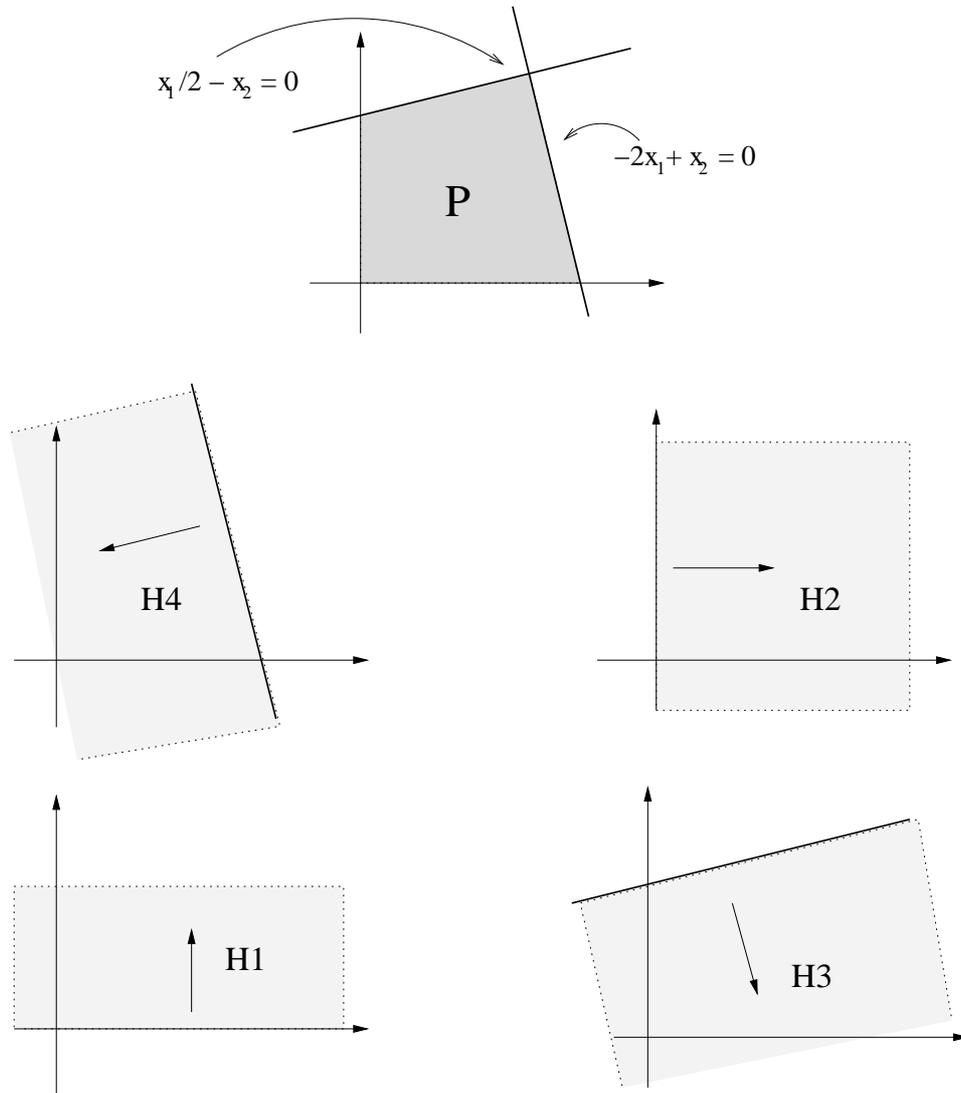


Figura 1.1: La región factible P de un sistema de desigualdades lineales, cada desigualdad lineal es un hiper-plano H y la intersección de éstas definen la región factible del sistema.

1.1.2 Complejidad computacional

Un *problema* usualmente está determinado por un conjunto de hipótesis y parámetros que lo especifican y un conjunto más de condiciones insolutas que al ser descritas nos dan la solución del problema. Muchos problemas requieren para su solución una serie de pasos, los cuales al agruparse, forman lo que se conoce como *algoritmo*. Los algoritmos se describen de manera conceptual y posteriormente se pueden llevar a su implementación en una computadora; normalmente los algoritmos se estudian para mejorar su eficiencia, pero ¿qué es la eficiencia en un algoritmo? Existen varias formas para describir la eficiencia de un algoritmo. Algunos estudios se enfocan a que el algoritmo sea más rápido para decir que es más eficiente; otros buscan que el algoritmo ahorre memoria de manera considerable. Una manera de estandarizar el análisis de la eficiencia de un algoritmo es comparando el número de ciclos de ejecución que éste representa; así los resultados dependerán del algoritmo de manera conceptual y no de la computadora en la que se implemente.

Un algoritmo de tiempo polinomial es aquél cuya complejidad temporal es $O(p(n))$, donde $p(n)$ es un polinomio. Este tipo de algoritmos describen problemas a los cuales se conoce como *problemas tratables*. Un algoritmo de tiempo exponencial es aquel que no puede ser acotado por una función polinomial. Este tipo de algoritmos describen problemas a los cuales se conoce como *problemas intratables*.

Otra clasificación de problemas, dada por Alan Turing en la primera mitad del siglo XX radica en que un problema pueda o no ser resuelto por una computadora (como conocemos a la computadora actual, es decir una máquina determinista). A aquellos para los cuales la afirmación es cierta se les conoce como problemas decidibles; a los que no pueden ser resueltos por una computadora se les llama problemas indecidibles.

A la clase que engloba a todos los problemas que son solubles en un tiempo polinomial le llamaremos clase P .

Un algoritmo *no determinista* es aquel en el que cada estado puede transitar a varios estados diferentes de manera simultánea y que no se puede predecir en un principio. Llamaremos NP a la clase que engloba a los problemas tales que se resuelven a través de un algoritmo no determinista en un tiempo polinomial.

Un problema se dice *polinomialmente transformable* en otro si existe una función f tal que en tiempo polinomial transforma los casos de un problema en el otro incluyendo las soluciones. Un problema se dice ser NP -completo si pertenece a la clase NP y todos los problemas de la clase NP son polinomialmente

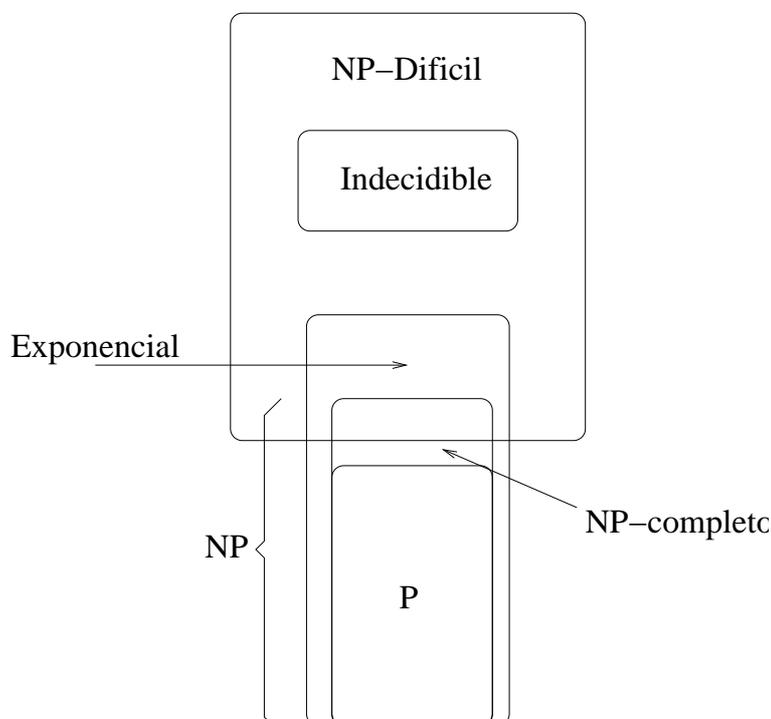


Figura 1.2: Jerarquía de la complejidad

transformables a él. Un problema se dice que es NP -difícil si cumple con que todos los problemas de la clase NP son polinomialmente transformables a él pero él no necesariamente pertenece a la clase NP .

1.1.3 Heurísticas

Una heurística es una técnica para aproximar la solución a un problema de optimización para el cual no existen métodos eficientes conocidos, ya sea porque el espacio de búsqueda es muy grande y la búsqueda exhaustiva es imposible, o porque el modelo del problema no se adapta a ninguna de las técnicas existentes (para las cuales sí se tienen algoritmos que nos llevan a la solución precisa). Es decir, las heurísticas se usan para lidiar con problemas cuya complejidad es tal, que el uso de algoritmos deterministas tiene un costo computacional prohibitivo. Una heurística busca aproximarse de manera intuitiva pero ordenada a la mejor solución, aunque no garantiza llegar al óptimo; para cada problema se puede construir

una heurística especial que dé mejores resultados que otras, ya sea en tiempo o en proximidad al óptimo. Se ha comprobado empíricamente que para cierto tipo de problemas existen heurísticas que funcionan muy bien. Sin embargo no existe la heurística que para cualquier problema sea más eficiente que todas las demás [63].

Para problemas de gráficas (por ejemplo hallar el corte máximo de una gráfica), el espacio de búsqueda es precisamente el conjunto de cortes posibles, el cual coincide con el conjunto potencia del conjunto de los vértices de la gráfica. En este caso hacer una búsqueda exhaustiva de la solución es imposible para gráficas con muchos nodos ya que crece de manera exponencial¹. En estos casos se justifica el uso de una heurística de búsqueda local o del recocido simulado [32] en la cual el tiempo de cálculo es muy aceptable y la solución aproximada es buena.

Las heurísticas no son adecuadas para todo tipo de problemas y son particularmente inadecuadas para problemas cuyo modelo se ajusta a técnicas ya muy estudiadas que nos arrojan soluciones suficientemente simples, como es el caso de los Programas Lineales; en éstos, el Método Simplex llega al óptimo de manera eficiente usando un método algorítmico de tiempo polinomial. Este tipo de técnica nos garantiza además, de manera determinista, que el óptimo (de existir) será alcanzado. Si es muy importante asegurar que efectivamente se alcanza el óptimo, las heurísticas no son recomendables.

1.1.4 Principales complicaciones

Como hemos mencionado existen muchos métodos y estudios dentro de la optimización lineal, la Investigación de Operaciones y la investigación en heurísticas que nos dan herramientas específicas para hallar las mejores soluciones con respecto a diferentes categorías de problemas. Sin embargo existen problemas que poseen características especiales que los hacen difíciles de tratar con los métodos clásicos.

La primera complicación que puede tener un problema de optimización es el hecho de que las funciones a optimizar no posean continuidad, dado que si buscamos el máximo o mínimo de una función continua, existen ya métodos precisos que encuentran los valores óptimos.

Otra característica que dificulta el uso de métodos clásicos es cuando la región factible es no convexa. Por ejemplo, puede demostrarse mediante un poco de topología de conjuntos que en regiones factibles convexas las imágenes bajo funciones lineales deben tener sus valores máximos y mínimos en la frontera; sin embargo la

¹La cardinalidad del conjunto potencia de un conjunto con n elementos (nodos de la gráfica) es 2^n .

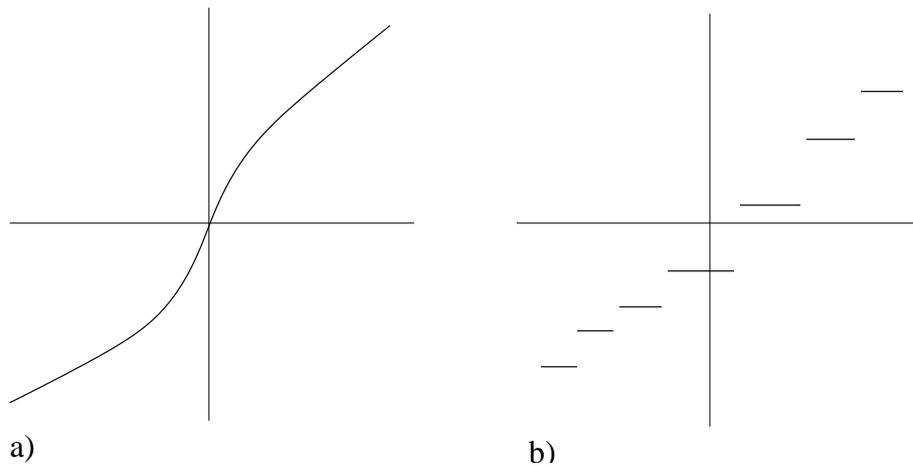


Figura 1.3: Ejemplo de a) una función continua, b) una función no continua

ausencia de convexidad nos limita y excluye de una gran cantidad de técnicas para optimizar basadas en este hecho.

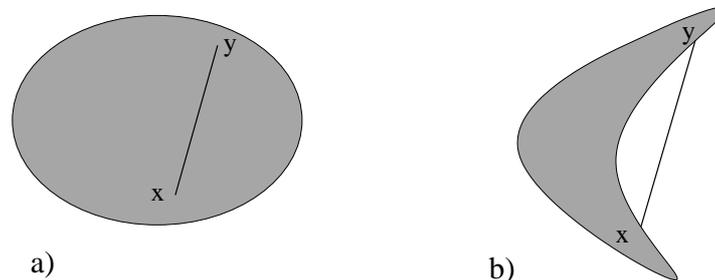


Figura 1.4: Ejemplo de a) conjunto convexo, b) conjunto no convexo

Algunas técnicas clásicas se basan en el estudio de la concavidad de las imágenes bajo funciones objetivo, en otras es necesario que las funciones a optimizar sean diferenciables; en el caso de funciones no diferenciables el repositorio de técnicas factibles a utilizar se reduce una vez más. Más aún, gran cantidad de problemas presentan poca o nula información acerca de su región factible. A veces las regiones no sólo son no convexas sino no conexas, lo cual deja el problema fuera del alcance de muchas de las heurísticas de búsqueda local, debido a que no podrían salir de óptimos locales si cayesen en uno.

El Conjunto de Cantor (Figura 1.6) es un ejemplo de regiones no continuas, cuya gráfica representa una función no diferenciable y es difícil de caracterizar

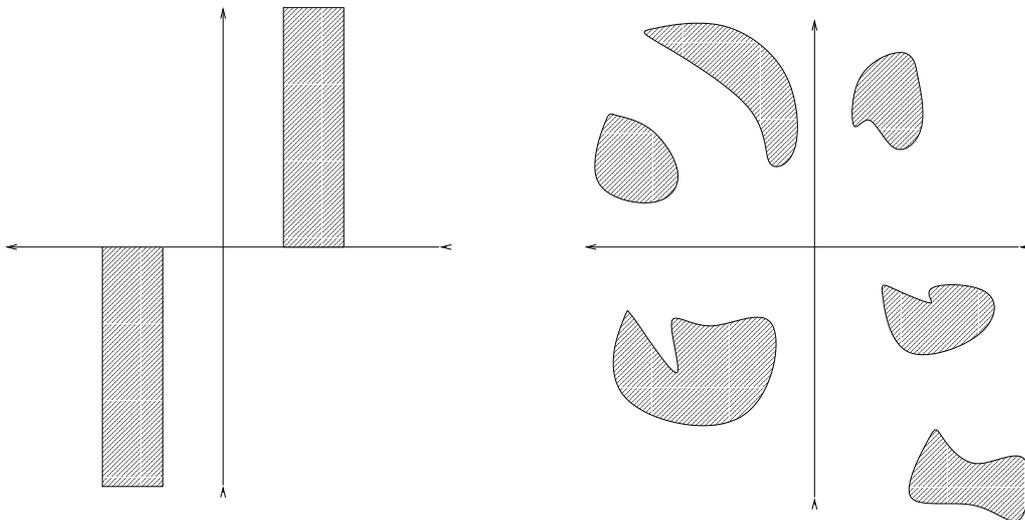


Figura 1.5: Dos regiones factibles no conexas; la región de la derecha cuenta además con componentes no convexas.

matemáticamente; se define como la intersección al infinito de todas sus etapas parciales.

Junto con las características descritas anteriormente puede citarse que algunos problemas cuentan con funciones objetivo difíciles de evaluar por un humano, por lo que el uso de una computadora es requerido. Ésta y las anteriores complicaciones hacen de los problemas que las poseen buenos candidatos para la aplicación de algoritmos evolutivos, puesto que para estas técnicas nada de lo anterior acarrea dificultad. Es por eso que cuando ningún método conocido aproxima razonablemente el óptimo o funciona en ciertos tipos de problemas, vale la pena intentar usar una técnica evolutiva.

En particular si utilizamos técnicas evolutivas en problemas con un alto costo computacional para la evaluación de las funciones de aptitud, debemos tener en cuenta que a cada generación es indispensable evaluar a cada uno de los individuos de la población. Es precisamente en estos casos en que elegir una versión paralela del algoritmo evolutivo es una buena opción (ejemplos de esto se muestran en [9]).

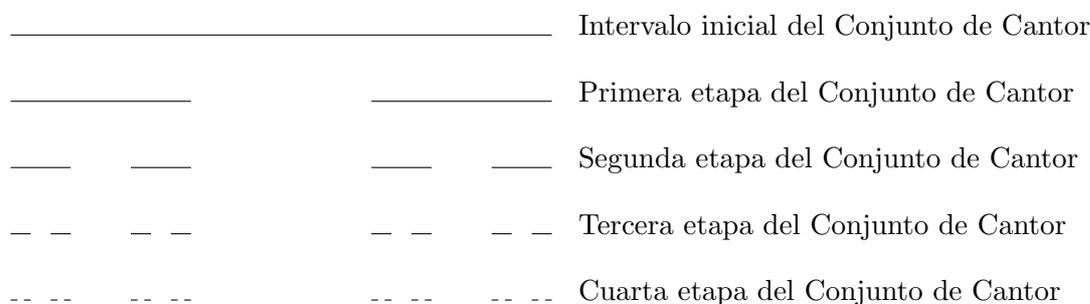


Figura 1.6: Algunas etapas del conjunto de Cantor

1.2 Problemas multi-objetivo

Para muchos campos dentro de las ciencias, tanto sociales como naturales, resolver un problema o tomar una decisión significa hallar la mejor solución común a un conjunto de relaciones (tal es el caso de la economía, el diseño en ingeniería, situaciones de la vida cotidiana, etc.). Un ejemplo sencillo de esto se presenta al tratar de determinar las proporciones de ciertos componentes en un medicamento tomando en cuenta los menores costos de producción cuando a la vez se busca que éste no reduzca su eficacia y sea atractivo para una mayor cantidad de consumidores; otro problema se presenta también si se requiere tomar la decisión de usar o no algún tipo de edulcorante en este medicamento, dado que esto afectaría en diferente factor a los aspectos anteriores. Muchos ejemplos de lo anterior se encuentran en el diseño y construcción de máquinas o piezas en la industria, en donde se deben tomar en cuenta diferentes características referentes a un mismo producto. Si cada una de estas relaciones puede expresarse como una función matemática, al referirnos a la mejora en conjunto, decimos que se debe *optimizar* a todas las funciones de manera simultánea, definiéndose entonces un problema del tipo descrito a continuación:

Definición 1.2.1 *Un problema multi-objetivo (MOP) puede definirse en el caso de minimización (y análogamente para el caso de maximización) como:*

Minimizar: $f(x)$

dado que $f : F \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^q$, $q \geq 2$.

y se evalúa en

$$A = \{a \in F : g_i(a) \leq 0, i \in \{1, \dots, m\}\} \neq \emptyset$$

El conjunto A es llamado la región factible del problema y se dice que el problema se encuentra sujeto a las restricciones g_i , donde

$$g_i : \mathbb{R}^n \longrightarrow \mathbb{R}$$

son funciones cualesquiera.

Algunas veces las funciones objetivo del problema deben escalarse o cambiar de signo, para presentar el problema de la forma anterior, ya que originalmente los problemas multi-objetivo vienen dados de tres formas: en el que todas las funciones se maximizan, en donde todas las funciones se minimizan y en donde algunas funciones se maximizan mientras que otras se minimizan.

A partir de la definición 1.2.1 el sentido común nos lleva a pensar en la optimización multi-objetivo como la búsqueda de un vector que representa el conjunto de variables de decisión y el cual optimiza (maximiza o minimiza) en conjunto a las funciones objetivo. Sin embargo en este caso es de principal importancia notar que dichas funciones pueden estar en conflicto unas con otras. Por ejemplo, (ver figura 1.7) en el caso de tener que optimizar las funciones $f_1(x) = x + 2$, $f_2(x) = x^2$ y $f_3(x) = 4 - x$ con $0 < x < 100$, mientras que $f_1(x) = x + 2$ y $f_2(x) = x^2$ crecen se tiene que $f_3(x) = 4 - x$ decrece, por lo que, la mejor solución será aquella que nos lleve a un mejor compromiso entre las tres funciones.

Además de la posibilidad de estar en conflicto unas con otras, las funciones en un problema multi-objetivo pueden representar cantidades que no son equivalentes entre sí. Por ejemplo, si el problema consiste en optimizar

$$f = (f_1, f_2, \dots, f_n)$$

que son las ganancias totales de una empresa, pero f_1 representa una ganancia actual en pesos, f_2 representa una ganancia en recursos humanos, f_3 representa un ahorro de energía, etc.

Debido a las situaciones anteriores, en optimización multi-objetivo no se utiliza el término óptimo para referirse a la mejor solución del problema, sino el término

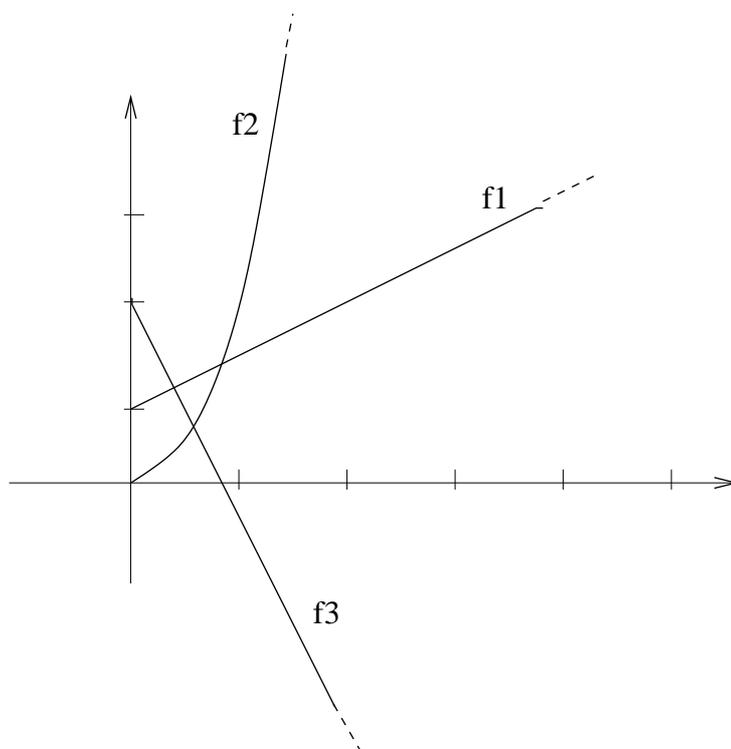


Figura 1.7: Funciones en conflicto, f_3 decrece mientras f_1 y f_2 crecen.

“conjunto de soluciones eficientes”; no necesariamente se debe encontrar una única solución mejoradora de todas las demás. Puesto que se está buscando un vector compromiso, la solución de un problema multi-objetivo consiste en dar un conjunto de vectores eficientes con los cuales se pueda tomar una decisión.

Una manera de definir este compromiso entre las diferentes funciones a optimizar es mediante el concepto de óptimo de Pareto (introducido en la Sección 2.4.2) el cual hace uso de los conceptos de la siguiente sección.

1.2.1 Conjuntos parcialmente ordenados

Pensar en el conjunto de vectores no dominados nos lleva de manera natural al concepto de conjunto parcialmente ordenado, el cual es estudiado por las matemáticas dentro de las áreas de Álgebra y Combinatoria. Para comprender mejor este concepto recordemos lo que es una relación y sus propiedades principales.

Definición 1.2.2 Sea A un conjunto. Al subconjunto $\mathcal{R} \subseteq A \times A$ le llamaremos

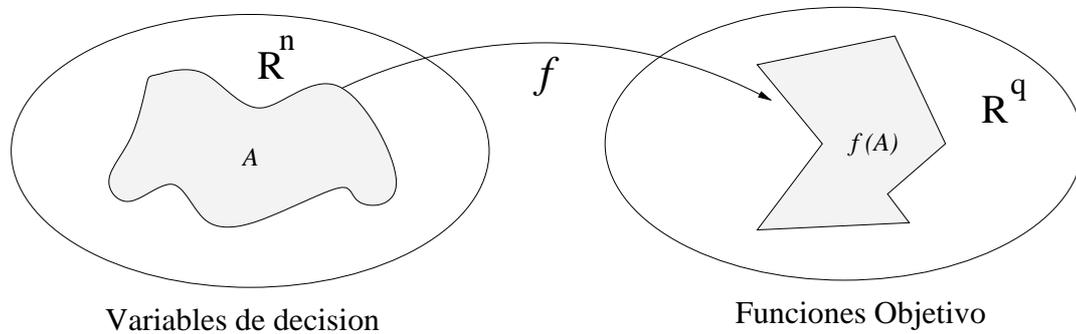


Figura 1.8: Problema multi-objetivo

relación binaria. A los elementos $(x, y) \in \mathcal{R}$ les denotaremos como $x\mathcal{R}y$ y decimos que x está relacionado con y .

La relación \mathcal{R} se llamará:

- a) reflexiva si ocurre que $x\mathcal{R}x$ para toda $x \in A$.
- c) simétrica si $x\mathcal{R}y$ entonces $y\mathcal{R}x$ para todos $x, y \in A$.
- d) antisimétrica si $x\mathcal{R}y$ & $y\mathcal{R}x$ entonces $x = y$ con $x, y \in A$.
- e) asimétrica si $x\mathcal{R}y$ entonces $\neg \mathcal{R}x$ para todos $x, y \in A$.
- f) transitiva si $x\mathcal{R}y$ & $y\mathcal{R}z$ entonces $x\mathcal{R}z$ para todos $x, y, z \in A$.

Definición 1.2.3 Una relación binaria que cumple con ser reflexiva, antisimétrica y transitiva es llamada una **relación de orden parcial** y se representa comúnmente mediante el símbolo \preceq .

A partir de un orden parcial puede definirse la relación de dominación (\prec) de la manera siguiente:

$$x \prec y \iff x \preceq y \wedge x \neq y.$$

Cuando ocurre que $x \not\preceq y \wedge y \not\preceq x$ decimos que son **no comparables**, lo cual es denotado por $x \parallel y$.

Ejemplo En \mathbb{R}^n puede definirse el orden parcial \leq de manera natural como

$$x \leq y \quad x, y \in \mathbb{R}^n \iff x_i \leq y_i \text{ con } i = 1, 2, \dots, n$$

Es claro que existen vectores en \mathbb{R}^n los cuales no pueden compararse bajo el orden parcial \leq , como por ejemplo los vectores:

$$(3.5, 7.8, 4, 6) \quad y \quad (4, 7, 4, 7).$$

El ejemplo anterior ilustra que no cualquier conjunto cumple con la propiedad de que todos sus elementos puedan compararse, como es el caso de \mathbb{R}^n . Esta característica es básica para el concepto que introducimos en la siguiente definición y con el cual más adelante modelaremos la jerarquización de Pareto.

Definición 1.2.4 Dado A un conjunto y (\preceq) una relación de orden parcial sobre él, llamamos a la pareja (A, \preceq) un **conjunto parcialmente ordenado** también referido como **Poset**.

Definición 1.2.5 Dado (A, \preceq) un Poset, el subconjunto $X \subseteq A$ se dice ser un **orden total** o **cadena** con respecto a (\prec) si y sólo si se cumple que $x \prec y$ ó $y \prec x$ para todos $x, y \in X$. En este caso decimos que (X, \preceq) es un **conjunto totalmente ordenado**.

Si se tiene que $x \not\prec y$ para todos $x, y \in X$ entonces decimos que X es una **anticadena**.

Las cadenas pueden ser finitas o infinitas y también pueden estacionarse en uno de sus extremos como:

$$x_0 \prec x_1 \prec x_2 \prec \dots \prec x_{i-1} \prec x_i \prec x_{i+1} \prec \dots$$

ó

$$\dots \prec x_{i+1} \prec x_i \prec x_{i-1} \prec \dots \prec x_2 \prec x_1 \prec x_0$$

Definición 1.2.6 Un elemento $x^* \in A$ es llamado un **elemento minimal** del Poset (A, \preceq) si no existe un elemento $x \in A$ tal que $x \prec x^*$. El conjunto de todos los elementos minimales de (A, \preceq) se denota como $\mathcal{M}(A, \preceq)$.

De esta definición se sigue claramente que el óptimo de Pareto (tratado en la definición 2.4.3) para un problema multi-objetivo dado es, visto de manera formal, un conjunto parcialmente ordenado. En la sección 5.1.1 trataremos algunos resultados en donde se plantea el principio de los algoritmos evolutivos para problemas multi-objetivo como la búsqueda de los elementos minimales del espacio solución \mathbb{R}^n visto como un Poset con la relación \leq definida de manera natural. En los resultados mencionados usaremos fuertemente el concepto de Poset completo, dado a continuación.

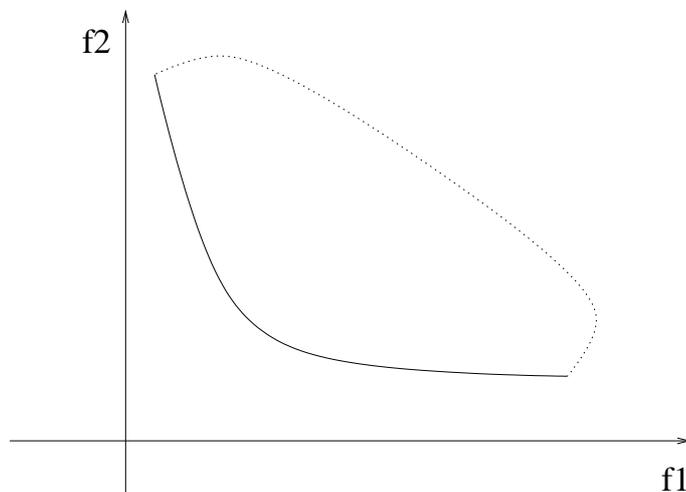


Figura 1.9: Un problema de minimización con dos funciones objetivo. Se indica el frente de Pareto con una línea continua.

Definición 1.2.7 El conjunto $\mathcal{M}(A, \preceq)$ de todos los elementos minimales de (A, \preceq) se dice **completo** si para cada $x \in A$ existe al menos un elemento $x^* \in \mathcal{M}(A, \preceq)$ tal que $x^* \preceq x$.

Lema 1.2.8 Dado un Poset (A, \preceq) tal que A es finito, se cumple que su conjunto de elementos minimales $\mathcal{M}(A, \preceq)$ es completo.

Demostración: Sea $x_0 \in A$, si $x_0 \in \mathcal{M}(A, \preceq)$ el lema es cierto por la propiedad de reflexividad de \preceq . Supongamos entonces que $x_0 \notin \mathcal{M}(A, \preceq)$, esto implica que existe un elemento $x_1 \in A$ tal que $x_1 \preceq x_0$. Dado que A es finito, existe un elemento x_n para alguna cadena $x_n \preceq \cdots \preceq x_1 \preceq x_0$ con $n \leq |A|$ en el cual la cadena transitiva se detiene, esto quiere decir que ningún elemento en A domina a x_n y por lo tanto se sigue directamente que $x_n \in \mathcal{M}(A, \preceq)$. ■

Lema 1.2.9 Sea $\mathcal{M}(A, \preceq) \neq \emptyset$ el conjunto de elementos minimales de algún conjunto parcialmente ordenado (A, \preceq) . Sea también para $x \in A$ el conjunto

$$\mathcal{G}(x) := \{y \in A : y \preceq x\};$$

entonces se cumplen las siguientes implicaciones:

$$a) \ x \in \mathcal{M}(A, \preceq) \iff \mathcal{G}(x) \setminus \{x\} = \emptyset.$$

b) Si $\mathcal{M}(A, \preceq)$ es completo y $x \notin \mathcal{M}(A, \preceq)$ entonces $(\mathcal{G}(x) \setminus \{x\}) \cap \mathcal{M}(A, \preceq) \neq \emptyset$

Demostración: (a) Por definición,

$$x \in \mathcal{M}(A, \preceq) \Leftrightarrow (\nexists y \in A : y \prec x) \Leftrightarrow (\nexists y \in A : y \preceq x \text{ y } y \neq x)$$

son implicaciones ciertas. La última expresión puede reescribirse como $\exists y \in \mathcal{G}(x) : y \neq x$. Esto entonces equivale a decir que $\mathcal{G}(x) \setminus \{x\} \neq \emptyset$. ■

(b) Sea $x \in A$ tal que $x \notin \mathcal{M}(A, \preceq)$. Considerando la parte (a), es un hecho que $\mathcal{G}(x) \setminus \{x\} \neq \emptyset$. Dado que $\mathcal{M}(A, \preceq)$ es completo podemos decir que existe un elemento $x^* \in \mathcal{M}(A, \preceq)$ tal que $x^* \neq x$ y $x^* \prec x$. Entonces empleando la definición de $\mathcal{G}(\cdot)$ concluimos que necesariamente $x^* \in \mathcal{G}(x) \setminus \{x\}$, por lo tanto

$$(\mathcal{G}(x) \setminus \{x\}) \cap \mathcal{M}(A, \preceq) \neq \emptyset$$

■

1.3 Sumario

Hemos presentado, a lo largo de este capítulo, conceptos básicos de optimización y heurísticas, resaltando las complicaciones que motivan el uso de técnicas evolutivas en lugar de técnicas clásicas de optimización.

Hablamos también del problema de optimización multi-objetivo puntualizando algunas de sus complicaciones e introduciendo el concepto de *conjunto parcialmente ordenado*, el cual será básico para las discusiones acerca de la convergencia de los algoritmos que se tratarán en este trabajo.

A continuación expondremos algunos aspectos introductorios de la computación evolutiva, sus paradigmas y usos principales.

Capítulo 2

Computación evolutiva

Las técnicas que conforman la computación evolutiva son procedimientos inspirados en el principio natural de la evolución de las especies, que modelan la evolución como un proceso particular de optimización. Describiremos a continuación de manera muy general sus características principales:

- i) Para dichas técnicas, existen entidades (individuos) las cuales tienen la habilidad de reproducirse, con una cierta probabilidad; cada una de estas entidades representa una solución al problema que se desea resolver. Al reproducirse éstas, se ejecuta un proceso de “explotación” del espacio de búsqueda para mejorar cada vez más la solución y con esto acercarnos al óptimo.
- ii) A cada una de estas entidades se le asocia una habilidad (aptitud) mediante la cual se evalúa su propia supervivencia y la de sus “genes” a cada generación. Esto se relaciona con el valor de la función objetivo para la correspondiente interpretación de la entidad en términos de soluciones del problema.
- iii) Existe una población que contiene a dichas entidades, la cual se renueva generación tras generación bajo el principio de “selección del más apto” lo que en computación evolutiva se interpreta como una mejora en la solución a nuestro problema.
- iv) La población cuenta con diversidad entre sus entidades. Esto se interpreta en computación evolutiva como un muestreo representativo del espacio de búsqueda, es decir, el algoritmo cuenta con un mecanismo de “exploración.”

Basados en lo anterior han sido desarrolladas técnicas, con múltiples variaciones, para resolver problemas cuyo objetivo es dar una aproximación a la solución

óptima del problema en cuestión. Existen varias ventajas al utilizar las técnicas evolutivas. Podemos comenzar mencionando que son una manera de incorporar el poder de cómputo actual para la resolución de un problema y al igual que la mayoría de las heurísticas son útiles cuando se tienen espacios de búsqueda demasiado grandes. Para el empleo de cualquiera de estas técnicas no es necesario que dicho problema cumpla condiciones de continuidad y/o derivabilidad en sus funciones objetivo o restricciones ni que éstas sean sencillas de evaluar para un humano.

Entre los problemas apropiados para resolver con estas técnicas se encuentran aquellos en los que la región factible no se puede caracterizar de manera sencilla, no cumple condiciones de convexidad o conexidad, o no se cuenta con un conocimiento previo de ella.

Una descripción acerca del surgimiento, operadores, métodos, vertientes y demás aspectos de las principales técnicas evolutivas puede consultarse en [13].

2.1 Antecedentes y usos

Las técnicas evolutivas basan sus desarrollos y mejoras en la imitación directa de la naturaleza a través de la representación de soluciones. El Neo-Darwinismo (Capítulo 2 de [20]) basa la evolución de la vida en nuestro planeta en los procesos de reproducción, mutación y competencia para la selección de los individuos en generaciones futuras. A su vez, las técnicas evolutivas toman estos principios del Neo-Darwinismo y tratan de reproducirlos estadísticamente a través de un algoritmo que asemejándose a la evolución nos lleve de manera natural a una buena solución.

Componentes del ciclo evolutivo:

- *Población*: es el conjunto de individuos que representan las soluciones al problema. En términos de implementación la población es un arreglo de datos en el cual se guardan de manera ordenada las cadenas de información genética de todos y cada uno de los individuos (soluciones potenciales del problema).
- *Operadores genéticos*: son aquellos procedimientos que manipulan la información genética de la población para con esto producir nuevos individuos vistos como variaciones de la población original. Los operadores genéticos más comunes en los algoritmos evolutivos (AE's) son:

- *Mutación*: puede verse como la perturbación (variación) sobre uno o más alelos de la cadena genética del individuo. Al hacerse de manera aleatoria nos transporta hacia una región del espacio de búsqueda diferente a la del individuo original; esto nos ayuda a salir en el caso de que la búsqueda se estanque en algún óptimo local; por esto se dice que la mutación es el procedimiento del algoritmo que se dedica a la *exploración* del espacio de búsqueda.
- *Recombinación*: es el procedimiento mediante el cual se “mezclan” los genes de los padres para formar las cadenas cromosómicas de los descendientes, preservando presumiblemente los “buenos” genes que nos han llevado a obtener mejores soluciones al problema; la recombinación es el procedimiento del algoritmo que se dedica a la *explotación* de las zonas más prometedoras en el espacio de búsqueda.
- *Proceso de selección*: es aquel mediante el cual se designa qué individuos pasarán a formar parte de la siguiente generación. Cada uno de los individuos cuenta con un valor que determina su aptitud de supervivencia ante el entorno. Dicho valor de aptitud determina qué tan bueno es el individuo como solución al problema que se trata de resolver. A la determinación de dichos valores se le conoce como evaluación de la *función de aptitud* y dado que el proceso de selección toma en cuenta esta función, se imita en el algoritmo el principio de supervivencia del más apto.

Notaremos aquí que la función de aptitud debe estar relacionada con la función objetivo a optimizar, es decir que mientras mejor cumpla el individuo con la función objetivo, mayor será su aptitud de supervivencia ante el entorno. La diferencia entre estos dos conceptos, función objetivo y función de aptitud, es más bien conceptual para diferenciar que las funciones objetivo son una característica intrínseca del problema a optimizar, mientras la función de aptitud es propia del algoritmo evolutivo. Nótese sin embargo que muchas veces en la práctica (para el caso de optimización mono-objetivo) estas dos funciones coinciden y en otras tantas ocasiones la función de aptitud es el resultado de escalar proporcionalmente la función objetivo. Cuando tratamos con optimización multi-objetivo la función de aptitud representa precisamente el compromiso buscado (ver sección 1.2) entre todas las funciones objetivo del problema.

Las técnicas evolutivas han tenido diversas implementaciones conforme su uso se ha extendido. Se han aplicado para mejorar programas de computadora [28, 58,

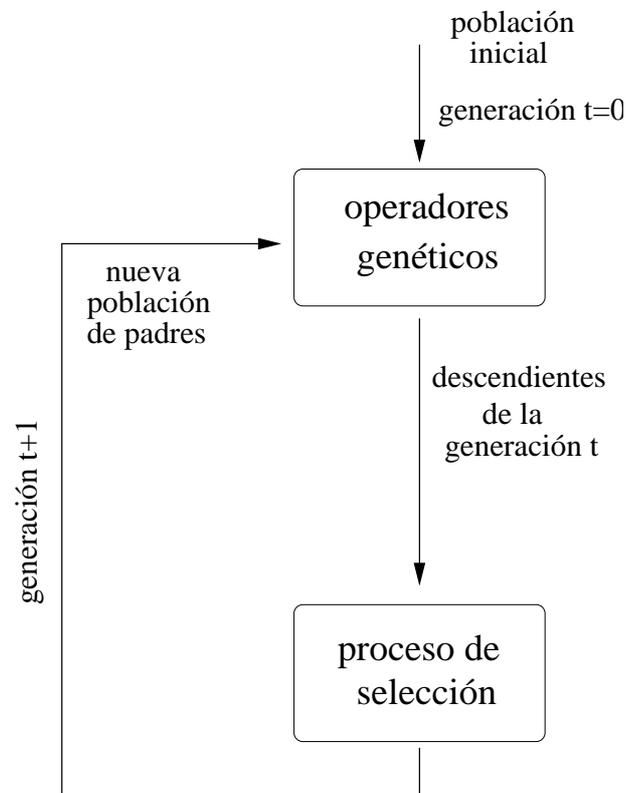


Figura 2.1: La base del ciclo evolutivo

34], diseño de circuitos y reconocimiento de patrones [33], aprendizaje de máquina [21], resolución de problemas combinatorios [38, 39], etc.

Cabe mencionar que la optimización no es el único uso que se le ha dado a los algoritmos evolutivos; Beyer, por ejemplo, desarrolla en [6] un algoritmo evolutivo que simula de manera evolutiva estados estacionarios en ciertos sistemas dinámicos.

2.2 Paradigmas principales

Existen tres paradigmas principales dentro de la computación evolutiva. Cada uno de éstos se desarrolló de manera independiente y con motivaciones distintas. A continuación se da una breve explicación de los principios de cada uno de ellos; un estudio detallado entre sus propiedades, diferencias y aplicaciones se encuentra en el libro de Thomas Bäck [3]. A lo largo de este trabajo nos referiremos a las

estrategias evolutivas (EE), específicamente sus propiedades de convergencia.

2.2.1 Estrategias evolutivas

El capítulo 4 de este trabajo se dedica para una descripción más amplia acerca de este paradigma; aquí presentamos un breve resumen de sus aspectos principales con el objeto de poderlo diferenciar de los otros dos paradigmas de CE descritos enseguida.

Las *estrategias evolutivas* (EE) se inspiran en la evolución que puede sufrir un individuo determinado afectado por la mutación y la recombinación de los genes de sus antepasados. Fueron concebidas en 1964 en la Universidad Técnica de Berlín, en Alemania, por un grupo de estudiantes de ingeniería: Ingo Rechenberg, Hans-Paul Schwefel y Paul Bienert. Dichos estudiantes necesitaban optimizar ciertas funciones para un diseño hidrodinámico el cual representaba un problema de alto grado de complejidad, y que además, no podía resolverse con técnicas de optimización tradicionales.

La versión original de la EE de dos miembros, denotada por $(1 + 1)$ -EE, usa un solo padre y produce a cada iteración un solo hijo. El mejor de entre los dos pasará a formar la siguiente generación en un ejemplo de cómo opera la selección extintiva¹. En la $(1 + 1)$ -EE, un individuo nuevo es producido usando:

$$\vec{x}^{t+1} = \vec{x}^t + N(0, \vec{\sigma})$$

donde $N(0, \vec{\sigma})$ es un vector de números Gaussianos independientes. Es decir, la mutación opera simplemente como una perturbación entre los individuos. Los números Gaussianos tienen media cero asemejándose a la naturaleza; es decir, en la naturaleza no se espera que ocurran mutaciones fuertes. La desviación estándar σ sirve para controlar el proceso denominado “autoadaptación” que consiste en variar σ dependiendo de si el número de éxitos obtenidos a través de la mutación es grande o no con respecto a un número fijo de generaciones de prueba.

Posteriormente Ingo Rechenberg propuso una variante de la estrategia evolutiva denominada $(\mu + 1)$ en la que cada generación consistía de μ padres que generaban un solo hijo, el cual podía reemplazar al peor de los μ padres de la población. El uso de múltiples hijos fue introducido por Schwefel en las estrategias denominadas $(\mu + \lambda)$ y (μ, λ) en las cuales cada población consta de μ padres los cuales generan λ hijos, Mediante la selección se eligen a los μ padres de la siguiente generación de entre los λ hijos en la estrategia (μ, λ) y de entre la unión de los μ padres y los λ

¹En la selección extintiva los individuos menos aptos tienen cero probabilidades de sobrevivir.

hijos en la $(\mu + \lambda)$.

La estrategia evolutiva multimiembro $(\mu + \lambda)$ puede resumirse en los pasos siguientes:

- I Se comienza con una población de μ individuos.
- II Se aplica algún operador de recombinación para generar λ hijos.
- III Se aplica el operador de mutación a los hijos.
- IV Sobreviven bajo el operador de selección los μ mejores individuos de la unión entre padres e hijos.

Dado que las estrategias evolutivas se basan en la mejora a nivel de los individuos, se permite la recombinación. En las estrategias evolutivas podemos mencionar, como un primer acercamiento a este operador, dos tipos de recombinación: Sexual y Panmítica. En la recombinación sexual intervienen dos individuos seleccionados para generar un descendiente; en la recombinación panmítica intervienen más de dos padres. En el caso de la recombinación panmítica el proceso es el siguiente:

1. Se tiene la población P_1, P_2, \dots, P_μ en la generación t con μ individuos.
2. Se eligen n de manera aleatoria P_{j_1}, \dots, P_{j_n} .
3. El cromosoma de cada padre P_{j_k} está dado por $(x_{j_k}^1, x_{j_k}^2, \dots, x_{j_k}^n) \in \mathbb{R}^n$
4. El hijo obtenido por recombinación H_i tendrá por cromosoma:

$$(x_{j_1}^1, x_{j_2}^2, \dots, x_{j_n}^n) \in \mathbb{R}^n$$

5. La recombinación se repite hasta tener un total de λ hijos

$$H_1, H_2, \dots, H_\lambda$$

6. Posteriormente se aplica la selección para obtener μ individuos de entre

$$\{P_1, P_2, \dots, P_\mu, H_1, H_2, \dots, H_\lambda\}$$

que formarán la población en la generación $t + 1$.

2.2.2 Programación evolutiva

La programación evolutiva es una técnica inspirada en el principio de la evolución al nivel de las especies. Utiliza selección probabilística y no requiere la cruce ni ningún tipo de recombinación puesto que una especie no puede mezclarse con otra. Propuesta en 1960, por Lawrence J. Fogel, la programación evolutiva ha sido utilizada para diferentes problemas de entre los que destacan: predicción, juegos y reconocimiento de patrones entre otras [20].

El algoritmo consiste en lo siguiente:

- I Generar una población inicial de manera aleatoria.
- II Aplicar el operador de mutación (existen muchos tipos diferentes).
- III Calcular la aptitud de cada descendiente y aplicar selección mediante torneo para retener a las mejores soluciones.

2.2.3 Algoritmos genéticos

Los algoritmos genéticos han sido ampliamente usados para la resolución de problemas diversos. Con ellos se introducen los conceptos, existentes en la naturaleza, de *genotipo* y *fenotipo*. El genotipo de un individuo representa la información genética que por sí mismo tiene, que ha heredado de sus antepasados y que a su vez transmitirá a sus descendientes. El fenotipo son las características visibles de cada individuo. Dentro del ámbito de la computación, llamaremos fenotipo a la representación natural de la solución del problema y genotipo a la cadena binaria que codifica dicha solución.

Desarrollado por John H. Holland [29, 30], el algoritmo genético opera entonces a nivel de genotipo de las soluciones mediante la siguiente secuencia:

- I Comenzar con una población inicial, la cual puede ser generada de manera aleatoria.
- II Calcular la aptitud de cada individuo.
- III Aplicar el operador de selección con base en la aptitud de la población.
- IV Aplicar los operadores genéticos de cruce y mutación a la población actual para con éstos generar a la población de la siguiente generación.

V Ir al paso II hasta que la condición de paro se satisfaga.

En el algoritmo genético el operador principal es la cruce, utiliza los conceptos de *esquema*² y *bloque constructor*³ y con ellos sostiene la llamada *hipótesis de los bloques constructores*:

“La cruce dentro de un algoritmo genético tiene la capacidad de recombinar las instancias de esquemas favorables, para formar otros igualmente buenos o mejores.”

Al igual que en muchas otras heurísticas, el comportamiento del algoritmo genético es altamente dependiente de los parámetros iniciales (tamaño de la población, porcentaje de cruce, porcentaje de mutación, número de generaciones, etc.).

2.3 Abordaje de problemas multi-objetivo con técnicas evolutivas

Las técnicas evolutivas son particularmente atractivas para el tratamiento de problemas multi-objetivo por varias razones; aquí comentaremos algunas de ellas.

La solución de un problema multi-objetivo no se resume a un único resultado óptimo, sino que consiste de una colección de valores maximales a la vez; las técnicas evolutivas se acoplan bien en este aspecto ya que en ellas se procesa de manera simultánea una población de individuos, los mismos que representan soluciones a mejorar en cada iteración. Al final de la ejecución del algoritmo evolutivo se obtiene una muy buena aproximación tanto al frente de Pareto como al conjunto de óptimos de Pareto (ver sección 2.4.2), caracterizando con esto perfectamente a las soluciones para ser usadas directamente por el tomador de decisiones.

En la vida real no es sencillo trabajar con problemas que no pueden caracterizarse (o cuya definición es muy compleja) algebraica o analíticamente, que tienen una región factible geoméricamente inusual, o cuyo espacio de búsqueda es demasiado grande. Este tipo de complicaciones se extienden con creces al caso de problemas multi-objetivo resultando en complejidades múltiples. Sin embargo, estas características que hacen a los problemas “difíciles” de manejar con técnicas

²Un esquema es un patrón de valores de los genes en un cromosoma.

³Un bloque constructor es un grupo pequeño y compacto de genes que han co-evolucionado de tal forma que su introducción en cualquier cromosoma tiene una alta probabilidad de incrementar la aptitud de dicho cromosoma.

clásicas son precisamente la inspiración para el desarrollo de las técnicas evolutivas, y en efecto, éstas han demostrado salvar dichas dificultades durante la búsqueda. Como una referencia acerca de las técnicas evolutivas con enfoque multi-objetivo puede consultarse [12].

2.3.1 La importancia del trabajo en convergencia

Dentro del área de Computación Evolutiva aumentan cada vez más las técnicas para implementar algoritmos nuevos y mejorar la eficiencia de los existentes. Hablando en términos de la optimización multi-objetivo, los investigadores avanzan para adaptar a esta rama las ventajas de las técnicas evolutivas. Sin embargo, paralelamente a estos esfuerzos debe trabajarse en la justificación teórica del funcionamiento de estas técnicas dentro del área. Esto no sólo es necesario para darle un carácter “formal” a los trabajos, sino también para poder comprender mejor las mismas técnicas que se han diseñado y generar ideas para implementar mejoras, a la par de desarrollar nuevas propuestas.

Dentro del desarrollo de cualquier teoría que justifique un método computacional es de suma importancia establecer las bases de lo que significa la convergencia de la técnica (algoritmo, procedimiento etc.). En el área de optimización siempre se han estudiado las condiciones para que un problema tenga solución y para que ésta sea hallada por un algoritmo determinado.

Por éstas y otras razones, estudiar la convergencia para las técnicas evolutivas se vuelve fundamental, pues sostiene la validez de las conclusiones del trabajo realizado en el área y puede dar ideas acerca de nuevos caminos a seguir; además, sin una teoría de convergencia de las técnicas evolutivas específica para problemas multi-objetivo, no se puede aplicar al máximo la maquinaria lógica existente para la formalización de sus resultados, aunado a que los conceptos desarrollados en otros aspectos teóricos, como pueden ser el manejo de preferencias o el estudio de métricas, no podrían explotarse al máximo.

Contribuir al estudio de los fundamentos teóricos en el área de la computación ayuda a entender mejor hacia dónde nos llevan los procedimientos que implementamos, y nos da ideas para desarrollar mejoras y/o buscar nuevas alternativas. Dentro de la optimización evolutiva multi-objetivo se requiere de un mayor esfuerzo en cuanto al desarrollo teórico por ser un área relativamente nueva y en la cual su origen (la computación evolutiva) se sustenta en principios eminentemente empíricos.

2.3.2 Estudios actuales sobre convergencia de AE's en problemas multi-objetivo

Por ahora los esfuerzos para el desarrollo de la teoría de convergencia de algoritmos evolutivos en problemas multi-objetivo se han enfocado al estudio, desde el punto de vista de los procesos estocásticos, del comportamiento en el límite asintótico $t \mapsto \infty$ de algoritmos evolutivos genéricos, es decir, sin especificar el efecto que los operadores de cruce, recombinación y mutación tienen en la convergencia. De esta manera, se han podido modelar los algoritmos evolutivos a través de la teoría de probabilidad para aplicar posteriormente los teoremas de convergencia en el tiempo, con los que estas teorías matemáticas cuentan desde hace varias décadas.

Dos enfoques principales han surgido, uno analizando espacios de búsqueda discretos bajo el cual se encuentran principalmente trabajos de los investigadores Günter Rudolph [48, 52] y David Van Veldhuizen [60]. El otro enfoque se da para los espacios de búsqueda continuos y existen trabajos también de Günter Rudolph [49] y Thomas Hanne [25].

2.4 Jerarquización de Pareto

Dentro de un algoritmo evolutivo (AE) se ejecuta un proceso de selección de individuos antes de la aplicación del operador de cruce. La mayor parte del tiempo se trata de seleccionar a los más aptos. La aptitud de cada individuo está estrechamente relacionada con la manera en que el individuo cumple la función objetivo; basados en dicha aptitud, se puede establecer un orden entre la población que indique qué individuo se prefiere sobre otro por representar una mejor solución al problema. En el ámbito de la optimización multi-objetivo y con el fin de aplicar la selección de individuos se tiene que decidir un cierto esquema de mejoría de una solución sobre otra, es decir cuáles se elegirán para la cruce por ser más aptas; a esta relación de mejoría de un individuo sobre otro es a lo que llamaremos *Esquema de Dominación*.

La definición de un Esquema de Dominación se basa principalmente en que la solución de un problema multi-objetivo no es única y por lo tanto el tomador de decisiones debe elegir de entre una gama de posibles soluciones que no se pueden mejorar entre sí, es decir que no se dominan, la que mejor convenga a sus necesidades. Si el algoritmo evolutivo basa su mecanismo de selección en dicho esquema de dominación entonces dará como resultado un conjunto de soluciones razonablemente buenas para elegir de entre ellas.

2.4.1 Esquema de dominación

Dentro del campo de los números reales se encuentra definido el orden \leq de manera natural. Para \mathbb{R}^n podemos extender el concepto mediante la siguiente definición

Definición 2.4.1 Dados x, y vectores en \mathbb{R}^n :

$$x \leq y \quad \text{si y sólo si} \quad x_k \leq y_k \quad \text{para todo} \quad k \in \{1, \dots, n\}.$$

y

$$x < y \quad \text{si y sólo si} \quad x \leq y \quad \text{y} \quad x \neq y.$$

2.4.2 Frente de Pareto

Una opción común a usarse como relación de dominación es la conocida dominación de Pareto definida a continuación:

Definición 2.4.2 Dado el problema multi-objetivo

$$\text{Minimizar:} \quad f(x)$$

donde

$$f : F \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^q, \quad q \geq 2$$

con $A \subseteq F$ la región factible. Decimos que un vector $x^* \in A$ es *no dominado* o un *óptimo de Pareto* si no existe un vector $x \in A$ tal que

$$f(x) < f(x^*).$$

Así, la respuesta al problema de hallar las mejores soluciones (las soluciones no dominadas, como quiera que se defina la dominación dentro de la técnica) en un problema multi-objetivo es a lo que llamaremos el *conjunto solución* del problema.

El conjunto de valores de la función objetivo con dominio restringido a los vectores del conjunto solución (es decir, los vectores no dominados) es lo que conoceremos como frente de Pareto (ver [26]). El concepto de frente de Pareto fue introducido en el siglo XIX por el economista italiano Vilfredo Pareto y se ha utilizado en el estudio de problemas multi-objetivo con técnicas clásicas de Investigación de Operaciones. En general y sobre todo para problemas de la vida real no es sencillo hallar el frente de Pareto de manera analítica (y en la mayoría de los casos es imposible).

Un concepto íntimamente relacionado con el Frente de Pareto es el de óptimo de Pareto. Tanto el óptimo de Pareto como el Frente de Pareto son el marco sobre el que se trabajaría dentro de la toma de decisiones multicriterio.

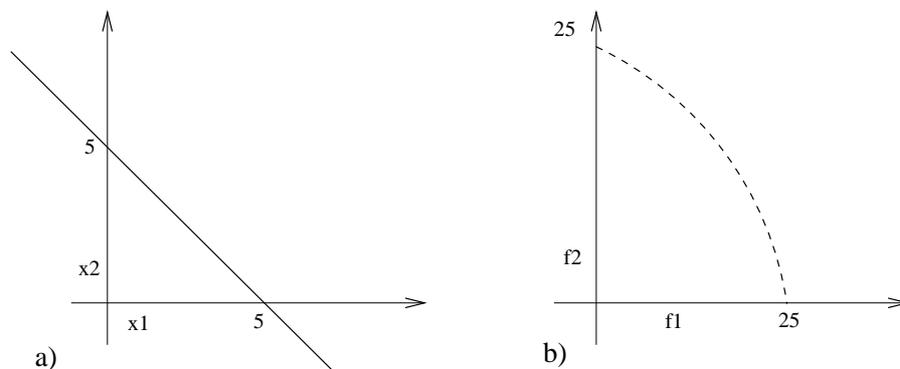


Figura 2.2: a) Los ejes coordenados y la recta mostrada determinan la región factible del problema, b) La línea punteada, en el espacio de las funciones objetivo, representa el frente de Pareto.

Definición 2.4.3 El conjunto $E(A, f)$ de soluciones de Pareto eficientes (también conocido como conjunto de óptimos de Pareto) se define de la manera siguiente:

$$E(A, f) := \{a \in A : \nexists b \in A \text{ que cumpla con } f(b) < f(a)\}.$$

Es decir, el conjunto de todos los vectores no dominados bajo el esquema de Pareto.

En resumen, el conjunto de óptimos de Pareto es el espacio solución del problema y el Frente de Pareto es su imagen con respecto a la función

$$f : F \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^q, \quad q \geq 2$$

a optimizar.

Definición 2.4.4 Dado un problema MOP, llamaremos *frente de Pareto verdadero* y denotaremos como \mathcal{F}^* al frente de Pareto teórico que cada problema posee en particular.⁴

Llamaremos *frente de Pareto conocido* y denotaremos como \mathcal{F}_t al frente de Pareto que se desprende⁵ del algoritmo evolutivo al término de la generación t .

⁴En la mayoría de los problemas este frente de Pareto teórico no se conoce. En algunos puntos de la literatura éste se denota como \mathcal{F}_{true} .

⁵En algunos puntos de la literatura el frente de Pareto arrojado al finalizar el AE se denota como \mathcal{F}_{known} .

Para ilustrar el concepto de frente de Pareto damos un ejemplo a continuación.

Ejemplo

Consideremos el problema multi-objetivo

$$\text{Maximizar } f := (f_1, f_2)$$

donde

$$f_i(x) \mapsto x_i^2 \text{ para } i = 1, 2$$

y se tiene la siguiente región factible (ver figura 2.2):

$$F := \{x \in \mathbb{R}^2 : x_1 \geq 0; x_2 \geq 0; x_1 + x_2 \leq 5\}$$

El vector $(0, 5)$ es un vector no dominado, al igual que el $(5, 0)$ y otros más.

2.5 Sumario

En este capítulo se presentaron, de manera introductoria, las ideas básicas de las técnicas de computación evolutiva dando la motivación para abordar con ellas a los problemas multi-objetivo. Se mencionaron varios aspectos acerca de sus aplicaciones y paradigmas principales, también establecimos los conceptos de óptimo y frente de Pareto.

En el capítulo siguiente expondremos los conceptos y resultados básicos de teoría de la probabilidad y procesos estocásticos necesarios para explicar los detalles de las técnicas matemáticas utilizadas para la demostración de la convergencia del algoritmo conocido como estrategia evolutiva.

Capítulo 3

Herramientas básicas

3.1 Espacios de probabilidad

Supongamos que Ω es cualquier conjunto, no necesariamente finito.

3.1.1 Conjuntos medibles y σ -álgebras

Definición 3.1.1 Una familia no vacía \mathcal{A} de subconjuntos de Ω se dice que es un **álgebra** si cumple con:

- i.* $A^c \in \mathcal{A}$ para todo subconjunto $A \in \mathcal{A}$.
- ii.* Si cualesquiera A_1, A_2 pertenecen a \mathcal{A} entonces se concluye que $A_1 \cup A_2 \in \mathcal{A}$.

Decimos que la familia \mathcal{A} es una **σ -álgebra** si cumple con las anteriores propiedades (*i* y *ii*) y además cumple:

- iii.* $\bigcup_{n=1}^{\infty} A_n \in \mathcal{A}$ siempre que $A_n \in \mathcal{A}$, $n \geq 1$.

Utilizando las leyes de De Morgan se deduce que el símbolo \cup en (*ii*) puede ser reemplazado fácilmente por \cap . Si la condición (*ii*) es satisfecha, se concluye de manera inmediata que la familia \mathcal{A} es cerrada bajo uniones finitas numerables, es decir, que para cada entero positivo n se cumple que

$$\bigcup_1^n A_j \in \mathcal{A} \quad \text{siempre que} \quad A_j \in \mathcal{A}, \quad 1 \leq j \leq n$$

y de igual forma

$$\bigcap_1^n A_j \in \mathcal{A} \quad \text{siempre que} \quad A_j \in \mathcal{A}, \quad 1 \leq j \leq n.$$

Si tomamos $A_n = A_2$ para todo $n \geq 2$ se ve que la condición (iii) implica (ii), por lo que para que una familia de subconjuntos sea una σ -álgebra sólo son necesarias (i) y (iii). Por estas propiedades se dice que una σ -álgebra es *numerablemente cerrada* bajo las operaciones de complemento, unión e intersección. También puede demostrarse de manera sencilla que una σ -álgebra es numerablemente cerrada bajo la operación de diferencia de conjuntos.

El hecho de que una σ -álgebra es no vacía implica que $\emptyset \in \mathcal{A}$ y $\Omega \in \mathcal{A}$ pues:

$$A \in \mathcal{A} \Rightarrow A^c \in \mathcal{A} \Rightarrow A \cup A^c = \Omega \in \mathcal{A} \Rightarrow \Omega^c = \emptyset \in \mathcal{A}.$$

Por lo tanto la mínima σ -álgebra que puede construirse es $\mathcal{A} = \{\emptyset, \Omega\}$.

Definición 3.1.2 *El conjunto Ω junto con una σ -álgebra \mathcal{F} de sus subconjuntos es a lo que llamaremos un **espacio medible**, y lo denotaremos por*

$$(\Omega, \mathcal{F}).$$

Podemos expresar la *recta real* como

$$\mathbb{R} = [-\infty, \infty] = \{\omega : -\infty \leq \omega \leq \infty\}.$$

considerando para la comparación a $-\infty$ y ∞ como números reales.

Ahora fijemos nuestra atención en el conjunto

$$(-\infty, \infty) = \{\omega : -\infty < \omega < \infty\}$$

al cual llamaremos *línea real finita*. A los elementos de ésta los referiremos como *números reales finitos*.

Para $x \in \mathbb{R}$, los conjuntos $[-\infty, x]$, $(-\infty, x)$, $(-\infty, x]$, $[-\infty, x)$, $[x, \infty]$, (x, ∞) , $[x, \infty)$ y $(x, \infty]$ son intervalos infinitos.

Definición 3.1.3 *Llamaremos **conjuntos de Borel** en \mathbb{R} a los elementos de la σ -álgebra \mathcal{B} generada ¹ por la colección de los intervalos finitos de la forma $[-\infty, x)$ con x un real finito. El espacio medible $(\mathbb{R}, \mathcal{B})$ es llamado la **línea de Borel**, o *espacio de Borel 1-dimensional*.*

¹La σ -álgebra generada por una colección S de subconjuntos es la intersección de todas las σ -álgebras que contienen a S , cumple con ser la σ -álgebra mínima que a su vez contiene a S .

Para cualquier intervalo

$$J \subset \mathbb{R}$$

la σ -álgebra generada por la familia de todos los subintervalos de J coincide con $\mathcal{B} \cap J$. Es natural referir dicha σ -álgebra como la familia de subconjuntos de Borel de J .

Las variables aleatorias juegan un papel importante en el análisis de resultados dentro de un experimento; en este trabajo las explotamos fuertemente al modelar las EE's. Para definir a las variables aleatorias es necesario introducir algunos de los siguientes conceptos:

Definición 3.1.4 Sean $(\Omega_1, \mathcal{F}_1)$ y $(\Omega_2, \mathcal{F}_2)$ espacios medibles, la aplicación

$$X : \Omega_1 \rightarrow \Omega_2$$

se dice ser una **transformación medible** o **función \mathcal{F}_1 medible** si cumple que

$$X^{-1}(\mathcal{F}_2) \subset \mathcal{F}_1.$$

Definición 3.1.5 Una función medible X del espacio (Ω, \mathcal{F}) a la línea de Borel $(\mathbb{R}, \mathcal{B})$ es llamada **función medible real en (Ω, \mathcal{F})** .

Dado que \mathcal{B} es la σ -álgebra generada por la colección de intervalos

$$[-\infty, x), -\infty < x < \infty, \quad \text{ó} \quad (x, \infty], -\infty < x < \infty$$

se sigue del lema 1 en [10] que con valores reales X en Ω es medible si y sólo si para cada valor finito real x se tiene

$$\{w : X(w) < x\} \in \mathcal{F} \quad \text{ó} \quad \{w : X(w) > x\} \in \mathcal{F}, \quad x \in (-\infty, \infty).$$

En el caso especial de una transformación medible f de la línea de Borel $(\mathbb{R}, \mathcal{B})$ a sí misma, referiremos a f como una **función Borel** o una **función Borel-medible**. Dado que cada conjunto abierto en $(-\infty, \infty)$ es una unión numerable de intervalos abiertos, y cada intervalo abierto es un conjunto de Borel, cualquier función continua en $(-\infty, \infty)$ es una función Borel. Así, cualquier función real en $[-\infty, \infty]$ con rango $[-\infty, \infty]$ la cual es continua en $(-\infty, \infty)$ es siempre una función Borel.

3.1.2 Funciones aditivas y espacio de probabilidad

Sea Ω un conjunto y \mathcal{A} una familia de subconjuntos de Ω con $\mathcal{A} \neq \emptyset$. Una función en \mathcal{A} es una función con valores reales definida sobre los elementos de \mathcal{A} .

Si $A \subset \Omega$ y $A = \bigcup_{n=1}^m A_n$ con $\{A_n, n = 1, \dots, m\}$ subconjuntos disjuntos a pares de \mathcal{A} , ésta es llamada una partición de A en \mathcal{A} . Si $A \subset \Omega$ y $A = \bigcup_{n=1}^{\infty} A_n$ con $\{A_n, n = 1, 2, \dots\}$ subconjuntos disjuntos a pares de \mathcal{A} , ésta es llamada una σ -partición de A en \mathcal{A} .

Definición 3.1.6 La función μ se dice ser σ -aditiva o **numerablemente aditiva** si para cualquier $A \in \mathcal{A}$ y cualquier σ -partición $\{A_n, n = 1, 2, \dots\}$ de A en \mathcal{A} en la que $\sum \mu(A_n)$ está definida, se cumple:

$$\mu(A) = \sum_{n=1}^{\infty} \mu(A_n).$$

Definición 3.1.7 Una función σ -aditiva no negativa μ en la familia \mathcal{A} donde $\emptyset \in \mathcal{A}$ se dice una **medida** si

$$\mu(\emptyset) = 0.$$

En el caso de una σ -álgebra \mathcal{F} se puede demostrar de su definición que siempre se tiene la contención $\emptyset \in \mathcal{F}$.

Definición 3.1.8 Sea \mathcal{A} un álgebra de subconjuntos de Ω . Una función $\mu = \mu(A), A \in \mathcal{A}$, en \mathcal{A} que toma valores en $[0, \infty]$, es llamada una **medida finitamente aditiva** definida sobre \mathcal{A} si

$$\mu(A \cup B) = \mu(A) + \mu(B)$$

para cada par de conjuntos disjuntos $A, B \in \mathcal{A}$.

Una medida finitamente aditiva μ con $\mu(\Omega) < \infty$ se dice ser finita. Si $\mu(\Omega) = 1$ se dice que es una medida de probabilidad finitamente aditiva, o simplemente una probabilidad finitamente aditiva.

Definición 3.1.9 Una terna $(\Omega, \mathcal{F}, \mu)$ donde μ es una medida en la σ -álgebra \mathcal{F} de subconjuntos de Ω es llamada un **espacio de medida**.

Un **espacio de Probabilidad** es un espacio de medida (Ω, \mathcal{F}, P) tal que

$$P(\Omega) = 1.$$

Se sigue de la definición de σ -aditividad que una medida μ de un álgebra \mathcal{A} es monótona, ésto es

$$\mu\{A_1\} \leq \mu\{A_2\} \text{ siempre que } A_1 \subset A_2, \quad A_1, A_2 \in \mathcal{A}$$

Dentro de un espacio de probabilidad (Ω, \mathcal{F}, P) los conjuntos $A \in \mathcal{F}$ son llamados **eventos** y al valor real que toma $P(A)$ se le llama la *probabilidad* del evento A . Dado que P por ser una medida es monótona se tiene que

$$0 = P\{\emptyset\} \leq P\{A\} \leq P\{\Omega\} = 1$$

El conjunto de referencia Ω normalmente es llamado **espacio muestra**. Desde el punto de vista de la probabilidad, un evento es sólo un subconjunto del espacio muestra. Desde el punto de vista práctico, decimos que un evento “*ocurre*” si el resultado de un experimento alatorio está contenido en él

Ejemplo 3.1.10 Supongamos el experimento aleatorio que consiste de lanzar un dado. El conjunto $\Omega = \{1, 2, 3, 4, 5, 6\}$ es el espacio muestra donde cada número representa que el dado muestra 1, 2, 3, 4, 5 ó 6 puntos respectivamente en su cara superior. Si $A = \{2, 3, 5\}$, entonces A es el evento que representa obtener con el dado un número primo, mientras que el evento $B = \{1, 3, 5\}$ representa obtener con el dado un número impar.

Ejemplo 3.1.11 Supongamos T el experimento aleatorio que consiste en determinar el número de generación (iteración) en que el algoritmo de la EE genera a un descendiente específico h . El espacio muestral está dado por $\Omega = \{1, 2, \dots, N\}$ donde N es el número máximo de generaciones que se ejecuta el algoritmo.

La ocurrencia del evento $A = \{1, 2, \dots, \lfloor \frac{N}{10} \rfloor\}$, con respecto a T , representa que h se generó en menos del 10% del tiempo total propuesto.

Los eventos que tienen probabilidad cero son llamados **eventos imposibles**. A los subconjuntos que se les asocia la probabilidad uno se les llama **eventos seguros**.

3.1.3 Variables aleatorias

Definición 3.1.12 Sean (Ω, \mathcal{F}, P) un espacio de probabilidad y (Ψ, \mathcal{G}) un espacio medible. Una función medible $\xi = \xi(w)$ de (Ω, \mathcal{F}, P) a (Ψ, \mathcal{G}) se referirá como **variable aleatoria**.

Supongamos (Ω, \mathcal{F}) un espacio medible, y sea $(\mathbb{R}, \mathcal{B})$ la línea de Borel.

Definición 3.1.13 Una función real $X = X(w)$ definida en (Ω, \mathcal{F}) se dice ser una variable aleatoria real si y sólo si X es una función \mathcal{F} -medible real, es decir del espacio (Ω, \mathcal{F}) a la línea de Borel $(\mathbb{R}, \mathcal{B})$.

Ejemplo El ejemplo más simple de una variable aleatoria es el indicador $I_A(w)$ de un conjunto arbitrario $A \in \mathcal{F}$ definida por

$$I_A(w) = \begin{cases} 1 & \text{si } w \in A \\ 0 & \text{en otro caso.} \end{cases}$$

Vectores aleatorios

Definición 3.1.14 Un **vector aleatorio** de dimensión n

$$X = (X_1, X_2, \dots, X_n),$$

es aquel en el cual cada una de sus componentes X_1, X_2, \dots, X_n son variables aleatorias sobre el mismo espacio de probabilidad (Ω, \mathcal{F}, P) con imagen en un mismo espacio numérico Ω'

3.2 Funciones de distribución

Sea (Ω, \mathcal{F}, P) un espacio de probabilidad y X una variable aleatoria del espacio (Ω, \mathcal{F}, P) a (Ω', \mathcal{F}') .

Definición 3.2.1 La función $F_X : \mathcal{F}' \rightarrow [0, 1] \subset \mathbb{R}$ definida por

$$F_X(A') := P(X^{-1}(A')) \text{ para todo } A' \in \mathcal{F}'$$

es llamada la **función de distribución de probabilidad** de X .

La función F_X es de hecho una medida de probabilidad para el espacio (Ω', \mathcal{F}') , por lo que a $(\Omega', \mathcal{F}', F_X)$ se le llama espacio de probabilidad inducido por X .

Definición 3.2.2 Sea $X = (X_1, X_2, \dots, X_n)$ un vector aleatorio. La función

$$f_X : \mathbb{R}^n \rightarrow [0, 1] \text{ definida para } v = (v_1, v_2, \dots, v_n) \text{ como}$$

$$\begin{aligned}
F_X(v) &:= F_{X_1, X_2, \dots, X_n}(v_1, v_2, \dots, v_n) \\
&:= F_{X_1, X_2, \dots, X_n}(X_1 \leq v_1, X_2 \leq v_2, \dots, X_n \leq v_n) \\
&:= P\{\cup_{i=1}^n \{w \in \Omega : X_i(w) \leq v_i\}\}
\end{aligned}$$

es llamada la función de **distribución de probabilidad combinada** para el vector aleatorio X .

Definición 3.2.3 Sean X_1, X_2, \dots, X_n variables aleatorias, todas ellas sobre un mismo espacio de probabilidad (Ω, \mathcal{F}, P) , se dicen ser **variables aleatorias independientes** si y sólo si cumplen:

$$F_{X_1, X_2, \dots, X_n}(v_1, v_2, \dots, v_n) = \prod_{i=1}^n F_{X_i}(v_i).$$

Los vectores aleatorios se clasifican principalmente en dos tipos:

Definición 3.2.4 (a) Un vector aleatorio X se dice ser **discreto**, si el conjunto de valores S , que pueden ser alcanzados por X con probabilidad positiva es a lo más numerable. El conjunto S es llamado en este caso el **soporte** de un vector aleatorio discreto.

(b) Un vector aleatorio X se dice ser **continuo** si existe una función no negativa $f_X(t_1, \dots, t_n)$, tal que para cualquier vector $v = (v_1, \dots, v_n) \in \mathbb{R}^n$ es válida la relación

$$F_X(v) = \int_{-\infty}^{v_1} \dots \int_{-\infty}^{v_n} f_X(t_1, \dots, t_n) dt_n \dots t_1.$$

Entonces la función f_X es llamada la función de **densidad de probabilidad** de un vector aleatorio X , y el conjunto

$$S = \{v \in \mathbb{R}^n : f_x(v) > 0\}$$

es llamado el **soporte** de un vector aleatorio continuo.

3.3 Esperanza condicional

Definición 3.3.1 Sea $g(\cdot)$ una función medible evaluada en los reales, y sea X una variable aleatoria discreta con valores en el conjunto $\{x_1, x_2, \dots\}$. Si

$$\sum_i |g(x_i)| P\{X = x_i\} < \infty$$

entonces definimos la **esperanza** de la variable aleatoria discreta $g(X)$ como

$$E[g(X)] := \sum_i |g(x_i)| P\{X = x_i\}.$$

Si X es una variable aleatoria continua con función de densidad f_X y

$$\int_{-\infty}^{\infty} |g(x)| f_X(x) dx < \infty$$

entonces definimos la **esperanza** de la variable aleatoria $g(X)$ como

$$E[g(X)] := \int_{-\infty}^{\infty} |g(x)| f_X(x) dx.$$

Si $g(x) = x$ entonces $E[X]$ es llamada simplemente la **esperanza** de X . Si $g(x) = x^k$ entonces $E[X^k]$ es llamada el k -ésimo momento de X .

Proposición 3.3.2 Sean X y Y dos variables aleatorias y, sean a, b constantes. Entonces:

- a) $E[a] = a$
- b) $E[aX + b] = aE[X] + b$
- c) $E[X + Y] = E[X] + E[Y]$
- d) $E[X] \leq E[Y]$ si $X \leq Y$ excepto a lo más en un conjunto con probabilidad cero.
- e) (*Desigualdad de Markov*) Si $X \geq 0$ excepto a lo más en un conjunto con probabilidad cero, entonces $E[X] \geq 0$ y para cualquier $\epsilon \in (0, \infty)$

$$P\{X > \epsilon\} \leq e^{-1} E[X].$$

- f) Si X y Y son independientes, entonces $E[X \cdot Y] = E[X] \cdot E[Y]$.

Demostración: Ver [10] y [47]

El operador esperanza puede extenderse para vectores de manera natural:

Definición 3.3.3 *El vector esperanza de un vector aleatorio*

$$X = (X_1, X_2, \dots, X_n)^T$$

está dado por

$$E[X] = (E[X_1], E[X_2], \dots, E[X_n])^T.$$

Definición 3.3.4 *Sea X una variable aleatoria \mathcal{F} -medible tal que $E[X] < \infty$ y sea \mathcal{G} una sub- σ -álgebra de \mathcal{F} . Entonces la variable aleatoria \mathcal{G} -medible definida como*

$$Y := E[X|\mathcal{G}]$$

que satisface

$$\int_G Y dP = \int_G X dP$$

para todo $G \in \mathcal{G}$ es llamada la **esperanza condicional** de X relativa a \mathcal{G} .

A partir de la esperanza condicional puede definirse la **probabilidad condicional** de X relativo a \mathcal{G} como :

$$P\{G|\mathcal{G}\} := E[1_G|\mathcal{G}].$$

Si X_1, \dots, X_n es una familia de variables aleatorias las cuales generan la σ -álgebra

$$\mathcal{F} = \sigma(X_1, \dots, X_n)$$

regularmente se escribe $E[X|X_1, \dots, X_n]$ en lugar de $E[X|\mathcal{F}]$.

3.4 Modelos estocásticos

Hablando en términos de variables aleatorias y sus esperanzas existen varias propiedades de convergencia que, para ser analizadas a fondo requieren el estudio de ciertas relaciones entre las mismas variables aleatorias. El estudio de estas relaciones y sus consecuencias puede ubicarse dentro del área de procesos estocásticos. Los modelos estocásticos son usados para estudiar cómo se comportan los fenómenos compuestos por flujos de eventos en el tiempo, especialmente aquellos que exhiben características variables, es decir variables aleatorias.

Definición 3.4.1 *Un **proceso estocástico** $X_\bullet = \{X_t\}$ con $t \in T$, con conjunto de índices T , se define como una familia de variables aleatorias discretas de un espacio de probabilidad (Ω, \mathcal{F}, P) a un espacio medible (Ω', \mathcal{B}) .*

Si el conjunto T de índices es igual a \mathbb{N}_0 se dice que tenemos un proceso estocástico **discreto en el tiempo**. El espacio imagen Ω' de $X_\bullet = \{X_t\}$, con $t \in T$, es llamado el **espacio de estados** del proceso.

3.4.1 Cadenas de Markov finitas

Las cadenas de Markov, al igual que las caminatas aleatorias son un proceso discreto en el tiempo

Dentro de un conjunto finito de estados (espacio de estados de un proceso estocástico), las cadenas de Markov son una herramienta que mediante la representación matricial nos da elementos para modelar los cambios de un estado a otro.

Supongamos que se tiene A , un conjunto finito numerado de 1 a n . Una cadena de Markov describe una trayectoria probabilística dentro del conjunto de estados A . Las probabilidades de que se de una transición del estado $i \in A$ al estado $j \in A$ se denotan por $p_{ij}(t)$ para todos $i, j \in A$.

Si $p_{ij}(t) = p_{ij}(s)$ para todos $t, s \in T$ entonces la cadena de Markov relacionada se dice ser una cadena de Markov **homogénea**. Dado que A es finito, las probabilidades de transición pueden agruparse en la matriz de transición

$$P = (p_{ij})_{i,j \in S}.$$

Definición 3.4.2 Si S es un conjunto finito y

$$\{X_t : t \in \mathbb{N}_0\}$$

es una S variable aleatoria con la propiedad de que

$$\begin{aligned} P\{X_{t+1} = j | X_t = i, X_{t-1} = t-1, \dots, X_0 = i_0\} \\ = P\{X_{t+1} = j | X_t = i\} = p_{ij} \end{aligned}$$

Para todo $t \geq 0$ y para cualquier pareja $(i, j) \in A \times A$ entonces la sucesión

$$\{X_t : t \in \mathbb{N}_0\}$$

es llamada una **cadena de Markov finita homogénea** del espacio de estados S .

Lema 3.4.3 Una cadena de Markov homogénea con espacio de estados finito y matriz de transición irreducible (propiedad definida en la siguiente sección) visita cada estado una infinidad de veces con probabilidad uno de manera independiente a la distribución inicial.

Demostración: Ver [31] pág. 89. ■

3.4.2 Análisis en mono-objetivo de algoritmos genéticos con cadenas de Markov

Con el fin de ejemplificar el uso de procesos de Markov en el análisis de algoritmos evolutivos hablaremos en esta sección acerca de los algoritmos genéticos y su modelo estocástico mono-objetivo. Este enfoque de cadenas de Markov para AG's ha sido estudiado ampliamente por autores como G. Rudolph [45] [47], Michael D. Vose [40] y L. Aarts [16]. Como un primer acercamiento a modelar los algoritmos genéticos como procesos estocásticos, daremos una breve descripción de la caracterización de los estados del proceso y sus probabilidades de transición.

El vector $\pi(t)$ con $\pi_i(t) = P\{X_t = i\}$ denota la distribución de la cadena de Markov en el paso $t \geq 0$ y P^t representa el producto t veces de la matriz P consigo misma, entonces

$$\pi(t) = \pi(t-1)P = \pi(0)P^t$$

para todo $t \geq 1$, por lo que una cadena de Markov finita homogénea está completamente especificada por su distribución inicial $\pi(0)$ y su matriz de transición P .

Las probabilidades de transición en el k -ésimo paso son:

$$p_{ij}(k) = P\{X_k = j | X_0 = i\} = e_i P^k e_j'$$

donde e_i es el i -ésimo vector unitario tal que

$$\pi_j(t) = \sum_{i \in S} \pi_i(0) \cdot p_{ij}(t).$$

Debido a que las cadenas de Markov están determinadas por sus matrices de transición, es de gran utilidad clasificarlas para simplificar los resultados.

Matrices de transición

Una matriz $M : n \times m$ se dice que es **no negativa** si $p_{ij} \geq 0$ y **positiva** si $p_{ij} > 0$ para todos $i = 1, 2, \dots, n$ y $j = 1, 2, \dots, m$.

Una matriz no negativa se dice ser **estocástica** si la suma de cada renglón es exactamente 1. Esto nos lleva a que las matrices de transición son estocásticas.

Una matriz estocástica P es *irreducible* si

$$\forall i, j \in S : \exists k \in N \text{ que cumple } p_{ij}(k) > 0.$$

Una matriz estocástica P es *primitiva* si

$$\exists k \in N : \forall i, j \in S \text{ se cumple } p_{ij}^{(k)} > 0.$$

A las cadenas de Markov con matriz de transición irreducible (positiva, etc.) se les llama cadenas de Markov irreducibles (positivas, etc.).

En [45], Günter Rudolph muestra cómo modelar el algoritmo genético simple (en mono-objetivo) a través de cadenas de Markov para después aplicar teoremas de convergencia ya establecidos en la teoría de procesos estocásticos. Primeramente, muestra que dado que los estados en el algoritmo genético simple dependen sólo de los genes de los individuos pertenecientes a la población, el espacio de estados se define como $A := \{0, 1\}^{n \cdot l}$ donde, n denota el tamaño de la población, y l la longitud del cromosoma de cada individuo. Luego, la matriz de transición de probabilidades P que representa al algoritmo genético simple puede representarse como el producto de tres matrices estocásticas $P = C \cdot M \cdot S$ que representan a los operadores de *cruza*, *mutación* y *selección* respectivamente.

El operador de cruza puede ser representado como una función aleatoria cuyo dominio y rango es A ; esto es, que cada estado de A se transforma de manera probabilística en otro estado de A . Normalmente se aplica con una probabilidad $p_c \in [0, 1]$ por lo que C es una matriz estocástica.

Dado que el operador de mutación se aplica de manera independiente a cada “gen” (bit) en la población, la probabilidad de que el estado i se convierta en el estado j después de la mutación está dado por

$$P\{i \mapsto j\} := p_m^{H_{ij}}(1 - p_m)^{N - H_{ij}} > 0 \text{ para todos } i, j \in A$$

donde H_{ij} es la distancia de *Hamming* entre la representación binaria del estado i y el estado j . De lo anterior se ve que la matriz M es una matriz positiva.

Cuando utilizamos selección proporcional, la población de la siguiente generación será determinada por n experimentos aleatorios independientes. La probabilidad, en cada experimento, de que el individuo b_i sea seleccionado de la tupla (b_1, b_2, \dots, b_n) para pertenecer a la siguiente generación está dada por:

$$P\{b_i \text{ es seleccionado}\} = \frac{f(b_i)}{\sum_{j=1}^n f(b_j)} > 0.$$

En [19] David Fogel analiza las propiedades asintóticas presentando también algunos experimentos y en [1] A. Agapie analiza condiciones mínimas para la convergencia.

3.5 Definiciones de convergencia

La convergencia es una propiedad que se asocia tanto a sucesiones como a funciones y nos garantiza la existencia de puntos o conjuntos a los cuales nuestros entes en estudio se aproximan después de un número infinito de términos. Cuando se trabaja con secuencias de variables aleatorias, ya que éstas están definidas sobre un espacio de probabilidad, debe tomarse en cuenta para la definición de convergencia la propia medida de probabilidad. A continuación definiremos cuáles son los principales tipos de convergencia estocástica que utilizaremos en este trabajo.

Definición 3.5.1 Sea X una variable aleatoria y (X_n) una sucesión de variables aleatorias definida sobre un espacio de probabilidad (Ω, \mathcal{F}, P) . Se dice que (X_n) converge **con probabilidad 1 (c.p.1)** o de manera **casi segura (c.s)** a X , lo que se denota como $X_n \xrightarrow{c.s} X$, si

$$P\left(\lim_{n \rightarrow \infty} |X_n - X| = 0\right) = 1;$$

o de manera explícita

$$P\left(\left\{w : \lim_{n \rightarrow \infty} X_n(w) = X(w)\right\}\right) = 1.$$

El sentido de la convergencia con probabilidad 1 es equivalente a decir que la propiedad se cumple para casi todos los eventos, excepto un conjunto de ellos el cual tiene probabilidad cero. Este concepto es análogo al del análisis clásico para propiedades de continuidad o derivabilidad excepto en un conjunto de medida (de Lebesgue) cero.

Otros sentidos de convergencia se definen enseguida:

Definición 3.5.2 Sea X una variable aleatoria y (X_n) una sucesión de variables aleatorias definida sobre un espacio de probabilidad (Ω, \mathcal{F}, P) . Se dice que (X_n) :

a) Converge **en promedio** a X , lo cual se denota como $X_n \xrightarrow{prom} X$, si

$$\lim_{n \rightarrow \infty} E[|X_n - X|] = 0.$$

- b) Converge **completamente** a X , lo cual se denota como $X_n \xrightarrow{comp} X$, si para cualquier $\epsilon > 0$

$$\lim_{n \rightarrow \infty} \sum_{i=1}^n P\{|X_i - X| > \epsilon\} < \infty.$$

- c) Converge **en probabilidad** a X , lo cual se denota como $X_n \xrightarrow{P} X$, si para cualquier $\epsilon > 0$

$$\lim_{n \rightarrow \infty} P\{|X_n - X| < \epsilon\} = 1,$$

o equivalentemente que

$$\lim_{n \rightarrow \infty} P\{|X_n - X| \geq \epsilon\} = 0.$$

Decir que tenemos convergencia completa es más general que decir que tenemos convergencia casi segura, la cual a su vez es más general que la convergencia en probabilidad. Por otra parte, la convergencia en promedio es también más general que la convergencia en probabilidad.

3.5.1 Convergencia de un algoritmo evolutivo multi-objetivo

Para el caso de los algoritmos evolutivos la convergencia va a representar qué tanto, conforme el número de iteraciones (generaciones) crece, la sucesión de soluciones que el algoritmo arroja se aproxima al verdadero óptimo global. En el caso de problemas multi-objetivo, queremos medir qué tanto el conjunto de elementos no dominados que arroja el algoritmo se acerca en cada iteración al verdadero frente de Pareto. Para poder hacer estimaciones es necesario primero definir una métrica, que en el caso de multi-objetivo, nos represente la cercanía entre dos conjuntos.

Definición 3.5.3 Una **norma** en \mathbb{R}^l se define como una aplicación $\|\cdot\| : \mathbb{R}^l \rightarrow \mathbb{R}_+$ que cumple con las tres propiedades siguientes:

$$\|x\| = 0 \text{ si y sólo si } x = 0,$$

$$\|\alpha x\| = |\alpha| \|x\|,$$

$$\|x + y\| \leq \|x\| + \|y\|.$$

con $x, y \in \mathbb{R}^l$ y $\alpha \in \mathbb{R}$.

Definición 3.5.4 Una *métrica* d en X se define como una función

$$d : X \times X \rightarrow \mathbb{R}$$

que cumple con las cuatro propiedades siguientes:

$$d(x, y) \geq 0 \text{ para todos } x, y \in X,$$

$$d(x, y) = 0 \text{ si y sólo si } x = y \text{ para todos } x, y \in X,$$

$$d(x, y) = d(y, x) \text{ para todos } x, y \in X,$$

$$d(x, z) \leq d(x, y) + d(y, z) \text{ para todos } x, y \in X.$$

Claramente la aplicación $\psi : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}_+$ definida como

$$\psi(x, y) = \|x - y\|$$

es una métrica en \mathbb{R}^l .

Si A y B son subconjuntos de un conjunto X , entonces puede probarse que

$$d(A, B) = |A \cup B| - |A \cap B|$$

es una métrica del conjunto potencia de X .

La medida $\delta_B(A) = |A| - |A \cap B|$ cuenta el número de elementos que hay en A pero no en B .

Definición 3.5.5 (ver definición 2.4.4)

- (a) Se dice que el algoritmo evolutivo **converge** con probabilidad 1 al conjunto de elementos minimales si

$$\delta_{\mathcal{F}^*}(\mathcal{F}_t) \rightarrow 0 \text{ con probabilidad 1 cuando } t \rightarrow \infty.$$

- (b) Sea A_t la población de algún algoritmo evolutivo en la iteración $t \geq 0$ y $\mathcal{F}_t := f(A_t)$ su imagen asociada. Se dice que el algoritmo evolutivo **converge** con probabilidad 1 al conjunto **completo** de elementos minimales si

$$d(\mathcal{F}_t, \mathcal{F}^*) \rightarrow 0 \text{ con probabilidad 1 cuando } t \rightarrow \infty$$

- (c) Se dice que la población A_t de un algoritmo evolutivo converge en la iteración t_0 si

$$A_{t_0} \subseteq \mathcal{F}^*.$$

3.6 Sumario

En este capítulo se presentaron los conceptos y resultados básicos de teoría de la probabilidad y procesos estocásticos necesarios para explicar los detalles de las técnicas matemáticas utilizadas en los capítulos 5 y 6. Se establecieron también las definiciones de convergencia adoptadas en este trabajo.

En el siguiente capítulo entraremos en materia definiendo las características y operadores de la estrategia evolutiva multi-objetivo.

Capítulo 4

Estrategias Evolutivas Multi-objetivo

A continuación expondremos los aspectos característicos de las *Estrategias Evolutivas* (EE), retomando la sección 2.2.1, y esclareciendo el lenguaje que las describe tanto en su forma original como en sus variantes multimiembro. Dentro de este capítulo analizaremos la adaptación de las Estrategias Evolutivas para el caso multiobjetivo; se tiene como segundo objetivo describir diversos aspectos referentes a sus operadores con el objeto de modelarlos estocásticamente.

A lo largo de este capítulo, al referirnos a optimización multi-objetivo, estaremos hablando del caso de minimización simultánea de todas las funciones objetivo de un MOP.

4.1 La estrategia (μ^+, λ) multi-objetivo

Tomemos una definición más general del problema multi-objetivo, esto es, reemplazando \mathbb{R} (en la definición 1.2.1) del contra-domino por cualquier conjunto K que posea un orden total (ver definición 1.2.5), por ejemplo $K = \mathbb{N}$, $K = \mathbb{Q}$ ó $K = \mathbb{Z}$; y reemplazando también \mathbb{R} del dominio por cualquier conjunto K_1 sin mayores restricciones que tener determinada una región factible $A \subseteq K_1^n$.

Sea entonces la función a minimizar:

$$\begin{aligned}
 f : \quad A &\longrightarrow K^m \\
 (x_1, \dots, x_n) &\mapsto (f_1(x_1), \dots, f_m(x_n)).
 \end{aligned}$$

Extendamos también, a K^n , la definición de orden (parcial) dada en 2.4.1 al igual que el concepto de no dominación de Pareto dado, para los reales, en 2.4.2. Decimos entonces, que para $x, y \in A$ con $x \neq y$, x tiene *mejor aptitud* que y o que x es *más apto* que y si $f(x)$ domina a $f(y)$, es decir

$$f(x) < f(y).$$

La solución a este problema de minimización multi-objetivo consiste en hallar un conjunto

$$X \subseteq A$$

tal que todos los elementos en $f(X)$ sean no dominados dentro de K^m . Equivalentemente, el problema se reduce a encontrar la anticadena correspondiente al conjunto de elementos minimales de K^m con respecto a la relación \leq , dado que ésta coincide con el frente de Pareto del problema.

Definición 4.1.1 Sea $f : A \longrightarrow (K^m, \preceq)$ una aplicación, entonces, para algún subconjunto $B \in A$ definimos el conjunto

$$\mathcal{M}_f(B, \preceq) := \{b \in B : f(b) \in \mathcal{M}(f(B), \preceq)\}.$$

En el caso de la EE tenemos que $\mathcal{M}_f(X, \preceq)$ coincide con el conjunto de óptimos de Pareto.

Un individuo en la EE mono-objetivo consiste de una tupla $\vec{x} = (x_1, \dots, x_n)$ y un conjunto $S \subseteq \mathbb{R}^m$ de parámetros “evolucionables”, llamados *endógenos* en la literatura de EE mono-objetivo[8]. Cada individuo tiene asociado el valor de la función $f(x)$ que determinará la jerarquía de su aptitud al compararse con otros individuos. El conjunto de parámetros S es utilizado para el proceso de **auto-adaptación** (ver sección 6.3) característico de las EE’s y la Programación Evolutiva.

En este estudio, utilizaremos como parámetro endógeno solamente a la desviación estándar en la mutación, es decir, $S = \sigma$. En la EE mono-objetivo el parámetro

σ es fijo, aunque para la EE de dos miembros se actualiza el valor de σ cada determinado número de generaciones (regularmente es a cada 20 generaciones) de acuerdo con la regla de auto-adaptación de Schwefel (ver sección 4.4).

Para la EE multi-objetivo, proponemos una regla de auto-daptación en la sección 6.3 que actualiza los parametros endógenos a cada generación. Para aplicar la auto-adaptación, propuesta en el capítulo 6.3, se debe considerar σ como vector en \mathbb{R}_+^n donde n es el número de variables en el problema multi-objetivo. Recordemos que los parámetros endógenos forman parte del cromosoma de cada individuo por lo que son operados en la recombinación, pero dado que son usados para la auto-adaptación a cada generación, no se afectan por la mutación ni intervienen dentro del cálculo de la aptitud de supervivencia del individuo.

Así, el **cromosoma** de un individuo $a_{\vec{x}}$ para la EE multi-objetivo está completamente definido por la pareja

$$(\vec{x}, \vec{\sigma})$$

y dados los individuos

$$a_{\vec{x}} = (\vec{x}, \vec{\sigma}) \quad \text{y} \quad a_{\vec{y}} = (\vec{y}, \vec{\sigma})$$

con

$$a_{\vec{x}}, a_{\vec{y}} \in \mathcal{A} = A \times \mathbb{R}_+^n,$$

se dice que $a_{\vec{x}}$ es *más apto* que $a_{\vec{y}}$ si sus cadenas de genes $x, y \in A$ cumplen que x es más apto que y . Al conjunto $\mathcal{A} = A \times \mathbb{R}_+^n$ lo llamaremos *espacio de estados de los individuos* de la población.

Una *población* en EE's está constituida por dos conjuntos de individuos, el conjunto de los padres con μ elementos y el conjunto de los descendientes (o hijos) con λ elementos. Se dice en este caso que la población es de tamaño $(\mu^+ \lambda)$.

Los parámetros μ y λ son fijos¹ en la EE original, puesto que no cambian durante el ciclo evolutivo; sin embargo en algunas variantes de la EE mono-objetivo, presentadas en [44] y [27], pueden también “evolucionarse”. En el Capítulo 6 explicaremos las ventajas y desventajas de mantener fijos estos parámetros como consecuencia del proceso de selección y la auto-adaptación propuesta para la EE multi-objetivo.

Las agrupaciones de padres e hijos en la generación t , con μ y λ individuos respectivamente, se simbolizan por:

¹Refiriéndose a esto, Beyer les llama *exógenos* en [8]

$$P_\mu^{(t)} := \{p_i^{(t)}\} = \{p_1^{(t)}, \dots, p_\mu^{(t)}\} := \{a_{x_1}, \dots, a_{x_\mu}\}$$

la población de padres en la iteración t de la EE, y

$$H_\lambda^{(t)} := \{h_i^{(t)}\} = \{h_1^{(t)}, \dots, h_\lambda^{(t)}\} := \{a_{y_1}, \dots, a_{y_\lambda}\}$$

la población de hijos (descendientes) en la iteración t de la EE.

4.2 Operadores

El paradigma evolutivo propuesto por Charles Darwin se encuentra sustentado en el principio de variación y selección, el cual, se representa en estrategias evolutivas a través de los operadores de mutación y selección. A continuación presentaremos una breve caracterización de los operadores evolutivos en *EEsM*. En la siguiente sección analizaremos diversos aspectos de cómo estos operadores contribuyen a la convergencia hacia el frente de Pareto de la EE en problemas multi-objetivo.

4.2.1 Mutación

Se puede decir que el operador de mutación es la base de la EE, debido a que produce las variaciones genéticas en los individuos de manera directa. El proceso de mutación utiliza parámetros que varían con el tiempo llevando al operador por un proceso de auto-adaptación.

Durante el proceso, la información genética del individuo, es decir su cadena cromosómica, muta usando una distribución normal. Esto representa la isotropía del espacio de búsqueda, lo que quiere decir que en principio no se tiene preferencia alguna para moverse a un determinado lugar del espacio de búsqueda, y esto se modela de manera natural con una distribución normal de Gauss.

Dado un padre p_i se produce mediante la mutación un hijo h_i por medio de la adición de un vector aleatorio Gaussiano (ruido estocástico aditivo) z . Es decir,

$$h_i = p_i + z \quad \text{con} \quad z := (z_1, \dots, z_n)^T$$

donde cada componente de z es estadísticamente independiente de los otros componentes

$$z := (N(0, \sigma_1^2), \dots, N(0, \sigma_n^2))^T.$$

Claramente la función de densidad del vector aleatorio z está dada por:

$$p(z) = \prod_{i=1}^n p(z_i) = \frac{1}{(\sqrt{2\pi})^n} \frac{1}{\prod_{i=1}^n \sigma_i} \cdot e^{\left(-\frac{1}{2} \sum_{i=1}^n \left(\frac{z_i}{\sigma_i}\right)^2\right)}.$$

Por analogía con otras estrategias de búsqueda determinística, llamaremos *longitud de paso* correspondiente a la variable i a cada uno de los σ_i . Notemos, por último, que en el caso de la EE multi-objetivo el conjunto de parámetros auto-adaptables S estará compuesto únicamente por $\vec{\sigma}$.

4.2.2 Selección

El operador de selección, dentro del paradigma de la estrategia evolutiva, tiene como objetivo determinar a cada generación t cuáles individuos sobrevivirán y formarán parte del conjunto de padres en la generación $t+1$. En el caso de estrategias evolutivas mono-objetivo el proceso de selección es muy simple, ya que se elige de manera inmediata el conjunto de los μ individuos con los mejores valores de aptitud. Cuando se trata con un problema mono-objetivo, el espacio fenotípico es un conjunto totalmente ordenado, por lo que el proceso de elegir los μ mejores valores es determinista.

La diferencia entre selección $(\mu + \lambda)$ y selección (μ, λ) radica en el conjunto de individuos sobre el cual opera la selección. En el primer caso se puede elegir de entre la unión $P_\mu^{(t)} \cup H_\lambda^{(t)}$ de padres e hijos, y en el segundo caso sólo se elige de entre la población $H_\lambda^{(t)}$ de hijos dando a los padres cero probabilidad de reproducirse por segunda vez.

En el caso de problemas multi-objetivo el proceso de selección difiere del propuesto originalmente para la estrategia evolutiva, ya que el espacio fenotípico no cuenta con un orden total y por tanto el proceso no puede llevarse a cabo de manera directa. En problemas multi-objetivo, la función de aptitud no acomoda a los individuos como una cadena, sino que solamente puede comparar si éstos se dominan o no entre sí.

A continuación describiremos tres posibles procedimientos para implementar el operador de selección en estrategias evolutivas, para problemas multi-objetivo, en ambos casos $(\mu + \lambda)$ y (μ, λ) .

Selección elitista de Peschel & Riedel

Este esquema de selección se desprende del algoritmo propuesto por Peschel & Riedel [41] en 1977, en el cual la población de padres se conforma, a cada genera-

ción, con todos y cada uno de los elementos minimales del conjunto formado por la población de padres en la generación anterior y los descendientes de éstos. Así, la nueva generación de padres $P_\mu^{(t+1)}$ será:

$$P_\mu^{(t+1)} = \mathcal{M}_{\mathcal{F}}(P_\mu^{(t)} \cup H_\lambda^{(t)}, \preceq),$$

donde $\mathcal{M}_{\mathcal{F}}(\cdot, \cdot)$ está definido en 4.1.1.

Claramente al utilizar este esquema de selección se modifica la estrategia evolutiva, de manera que el número de padres en cada generación varía conforme al número de elementos minimales de la población, resultante de la unión de padres e hijos, en esa generación. Lo anterior es una dificultad obvia en términos de la implementación del algoritmo, pues en teoría el conjunto de elementos minimales puede ser tan grande como la representación del espacio de búsqueda.

Dominación por torneo binario

A continuación proponemos un método para llevar el proceso de selección de los μ padres de la *EE* en *MOP's*. Está basado en la técnica de selección mediante torneo propuesta por Wetzel en [62] para AG's en problemas de optimización combinatoria mono-objetivo. La referiremos como *técnica de dominación por torneo binario* y consiste en lo siguiente:

1. Determinar la población C de candidatos de acuerdo al tipo de selección que se utilizará:

$$\begin{aligned} (\mu, \lambda) &\longrightarrow C = \{c_1, \dots, c_\lambda\} = H_\lambda^{(t)} \\ (\mu + \lambda) &\longrightarrow C = \{c_1, \dots, c_{\mu+\lambda}\} = P_\mu^{(t)} \cup H_\lambda^{(t)} \end{aligned} \quad \text{Cuidar que } |C| \text{ resulte un número par.}$$

2. Crear un nuevo conjunto C' formado por μ parejas de individuos de C seleccionados aleatoriamente.
3. Comparar a los individuos de cada pareja con respecto a la relación de dominación de Pareto. Si los individuos son comparables el individuo que domina formará parte de la nueva generación de padres $P_\mu^{(t+1)}$. Si no son comparables, se elige aleatoriamente entre los dos a uno para formar parte de $P_\mu^{(t+1)}$. Después de esto se habrán seleccionado los μ nuevos padres de la generación $t + 1$.

La técnica propuesta por Wetzel no puede aplicarse aquí en su versión original porque supone que se deben seleccionar tantos nuevos padres como candidatos a ser

seleccionados hay previamente, mientras que aquí queremos seleccionar a μ padres de entre $|C|$ candidatos. Originalmente esta técnica tiene dos enfoques, uno determinista y el otro probabilístico. El enfoque determinista no puede trasladarse a este caso debido a que no necesariamente se compararán todos los individuos de C , además de que aunque así ocurriese, en el paso 3 no necesariamente los individuos de la pareja son comparables uno con el otro y debe introducirse una probabilidad para elegir uno o el otro; la única parte que puede verse como “determinista” en el proceso se da cuando los individuos de una pareja pueden compararse ya que siempre se rechaza el que es dominado.

La mayoría de los estudios teóricos se han concentrado en las estrategias evolutivas del tipo $(1 + 1)$ -EE en [43] y [7], $(1 + \lambda)$ -EE y $(1, \lambda)$ -EE en [55], [7]. Esto es debido a las complicaciones matemáticas, las cuales presentaremos más adelante, que surgen en la determinación de las distribuciones poblacionales para hacerlas consistentes.

4.2.3 Recombinación

En la estrategia evolutiva original se considera únicamente el uso de los operadores de mutación y selección. Al introducir el operador de recombinación para una EE se añade un parámetro ρ al conjunto S de parámetros fijos; el parámetro ρ representa el número de padres que intervienen en el proceso de recombinación de genes para producir un mismo descendiente. Las EE con este parámetro se denotan por $(\mu/\rho \ddagger \lambda) - EE$. El símbolo \ddagger está relacionado con el operador de selección usado.

Adicionando el operador de recombinación al principio de variación y selección se completan los operadores genéticos del ciclo evolutivo (figura 2.1 de la página 22). Aún no se tiene una justificación contundente del porque el operador de recombinación es benéfico en el proceso de búsqueda. Existen trabajos que muestran que la recombinación no incrementa sustancialmente el desarrollo de la técnica [18], está también la *hipótesis de los bloques constructores* en AG [24], y para EE se tiene la hipótesis del *reparo genético*, de la cual se hablará más adelante y se darán sus fundamentos para multi-objetivo.

Para las EE con valores en los reales (\mathbb{R}) o en un subconjunto de ellos ($\mathbb{Z}, \mathbb{N}, \mathbb{Q}$) existen dos tipos de recombinación, la recombinación *intermedia* y la recombinación *dominante/discreta*.

Para describir cómo opera la recombinación, supongamos que el operador trabajará sobre una ρ -tupla de vectores (p_1, \dots, p_ρ) .

Para el caso en que $\rho > 2$ decimos que se trabaja con una *multirecombinación*.

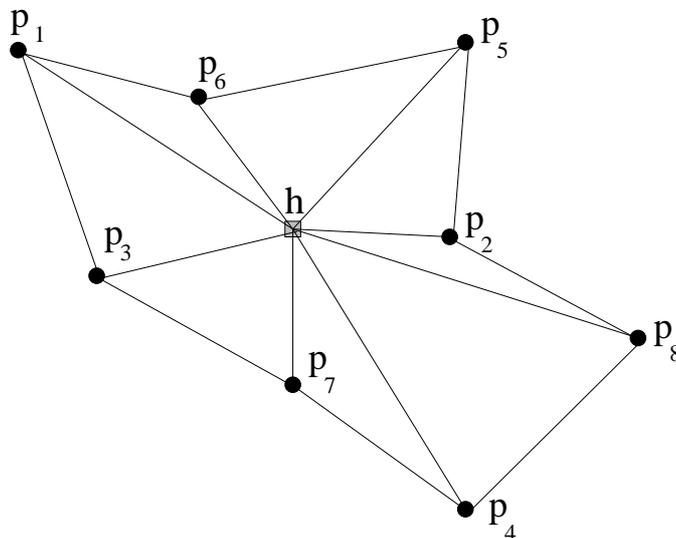


Figura 4.1: En la recombinación intermedia el cromosoma del descendiente es el centro de gravedad de las coordenadas del conjunto de cromosomas de sus padres.

- *Recombinación Intermedia.*- El descendiente resultante de la recombinación es generado vectorialmente como el centro de gravedad de los ρ individuos padre previamente seleccionados de manera aleatoria

$$h := \langle p \rangle_{\rho} := \frac{1}{\rho} \sum_{i=1}^{\rho} p_i.$$

- ρ -*Recombinación Dominante.*- Cada una de las componentes i -ésimas del descendiente de la recombinación h es obtenida seleccionando de manera aleatoria de entre los ρ individuos padre

$$h := \sum_{i=1}^n (e_i^T p_{m_i}) e_i \quad \text{con} \quad m_i := \text{random}\{1, \dots, \rho\}.$$

El símbolo e_i representa el vector unitario en la i -ésima dirección. El caso más común de la recombinación dominante es cuando $\rho = 2$, el cual es llamado *cruza uniforme* y utilizado en AG's.

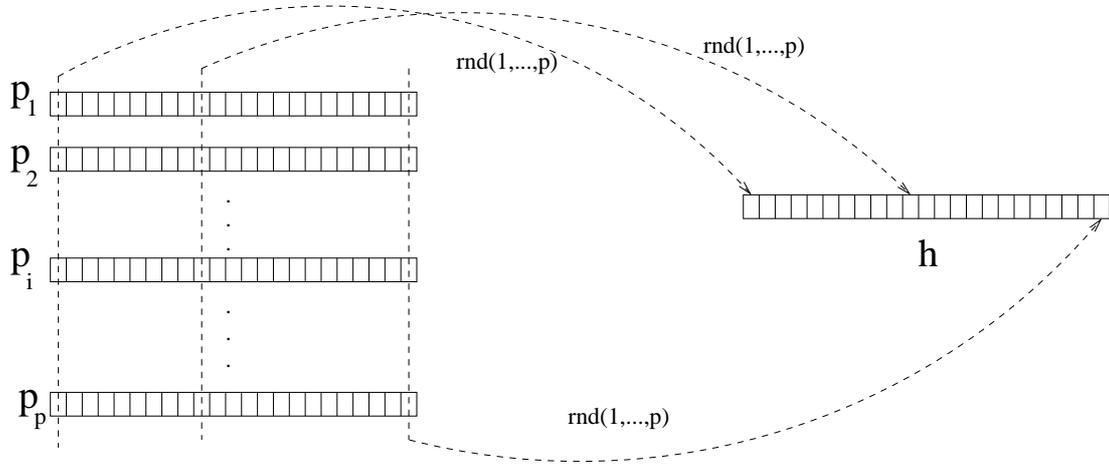


Figura 4.2: Cadenas cromosómicas del descendiente y de sus padres en la recombinación dominante.

4.3 Kernel de variación

El concepto de Kernel positivo de variación nos ayuda a estudiar el proceso evolutivo y principalmente a garantizar la creación de ciertos elementos específicos, como se verá en la sección 5.1, por medio de los operadores evolutivos.

Definición 4.3.1 Sea ρ , con $1 \leq \rho \leq \mu$, el número de padres que intervienen para producir un descendiente específico $(y, \sigma) \in A \times \mathbb{R}^m$, donde A es el espacio de búsqueda y $\{\sigma_1, \dots, \sigma_\rho\} \subseteq \mathbb{R}$ el conjunto de los parámetros para la autoadaptación. Una función de probabilidad de transición

$$P_y : (A \times \mathbb{R})^\rho \times (A \times \mathbb{R}) \longrightarrow [0, 1]$$

con la propiedad de que

$$P_y((x_1, \sigma_1), (x_2, \sigma_2), \dots, (x_\rho, \sigma_\rho); (y, \sigma)) \geq \delta > 0$$

se conocerá como kernel de variación **positivo**.

Para demostrar algunos de los teoremas en el capítulo siguiente será de gran utilidad la siguiente propiedad:

Lema 4.3.2 Sea T la variable aleatoria que representa el número de generaciones necesarias para producir, con un kernel de variación positiva, un descendiente específico y a partir de una colección arbitraria de padres $\{x_1, x_2, \dots, x_\rho\}$ con $y, x_1, \dots, x_\rho \in A \times \mathbb{R}$. Entonces:

$$P[\{T < \infty\}] = 1$$

Demostración: Retomemos el ejemplo 3.1.11 suponiendo ahora un espacio muestra infinito $\Omega = \{1, \dots, \infty\}$. Interpretamos $P[\{t\} | \{1, \dots, t-1\}^c]$ como la probabilidad de que T tome el valor de t dado que no ha tomado ninguno de los valores entre $\{1, \dots, t-1\}$. Dado que el kernel de variación es positivo se tiene que:

$$P[\{t\} | \{1, \dots, t-1\}^c] = P[\{t\} | \{t, \dots, \infty\}] \geq \delta > 0. \quad (4.1)$$

Primero, probaremos por inducción matemática que

$$P[\{T > t\}] \leq (1 - \delta)^t. \quad (4.2)$$

Caso base:

$$P[\{1\}] = P[\{1\} | \Omega] \geq \delta$$

por la hipótesis 4.1; entonces

$$P[\{T > 1\}] = P[T \in \{2, 3, \dots\}] \quad (4.3)$$

$$= P[\{1\}^c] \quad (4.4)$$

$$\leq 1 - \delta. \quad (4.5)$$

Paso inductivo:

$$\begin{aligned} P[\{1, \dots, t-1\}^c] &= P[\{1, \dots, t\}^c] - P[\{t+1\}] \\ &= P[\{1, \dots, t\}^c] - P[\{t+1\} \cap \{\{1, \dots, t\} \cup \{1, \dots, t\}^c\}] \\ &= P[\{1, \dots, t\}^c] - P[\emptyset \cup \{\{t+1\} \cap \{1, \dots, t\}^c\}] \\ &= P[\{1, \dots, t\}^c] - P[\{t+1\} \cap \{1, \dots, t\}^c] \\ &= P[\{1, \dots, t\}^c] - P[\{t+1\} | \{1, \dots, t\}^c] P[\{1, \dots, t\}^c] \\ &= P[\{1, \dots, t\}^c] [1 - P[\{t+1\} | \{1, \dots, t\}^c]]. \end{aligned}$$

Por hipótesis de inducción se tiene que

$$P[\{1, \dots, t\}^c] \leq (1 - \delta)^t$$

y por la hipótesis 4.1 tenemos

$$P[\{t+1\} | \{1, \dots, t\}^c] \leq 1 - \delta,$$

así que de lo anterior se sigue

$$P[\{1, \dots, t-1\}^c] \leq (1 - \delta)^{t+1} \quad (4.6)$$

concluyendo entonces la prueba por inducción de 4.2.

Como una consecuencia inmediata se observa que

$$\begin{aligned} P[\{T < \infty\}] &= 1 - \lim_{t \rightarrow \infty} P[\{T > t\}] \\ &\geq 1 - \lim_{t \rightarrow \infty} (1 - \delta)^t \\ &= 1. \end{aligned}$$

y en vista de la definición de medida de probabilidad concluimos que

$$P[\{T < \infty\}] = 1. \quad \blacksquare$$

Corolario 4.3.3 Sea T la variable aleatoria que representa el número de generaciones necesarias para producir, con un kernel de variación positiva, un descendiente específico y a partir de un padre x con $x, y \in A \times \mathbb{R}$. Entonces:

$$P[\{T < \infty\}] = 1$$

Demostración: Inmediata del lema anterior. ■

4.4 Auto-adaptación de EEs mono-objetivo.

Teóricamente se le llama auto-adaptación al hecho de operar los parámetros endógenos durante el proceso evolutivo, es decir, que éstos como parte que son del cromosoma intervienen dentro del proceso de recombinación, y mutación de la Estrategia Evolutiva.

Sin embargo, durante el proceso de “evolucionar” estos parámetros puede ajustarse la longitud de paso (desviación estándar de la mutación) para así garantizar la convergencia del algoritmo.

En [42], Ingo Rechenberg establece la regla de éxito conocida como regla $\frac{1}{5}$ para la (1 + 1)-EE mono-objetivo y demuestra, para dos problemas en particular, la convergencia de la técnica hacia el óptimo global.

Los problemas que Rechenberg estudia son conocidos como modelo de *esfera* (f_1) y modelo de *corredor* (f_2) dados por las funciones:

$$f_1(\vec{x}) = c_0 + c_1 \cdot \sum_{i=1}^n (x_i - x_i^*)^2 = c_0 + c_1 \cdot r^2,$$

donde $\vec{x}^* = (x_1^*, \dots, x_n^*)$ es el mínimo global y r es la distancia euclídeana entre \vec{x} y \vec{x}^* ,

$$f_2(\vec{x}) = c_0 + c_1 \cdot x_1 \text{ con } i \in \{2, \dots, n\} : \frac{-b}{2} \leq x_i \leq \frac{b}{2}.$$

Rechenberg define el *factor de convergencia* de la técnica como

$$\varphi = \int p(k) \cdot k dk,$$

donde $p(k)$ representa la probabilidad para una mutación de cubrir cierta distancia k con respecto al óptimo (para un análisis detallado de esto pueden consultarse [56, 54]). Los límites de integración dependen de la función objetivo en particular.

Optimizando φ para ambos modelos, la esfera y el corredor, se deriva la siguiente regla: Para un cierto número de generaciones dentro de la búsqueda del óptimo se calcula la frecuencia de éxitos, es decir, el cociente del número de mutaciones exitosas entre el número total de mutaciones dentro del periodo fijo de generaciones; si este cociente es mayor que $\frac{1}{5}$, aumentar la varianza, en caso contrario disminuirla.

Esto se puede ver dentro del algoritmo como que a cada n generaciones se modifica σ de acuerdo a:

$$\sigma(t) = \begin{cases} \sigma(t-n) \cdot \frac{1}{c}, & \text{si } p_s > \frac{1}{5} \\ \sigma(t-n) \cdot c, & \text{si } p_s < \frac{1}{5} \\ \sigma(t-n), & \text{si } p_s = \frac{1}{5} \end{cases}$$

donde p_s es la frecuencia relativa de mutaciones exitosas y $0.817 \leq c \leq 1$. Este valor para c fue derivado de manera teórica, posteriormente, por Schwefel [54]. La regla $\frac{1}{5}$ aplica para estrategias evolutivas mono-objetivo sin recombinación ni mutación del parámetro σ .

Para ambos modelos, esfera y corredor, Schwefel derivó una aproximación de esta teoría para el caso de las estrategias $(\mu+\lambda)$ y (μ, λ) ([54] pág. 142-145). La idea

básica que Schwefel utiliza es estudiar el promedio de los μ mejores individuos que forman la siguiente generación, donde cada descendiente cubre una cierta distancia k ; siendo $p_\nu(k)$ la probabilidad de que los ν -ésimos mejores descendientes cubran la distancia k , la probabilidad correspondiente al promedio de los μ mejores es

$$p(k) = \frac{1}{\mu} \sum_{\nu=1}^{\mu} p_\nu(k).$$

4.5 Sumario

En este capítulo se definieron los operadores y características específicos de la estrategia evolutiva como paradigma de la computación evolutiva multi-objetivo. Se estableció el concepto de “kernel de variación”, el cual nos ayudará a continuación a probar la convergencia del algoritmo, al igual que las ideas del algoritmo de autoadaptación para mono-objetivo, tema que se extenderá a multi-objetivo en el Capítulo 6.

En el capítulo siguiente se propondrán algoritmos para las estrategias evolutivas multiobjetivo $(1+1)$, (μ, λ) y $(\mu+\lambda)$ estudiando el comportamiento límite de dichos algoritmos.

Capítulo 5

Convergencia

Supongamos un problema que consta de una función multi-objetivo a optimizar (maximizar o minimizar) f definida sobre un conjunto S (no necesariamente finito), el cual representa el espacio de búsqueda, también llamado espacio solución.

Si logramos modelar los elementos de S de manera que podamos representar una buena parte de ellos en una computadora aproximando con esto al conjunto S , entonces puede utilizarse una técnica evolutiva para hallar valores cercanos al óptimo de Pareto y obtener así una buena solución al problema Multi-objetivo original.

A partir de esto surgen dos interrogantes:

¿Cómo podemos garantizar que las técnicas evolutivas nos aproximarán a la solución del problema?

¿Qué tan rápido nos acercan las técnicas evolutivas al óptimo de Pareto del problema en cuestión?

Para comenzar a responder estas preguntas analizaremos en este capítulo convergencia asintótica de la EE multi-objetivo, posteriormente hablaremos de la velocidad de convergencia en el Capítulo 6.

5.1 Aspectos globales para la convergencia de EEsM

Para demostrar la convergencia de la EE, primeramente debemos definir una manera de medir la distancia entre dos conjuntos: la imagen de nuestro conjunto

solución arrojado por la técnica en cuestión, y el verdadero frente de Pareto \mathcal{F}^* teórico del problema. Comenzamos entonces esta sección con la definición de dicha métrica.

Definición 5.1.1 *Cualquier conjunto no vacío \mathcal{F} puede ser dotado con la métrica discreta definida para $a, b \in \mathcal{F}$ como*

$$d(a, b) := \begin{cases} 0 & \text{si } a = b \\ 1 & \text{si } a \neq b. \end{cases}$$

Usando la métrica discreta, la distancia

$$d(a, A) := \min \{d(a, b) : b \in A\}$$

de un punto $a \in \mathcal{F}$ cualquiera a un subconjunto $A \subseteq \mathcal{F}$ está bien definida, ya que es cero si $a \in A$ y uno si $a \notin A$.

El primer aspecto que debe atacarse en el estudio de la convergencia de una EE es, en el sentido global, si el algoritmo tiene la capacidad de llegar al estado solución óptimo; es decir, como es un algoritmo probabilístico debemos poder garantizar que se tendrá al menos una solución aceptable al final de su ejecución.

En los teoremas posteriores 5.1.5 y 5.1.7, se muestra que para un tiempo arbitrariamente grande (cuando $t \rightarrow \infty$) las estrategias evolutivas $(1 + 1)$ y $(\mu + \lambda)$ garantizan al menos una solución en el conjunto óptimo de Pareto.

Esto es:

$$\lim_{t \rightarrow \infty} P [f(x_t) \in \mathcal{M}(f(S), \preceq)] = 1$$

5.1.1 Convergencia de la $(1 + 1)$ -EE multiobjetivo

A continuación analizaremos cómo en teoría una estrategia evolutiva $(1+1)$ alcanza el óptimo de Pareto en alguna generación y sin importar cuántas generaciones más se ejecute, la solución no se perderá.

Suponemos que el espacio de búsqueda $S \neq \emptyset$ es finito; lo anterior se justifica puesto que la estrategia evolutiva se implementa en una computadora; sin importar cuál sea su precisión con los números reales, una solución puede tomar sólo un número finito de valores.

Supongamos también una aplicación $f : S \rightarrow f(S)$ resultando que $(f(S), \preceq)$ es un Poset. Dado que la imagen de un conjunto finito es un conjunto finito, podemos garantizar, que $\mathcal{M}(f(S), \preceq)$ es completo.

Algoritmo 5.1.2 El algoritmo en general de la estrategia evolutiva (1+1) se puede describir de la siguiente forma:

- I) Generar de manera aleatoria un individuo $x_0 \in S$, inicializar el contador de generaciones $t = 0$.
- II) Aplicar algún operador de variación (mutación) para perturbar el valor de x_t y obtener así un descendiente $y_t \in S$.
- III) Evaluar, si $f(y_t) \preceq f(x_t)$ entonces $x_{t+1} = y_t$, de lo contrario $x_{t+1} = x_t$ (mecanismo de selección).
- IV) Incrementar t y a menos que se cumpla alguna condición de terminación volver al paso (II).

Con este algoritmo generamos una sucesión $(x_t : t \geq 0)$. Hablando en términos de la convergencia del algoritmo, el objetivo es mostrar que a partir de un cierto valor t_0 la sucesión de las imágenes $(f(x_t) : t \geq t_0)$ pertenece a $\mathcal{M}(f(S), \preceq)$. Para esto, supondremos que el operador de variación referido en el paso (II) tiene asociada una función de probabilidad de transición P_v para pasar de un estado (un padre) $x \in S$ del espacio de búsqueda a otro estado $y \in S$ (un descendiente); supongamos también que P_v cumple con la propiedad:

$$P_v(x, y) \geq \delta > 0 \text{ para todos } x, y \in S.$$

A lo largo de este trabajo, a la función de probabilidad de transición le llamamos también (ver definición 4.3.1) *kernel de variación*.

Dado que el kernel de variación está acotado por cero de manera estricta, la probabilidad de generar un descendiente específico $y \in S$ a través de una variación relativa a un padre arbitrario $x \in S$, tal que

$$f(y) \in \mathcal{M}(f(S), \preceq),$$

es estrictamente positiva; es decir, al menos $\delta > 0$. Lo anterior se interpreta de manera práctica como que todo elemento minimal de $f(S)$ puede ser producido a partir de cualquier estado de S .

Suponiendo que ha ocurrido lo anterior, es decir, se dio una transición de $x \in S$ a $y \in S$ con $y \in \mathcal{M}(f(S), \preceq)$, esto no garantiza que el paso (III) del algoritmo acepte a y como el individuo de la nueva población. Lo anterior es debido a que en

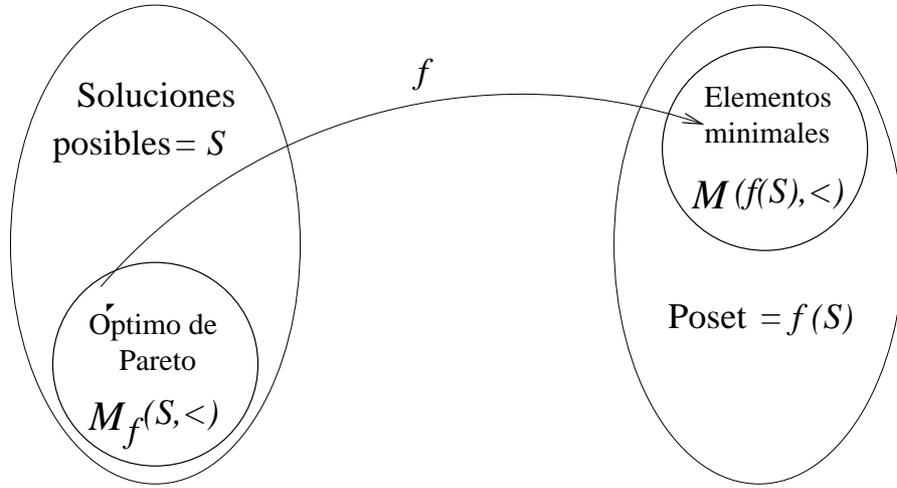


Figura 5.1: El frente de Pareto coincide con el conjunto de los elementos minimales de un conjunto parcialmente ordenado, por definición. En el caso de dominación de Pareto el Poset coincide con \mathbb{R}^n bajo la relación usual ($<$).

un orden parcial no necesariamente cada pareja de elementos debe ser comparable. Esto nos lleva a que el elemento y puede ser rechazado a pesar de ser un elemento minimal. Para superar esta inconveniencia notemos el siguiente lema:

Lema 5.1.3 *Dadas las hipótesis tales que $S \neq \emptyset$ es un conjunto finito y*

$$f : S \longrightarrow f(S)$$

una aplicación para la cual $(f(S), \preceq)$ resulta ser un Poset, (ver página 17 para notación) es cierto que:

- a) $\mathcal{T}(f(x)) := \mathcal{G}(f(x)) \cap \mathcal{M}(f(S), \preceq) \neq \emptyset$ para cualquier $x \in S$.
- b) $\mathcal{I}(f(x)) := \{y \in S : f(y) \in \mathcal{T}(f(x))\} \neq \emptyset$ para cualquier $x \in S$.

Demostración: Dado que $\mathcal{M}(f(S), \preceq)$ es completo, el lema 1.2.9(b) nos asegura que para cada $x \in S$ se tiene que $\mathcal{T}(f(x)) \neq \emptyset$. A consecuencia de esto el conjunto $\mathcal{I}(f(x))$, el cual es imagen inversa de $\mathcal{T}(f(x))$, debe también ser no vacío. ■

Teorema 5.1.4 (teorema 1 de EEsM) *Dado S un espacio de búsqueda finito, la población de la estrategia evolutiva multiobjetivo (1 + 1) sobre S , descrita en el algoritmo 5.1.2 converge (ver def 3.5.5) en un número finito de generaciones.*

Demostración: En vista del lema anterior 5.1.3(a) podemos asegurar la existencia de elementos minimales contenidos también en el conjunto $\mathcal{G}(f(x))$ para cada $x \in S$, y el conjunto $\mathcal{T}(f(x))$ es de hecho donde se encuentran dichos elementos. Dado que $\mathcal{I}(f(x)) \neq \emptyset$, claramente contiene a todos los descendientes que cumplen con ser minimales y que serán aceptados por el algoritmo para reemplazar a la población actual; la probabilidad con la que el algoritmo $(1+1)$ -EE genera y a la vez acepta un descendiente del conjunto de elementos minimales en una iteración puede ser acotado por:

$$p(x, \mathcal{I}(f(x))) \geq \delta \cdot |\mathcal{I}(f(x))| \geq \delta > 0 \text{ para cada } x \in S. \quad (5.1)$$

Ahora veamos qué ocurre después de que un elemento de la sucesión

$$(f(x_t) : t \geq 0)$$

forma ya parte del conjunto $\mathcal{M}(f(S), \preceq)$ de elementos minimales:

Tomemos $t_0 \in \mathbb{N}$ el primero tal que se cumple

$$f(x_{t_0}) \in \mathcal{M}(f(S), \preceq).$$

El conjunto de elementos que podrían ser aceptados en una iteración a partir de que la población consiste del individuo x_{t_0} es $\mathcal{G}(f(x_{t_0})) = \{f(x_{t_0})\}$. Por esto y en vista del lema 1.2.9(a) podemos garantizar que

$$f(x_t) \in \mathcal{M}(f(S), \preceq) \text{ para todo } t \geq t_0.$$

Notemos que a consecuencia de lo anterior, la probabilidad de que la población “salga” del conjunto de elementos minimales durante la aplicación del algoritmo es cero. ■

Lo anterior, puede retomarse cuando se trata de generalizar este resultado a espacios métricos arbitrarios, siguiendo el trabajo en 5.1. De la ecuación 5.1 y el Lema 1 en [46]¹ se sigue que para $t \geq 1$

$$P\{f(x_t) \in \mathcal{M}(f(S), \preceq)\} \geq 1 - (1 - \delta)^t. \quad (5.2)$$

¹El Lema 1 en [46] afirma que dado el conjunto de estados óptimos A , si el kernel de variación $K(x, A)$, es decir la probabilidad de transición del estado x al conjunto A está estrictamente acotado por cero

$$K(x, A) \geq \delta > 0$$

para todo $x \in E \setminus A$ y $K(x, A) = 1$ para $x \in A$ entonces la probabilidad de transición del estado x al conjunto A en el tiempo t está estrictamente acotada por

$$K^{(t)}(x, A) \geq 1 - (1 - \delta)^t$$

para $t \geq 1$.

Teorema 5.1.5 (Rudolph [48]) sea $S \neq \emptyset$ un conjunto finito y

$$f : S \longrightarrow \mathcal{F} = \{f(x) : x \in S\}$$

una función, donde (\mathcal{F}, \preceq) es un conjunto parcialmente ordenado dotado con la métrica discreta $d(\cdot, \cdot)$. Sea $(x_t : t \geq 0)$ con $x_t \in S$ la sucesión aleatoria generada por la estrategia evolutiva (1+1) cuyo kernel de variación invariante en el tiempo tiene la propiedad de acotarse como:

$$P_v(x, y) \geq \delta > 0 \text{ para todos } x, y \in S.$$

Entonces la sucesión

$$d(f(x_t), \mathcal{M}(f(S), \preceq))$$

de distancias aleatorias entre $f(x_t)$ y el conjunto de elementos minimales

$$\mathcal{M}(f(S), \preceq)$$

converge completamente y en promedio a cero cuando $t \rightarrow \infty$.

Demostración:

Utilizando el concepto de distancia derivado de la métrica discreta (definición 5.1.1) podemos definir la variable aleatoria D_t como

$$D_t = d(f(x_t), \mathcal{M}(f(S), \preceq))$$

así la expresión 5.2 puede reescribirse en términos de D_t para $\varepsilon > 0$ de la forma siguiente:

$$P\{D_t \leq \varepsilon\} \geq 1 - (1 - \delta)^t. \quad (5.3)$$

Dado que la variable aleatoria en la ecuación 5.2 converge a uno cuando $t \rightarrow \infty$ puede mostrarse que la sucesión $(D_t : t \geq 0)$ converge en probabilidad (ver definición 3.5.2 de la página 45) a cero. Dado que la convergencia es geoméricamente mas rápida en 5.3 que en 5.2 se sigue que $(D_t : t \geq 0)$ converge completamente a cero, por lo cual se implica la convergencia con probabilidad 1. La convergencia en promedio se asegura a través de los márgenes que acotan a D_t aunado a la convergencia en probabilidad. ■

Si tomamos una población con más de un individuo por generación y lo dotamos con una *estrategia de preservación elitista*, el teorema anterior nos garantiza que la imagen de al menos un elemento de la población alcanza el conjunto de elementos minimales después del ciclo evolutivo.

Definición 5.1.6 estrategia de preservación elitista: Sea

$$P = (p_1^{(t)}, p_2^{(t)}, \dots, p_\mu^{(t)})$$

la población de padres y

$$H = (h_1^{(t)}, h_2^{(t)}, \dots, h_\lambda^{(t)})$$

la colección de descendientes en la generación $t > 0$. Sin pérdida de generalidad llamaremos padre elitista a $x_t = p_1^{(t)}$.

Si

$$B_t = \{h \in H_k : f(h) \in \mathcal{G}(f(x_t)) \setminus \{f(x_t)\}\}$$

es vacío, entonces asignamos el valor de el padre elitista en la generación $t + 1$ como $x_{t+1} = x_t$. De otro modo, elegimos un individuo $h_t^* \in B_t$ arbitrario cuya imagen sea mínima² en el Poset $(f(B_t), \preceq)$ y determinamos $x_{t+1} = h_t^*$.

Usando esta definición surge el siguiente teorema:

Teorema 5.1.7 (Rudolph [48]) Sea $S \neq \emptyset$ un conjunto finito y

$$f : S \longrightarrow \mathcal{F} = \{f(x) : x \in S\}$$

una función, donde (\mathcal{F}, \preceq) es un conjunto parcialmente ordenado dotado con la métrica discreta $d(\cdot, \cdot)$. Sea un algoritmo evolutivo con $\mu < \infty$ padres y $\lambda < \infty$ hijos en S que toma $f(\cdot)$ como función multi-objetivo a optimizar, y está provisto con una estrategia de preservación elitista. Entonces podemos garantizar, con dicho algoritmo evolutivo, la generación de al menos una secuencia x_t aleatoria de padres tales que la sucesión de distancias

$$d(f(x_t), \mathcal{M}(f(S), \preceq))$$

converge completamente y en promedio a cero cuando $t \rightarrow \infty$, bajo la suposición de que el kernel de variación invariante en el tiempo es estrictamente positivo en $S^\mu \times S^\lambda$, esto es,

$$P_v(x, y) \geq \delta > 0 \quad \text{para todos } x \in S^\mu, y \in S^\lambda.$$

²Siempre puede elegirse un elemento cuya imagen es mínima porque B es finito.

Demostración: Claramente la sucesión $\{x_t\}$ resultante de la estrategia de preservación de élite aplicada cumple con las condiciones del teorema 5.1.5, por lo que el resultado es inmediato. ■

En el caso de la $(1 + 1)$ -EE multiobjetivo, el teorema 1 de EEsM prueba la convergencia por completo ya que la población a cada generación consiste de un solo elemento, el mismo para el cual se ha probado la convergencia al conjunto de óptimos de Pareto. Para demostrar la convergencia de la población completa en las estrategias evolutivas $(\mu + \lambda)$ y (μ, λ) multiobjetivo, se requiere de un estudio especial y aplican algunas restricciones; para mostrar esto, dedicamos las secciones siguientes.

5.2 Convergencia de la $(\mu + \lambda)$ -EE multi-objetivo

En la $(\mu + \lambda)$ -EE mono-objetivo existe un elitismo de manera implícita dentro del operador de selección, esto es porque una vez que el óptimo global se encuentra dentro de la población de padres podemos asegurar, por la construcción del algoritmo, que el óptimo global se seleccionará como padre en todas las generaciones posteriores.

Si logramos trasladar esta propiedad al caso de multi-objetivo podríamos garantizar la convergencia de toda la población de individuos hacia el frente de Pareto en un número finito de generaciones.

Aplicando a la EE multi-objetivo el esquema de selección propuesto por Peschel & Riedel (sección 4.2.2), se obtiene el siguiente algoritmo:

Algoritmo 5.2.1 Denotamos por $P_\mu^{(t)}$ a la población de padres³, $H_\lambda^{(t)}$ es el conjunto de descendientes⁴ y $\mathcal{M}_f(\cdot, \cdot)$ está definido en 4.1.1.

- I) Se inicializa $P_{\mu_0}^{(0)}$ de manera aleatoria.
- II) $t \leftarrow 0$
- III) $H_\lambda^{(t)} \leftarrow \text{recombinación}(P_{\mu_t}^{(t)})$
- IV) $H_\lambda^{(t)} \leftarrow \text{mutación}(H_\lambda^{(t)})$
- V) $P_{\mu_{t+1}}^{(t+1)} = \mathcal{M}_f(P_{\mu_t}^{(t)} \cup H_\lambda^{(t)}, \preceq)$

³de tamaño μ en la iteración (generación) t .

⁴de tamaño λ en la iteración (generación) t .

VI) $t \leftarrow t + 1$

VII) Si el criterio de terminación no se satisface, ir al paso III.

A continuación haremos unas observaciones, en forma de lemas, en torno al funcionamiento del algoritmo anterior

Lema 5.2.2 *Si en la población $P_{\mu_t}^{(t)}$ ha entrado un elemento óptimo de Pareto, entonces éste pertenecerá a cada población de padres subsecuente. Lo anterior quiere decir que dado $x \in S : f(x) \in \mathcal{M}_f(S, \preceq)$, para todo $t \geq 0$ se cumple*

$$x \in P_{\mu_t}^{(t)} \implies x \in P_{\mu_{t+1}}^{(t+1)}.$$

Demostración: Se sigue inmediatamente, pues por ser x minimal, no existe elemento alguno en S que lo domine bajo el orden \preceq . En particular, para $t \geq 0$ no existe elemento alguno de $P_{\mu_t}^{(t)} \cup H_{\lambda}^{(t)} \subseteq S$ que lo domine, así que en el paso V del algoritmo se garantiza que $x \in P_{\mu_{t+1}}^{(t+1)}$. ■

Lema 5.2.3 *Si el elemento $y \in P_{\mu_t}^{(t)}$ no es un óptimo de Pareto, entonces existe un elemento $x \in S$ tal que $x \preceq y$.*

Demostración: Esto se sigue de que la representación de S es un conjunto finito, y por el lema 1.2.8 se tiene que $\mathcal{M}(f(S), \preceq)$, es completo. ■

El elitismo implícito de este algoritmo puede verse como una generalización de la estrategia de preservación elitista (Definición 5.1.6) mencionada en el teorema 5.1.7; es precisamente este elitismo implícito lo que nos ayuda a probar, en el siguiente teorema, la convergencia del algoritmo hacia el frente de Pareto.

Teorema 5.2.4 (teorema 2 de EEsM) *Si el Kernel de variación de la EE multi-objetivo descrita en el algoritmo 5.2.1 es positivo, entonces la población $P_{\mu}^{(t)}$ converge (ver def. 3.5.5) en un número finito de generaciones.*

Demostración: Supongamos que en el paso $t \geq 0$ ningún miembro de la población $P_{\mu}^{(t)}$ es un óptimo de Pareto. El lema 5.2.3 nos asegura que podemos encontrar un elemento $x \in S$ tal que domina a alguno de los padres en $P_{\mu}^{(t)}$. Como el kernel de variación es positivo, se puede aplicar el lema 4.3.2 y con él probar que dicho elemento $x \in \mathcal{M}(f(S), \preceq)$ se generará en un número finito de generaciones con probabilidad uno, de forma que para algún $t_0 < \infty$ se tendrá que $x \in H_{\lambda}^{(t_0)}$. Claramente

$$x \in H_{\lambda}^{(t_0)} \implies x \in P_{\mu_{t_0+1}}^{(t_0+1)}$$

y por el lema 5.2.2, este individuo permanecerá en la población de padres para todas las generaciones posteriores. Repitiendo este argumento para cada elemento de la población actual que no sea un óptimo de Pareto, llegaremos, después de un número finito de generaciones, a la conclusión de que todos los elementos de la población son óptimos de Pareto. ■

A pesar de que se ha demostrado la convergencia de la población, no podemos decir que este algoritmo adapta la $(\mu + \lambda)$ -EE al caso de multiobjetivo, puesto que el parámetro μ cambia de generación en generación. Mas aún, ésta es la mayor desventaja en cuanto al uso de este algoritmo, ya que si el tamaño de la población de padres varía con respecto al tiempo, ésta puede llegar a ser demasiado grande, tanto como el mismo frente de Pareto teórico \mathcal{F}^* del problema.

Debido a lo anterior presentamos, como alternativa para definir la $(\mu + \lambda)$ -EE multiobjetivo, una adaptación del algoritmo propuesto por Rudolph en [51]; Aquí se utiliza un esquema de selección muy parecido al elitismo implícito de Peschel & Riedel, pero a la vez se acota el tamaño de la población a un número fijo de μ padres como máximo.

5.2.1 Restringiendo el tamaño de la población de padres

Algoritmo 5.2.5 Denotamos por $P_\mu^{(t)}$ a la población de padres, $H_\lambda^{(t)}$ es el conjunto de descendientes, $\mathcal{M}_f(\cdot, \cdot)$ está definido en 4.1.1, $P_{\mu_t}^{(t)}$, $H_{\lambda_t}^{*(t)}$ y $\text{AUX}(t)$ son conjuntos tales que:

$$\lambda_t := |H_{\lambda_t}^{*(t)}| \leq \lambda, \quad \mu_t := |P_{\mu_t}^{(t)}| \leq \mu \quad \text{y} \quad |\text{AUX}(t)| \leq |H_{\lambda_t}^{*(t)}| \leq \lambda.$$

La función $\text{draw}(k, A)$ elige de manera estocástica k individuos del conjunto A .

- I) Se inicializa $P_\mu^{(0)}$ de manera aleatoria.
- II) $P_{\mu_0}^{(0)} \leftarrow \mathcal{M}_f(P_\mu^{(0)})$.
- III) $t \leftarrow 0$
- IV) $H_\lambda^{(t)} \leftarrow \text{recombinación}(P_{\mu_t}^{(t)})$
- V) $H_\lambda^{(t)} \leftarrow \text{mutación}(H_\lambda^{(t)})$
- VI) $H_{\lambda_t}^{*(t)} = \mathcal{M}_f(H_\lambda^{(t)}, \preceq)$
- VII) $\text{AUX}(t) \leftarrow \emptyset$

VIII) para cada $h \in H_{\lambda_t}^{*(t)}$ ejecutar:

a) $D_h = \{p \in P_{\mu_t}^{(t)} : f(h) \prec f(p)\}$

b) si $D_h \neq \emptyset$ entonces $P_{\mu_t}^{(t)} \leftarrow (P_{\mu_t}^{(t)} \setminus D_h) \cup \{h\}$

c) de lo contrario, si $f(h) \parallel f(p)$ para cualquier $p \in P_{\mu_t}^{(t)}$
entonces $AUX(t) \leftarrow AUX(t) \cup \{h\}$

IX) $k = \min\{\mu - \mu_t, |AUX(t)|\}$

X) $P_{\mu_{t+1}}^{(t+1)} = P_{\mu_t}^{(t)} \cup \text{draw}(k, AUX(t))$

XI) $t \leftarrow t + 1$

XII) Si el criterio de terminación no se satisface, ir al paso IV.

En este algoritmo el tamaño de la población de padres oscila, durante la ejecución del algoritmo, entre 1 y μ ; en la demostración del siguiente teorema queda claro que al final del algoritmo la población final constará de $\min\{\mu, \mathcal{F}^*\}$ elementos.

Teorema 5.2.6 (teorema 3 de EEsM) *Si el Kernel de variación de la EE multi-objetivo descrita en el algoritmo 5.2.5 es positivo, entonces la población $P_{\mu}^{(t)}$ converge (ver def. 3.5.5) en un número finito de generaciones.*

Demostración: Claramente en este algoritmo se sigue cumpliendo el lema 5.2.2, es decir si un óptimo de Pareto pertenece a la población $P_{\mu_t}^{(t)}$ pertenecerá también a la población de padres en cada una de las generaciones posteriores.

Si un elemento $h \in H_{\lambda_t}^{*(t)}$ domina a alguno de los padres en $P_{\mu_t}^{(t)}$ entonces h formará parte de $P_{\mu_t}^{(t)}$ y por lo tanto de $P_{\mu_{t+1}}^{(t+1)}$, mientras que los padres en $P_{\mu_t}^{(t)}$ que fueron dominados por h ya no pasarán a la siguiente generación.

Si un elemento $h \in H_{\lambda_t}^{*(t)}$ es un óptimo de Pareto entonces éste pasará ya sea a $P_{\mu_t}^{(t)}$, si domina a alguno de los padres, o a AUX si no lo hace. Lo anterior hace posible perder un elemento óptimo de Pareto mientras se conserva un elemento no óptimo en el caso en que $|P_{\mu_t}^{(t)}| = \mu$. Lo anterior no afecta a la convergencia, pues supongamos que existe un elemento $p \in P_{\mu_t}^{(t)}$ el cual no es un óptimo de Pareto, el lema 5.2.3 nos asegura que podemos encontrar un elemento $x \in \mathcal{M}(f(S), \preceq)$ que lo domina, luego, el lema 4.3.2 nos asegura que dicho elemento se generará en un número finito de iteraciones con probabilidad uno. Por último, el paso VII

del algoritmo nos asegura que pertenecerá al conjunto $H_{\lambda_t}^{*(t)}$ para algún $t > 0$, reemplazando en esta iteración al elemento p .

Repetiendo este argumento para cada elemento de la población actual que no sea un óptimo de Pareto, llegaremos, después de un número finito de generaciones, a la conclusión de que todos los elementos de la población son óptimos de Pareto.

■

5.3 Convergencia de la (μ, λ) –EE multi-objetivo

En el caso de la (μ, λ) –EE el elitismo no está implícito en el mecanismo de selección puesto que las soluciones de los padres de la generación anterior no se conservan. Si existe un óptimo de Pareto entre los padres, éste puede perderse al pasar a la siguiente generación. El elitismo en este caso se implementa de manera “explícita” utilizando lo que se conoce como **población secundaria**, PS^t en la cual se conservan las mejores soluciones (soluciones no dominadas) con respecto a la población de descendientes, la cual constituye la población de la siguiente generación.

En esta estrategia puede demostrarse de manera directa la convergencia de toda la población secundaria hacia el frente de Pareto. El siguiente algoritmo de la (μ, λ) –EE multi-objetivo servirá para ejemplificar la demostración:

Algoritmo 5.3.1 P^t es la población del algoritmo y cumple $0 < |P^t| \leq \max\{\lambda, \mu.\}$

1. Se inicializa P_μ^0 de manera aleatoria.
2. $P_\mu^0 \leftarrow \mathcal{M}_{\mathcal{F}}(P_\mu^{(0)}, \preceq)$
3. $t=0$, $PS^0 = P_\mu^0$
4. $P_\lambda^{t+1} = \text{generación}(P_\mu^{(t)})$
5. $PS^{t+1} = \mathcal{M}_{\mathcal{F}}(P_\lambda^{(t+1)} \cap PS^{(t)}, \preceq)$
6. $t \leftarrow t + 1$
7. Si el criterio de terminación no se satisface, ir al paso 4.

Teorema 5.3.2 (teorema 4 de EEsM) *Si la secuencia poblacional P_λ^t , $t \geq 0$ de la (μ, λ) -EE anterior (algoritmo 5.3.1) es una cadena de Markov finita homogénea con matriz de transición irreducible (ver sección 3.4.2) entonces el algoritmo completo converge al frente de Pareto completo de la representación del problema.*

Demostración: La convergencia para este algoritmo puede también derivarse de la convergencia del algoritmo presentado por David Van Veldhuizen en [60].

Por la construcción del algoritmo podemos garantizar que el conjunto imagen $f(PS^t)$ de PS^t es una anticadena la cual a cada generación coincide con el conjunto de elementos minimales de $(f(PS^t), \preceq)$. Tan pronto como un elemento de \mathcal{F}^* entra en PS^t podemos garantizar que no saldrá de este conjunto. Ahora sólo resta mostrar que el conjunto de elementos minimales teórico del problema está completamente contenido (con probabilidad uno) en $f(PS^t)$ en un número finito de generaciones.

Sea $y \in PS^{t_0}$ tal que $f(y) \notin \mathcal{F}^*$. Dado que (\mathcal{F}, \preceq) es completo (Lema 1.2.8) se garantiza que existe $x \in S$ tal que $f(x) \prec f(y)$, es decir, para alguna iteración $t \geq t_0$ el elemento no óptimo será reemplazado por un elemento óptimo generado en la secuencia $P_\lambda^{(t)}$ y por lo tanto en PS^t . Dado que la cadena de Markov es irreducible, el lema 3.4.3 asegura que cada elemento de S^n será visitado una infinidad de veces, lo que nos lleva a que el tiempo de espera para que se genere (con probabilidad 1) x es finito. De esta forma, los elementos no óptimos serán eliminados de PS^t , en un tiempo finito, con probabilidad uno.

Si un individuo es incomparable con los elementos de $P_\lambda^{(t)}$, éste pasará a formar parte de PS^t y si es un óptimo, formará parte de cada uno de los PS^{t+k} , $k \geq 0$ subsecuentes. Si el elemento mencionado no es un óptimo, entonces será reemplazado por un óptimo en un tiempo finito. En conclusión,

$$d(f(PS^t), \mathcal{F}^*) \longrightarrow 0$$

con probabilidad uno cuando

$$t \longrightarrow \infty.$$

■

5.3.1 Restringiendo el tamaño de la población secundaria

Una de las principales desventajas del algoritmo 5.3.1 es que el tamaño de la población PS^t puede llegar a crecer tanto como el óptimo de Pareto teórico del problema el cual está solamente acotado por el espacio de búsqueda en cuestión.

Lo anterior claramente hace que este algoritmo se pueda implementar solamente para cierto tipo de problemas. A continuación presentamos un algoritmo basado en el propuesto por G. Rudolph y A. Agapie en [52].

Algoritmo 5.3.3 Denotamos por $P^{(t)}$ a la población. PS^t la población secundaria tal que $|PS^t| \leq m$, $m \in \mathbb{N}$. $\mathcal{M}_f(\cdot, \cdot)$ esta definido en 4.1.1 y $AUX(t)$ y $P_{\lambda_t}^{*(t)}$ son conjuntos tales que:

$$\lambda_t \leq \lambda \text{ y } |AUX(t)| \leq \lambda.$$

La función $draw(k, A)$ elige de manera estocástica k individuos del conjunto A .

- I) Se inicializa $P_{\mu}^{(0)}$ de manera aleatoria.
- II) $PS^{(0)} \leftarrow \mathcal{M}_f(P_{\mu}^{(0)})$.
- III) $t \leftarrow 0$
- IV) $P_{\lambda}^{(t)} \leftarrow \text{recombinación}(P_{\mu}^{(t)})$
- V) $P_{\lambda}^{(t)} \leftarrow \text{mutación}(P_{\lambda}^{(t)})$
- VI) $P_{\lambda_t}^{*(t)} = \mathcal{M}_f(P_{\lambda}^{(t)}, \preceq)$
- VII) $AUX(t) \leftarrow \emptyset$
- VIII) para cada $h \in P_{\lambda_t}^{*(t)}$ ejecutar:
 - a) $D_h = \{p \in PS^{(t)} : f(h) \prec f(p)\}$
 - b) si $D_h \neq \emptyset$ entonces $PS^{(t)} \leftarrow (PS^{(t)} \setminus D_h) \cup \{h\}$
 - c) de lo contrario, si $f(h) \parallel f(p)$ para cualquier $p \in PS^{(t)}$ entonces $AUX(t) \leftarrow AUX(t) \cup \{h\}$
- IX) $k = \min\{m - |PS^t|, |AUX(t)|\}$
- X) $PS^{(t+1)} = PS^{(t)} \cup draw(k, AUX(t))$
- XI) $t \leftarrow t + 1$, $P_{\mu}^{(t)} = draw(\mu, P_{\lambda}^{(t-1)})$.
- XII) Si el criterio de terminación no se satisface, ir al paso IV.

En este algoritmo el tamaño de la población secundaria oscila, durante la ejecución del algoritmo, entre 1 y m . En la demostración del siguiente teorema queda claro que al final del algoritmo la población final constará de $\min\{m, \mathcal{F}^*\}$ elementos.

Este algoritmo presenta un esquema de selección más complejo pero garantiza la convergencia de la población secundaria restringiendo su cardinalidad, lo cual se prueba en el siguiente teorema:

Teorema 5.3.4 (teorema 5 de EEsM) *Si la secuencia poblacional P_λ^t , $t \geq 0$ de la (μ, λ) -EE anterior (algoritmo 5.3.3) es una cadena de Markov finita homogénea con matriz de transición irreducible (ver sección 3.4.2) entonces el algoritmo completo converge al frente de Pareto de la representación del problema. También se tiene que*

$$|f(PS^t)| \longrightarrow \min\{\mu, |F^*|\}.$$

Demostración: Por la construcción del algoritmo podemos garantizar que el conjunto $f(PS^t)$ es una anticadena y claramente

$$PS^t = \mathcal{M}_f(PS^t, \preceq) \quad \text{para todo } t \geq 0.$$

Un elemento $y \in PS^t$ es eliminado si existe un elemento $x \in P_\lambda^{t_0}$ que lo domine; entonces x entra en PS_0^t y al menos un elemento de PS_0^t será eliminado. Si el elemento x pertenece a \mathcal{F}^* entonces $x \in PS^t$, para todo $t \geq t_0$.

Notemos que los elementos de AUX son incomparables con los elementos de PS^t y éstos se utilizan para completar la cardinalidad de PS^t a m . Usando el mismo argumento de la prueba del teorema 5.3.2 podemos garantizar que si hay un elemento no óptimo en PS^t , éste será reemplazado en un tiempo finito por uno óptimo. Los elementos óptimos entrarán a PS^t directamente desde el paso VIII(b) o a través de AUX en el paso X. Entonces concluimos que:

$$\delta_{\mathcal{F}^*}(f(PS^t)) \longrightarrow 0 \quad \text{con probabilidad uno y en promedio cuando } t \longrightarrow \infty.$$

■

5.4 Sumario

A lo largo de este capítulo hemos propuesto algoritmos para las estrategias evolutivas multiobjetivo $(1+1)$, $(\mu, \lambda)y(\mu + \lambda)$; hemos estudiado el comportamiento

límite de los algoritmos propuestos basándonos en los trabajos de David Van Veldhuizen [61, 60], G. Rudolph [46, 49, 48, 52, 50, 47] y A. Agapie [52]. Presentamos cinco algoritmos y las pruebas correspondientes de su convergencia hacia el frente de Pareto.

Para la estrategia $(1 + 1)$ -EE multiobjetivo (algoritmo 5.1.2) se prueba, en el teorema 5.1.4, que la población (compuesta por un elemento) pertenece al conjunto de óptimos de Pareto en un número finito de iteraciones del algoritmo, y se estaciona ahí.

Para la estrategia $(\mu + \lambda)$ -EE multiobjetivo (algoritmo 5.2.1) se prueba, en el teorema 5.2.4, que la población pertenece, y de hecho iguala, al conjunto de óptimos de Pareto en un número finito de iteraciones del algoritmo.

Para la estrategia (μ, λ) -EE multiobjetivo (algoritmo 5.3.1) se mencionan varios aspectos acerca de su funcionamiento y el elitismo explícito que se implementa a través de la población secundaria. Se prueba, en el teorema 5.3.2, que la población secundaria pertenece, y de hecho iguala, al conjunto de óptimos de Pareto en un número finito de iteraciones del algoritmo.

Para ambas estrategias $(\mu + \lambda)$ -EE multiobjetivo y (μ, λ) -EE multiobjetivo se dan algoritmos alternativos, 5.2.5 y 5.3.3, para restringir el tamaño de la población para la $(\mu + \lambda)$ -EE multiobjetivo y de la población secundaria para la (μ, λ) -EE multiobjetivo. También se dan las pruebas de dichos algoritmos alternativos.

Capítulo 6

Aspectos locales en la convergencia de EEsM

En el capítulo anterior se han dado las pruebas de convergencia global para un AE multi-objetivo del tipo $(1 + 1)$ -EE multi-objetivo, (μ, λ) -EE multi-objetivo y $(\mu + \lambda)$ -EE multi-objetivo con presencia implícita o explícita de elitismo. En dichas pruebas, por su carácter general, no se toma en cuenta mucha de la influencia que tienen los operadores de selección, mutación y recombinación específicos de cada AE dentro de la convergencia en multi-objetivo.

A continuación haremos un análisis de las características de estos operadores MOP's encaminado a desarrollar una teoría específica de convergencia en EE. Estimar cuáles mecanismos intervienen dentro de las EE's para acercarnos al frente de Pareto no sólo nos garantiza la convergencia, sino que nos da ideas acerca del desempeño y las capacidades del algoritmo.

Otra forma de estudiar la convergencia se presenta al buscar una razón de avance para la convergencia, es decir, tratar de acotar el número de generaciones suficientes para que el algoritmo tenga un buen desempeño. Cabe mencionar que esto no parece ser una tarea sencilla puesto que habrá que entender la dinámica de la EE y hacer un cuidadoso análisis de las características estocásticas de sus estados y sus parámetros.

Una buena forma de comenzar a escribir una teoría de convergencia para EE'sM es llevando al ámbito multi-objetivo el análisis hecho previamente para la convergencia de EE hacia un óptimo global. Así pues, comenzaremos analizando algunos conceptos característicos para los cuales se puede escribir una teoría en multi-objetivo.

A lo largo de este capítulo vamos a analizar la velocidad de convergencia de la

Estrategia Evolutiva multi-objetivo a través de un problema particular.

6.1 Problema de prueba

Por tener un frente de pareto convexo, usaremos el problema tres que Valenzuela-Rendón y Uresti-Charre proponen en [59]. Este problema se describe en el ejemplo a continuación por dos funciones objetivo sobre dos variables.

Ejemplo 6.1.1 *Minimizar:*

$$f_1(x, y) = x + y + 1$$

$$f_2(x, y) = x^2 + 2y - 1$$

con $x \in [-3, 3], y \in [-3, 3]$.

De manera analítica se encuentran los Óptimos y el Frente de Pareto, los cuales se muestran en la Figura 6.1 respectivamente.

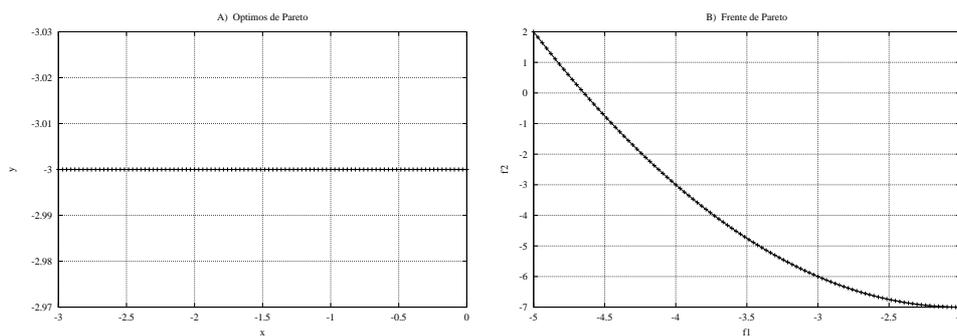


Figura 6.1: Conjunto de óptimos de Pareto y frente de Pareto para el problema estudiado.

6.2 Poblaciones de tamaño variable

Como se vio en las demostraciones del capítulo anterior, usar jerarquización de Pareto en el proceso de selección de individuos que sobreviven de una generación a otra garantiza que: Si se ha hallado un óptimo de Pareto en una determinada generación, éste sobrevivirá por siempre en todas las generaciones posteriores en forma de individuo dentro de la población. Sin embargo, una de las principales desventajas del uso de la jerarquización de Pareto es que la complejidad computacional en el proceso de selección puede llegar a $O(n^2)$. Pero esto no es lo único que debe tomarse en cuenta, pues al considerar a la jerarquización de Pareto como el único proceso de selección, el tamaño de la población variará a cada generación acarreado con ello algunas complicaciones:

En el caso del Ejemplo 6.1.1 el número de individuos en la anti-cadena de la población inicial es dramáticamente pequeño con respecto al de la población misma (ver Figura 6.2). Recordemos que estos pocos individuos serán los únicos que pasarán como padres a la siguiente generación; esto contribuye a que el algoritmo llegue en este caso a estancamientos parciales en un determinado momento y afecta como consecuencia a su velocidad de convergencia. Este problema se superó inmediatamente al mejorar el mecanismo de búsqueda incorporando al algoritmo el mecanismo de auto-adaptación propuesto en la sección 6.3.1. En la sección 6.4.2 expondremos cómo el tamaño de la población podría ayudar para la convergencia.

En la siguiente sección analizaremos un aspecto que contribuye a la distribución de los puntos en el espacio fenotípico observada en la Figura 6.3 donde se ilustran los pocos individuos que quedan en la población después de la jerarquización de Pareto.

La segunda complicación ocurre cuando se salva la primera: Sabemos que teóricamente el frente de Pareto puede llegar a ser tan grande como la cardinalidad de la imagen del espacio de búsqueda; como la representación de \mathbb{R}^n en una computadora. Particularmente en el Ejemplo 6.1.1 al incorporar mecanismos de mejora en la búsqueda, como la adaptación (sección 6.3), el algoritmo con una población inicial de 100 padres sólo pudo ejecutarse cuatro o cinco iteraciones, dependiendo de la semilla inicial, puesto que el frente de Pareto crecía a cada generación y se llegaba a un desbordamiento de memoria; esto se explicará más a detalle en la sección 6.3.2.

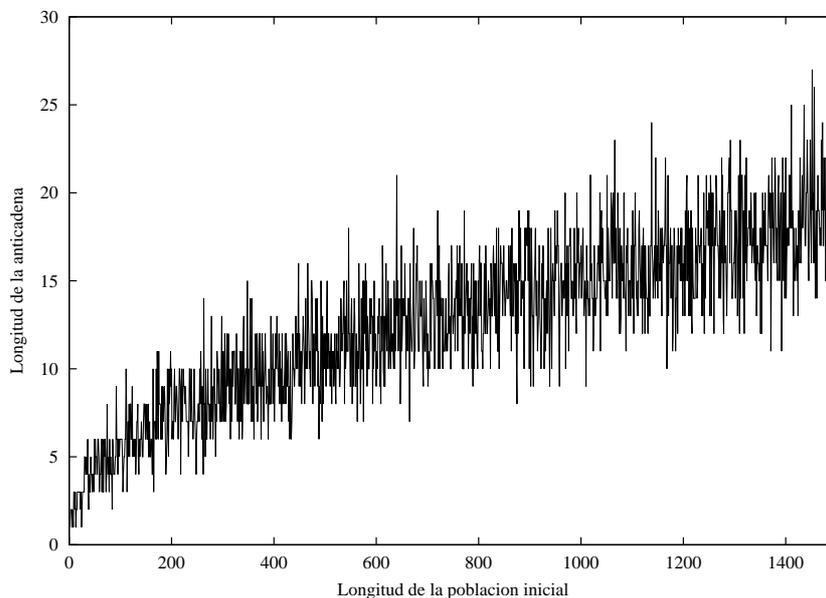


Figura 6.2: Tamaño μ de la población inicial para la EE multi-objetivo en el problema del Ejemplo 6.1.1 contra el de su respectiva anti-cadena.

6.2.1 Correlación entre las funciones objetivo

Otra característica interesante en el espacio de las funciones objetivo es la correlación que puede existir entre ellas. El coeficiente de correlación es una medida de dependencia estadística lineal entre dos variables X y Y y está definido por:

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma(X)\sigma(Y)}$$

donde

$$\text{Cov}(X, Y) = E((X - E(X))(Y - E(Y))).$$

Una de las propiedades de $\rho(X, Y)$ es que se encuentra en el intervalo $[-1, 1]$. Un valor cerca de uno indica que a medida que el valor de una variable aumenta, el valor de la otra también tiende a aumentar. Si el valor está cerca de -1 a medida que el valor de una variable aumenta, el valor de la otra variable disminuye. Una correlación de cero indica que no hay dependencia lineal estadística entre las dos variables, aunque no indica que necesariamente las variables sean independientes. Una correlación distinta de cero tampoco es evidencia suficiente para concluir que

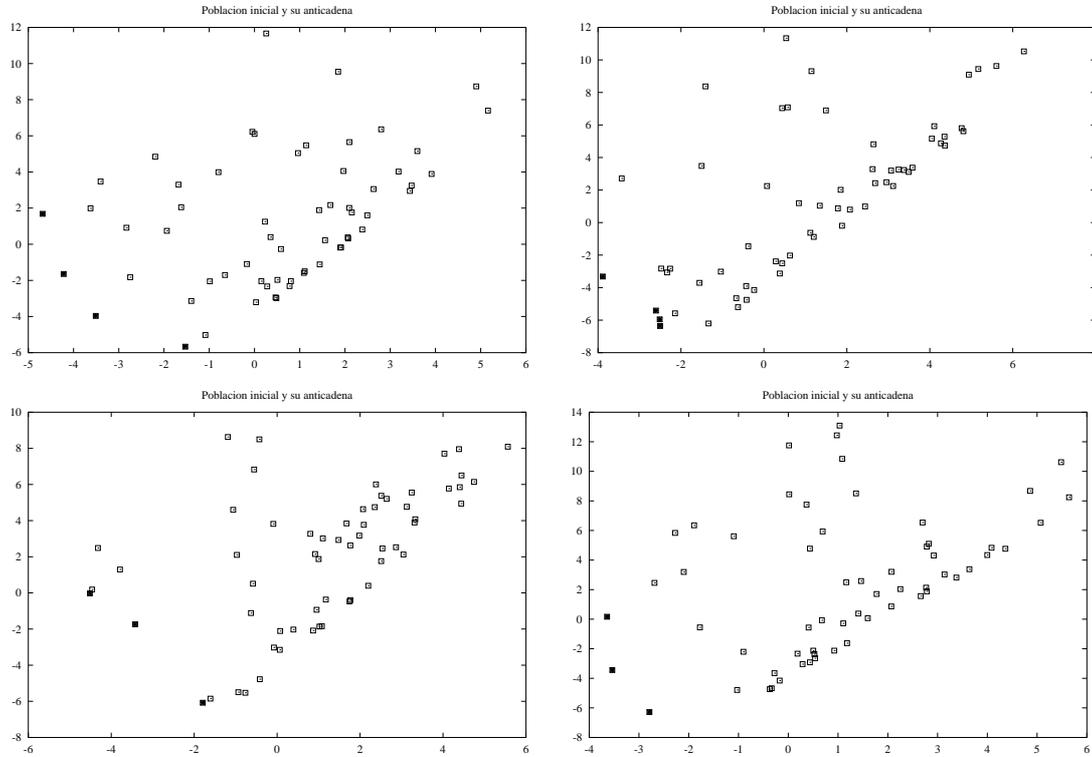


Figura 6.3: Poblaciones iniciales de 60 padres y sus respectivas anti-cadenas (representadas por ■) para diferentes semillas aleatorias iniciales.

hay una relación de causa y efecto entre las variables.

Cuando existe una correlación entre las funciones objetivo del problema ésta puede intervenir sobre el proceso de generación aleatorio de individuos afectando la forma en que éstos se distribuyen a través del espacio fenotípico. Un ejemplo de esto se da en el caso particular que estudiamos, bajo una distribución de probabilidad uniforme para $x, y \in [-3, 3]$ independientes entre si, las funciones $f_1(x, y) = x + y + 1$ y $f_2(x, y) = x^2 + 2y - 1$ están correlacionadas por:

$$\rho(f_1, f_2) = \frac{\text{Cov}(x + y + 1, x^2 + 2y - 1)}{\sigma(x + y + 1)\sigma(x^2 + 2y - 1)} = \frac{\sqrt{5}}{4} \approx 0.56$$

puesto que:

$$E(x + y + 1) = 1,$$

$$E(x^2 + 2y - 1) = 2,$$

$$\sigma^2(x + y + 1) = 6,$$

$$\sigma^2(x^2 + 2y - 1) = \frac{96}{5}$$

y

$$\begin{aligned} \text{Cov}(x + y + 1, x^2 + 2y - 1) &= E[(x + y)(x^2 + 2y - 3)] \\ &= E[x^3 + 2xy - 3x + yx^2 + 2y^2 - 3y] \\ &= 2E[y^2] \\ &= 6. \end{aligned}$$

Para este problema particular es sencillo observar que

$$\begin{aligned} f_2 &= x^2 + 2y - 1 \\ &= 2(x + y + 1) + x^2 - 2x - 3 \\ &\geq 2(x + y + 1) - 4 \\ &= 2f_1 - 4, \end{aligned}$$

por lo cual, al generar la población inicial del algoritmo a pesar de ser de manera aleatoria, todos los fenotipos de los individuos en la población permanecerán en el semi-plano generado por la desigualdad anterior. La Figura 6.4 ilustra lo anterior.

6.3 Auto-adaptación

La región en la cual se tiene alcance a través de la mutación, comúnmente llamada “ventana evolutiva” depende directamente de la topología del espacio fenotípico, un aspecto que no es del todo conocido de manera “a priori”. Por ejemplo, en [49], Rudolph prueba que un algoritmo sencillo converge si el alcance de la mutación está relacionado de manera proporcional a la distancia con respecto a \mathcal{F}^* .

En [35], Frank Kursawe propone una estrategia evolutiva que utiliza la estructura de diploides, es decir el uso de genes recesivos y genes dominantes, para manejar la información cromosómica de los individuos. Kursawe no utiliza dominancia de Pareto para la selección, sino un vector de probabilidades para cada

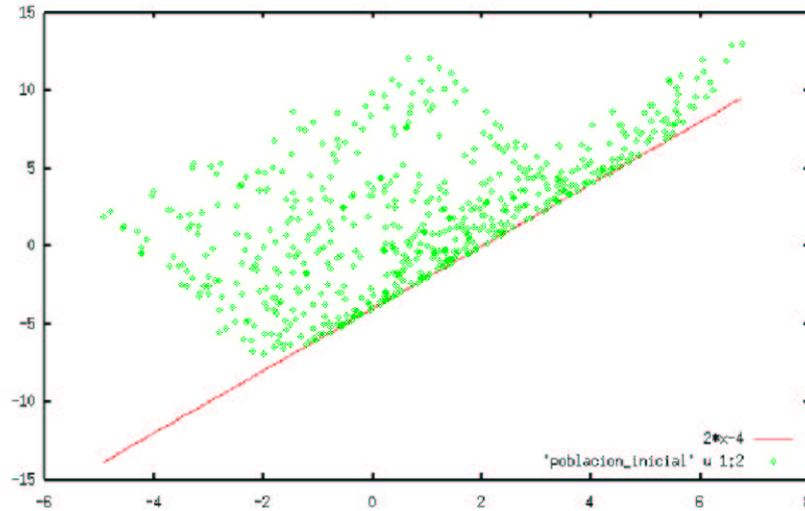


Figura 6.4: Población inicial para la EE multi-objetivo aplicada al problema 6.1.1.

objetivo el cual va cambiando en el tiempo con respecto al entorno poblacional y a través de las estructuras diploides se propone guardar los cambios en este vector de una generación a otra con el fin de perder la menor cantidad posible de buenas soluciones.

Otro tipo de auto-adaptación se presenta en el modelo “cazador-presa” [37], el cual puede verse como un tipo especial de $(\mu + 1)$ -EE; en dicho modelo el tamaño de paso para la mutación es decrementado por un factor constante (conservativo) cada vez que es producido un descendiente. Podemos decir que en este caso el alcance de la mutación no se ajusta con respecto al entorno además de que no se tiene la posibilidad de incrementar el tamaño de paso, sólo de disminuirlo, esto aunado a que deben introducirse nuevos parámetros al algoritmo correspondientes al factor de disminución.

6.3.1 Algoritmo propuesto

El conjunto de parámetros endógenos del estado de un individuo no intervienen en la determinación de su aptitud. Sin embargo, son transmitidos a sus descendientes y evolucionados como parte de su cadena genética a cada generación que el individuo sobrevive.

Al igual que en mono-objetivo, esta auto-adaptación para la EE multi-objetivo busca modificar la longitud de paso para la mutación, es decir, restringir o expandir el espacio hacia donde se exploran los nuevos individuos. El alcance de la mutación se determina de acuerdo con el entorno poblacional y se ajusta a cada generación; este ajuste se hace imitando el razonamiento de la (1 + 1)EE mono-objetivo de que el máximo desempeño se alcanza mediante longitudes de paso guiadas por una probabilidad de éxito de aproximadamente 20% .

Supongamos un problema multi-objetivo que consiste en minimizar (en el sentido de Pareto) la función

$$f = (f_1, \dots, f_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m.$$

Para implementar la auto-adaptación que proponemos en la EE multi-objetivo, debemos modificar ligeramente el algoritmo tradicional: Para cada iteración del ciclo evolutivo, cada pareja de padres (a, b) debe tener al menos un hijo $h^{(a,b)}$. Durante el proceso que llamaremos “cálculo de la auto-adaptación”, cada $h^{(a,b)}$ producirá un total de k_i , $i = 1, \dots, n$ individuos mutados en la variable i para cada una de las variables x_1, \dots, x_n , es decir, aplicando a su cromosoma únicamente el ruido aditivo correspondiente a σ_i ; denotaremos a estos k_i individuos mutados en la variable i por $h_{i,k}^{(a,b)}$ $k = 1, \dots, k_i$. Posteriormente calculamos:

$$s_{i,j} = \frac{\sum_{k=1}^{k_i} \left(f_j(h_{i,k}^{(a,b)}) - f_j(h^{(a,b)}) \right)}{\sum_{k=1}^{k_i} \left| f_j(h_{i,k}^{(a,b)}) - f_j(h^{(a,b)}) \right|}$$

para cada variable x_i y cada función objetivo f_j . Con los valores anteriores podremos establecer, para cada pareja de padres y cada una de las variables x_1, \dots, x_n del genotipo, el factor:

$$\alpha_{x_i}^{(a,b)} = e^{-\sum_{j=1}^m s_{i,j} + \beta}$$

mediante el cual “adaptaremos”, a cada generación, los parámetros endógenos de cada uno de los individuos (padres y/o hijos) que intervienen en el proceso de selección como candidatos a formar parte de la siguiente generación. El parámetro $\beta \in (0, 1)$ es dependiente del problema. Así, la rutina de auto-adaptación propuesta es la siguiente:

Algoritmo 6.3.1 Mutación con auto-adaptación de la EE Multi-objetivo.

Denotamos por P_μ a la población de padres¹, $h^{(a,b)}$ denota un descendiente producto de la recombinación de a y b con $a, b \in P_\mu$, $H^{(a,b)}$ serán las clases² que forman a la población de descendientes $H_\lambda = \cup H^{(a,b)}$ para $a, b \in P_\mu$ en la generación t .

1. Tomar una copia de cada clase de la población de hijos,
 $M^{(a,b)} = H^{(a,b)}$.
2. Para cada x_i en el espacio genotípico:
 - (a) Para cada $h^{(a,b)} \in M_\lambda = \cup M^{(a,b)}$:
Mutar sólo el gen correspondiente a x_i , es decir,
 $h_i^{(a,b)} = \text{mutación en } x_i \text{ de } h^{(a,b)}$
 - (b) Para cada f_j en el espacio fenotípico:
calcular $s_{i,j} = \frac{\sum_{k=1}^{k_i} (f_j(h_{i,k}^{(a,b)}) - f_j(h^{(a,b)}))}{\sum_{k=1}^{k_i} |f_j(h_{i,k}^{(a,b)}) - f_j(h^{(a,b)})|}$
 - (c) Para cada $a, b \in P_\mu$ calcular $\alpha_{x_i}^{(a,b)} = e^{-\sum_{j=1}^m s_{i,j} + \beta}$.
3. $H_\lambda \leftarrow \text{mutación}(H_\lambda)$.
4. Para cada $ind^{(a,b)} \in P_\mu \cup H_\lambda : ind^{(a,b)} = (x_1, \dots, x_n, \sigma_1, \dots, \sigma_n)$:
 - (a) Actualizar los genes σ_i $i=1, \dots, n$ de su cadena cromosómica con el factor de auto-adaptación
 $\sigma_i \leftarrow \alpha_{x_i}^{(a,b)} * \sigma_i$
 - (b) Actualizar las referencias a sus padres dejándolos como descendientes de sí mismos, esto es, para cada $ind \in P_\mu \cup H_\lambda$ se tiene que $ind = ind^{(ind, ind)}$.

Cuando se genera la primera población de padres se toma como convención que éstos descienden de la recombinación con sí mismos, es decir, para cada $p \in P_\mu$ se tiene que $p = p^{(p,p)}$.

El proceso $\text{mutación}(H_\lambda)$ es el habitual, aplicando al cromosoma de cada individuo $ind^{(a,b)} = (x_1, \dots, x_n, \sigma_1, \dots, \sigma_n) \in H_\lambda$ el ruido estocástico correspondiente a sus parámetros endógenos: $x_i = x_i + N(0, \sigma_i)$, para todo $i = 1, \dots, n$.

¹De tamaño μ en la iteración (generación) t .

²Las $H_{(a,b)}$ forman una partición de H_λ como clases de equivalencia bajo la relación $h^{(a_1, b_1)} \sim h^{(a_2, b_2)} \Leftrightarrow a_1 = a_2 \ \& \ b_1 = b_2$ para todos los padres $a, b \in P_\mu$

6.3.2 Restringiendo a la población de padres

Con respecto al proceso de auto-adaptación propuesto se deben hacer algunos comentarios:

Notemos que el tamaño λ de la población de descendientes variará en función del tamaño de la población de padres μ el cual, a su vez, está determinado por la longitud de la mayor anti-cadena (conjunto de elementos minimales) de la unión de las poblaciones de padres e hijos, en el caso de selección $(\mu + \lambda)$, o únicamente la población de hijos en el caso de selección (μ, λ) .

Sin importar el tipo de selección $(\mu + \lambda)$ ó (μ, λ) dado que cada pareja de padres debe tener al menos un hijo, el valor de λ será proporcional a $\frac{\mu(\mu+1)}{2}$ que es el número de parejas de padres que pueden formarse a cada generación; para cada pareja de padres se puede crear un número fijo k de hijos. Esto, como veremos más adelante nos ayuda a asegurar la convergencia. Así, en el peor de los casos (que a su vez sería el mejor en términos de la búsqueda) el tamaño de la anti-cadena en la selección estará determinado a cada iteración t por

$$\left| \mathcal{M}(f(P_{\mu_t}^{(t)} + H_{\lambda_t}^{(t)}), \leq) \right| = k \frac{\mu_t(\mu_t + 1)}{2}.$$

Dado entonces que

$$\lambda_t \propto \mu_t^2$$

y

$$\mu_{t+1} \propto \lambda_t$$

el tamaño de la población de padres crecerá de manera exponencial

$$\mu_t \propto \mu_0^{2^t}$$

razón por la cual, al implementar del programa se produce en tiempo de ejecución un desbordamiento de memoria.

Para poder ejecutar en el programa una cantidad considerable de iteraciones del ciclo evolutivo, debemos modificarlo restringiendo el crecimiento de la población de padres. La manera más sencilla es tomar un valor fijo para μ al inicio de cada generación.

En las demostraciones del Capítulo 5 se considera que la selección es de Pareto, por lo que si un elemento óptimo de Pareto llega a la población de padres, éste sobrevivirá en todas las iteraciones; pero resulta claro, de las aproximaciones anteriores, que si permitimos variabilidad en el tamaño de la población es imposible guardar todos los valores de la anti-cadena a cada generación. Modificaremos

entonces el algoritmo para usar un número fijo de padres en vista de las demostraciones 5.2.6 y 5.3.4, que garantizan la convergencia al límite en este caso y de que los valores experimentales, para este caso particular, confirman que se pueden obtener buenos resultados.

6.4 Velocidad de Convergencia

Si queremos medir el acercamiento, en cada iteración t , del conjunto de óptimos arrojado por nuestro algoritmo al conjunto de óptimos teóricos del problema debemos definir una distancia (ver definición de métrica 3.5.4) entre estos conjuntos. La forma en que estudiaremos la velocidad de convergencia de la EE multi-objetivo, para el caso del Ejemplo 6.1.1, será midiendo dicha distancia del conjunto de imágenes de los óptimos que arroja el algoritmo a cada generación con respecto al frente de Pareto teórico del problema.

Dentro de la sección 3.5.1 se propusieron, para fines de demostrar convergencia global, un par de métricas para determinar la distancia entre dos conjuntos; ahora tratamos de “trazar” el acercamiento a cada iteración del algoritmo, es por eso que en esta sección usaremos una tercera métrica.

Definición 6.4.1 *Llamaremos **norma supremo** a la aplicación*

$$\|\cdot\|_{\infty} : \mathbb{R}^p \rightarrow \mathbb{R}_+$$

definida por:

$$\|x\|_{\infty} \mapsto \sup\{|x_1|, |x_2|, \dots, |x_p|\} \text{ para todo } x = (x_1, x_2, \dots, x_p) \in \mathbb{R}^p.$$

Es sencillo comprobar que la norma supremo cumple con la definición 3.5.3, desprendiendo de ella la métrica correspondiente:

$$d_{\infty}(x, y) = \|x - y\|_{\infty} = \sup\{|x_i - y_i|\} \text{ para } i = 1, \dots, p$$

y para todos $x, y \in \mathbb{R}^p$.

La métrica d_{∞} no pierde generalidad con respecto a la métrica que se deriva de la **norma euclidiana**

$$\|x\|_2 \mapsto \sqrt{x_1^2 + x_2^2 + \dots + x_p^2} \text{ para todo } x = (x_1, x_2, \dots, x_p) \in \mathbb{R}^p,$$

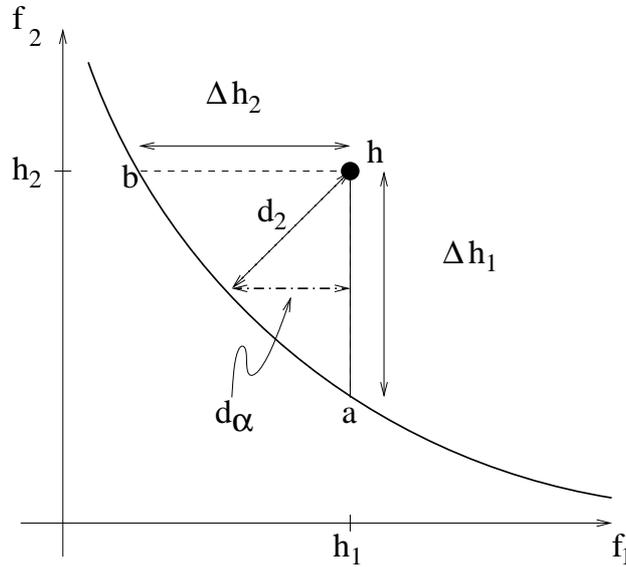


Figura 6.5: Individuo en el espacio fenotípico, su distancia euclídeana d_2 al frente de Pareto y su distancia d_∞ .

puesto que el Teorema 8.10 de [5] nos da una equivalencia³ entre ambas.

Usar una en vez de la otra se hizo para simplificar la implementación y ahorrar cálculos en tiempo de ejecución.

Dado que d_∞ se puede extender a la distancia entre un punto x y un conjunto B como

$$d_\infty(x, B) = \inf\{d_\infty(x, b) : b \in B\}$$

tomaremos entonces para fines de demostrar el acercamiento, de todos los elementos de \mathcal{F}^t a \mathcal{F}^* , la distancia entre ellos como:

$$d_\infty(\mathcal{F}^t, \mathcal{F}^*) = \sup\{d_\infty(a, \mathcal{F}^*) : a \in \mathcal{F}^t\}.$$

El siguiente lema nos será útil para el estudio de la probabilidad de convergencia de la sección 6.4.3:

Lema 6.4.2 *Sea \mathcal{F}^* el frente de Pareto de un problema multi-objetivo cualquiera y h un elemento en el espacio fenotípico con Δh_1 y Δh_2 como en la Figura 6.5,*

³El Teorema 8.10 de [5] afirma que si $x = (x_1, x_2, \dots, x_p) \in \mathbb{R}^p$ entonces

$$|x_i| \leq \|x\|_2 \leq \sqrt{p} \|x\|_\infty.$$

entonces

$$d_\infty(h, \mathcal{F}^*) \leq \min\{\Delta h_1, \Delta h_2\}.$$

Demostración: Utilizando la nomenclatura de la Figura 6.5, notemos que

$$d_\infty(h, \mathcal{F}^*) \leq d_\infty(h, c) \text{ para todo } c \in \mathcal{F}^*,$$

particularmente

$$d_\infty(h, a) = \Delta h_1 \text{ con } a = (h_1, f_2|_{(f_1=h_1)}), \text{ y}$$

$$d_\infty(h, b) = \Delta h_2 \text{ con } b = (f_1|_{(f_2=h_2)}, h_2);$$

por lo que

$$d_\infty(h, \mathcal{F}^*) \leq \min\{\Delta h_1, \Delta h_2\}.$$

■

6.4.1 Distancia genotípica vs. distancia fenotípica

En [49] Rudolph analiza la convergencia de una Estrategia Evolutiva (1 + 1) para un problema particular con dos variables y dos funciones objetivo. En ese artículo Rudolph prueba, para dicho problema, la convergencia asintótica en el espacio genotípico, es decir de \mathcal{F}_t^{-1} hacia \mathcal{F}^{*-1} (Ver definición 2.4.4) cuando $t \rightarrow \infty$. Las pruebas en general de la convergencia asintótica que presentamos en el Capítulo 5 están hechas en el espacio fenotípico, es decir que se prueba el acercamiento con probabilidad 1 de \mathcal{F}_t hacia \mathcal{F}^* cuando $t \rightarrow \infty$.

La razón por la que seguiremos trabajando en el espacio fenotípico para estudiar la velocidad de convergencia es simplemente porque la comparación entre los individuos para determinar supervivencia a través del tiempo se da en este espacio. Al usar el espacio genotípico se encontraron casos como el siguiente:

Ejemplo 6.4.3 En el problema expuesto en el Ejemplo 6.1.1 se hallaron casos en los cuales a pesar de que los individuos mejoraban en el sentido de Pareto, es decir en el espacio fenotípico, su distancia en el espacio genotípico con respecto al conjunto de óptimos aumentaba:

Generación 14

<i>Ind.</i>	x_1	x_2	$f_1(x_1, x_2)$	$f_2(x_1, x_2)$
a	1.764	-2.239	0.525	-2.366
b	1.531	-1.992	0.539	-2.640

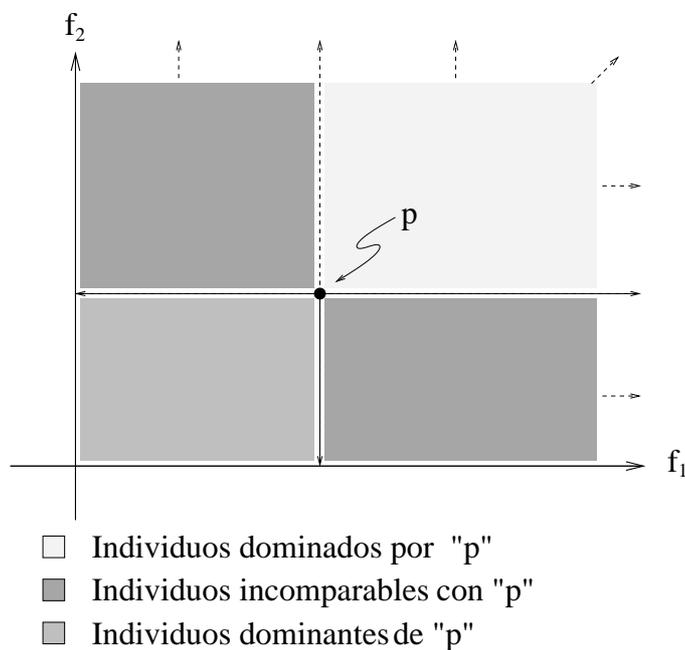


Figura 6.6: Representación, para dos funciones objetivo, de las regiones de dominancia e incomparabilidad de un individuo p .

Generación 15

<i>Ind.</i>	x_1	x_2	$f_1(x_1, x_2)$	$f_2(x_1, x_2)$
c	0.556	-1.168	-0.72	-3.03

A pesar de que toda la generación 14 fue reemplazada por la siguiente, la distancia en el espacio genotipo aumentó pues⁴:

$$d_{\infty}(\mathbf{a}, \mathcal{F}^{-1}) = \max\{|1.764 - 0.0|, |-2.239 - (-3)|\} = \max\{(1.764), (0.76)\} = \mathbf{1.764}$$

$$d_{\infty}(\mathbf{b}, \mathcal{F}^{-1}) = \max\{|1.531 - 0.0|, |-1.992 - (-3)|\} = \max\{(1.531), (1.01)\} = \mathbf{1.531}$$

$$d_{\infty}(\mathbf{c}, \mathcal{F}^{-1}) = \max\{|0.556 - 0.0|, |-1.168 - (-3)|\} = \max\{(0.556), (1.83)\} = \mathbf{1.83}$$

⁴De la geometría del problema, y para este ejemplo particular, es claro que

$$d_{\infty}(\mathbf{a}, \mathcal{F}^{-1}) = d_{\infty}(\mathbf{a}, (0, -3)) = \max\{(a_x - 0), (a_y - (-3))\},$$

igualmente para $d_{\infty}(\mathbf{b}, \mathcal{F}^{-1})$ y $d_{\infty}(\mathbf{c}, \mathcal{F}^{-1})$.

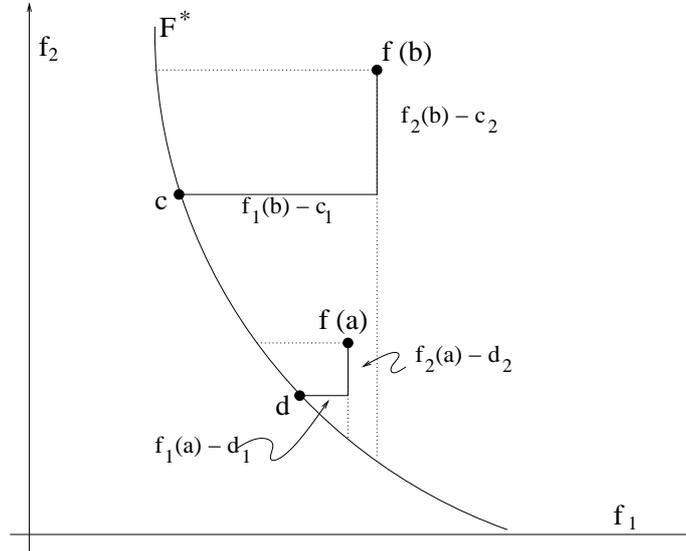


Figura 6.7: $d_{\infty}(f(b), \mathcal{F}^*) = d_{\infty}(f(b), c)$ y $d_{\infty}(f(a), \mathcal{F}^*) = d_{\infty}(f(a), d)$

$$\begin{aligned}
 d_{(\mathcal{F}_{14}^{-1}, \mathcal{F}^{*-1})} &= \max\{d_{\infty}(\mathbf{a}, \mathcal{F}^{-1}), d_{\infty}(\mathbf{b}, \mathcal{F}^{-1})\} \\
 &= \max\{1.764, 1.531\} \\
 &= \mathbf{1.764}
 \end{aligned}$$

$$\begin{aligned}
 d_{(\mathcal{F}_{15}^{-1}, \mathcal{F}^{*-1})} &= d_{\infty}(\mathbf{c}, \mathcal{F}^{-1}) \\
 &= \mathbf{1.83}
 \end{aligned}$$

implican que: $d_{(\mathcal{F}_{15}^{-1}, \mathcal{F}^{*-1})} > d_{(\mathcal{F}_{14}^{-1}, \mathcal{F}^{*-1})}$. Las figuras 6.8 y 6.9 ilustran a los individuos a, b y c en ambos espacios.

Lo anterior no ocurre, en general para frentes de Pareto en la frontera de conjuntos convexos, cuando las distancias se miden en el espacio fenotípico pues

$$\begin{aligned}
 a \leq b &\Leftrightarrow f_1(a) \leq f_1(b), f_2(a) \leq f_2(b), \dots, f_m(a) \leq f_m(b) \\
 &\Leftrightarrow f_i(a) - c_i \leq f_i(b) - c_i \text{ para todo } i = 1, \dots, m \text{ y para todo } c \in \mathcal{F},
 \end{aligned}$$

dado $c \in \mathcal{F}^*$ tal que $d_{\infty}(f(b), \mathcal{F}^*) = d_{\infty}(f(b), c)$ se tienen dos casos:

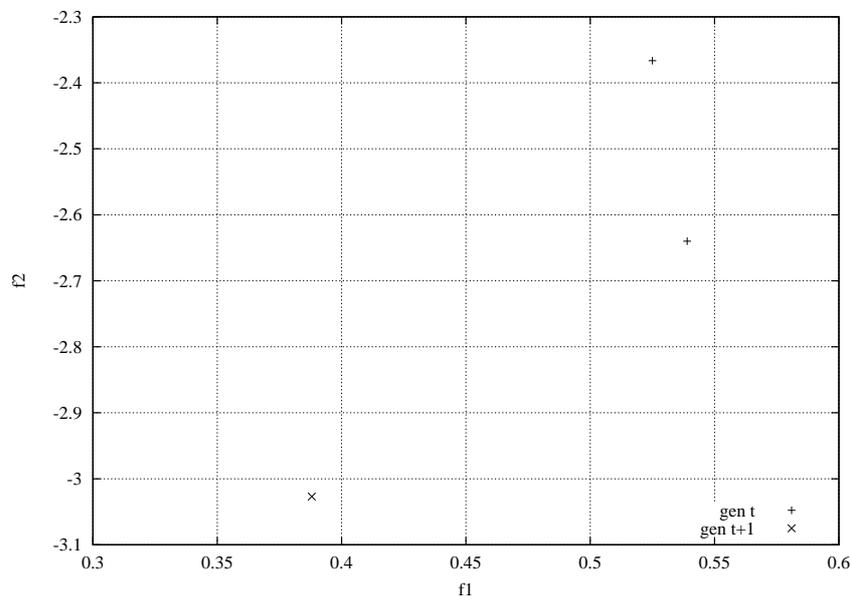


Figura 6.8: Fenotipos de los individuos del Ejemplo 6.4.3, la generación $t + 1 = 15$ reemplaza completamente a la generación $t = 14$.

I) $c \leq a$:

$$\begin{aligned}
 c \leq a &\Rightarrow |f_i(a) - c_i| \leq |f_i(b) - c_i| \text{ para todo } i = 1, \dots, m \\
 &\Rightarrow \sup\{|f_i(a) - c_i|\} \leq \sup\{|f_i(b) - c_i|\} \text{ con } i = 1, \dots, m \\
 &\Leftrightarrow d_\infty(f(a), c) \leq d_\infty(f(b), c) \\
 &\Rightarrow d_\infty(f(a), \mathcal{F}^*) \leq d_\infty(f(a), c) \leq d_\infty(f(b), c) = d_\infty(f(b), \mathcal{F}^*).
 \end{aligned}$$

II) c es incomparable con a : Dada la geometría del problema, (ver Figura 6.7), $d_\infty(f(a), \mathcal{F}) = d_\infty(f(a), d)$ y si $d_\infty(f(a), \mathcal{F}) = |f_i(a) - d_i|$ para algún $i = 1, \dots, m$, entonces

$$\begin{aligned}
 |f_i(a) - d_i| &\leq |f_i(a) - c_i| \\
 &\leq |f_i(b) - c_i| \\
 &= |f_j(b) - c_j| \text{ para } j = 1, \dots, m \\
 &\leq d_\infty(f(b), \mathcal{F}^*)
 \end{aligned}$$

implica que $d_\infty(f(a), \mathcal{F}^*) \leq d_\infty(f(b), \mathcal{F}^*)$ ■

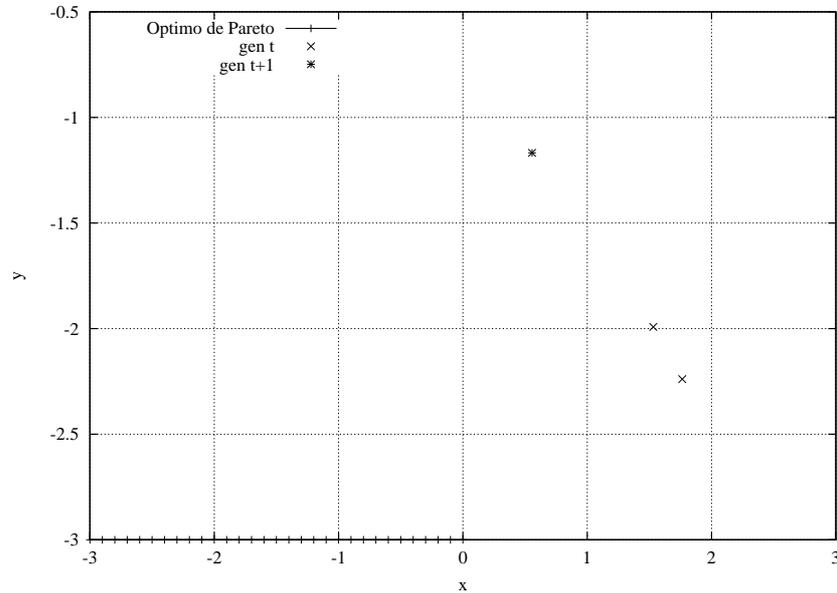


Figura 6.9: Genotipos de los individuos del Ejemplo 6.4.3. La distancia máxima hacia el conjunto de óptimos para la generación $t + 1 = 15$ aumenta con respecto a la encontrada en la generación anterior $t = 14$.

6.4.2 Ventajas poblacionales

Dado un problema con m funciones objetivo, se tiene que cada individuo p divide el espacio fenotípico en 4 regiones delimitadas por la intersección de m hiperplanos⁵ cada una. Estas regiones representan el espacio en el que se encuentran los fenotipos de individuos que dominan a p , los fenotipos de individuos que son dominados por p y los fenotipos de individuos que son incomparables con respecto a p (Ver Figura 6.6).

Independientemente del espacio en el que se estudie (genotípico o fenotípico) la velocidad de convergencia, la distancia entre la anti-cadena y el frente de Pareto no siempre es decreciente (Figura 6.10); esto es porque en cualquier momento pueden entrar elementos nuevos en la anti-cadena, los cuales pueden ser incomparables con

⁵Estas regiones también son llamadas “conos poliédricos” [36]. Un conjunto no vacío C en \mathbb{R}^n se dice un cono generado por y_1, \dots, y_m si

$$C = \{\lambda_1 y_1 + \dots + \lambda_m y_m \mid \lambda_1, \dots, \lambda_m \geq 0\}$$

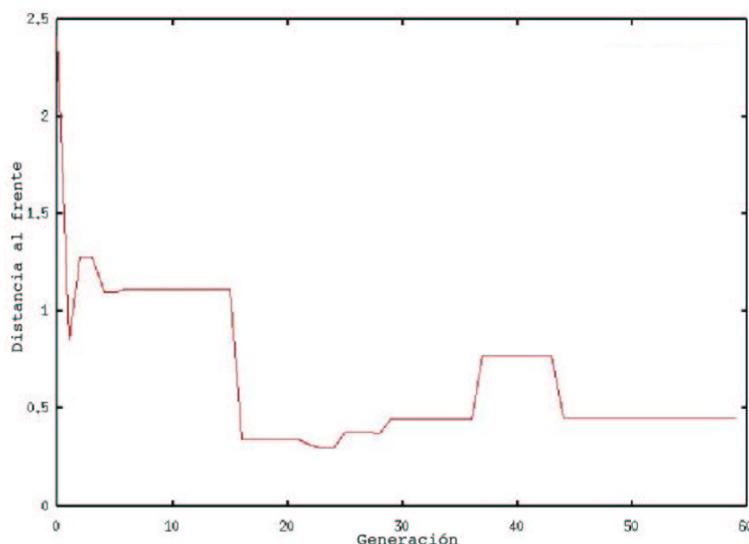


Figura 6.10: Velocidad de convergencia, en 60 generaciones del algoritmo propuesto. Para una población de 100 individuos muestra la distancia fenotípica al frente de Pareto para cada generación.

los actuales, por lo tanto y como no entraron a la anti-cadena dominando a alguno de los ya existentes entonces no necesariamente su distancia va a ser menor.

Sin embargo, la distancia sí decrece conforme pasa el tiempo y esto es gracias a que no se trabaja con un solo individuo, sino con toda una población de individuos elitistas. En la Figura 6.11 se ve que aunque la distancia no necesariamente decrece, el efecto conjunto de la mejora, en general, de la población ayuda a que el rango entre el cual la distancia puede medirse disminuya de generación en generación.

6.4.3 Probabilidad de acercamiento

Supongamos que para una cierta generación n la población elitista (ya sea la de padres o la secundaria) se encuentra bien determinada. Tomemos entonces, de entre los individuos de dicha población, el punto (K_1, K_2) como el fenotipo del individuo más alejado (d_∞) del frente de Pareto (ver Figura 6.12). En vista del Lema 6.4.2 consideraremos al valor ΔK_2 , con objeto de simplificar el análisis, como una cota para la distancia (d_∞) al frente de Pareto.

A continuación calcularemos el valor esperado para el valor de la distancia Δd

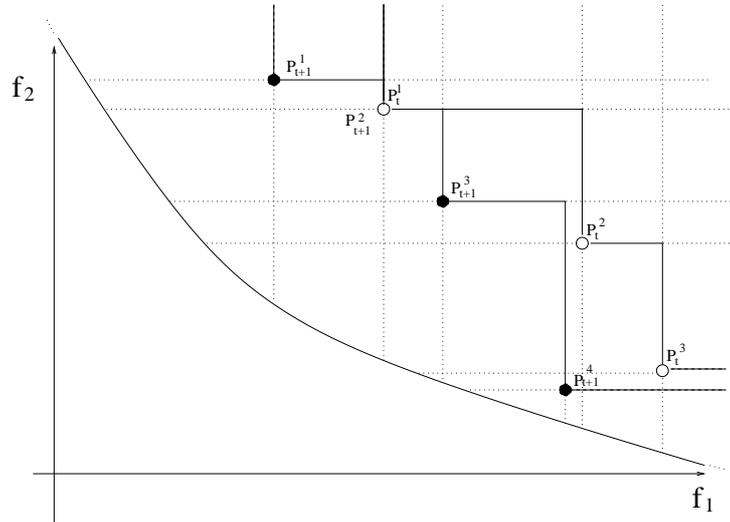


Figura 6.11: Se observa la región Γ en que pueden encontrarse los nuevos individuos y la cual es una cota para la distancia generacional. Para los individuos de la generación siguiente el área se acota aún más.

mostrada en la Figura 6.12:

$$\begin{aligned} E\{\Delta d\} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x,y)(\Delta d)(x,y)dA \\ &\geq f_{min} \int_{K_2-\Delta K_2}^{K_2} (K_2 - y) \left(1 - \frac{K_2 - y}{\Delta K_2}\right) \Delta K_1 dy. \end{aligned} \quad (6.1)$$

El cálculo del elemento dA se ilustra con la Figura 6.12 dado que

$$\frac{b}{\Delta K_1} = \frac{\Delta K_2 - \Delta d}{\Delta K_2} \Rightarrow b = \left(1 - \frac{\Delta d}{\Delta K_2}\right) \Delta K_1 dy$$

y

$$\Delta d = K_2 - y$$

tenemos

$$dA = b \cdot dy \Rightarrow dA = \left(1 - \frac{K_2 - y}{\Delta K_2}\right) \Delta K_1 dy.$$

El valor de la integral

$$\int_{K_2-\Delta K_2}^{K_2} (K_2 - y) \left(1 - \frac{K_2 - y}{\Delta K_2}\right) \Delta K_1 dy = \frac{\Delta K_1 (\Delta K_2)^2}{6} \quad (6.2)$$

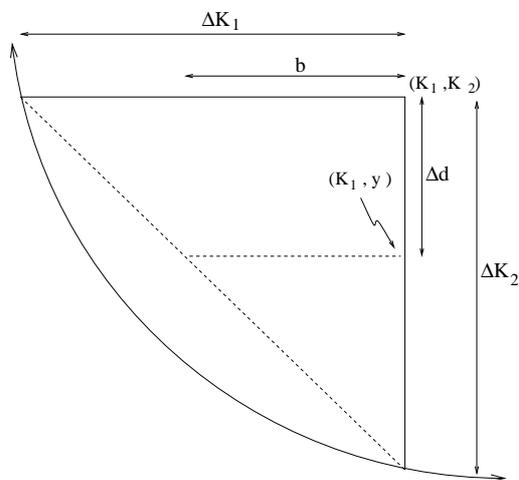


Figura 6.12: (K_1, K_2) representa el fenotipo del individuo más alejado (con respecto a d_∞) del frente para una cierta generación n .

puede aproximarse con

$$\frac{\Delta K_1 (\Delta K_2)^2}{6} \approx \frac{(\Delta K_2)^3}{36} \quad (6.3)$$

pues por el teorema del valor medio

$$\Delta K_1 \geq \alpha_{min} \Delta K_2$$

con

$$\alpha_{min} = \frac{1}{\max\{|FP'\|}\}$$

donde FP' es el valor de la derivada de la función objetivo f_2 con respecto a f_1

$$f_2(f_1) = f_1^2 + 4f_1 - 3 \Rightarrow FP' = 2f_1 + 4$$

por lo que

$$\max\{|FP'|\} = 6 \text{ para } f_1 \in [-5, -2].$$

Por otra parte, tomando un individuo $h \in H_{\lambda^t}$ obtenido durante el proceso de recombinación, sean Y_1, Y_2 las variables aleatorias definidas como $Y_1(h) = h_1 + N(0, \sigma_1)$ y $Y_2(h) = h_2 + N(0, \sigma_2)$. Estas variables representan a los individuos mutados a partir de h .

Claramente la función de distribución de probabilidad conjunta de Y_1, Y_2 está dada por:

$$\begin{aligned} F_{Y_1 Y_2}(y_1, y_2) &= P(\{Y_1 \leq y_1\} \cap \{Y_2 \leq y_2\}) \\ &= \int_{-\infty}^{y_1} \int_{-\infty}^{y_2} \frac{1}{2\pi} \frac{1}{\sigma_{Y_1} \sigma_{Y_2}} \cdot e^{-\left(\frac{(Y_1-h_1)^2}{2\sigma_{Y_1}^2} + \frac{(Y_2-h_2)^2}{2\sigma_{Y_2}^2}\right)} dY_2 dY_1. \end{aligned}$$

En el caso de la EE multi-objetivo sin autoadaptación, se tiene que la longitud de paso σ es fija, por lo que

$$\sigma_{Y_2} = \sigma_{Y_1} = \sigma,$$

fijando $\sigma = 0.5$, la función de densidad de probabilidad puede también acotarse por debajo con su valor mínimo dentro del área factible como:

$$f = \frac{1}{2\pi} \frac{1}{\sigma^2} \cdot e^{-\left(\frac{(x-h_1)^2}{2\sigma^2} + \frac{(y-h_2)^2}{2\sigma^2}\right)} \geq \frac{1}{2\pi} \frac{1}{0.5^2} \cdot e^{-\left(\frac{(6\sqrt{2})^2}{2 \cdot (0.5)^2} + \frac{(6\sqrt{2})^2}{2 \cdot (0.5)^2}\right)} = 5.3 \times 10^{-126},$$

siendo $f_{min} = 5.3 \times 10^{-126}$ resulta de 6.2 y 6.3 que

$$E\{\Delta d\} \geq \frac{f_{min}}{36} E\{(\Delta k_2^{(n)})^3\}$$

por lo anterior y considerando que por el crecimiento de las normas L^p con p ,

$$E\{(\Delta k_2^{(n)})^3\} \geq E\{\Delta k_2^{(n)}\}^3$$

tenemos que

$$\begin{aligned} E\{\Delta k_2^{(n+1)}\} &= E\{\Delta k_2^{(n)}\} - E\{\Delta d\} \\ &\leq E\{\Delta k_2^{(n)}\} - \frac{f_{min}}{36} E\{(\Delta k_2^{(n)})^3\} \\ &\leq E\{\Delta k_2^{(n)}\} - \frac{f_{min}}{36} E\{\Delta k_2^{(n)}\}^3. \end{aligned} \tag{6.4}$$

Suponiendo que para alguna iteración n se cumple que

$$E\{\Delta k_2^{(n)}\} \leq \frac{1}{\sqrt{\frac{f_{min}}{36}n + c}} \text{ para algún } c \tag{6.5}$$

tenemos por 6.4 que

$$E\{\Delta k_2^{(n+1)}\} \leq \frac{1}{\sqrt{\frac{f_{min}}{36}n + c}} - \frac{f_{min}}{36} \frac{1}{\left(\frac{f_{min}}{36}n + c\right)^{\frac{3}{2}}} \quad (6.6)$$

de la ecuación 6.6 y dado que

$$\begin{aligned} \frac{(n^2 - 1)(n - 1)}{n^3} \leq 1 &= \frac{(n - 1)^2}{n^2} \leq \frac{n}{n + 1} \\ \Rightarrow \left(1 - \frac{1}{n}\right)^2 &\leq \frac{n}{n + 1} \\ \Rightarrow 1 - \frac{1}{n} &\leq \sqrt{\frac{\frac{f_{min}}{36}n}{\frac{f_{min}}{36}(n + 1)}} \\ \Rightarrow 1 - \frac{\sqrt{\frac{f_{min}}{36}n}}{\sqrt{\frac{f_{min}}{36}(n + 1)}} &\leq \frac{\frac{f_{min}}{36}}{\frac{f_{min}}{36}n} \\ \Rightarrow \frac{1}{\sqrt{\frac{f_{min}}{36}n}} - \frac{1}{\sqrt{\frac{f_{min}}{36}(n + 1)}} &\leq \frac{f_{min}}{36} \frac{1}{\left(\frac{f_{min}}{36}n\right)^{\frac{3}{2}}} \\ \Rightarrow \frac{1}{\sqrt{\frac{f_{min}}{36}n}} - \frac{f_{min}}{36} \frac{1}{\left(\frac{f_{min}}{36}n\right)^{\frac{3}{2}}} &\leq \frac{1}{\sqrt{\frac{f_{min}}{36}(n + 1)}} \end{aligned}$$

vemos que $c = 0$ cumple tanto que $c \leq \frac{1}{(\Delta K_2^{(0)})^2}$ como que

$$\frac{1}{\sqrt{\frac{f_{min}}{36}n + c}} - \frac{f_{min}}{36} \frac{1}{\left(\frac{f_{min}}{36}n + c\right)^{\frac{3}{2}}} \leq \frac{1}{\sqrt{\frac{f_{min}}{36}(n + 1) + c}},$$

Con todo lo anterior, queda entonces demostrado que

$$E\{\Delta k_2^{(n)}\} \leq \frac{1}{\sqrt{\frac{f_{min}}{36}n}}.$$

Hemos probado que la velocidad de convergencia de la estrategia evolutiva multiobjetivo planteada en un principio es hiperbólica con exponente $\frac{1}{2}$. Esto muestra que el proceso de convergencia es muy lento, lo cual coincide con los resultados

experimentales obtenidos y presentados en la sección 6.6.1. El orden de convergencia no puede mejorar si se mantiene una longitud de paso con σ_i constante, ya que $f \leq \frac{1}{2\pi\sigma^2}$ por lo que si en el análisis previo, en vez de minimizar el valor esperado lo maximizamos, la velocidad calculada no variaría mas que en el valor de la constante f_{min} (la cual esta acotada por f), es decir, conserva el mismo orden $O(\frac{1}{\sqrt{n}})$.

Es por lo anterior, que propusimos en la sección 6.3 un proceso de auto-adaptación basado en una variación controlada para los valores de los σ'_i s. Los resultados que obtuvimos al implementarlo en el algoritmo de la EE multiobjetivo se presentan en la sección 6.6.2 y muestran una mejora significativa sobre los arrojados por el algoritmo sin auto-adaptación.

6.5 Algoritmo para la EE Multi-objetivo

El algoritmo al cual analizamos su velocidad de convergencia fue:

Algoritmo 6.5.1 EE-Multi-objetivo: Denotamos por P_μ^t a la población de padres a cada generación t , el subíndice de P^t denotará el tamaño de la población y variará como μ_t durante el ciclo evolutivo de cada generación, pero al pasar de una generación a otra se tomará siempre un número fijo de padres μ . La población de descendientes H_λ^t también variará de tamaño pues λ estará determinado por $\lambda = k \cdot \mu_t$ donde k es el número de hijos para cada pareja de padres; k no varía con el tiempo.

- I) Se inicializa $P_\mu^{(0)}$ de manera aleatoria.
- II) $P_{\mu_0}^{(0)} \leftarrow \mathcal{M}_f(P_\mu^{(0)})$.
- III) $t \leftarrow 0$.
- IV) $H_\lambda^{(t)} \leftarrow \text{recombinación}(P_{\mu_t}^{(t)})$.
- V) $H_\lambda^{(t)} \leftarrow \text{mutación}(H_\lambda^{(t)})$.
- VI) $P_{\mu_{t+1}}^{(t+1)} = \mathcal{M}_f(P_{\mu_t}^{(t)} \cup H_\lambda^{(t)})$.
- VII) Restringir a μ el tamaño de la población para la generación siguiente

$$P_\mu^{(t+1)} \leftarrow P_{\mu_{t+1}}^{(t+1)}.$$

VIII) $t \leftarrow t + 1$.

IX) Si el criterio de terminación no se satisface, ir al paso IV.

6.6 Resultados

6.6.1 Resultados sin auto-adaptación

Las figuras 6.13, 6.14, 6.15 y 6.16 muestran, para diferentes semillas aleatorias, la distancia máxima de la población de padres con respecto al frente de Pareto teórico del ejemplo 6.1.1 a cada generación. Cada gráfica muestra el comportamiento del algoritmo 6.5 usando una desviación estándar fija $\sigma = 0.01$, 0.1 , 1.0 y 2.0 respectivamente; en todos los casos la cantidad de padres que intervienen en la creación de cada hijo es $\rho = 2$. Notemos que en ninguno de los casos se alcanza la convergencia total antes de 6000 generaciones. Se puede observar también que para una σ muy pequeña (Figura 6.13) la velocidad de convergencia es muy mala, esto se explica porque el tamaño de paso con que la población se acerca al frente es demasiado pequeño por lo que la velocidad empeora.

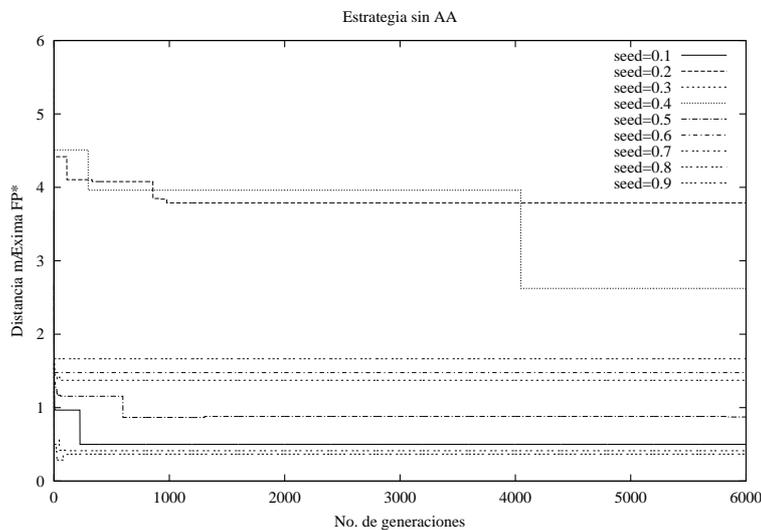


Figura 6.13: Gráfica de la velocidad de convergencia para nueve diferentes semillas iniciales, durante 6000 generaciones con $\sigma = 0.01$, del algoritmo con $\mu = 100$ padres y $\lambda = 150$ hijos.

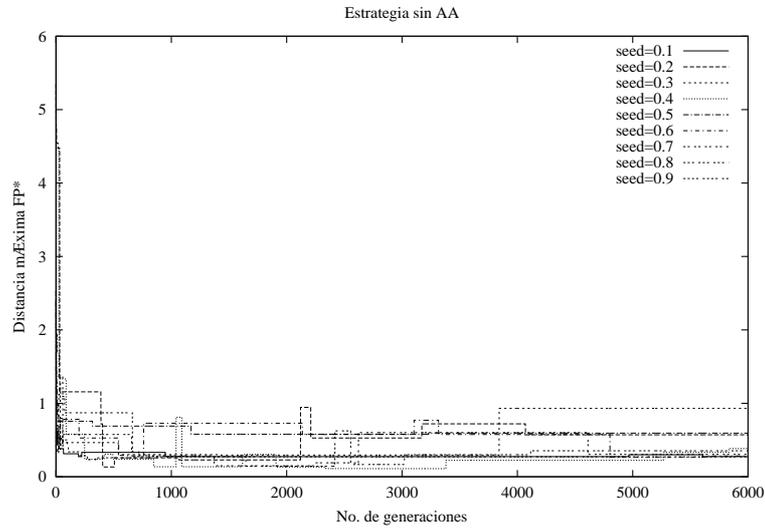


Figura 6.14: Gráfica de la velocidad de convergencia para nueve diferentes semillas iniciales, durante 6000 generaciones con $\sigma = 0.1$, del algoritmo con $\mu = 100$ padres y $\lambda = 150$ hijos.

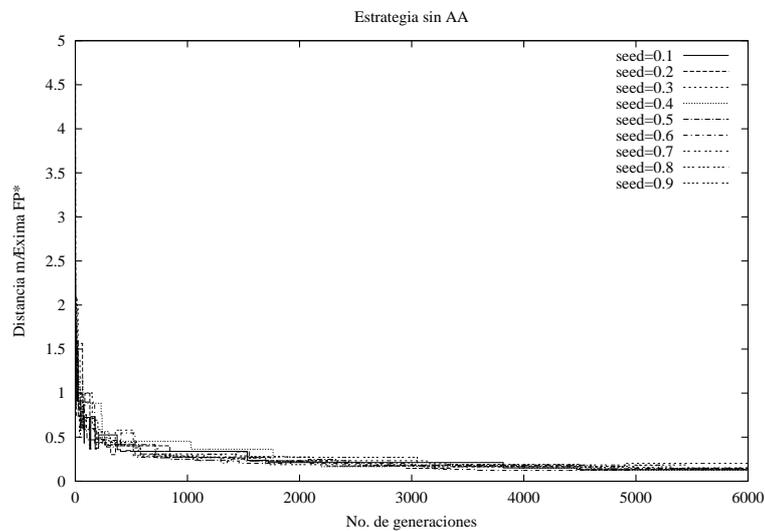


Figura 6.15: Gráfica de la velocidad de convergencia para nueve diferentes semillas iniciales, durante 6000 generaciones con $\sigma = 1.0$, del algoritmo con $\mu = 100$ padres y $\lambda = 150$ hijos.

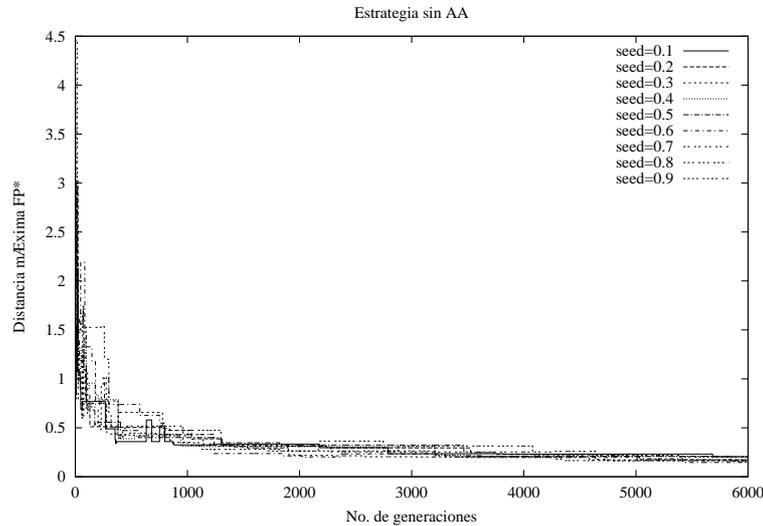


Figura 6.16: Gráfica de la velocidad de convergencia para nueve diferentes semillas iniciales, durante 6000 generaciones con $\sigma = 2.0$, del algoritmo con $\mu = 100$ padres y $\lambda = 150$ hijos.

6.6.2 Resultados con auto-adaptación

Las figuras 6.17 y 6.18 muestran, para diferentes semillas aleatorias, la distancia máxima de la población de padres de 100 y 200 individuos respectivamente con respecto al frente de Pareto teórico del ejemplo 6.1.1 para el algoritmo 6.5 adicionándole el mecanismo de autoadaptación propuesto en 6.3. Cada gráfica muestra el comportamiento del algoritmo con 100 y 200 padres respectivamente. La cantidad de padres que intervienen en la creación de cada hijo es $\rho = 2$. Podemos notar que la convergencia total se alcanza para el caso $\mu = 200$ en menos de 40 generaciones y para el caso $\mu = 100$ en menos de 100 generaciones; cabe mencionar que la ventaja de un caso sobre otro puede deberse al efecto poblacional mencionado en la sección 6.4.2.

Aunque no se ha demostrado una cota para la velocidad de convergencia del algoritmo con autoadaptación, se desprende de los presentes resultados experimentales (para este ejemplo en particular) que el orden de la velocidad de este algoritmo es menor que $O(\frac{1}{\sqrt{n}})$ encontrado en la sección 6.4.3 para el algoritmo sin autoadaptación.

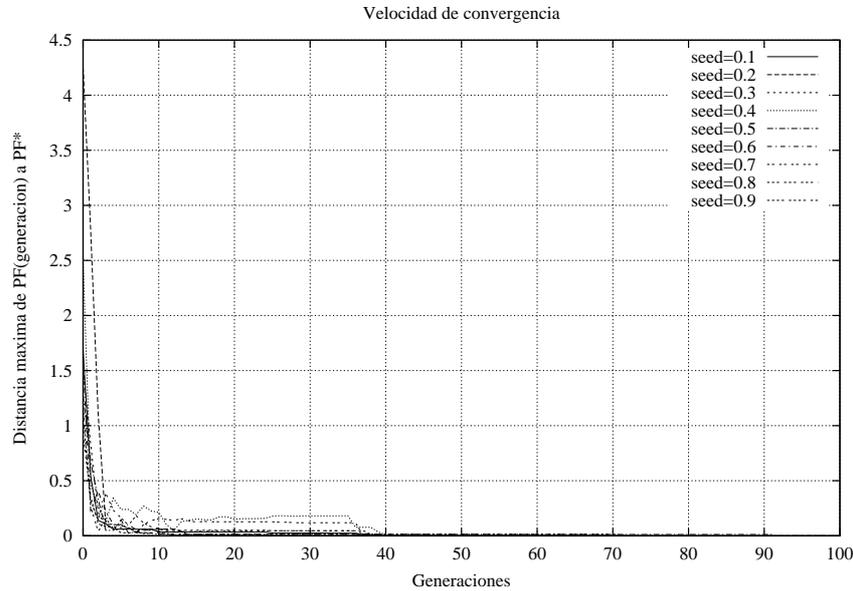


Figura 6.17: Velocidades de convergencia para cada generación de la estrategia con auto-adaptación. $\mu = 100$

6.7 Sumario

En este capítulo se estudiaron varios aspectos de la estrategia evolutiva multi-objetivo sobre un caso de estudio particular para el cual se determinó su velocidad de convergencia al frente de Pareto. También se propuso un mecanismo de auto-adaptación en el operador de mutación y se comprobó experimentalmente que dicho mecanismo mejora la velocidad de convergencia del algoritmo.

La prueba para la velocidad de convergencia presentada en la sección 6.4.3 se aplica no solo al este caso de estudio, sino que funciona para la estrategia evolutiva aplicada a cualquier otro problema con dos funciones objetivo y frente de Pareto convexo simple que se cuente con un operador de mutación cuya función de densidad de probabilidad no se anule.

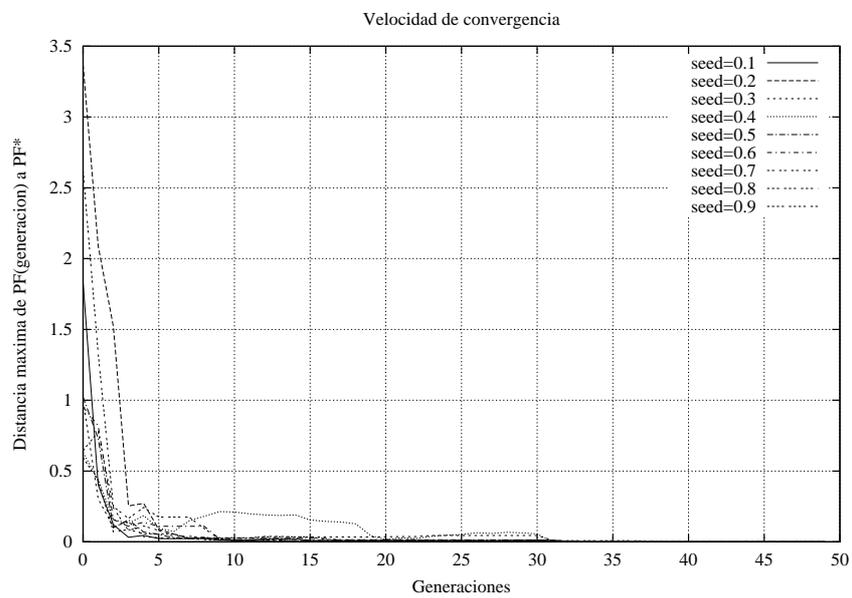


Figura 6.18: Velocidades de convergencia para cada generación de la estrategia con auto-adaptación. $\mu = 200$

Conclusiones

A lo largo de este trabajo se establecieron la notación y los conceptos principales para el estudio teórico de las Estrategias Evolutivas $(1 + 1)$, (μ, λ) y $(\mu + \lambda)$ usadas en problemas de optimización multi-objetivo. Se analizaron sus operadores desde el punto de vista probabilístico ejemplificando su efecto en la búsqueda mediante un problema específico.

Se presentaron, en el Capítulo 5, los algoritmos propuestos para adaptar la EE original al caso multi-objetivo. Se demostró que, para espacios de búsqueda finitos, dichos algoritmos multi-objetivo convergen hacia el frente de Pareto en un número finito de iteraciones. En este capítulo se utilizaron dos enfoques, uno restringiendo y otro no restringiendo el tamaño de la población de padres; el uso de estos dos enfoques se justificó y discutió en la sección 6.3.2.

La idea de convergencia que aquí utilizamos, para la Estrategia Evolutiva multi-objetivo, se basa en que la población elitista quede totalmente contenida dentro del frente de Pareto en alguna iteración del algoritmo y en iteraciones posteriores las soluciones no se pierdan. En el caso de la estrategia $(\mu + \lambda)$ mutiobjetivo la población elitista coincide con la población original de padres a cada generación, debido al elitismo implícito de este tipo de selección; en el caso de la estrategia $(\mu + \lambda)$ mutiobjetivo la población elitista se maneja a través de una población secundaria. Para el análisis de la velocidad de convergencia de los algoritmos, consideramos a cada generación la distancia de la población al frente de Pareto como el peor valor de las distancias de cada uno de los individuos (elitistas) con respecto al frente; así, la “convergencia total” del algoritmo depende de esta distancia, y asegura el acercamiento de la población total hacia el frente.

Se analizaron, en el Capítulo 6, varios aspectos acerca del uso de jeraquización de Pareto en el proceso de selección, como son, el aumento o disminución excesivos en el tamaño de la población así como las consecuencias que éstos acarrearán al problema específico y en general, para la clase de problemas en los que sus funciones objetivo mantienen alguna correlación estadística. Para el caso de estudio elegido

se halló, en la sección 6.4.3, una relación para la velocidad de convergencia de la EE multiobjetivo sin auto-adaptación, sin embargo la prueba puede extenderse sin mayores dificultades a la clase de problemas multiobjetivos con frentes de Pareto convexos bajo mutaciones cuya densidad de probabilidad no se anule. Se mostró que el acercamiento al frente de Pareto se hace con orden $O(\frac{1}{\sqrt{n}})$ con n el número de iteración del algoritmo, siendo esta una velocidad de convergencia muy lenta, lo cual se ilustra con resultados experimentales en la sección 6.6.1. Se propuso entonces un mecanismo de auto-adaptación de los parámetros de la mutación en el algoritmo, el cual, es perfectamente escalable para problemas de dimensiones mayores.

Ayudados por el mecanismo de auto-adaptación propuesto en 6.3, los algoritmos de la EE Multiobjetivo que usan la jerarquización de Pareto dieron buenos resultados, para el caso de estudio, además de que garantizan teóricamente su convergencia asintótica hacia el frente de Pareto. Los resultados comparados fueron para el algoritmo sin el mecanismo de auto-adaptación, que no logró una convergencia completa en menos de 6000 generaciones, contra el algoritmo que usa la auto-adaptación propuesta, el cual logró la convergencia completa en menos de 100 generaciones, con lo que se ve experimentalmente una mejoría notable.

Trabajo Futuro

Fuera de los alcances de esta tesis queda estudiar varios problemas concretos resueltos previamente con algoritmos evolutivos multiobjetivo que usan selección de tipo $(\mu + \lambda)$, como son NPGA2 [17] y NSGA-II [15], estudiando las variantes de los operadores genéticos en estos algoritmos y cómo afectan a la convergencia para ciertos problemas en particular.

Analizar cómo afecta estocásticamente a la convergencia el uso de otros tipos de selección, por ejemplo, la adaptación propuesta en la sección 4.2.2 para multiobjetivo, del torneo binario.

Un problema abierto para varios algoritmos evolutivos multiobjetivo consiste en determinar el tiempo esperado de convergencia del algoritmo para diversas clases de problemas.

Para estimar la velocidad de convergencia del algoritmo de autoadaptación propuesto, es necesario hacer un análisis cuidadoso de cómo interviene generación a generación el ajuste de la desviación estandar para cada una de las variables; se puede iniciar haciendo un análisis estadístico para ciertos casos particulares de estudio con características definidas previamente.

Bibliografía

- [1] A. Agapie. Genetic algorithms: Minimal conditions for convergence. In J.K. Hao, E. Lutton, E. Ronald, and M. Schoenauer, editors, *Artificial Evolution: Third European Conference; selected papers/ AE'97*, pages 183–193, Berlin, 1998. Springer.
- [2] Niels Arley and K. Rander Buch. *Introduction to the Theory of Probability and Statistics*. Applied Mathematics Series. John Wiley & Sons, Inc., 1961.
- [3] Thomas Bäck. *Evolutionary algorithms in Theory and Practice*. Oxford University Press, 1996.
- [4] Norman T. J. Bailey. *The Elements of Stochastic Processes*. John Wiley & Sons, Inc., 1964.
- [5] Robert G. Bartle. *Introducción al análisis matemático*. Noriega, 1989.
- [6] Hans-Georg Beyer. *Ein Evolutionsverfahren zur mathematischen Modellierung stationärer Zustände in dynamischen Systemen*. PhD thesis, Hochschule für Architektur und Bauwesen, Weimar, 1989.
- [7] Hans-Georg Beyer. Toward a theory of evolution strategies some asymptotical results from the $(1+\lambda)$ -theory. *Evolutionary Computation*, 1(2):165–188, 1993.
- [8] Hans-Georg Beyer. *The Theory of Evolution Strategies*. Springer, 1998.
- [9] Erick Cantú Paz. *Efficient and accurate parallel genetic algorithms*. Kluwer Academic Publishers, Massachusetts, USA, 2000.
- [10] Yuan Shih Chow and Henry Teicher. *Probability Theory, Independence-Interchangeability-Martingales*. Springer-Verlag, 1978.

-
- [11] Kai Lai Chung. *Elementary Probability Theory with Stochastic Processes*. UTM. Springer-Verlag, 1979.
- [12] Carlos A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems.*, 3(1):269–308, August 1999.
- [13] Carlos A. Coello Coello. Introducción a la computación evolutiva. Notas de curso, mayo 2001.
- [14] George B. Dantzig and Mukund. Thapa. *Linear Programming*. Springer Series in Operations Research, 1997.
- [15] Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.
- [16] A.E. Eiben, E.H.L. Aarts, and K.M. van Hee. Global convergence of genetic algorithms: A markov chain analysis. In H.P. Scheffel and R. Männer, editors, *Parallel problem solving from nature*, pages 4–12, Berlin and Heidelberg, 1991.
- [17] Mark Erickson, Alex Mayer, and Jeffrey Horn. The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 681–695. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
- [18] D. Fogel and L. Stayton. On the effectiveness of crossover in simulated evolutionary optimization. *BioSystems*, 32:171–182, 1994.
- [19] David B. Fogel. Asymptotic convergence properties of genetic algorithms and evolutionary programming. *Cybernetics and systems*, 25(3):389–407, 1994.
- [20] David B. Fogel. *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*. The Institute of Electrical and Electronic Engineers, New York, 1998.
- [21] Lawrence J. Fogel. *Artificial intelligence through simulated evolution*. John Wiley, New York, 1966.

-
- [22] Bert Fristedt and Laurence Gray. *A Modern Approach to Probability Theory*. Birkhäuser, 1997.
- [23] Saul I. Gass. *Linear Programming, Methods and applications*. McGraw-Hill, 1975.
- [24] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [25] Thomas Hanne. On the convergence of multiobjective evolutionary algorithms. *European Journal of Operational Research*, 117:553–564, 1999.
- [26] Thomas Hanne. Global multiobjective optimization using evolutionary algorithms. *Journal of Heuristics*, 6:347–360, 2000.
- [27] M. Herdy. Reproductive isolation as strategy parameter in hierarchically organized evolution strategies. In *Männer & Manderick*, pages 207–117. Berlin, Springer, 1992.
- [28] J. F. Hicklin. Application of the genetic algorithm to automatic program generation. Master’s thesis, University of Idaho, Moscow, Idaho, 1986.
- [29] John H. Holland. *Concerning efficient adaptative systems*, pages 215–230. Self-organizing Systems. Spartan Books, Washington, D.C., 1962.
- [30] John H. Holland. Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery*, 9:297–314, 1962.
- [31] M. Iosifescu. *Finite Markov process and their applications*. Chichester:Wiley, 1980.
- [32] S. Kirkpatrick, C. Gellatt, and M. Vecchi. Optimization and simulated annealing. *Science*, 4598(220):671–680, 1983.
- [33] John R. Koza. *Genetic programming. On the programming of computers by means of natural selection*. MIT, Press, Cambridge, massachusetts, 1992.
- [34] John R. Koza. *Genetic programming II. Automatic discovery of reusable programs*. MIT, Press, Cambridge, massachusetts, 1994.
- [35] Frank Kursawe. A variant of evolution strategies for vector optimization. In H.P. Scheffel and R. Männer, editors, *Parallel problem solving from nature*, pages 193–197, Berlin, 1991.

-
- [36] Adriana Lara López. Temas selectos en teoría de gráficas, geometría de poliedros y programación lineal . Escuela Superior de Física y Matemáticas, IPN México, Noviembre 2001. Bachelor's Thesis.
- [37] Marco Laumanns, Günter Rudolph, and H.P. Scheffel. A spatial predator-prey approach to multiobjective optimization: A preliminary study. In Agoston E. Eiben et al, editor, *Parallel Problem Solving from Nature*, pages 241–249, Berlin, 1998.
- [38] G. E. Liepins, M. R. Hilliard, J. Richardson, and Palmer M. Genetic algorithms application to set covering and travelling salesman problems. In D. E. Brown and White C. C., editors, *Operations research and Artificial Intelligence: The integration of problem-solving strategies*, pages 29–57. Kluwer Academic, Massachusetts, 1990.
- [39] George G. Mitchell, Diarmuid O'Donoghue, and Adrian Trenaman. A new operator efficient evolutionary solutions to the travelling salesman problem. In *IASTED. Applied Informatics*, pages 98–103, Innsbruck, Austria, 2000.
- [40] A. E. Nix and M.D. Vose. Modeling genetics algorithms with markov chains. In *Annals of Mathematics and Artificial Intelligence*, volume 5, pages 79–88. Kluwer Academic, 1992.
- [41] M. Peschel and C. Riedel. Use of vector optimization in multiobjective decision making. In D. E. Bell, Keeney R. L., and H. Raiffa, editors, *Conflicting Objectives in Decisions*, pages 97–121. Wiley, Chichester, 1977.
- [42] Ingo Rechenberg. Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution. Technical report, Frommann-Holzboog, Stuttgart, 1973.
- [43] Ingo Rechenberg. Evolutionsstrategie: Optimierung technischer systemenach prinzipien de biologischen evolution. Technical report, Stuttgart: Frommann-Holzboog, 1973.
- [44] Ingo Rechenberg. Evolutionsstrategien. In *Simulationmethoden in der Medizin und Biologie*, pages 83–114. Berlin:Springer, 1978.
- [45] Günter Rudolph. Convergence properties of canonical genetic algorithms. *IEEE Transactions on Neural Networks* 5, 1:96–101, 1994.

-
- [46] Günter Rudolph. Convergence of evolutionary algorithms in general search spaces. In *Proceedings of the 3rd IEEE International Conference on Evolutionary Computation*, pages 50–54. IEEE Press, Piscataway (NJ), 1996.
- [47] Günter Rudolph. *Convergence Properties of Evolutionary Algorithms*. Verlag Dr. Kovač, 1997.
- [48] Günter Rudolph. Evolutionary search for minimal elements in partially ordered finite sets. In V.W. Porto, N. Saravanan, and D. Waagen, editors, *Proceedings of the 7th Annual Conference on Evolutionary programming (EP'98)*, pages 345–353. Springer, Berlin and Heidelberg, 1998.
- [49] Günter Rudolph. On a multiobjective evolutionary algorithm and its convergence to the pareto set. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, pages 511–516, Piscataway (NJ), 1998. IEEE Press.
- [50] Günter Rudolph. Evolutionary search under partially ordered fitness sets. In *Proceedings of the International Symposium on Information Science Innovations (ISI 2001) in Engineering of Natural and Artificial Intelligent Systems ENAIS*, pages 818–822. ICSC Academic Press:Millet/Sliedrecht, 2001.
- [51] Günter Rudolph. Some theoretical properties of evolutionary algorithms under partially ordered fitness values. In *Proceedings of the Evolutionary Algorithms Whorkshop (EAW 2001)*, pages 9–22. INFOREC Printing House, Bucharest Romania, 2001.
- [52] Günter Rudolph and Alexandru Agapie. Convergency properties of some multiobjective evolutionary algorithms. In *Proceedings of the 2000 Congress on Evolutionary Computation (CEC 2000)*, volume 2, pages 1010–1016, Piscataway (NJ), 2000. IEEE Press.
- [53] Günter Rudolph and Alexandru Agapie. A partial order approach to noisy fitness function. In A Zalzala and et al., editors, *Proceedings of the 2000 congress on evolutionary computation (CEC 2000)*, volume 2, pages 1010–1016, Piscataway (NJ), 2001. IEEE Press.
- [54] H.-P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels der evolutionstrategie*, volume 26 of *Interdisciplinary Systems Research*. Birkhäuser, Basel, 1977.

-
- [55] H.-P. Schwefel. *Numerical Optimization of Computer Models*. Chichester, UK, Wiley, 1981.
- [56] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley Inter-science, 1995.
- [57] A.N. Shiryaev. *Probability*. GTM. Springer, 1996.
- [58] S. Ray Thomas. An approach to the synthesis of life. In C.G. Langton, C. Taylor, J.D. Farmer, and S. Rasmussen, editors, *Artificial Life*, Reading, pages 371–408. Addison-Wesley, Massachusetts, 1992.
- [59] Manuel Valenzuela-Rendón and Eduardo Uresti-Charre. A non-generational genetic algorithm for multiobjective optimization. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 658–665, San Mateo, California, July 1997. Michigan State University, Morgan Kaufmann Publishers.
- [60] David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Faculty of the Graduate School of Engineering of the Air Force Institute of Technology, 1999.
- [61] David A. Van Veldhuizen and Gary B. Lamont. Evolutionary computation and convergence to a pareto front. In *Late Breaking Papers at the Genetic Programming*, pages 221–228, Stanford University, California, 1998. Stanford University Bookstore.
- [62] A. Wetzel. Evaluation of the effectiveness of genetic algorithms in combinatorial optimization. Technical report, University of Pittsburg, Pittsburg (unpublished), 1973.
- [63] D.H. Wolpert and Macready W.G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary computation*, 1(1):67–82, 1997.

Índice de Materias

$\mathcal{G}(x)$, 17

aleatoria

variable, 37

aleatorio

camino, 42

vector, 38

álgebra, 33

σ -álgebra, 33

algoritmo, 7

anticadena, 16, 50

aptitud, 21

mejor, 50, 51

auto-adaptación, 50

binaria

relación, 15

bloque constructor, 26

hipótesis, 26, 55

cadena, 16

cadena de Markov, 42

completo

Poset, 16, 17

conjunto

de Borel, 34

de elementos minimales, 16, 50

de padres, 51, 52

parcialmente ordenado, 16

solución, 29

convergencia

casi segura, 45

con probabilidad 1, 45

cromosoma

de un individuo, 51

cruza uniforme, 56

desigualdad

lineal, 4

discreto

en el tiempo, 42

vector aleatorio, 39

distancia

de Hamming, 44

dominación

de Pareto, 29, 54

esquema de, 28

por torneo binario, 54

relación de, 15

ecuación

lineal, 3

EE, 23

elemento

minimal, 16

endógenos, 50, 85

espacio

de medida, 36

de probabilidad, 36

fenotípico, 53

muestra, 37

esperanza, 40

- condicional, 41
- vector, 41
- esquema, 26
- estados
 - espacio de, 42, 51
- estocástica
 - matriz, 43
- estocástico
 - proceso, 41
- estrategia
 - elitista, 68
 - evolutiva, 23
- evento
 - imposible, 37
 - seguro, 37
- eventos, 37
- evolutivas
 - estrategias, 23
- exógenos, 51
- exploración, 21
- explotación, 21
- finita
 - cadena de Markov, 42
- frente
 - conocido, 31, 47
 - verdadero, 30, 47
- función
 - de aptitud, 21
 - de densidad, 39
 - de distribución, 38
 - objetivo, 3, 21
- genéticos
 - algoritmos, 25
 - operadores, 20
- generación, 51
- global
 - óptimo, 3
- homogénea
 - cadena de Markov, 42
- igualdad
 - lineal, 3
- individuo, 20, 50
 - cromosoma de un, 51
- inecuación
 - lineal, 4
- irreducible
 - matriz, 44
- isotropía, 52
- kernel de variación, 57, 65
- lineal
 - desigualdad, 4
 - ecuación, 3
- longitud de paso, 53
- métrica, 47
 - discreta, 64
- mínimo
 - global, 3
- medible
 - función, 35
- medida, 36
 - de probabilidad, 37
 - espacio de, 36
- MOP, 12, 49
- multirecombinación, 55
- mutación, 21, 52
- no comparables, 15
- no dominado, 29, 50
- no negativa
 - matriz, 43
- norma, 46

- supremo, 89
- objetivo
 - función, 3
- operadores genéticos, 20
- optimización, 1
- óptimo, 29
 - de Pareto, 29
 - global, 3
- orden
 - parcial, 15, 50
 - total, 16
- Pareto
 - óptimo, 30
 - frente, 29
 - no dominación, 50
 - óptimo, 29
- partición, 36
- población, 20, 51
 - secundaria, 74
- Poset, 16
 - completo, 17
 - elemento minimal en un, 16
- positiva
 - matriz, 44
- probabilidad
 - condicional, 41
 - del evento, 37
 - distribución de, 38
 - espacio de, 36
 - espacio inducido de, 38
 - función de densidad, 39
 - medida de, 37
- problema, 7
- recombinación, 21, 55
- recta real, 34
- relación
 - de dominación, 15
 - binaria, 15
 - de orden parcial, 15
- selección, 21, 52
- solución, 4
- variable aleatoria, 37
 - distribución de una, 38
 - esperanza de una, 40
 - independiente, 39
- vector
 - no dominado, 29
- vector aleatorio, 38
 - continuo, 39
 - discreto, 39
 - soporte, 39