

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY

CAMPUS MONTERREY

DIVISION DE INGENIERIA Y ARQUITECTURA  
PROGRAMA DE GRADUADOS EN INGENIERIA



OPTIMIZACION MULTIOBJETIVO DE PROCESOS  
USANDO ALGORITMOS GENETICOS

TESIS

PRESENTADA COMO REQUISITO PARCIAL  
PARA OBTENER EL GRADO ACADÉMICO DE:  
MAESTRO EN CIENCIAS EN SISTEMAS  
AMBIENTALES CON ESPECIALIDAD EN INGENIERIA  
DE PROCESOS

POR:

MARIA DEL CARMEN BARCENA RAMIREZ

MONTERREY, N. L.

MARZO DEL 2003

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY**

**CAMPUS MONTERREY**

DIVISIÓN DE INGENIERÍA Y ARQUITECTURA  
PROGRAMA DE GRADUADOS EN INGENIERÍA



OPTIMIZACIÓN MULTIOBJETIVO DE PROCESOS  
USANDO ALGORITMOS GENÉTICOS

**TESIS**

PRESENTADA COMO REQUISITO PARCIAL  
PARA OBTENER EL GRADO ACADÉMICO DE

MAESTRO EN CIENCIAS EN SISTEMAS AMBIENTALES  
CON ESPECIALIDAD EN INGENIERÍA DE PROCESOS

**POR**

MARÍA DEL CARMEN BÁRCENA RAMÍREZ

MONTERREY, N.L.

MARZO DEL 2003

**OPTIMIZACIÓN MULTIOBJETIVO DE PROCESOS  
USANDO ALGORITMOS GENÉTICOS**

por

Ing. María del Carmen Bárcena Ramírez

**TESIS**

Presentada al Programa de Graduados en Ingeniería

del

Instituto Tecnológico y de Estudios Superiores de Monterrey  
Campus Monterrey

Como requisito parcial para obtener el grado académico de

Maestro en Ciencias en Sistemas Ambientales  
con especialidad en Ingeniería de Procesos

Monterrey, Nuevo León

Marzo del 2003

## **Dedicatoria**

*A Dios por sus bendiciones y por darme el regalo de la vida.*

*A mis Padres Saúl Bárcena García y María del Carmen Ramírez García por su amor, confianza y apoyo en todo momento para la realización de mis proyectos.*

*A mis hermanos Saúl Fernando, Berenice y Belinda por su cariño.*

## **Agradecimientos**

A mi comité de tesis Dr. Joaquín Acevedo Mascarúa, Dr. Eduardo Gómez-Maqueo Aréchiga y Dr. Enrique Ortiz Nadal por el interés prestado durante la realización de este trabajo.

Al departamento de Ingeniería Química por las facilidades prestadas para la realización de esta tesis.

Al Dr. Federico Viramontes Brown por todas sus atenciones.

A mi novio y amigo Teodoro Delgado Herrera.

A mi amiga Claudia del Río por estar siempre al pendiente de Mí.

A la familia Sánchez Mejía, por todas sus atenciones y muestras de cariño desde mi llegada a esta ciudad.

Al Sr. Marcelo Castillo Prado, por su valioso apoyo.

A Oscar Balcázar, Ana D. Sánchez , Claudia Martínez, Jesús Mireles, Alma Flores, Francisco Calleja, Francisco Montes, Angie Miranda, Luis Rosas, Miguel Angel Medina, Octavio Rangel y Dora Elia Bernal por brindarme su apoyo y su amistad.

A todos los que me apoyaron con sus oraciones y buenos deseos.

---

## Resumen

Actualmente existen en la literatura diversas técnicas de optimización multiobjetivo las cuales han sido implementadas para la solución de problemas multicriterio, que ha menudo se presentan en ingeniería química. Una herramienta útil en la implantación de estas técnicas cuando se usan simuladores secuenciales son los algoritmos genéticos (AG), los cuales están basados en la evolución natural. Los simuladores de procesos representan un recurso de gran utilidad para el ingeniero de proceso ya que le permiten modelar fácilmente las diversas operaciones unitarias mediante técnicas matemáticas complejas.

En este trabajo se integra una técnica de optimización multiobjetivo (con algoritmos genéticos) con el simulador de procesos Aspen Plus para la solución de problemas de separación. Se presenta una guía para la selección de valores adecuados de los parámetros del AG y se propone un nuevo criterio de convergencia, basado en la clasificación de los individuos en intervalos según sus aptitudes; esto con el fin de mejorar el desempeño del AG.

Para verificar el desempeño del algoritmo genético con el criterio de paro implementado, se resolvieron tres problemas de optimización. El primero consiste en un problema numérico sencillo con dos funciones objetivo. El segundo involucra el acoplamiento del AG con el simulador para la optimización de un sistema de separación en donde se busca minimizar el costo del equipo y maximizar la pureza de los productos. El tercero consiste en incorporar como criterio de optimización la cuantificación del riesgo asociado a un sistema de separación, el otro criterio de optimización fueron las ganancias totales

En los ejemplos anteriores el criterio de paro demostró buen desempeño ya que se logró reducir el tiempo requerido para obtener un conjunto de soluciones óptimas, disminuyendo el tiempo de cómputo hasta en un 80%.

Este trabajo de investigación representa el primer acoplamiento (del cual el autor tiene conocimiento) de una técnica de optimización multiobjetivo (con algoritmos genéticos) con el simulador de procesos Aspen Plus, esto puede considerarse como una línea importante de investigación para trabajos futuros. Los objetivos propuestos fueron alcanzados, obteniéndose resultados satisfactorios.

---

---

---

## Índice general

<b>RESUMEN</b> .....	<b>1</b>
<b>ÍNDICE GENERAL</b> .....	<b>3</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>5</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>7</b>
<b>CAPÍTULO 1 INTRODUCCIÓN</b> .....	<b>9</b>
1.1 DESCRIPCIÓN DEL PROBLEMA .....	9
1.2 EJEMPLO MOTIVADOR.....	9
1.3 OBJETIVOS.....	11
1.3.1. <i>Objetivo general</i> .....	11
1.3.2. <i>Objetivo específico</i> .....	11
1.4 DESCRIPCIÓN DE LA TESIS.....	12
<b>CAPÍTULO 2 ALGORITMOS GENÉTICOS Y LA OPTIMIZACIÓN MULTIOBJETIVO</b> .....	<b>13</b>
2.1 ASPECTOS BÁSICOS DE UN ALGORITMO GENÉTICO.....	13
2.2 OPERADORES GENÉTICOS .....	14
2.3 LOS AG'S EN INGENIERÍA QUÍMICA .....	16
2.4 OPTIMIZACIÓN MULTIOBJETIVO.....	17
2.4.1. <i>El problema multiobjetivo</i> .....	18
2.4.2. <i>Técnicas para la optimización multiobjetivo propuestas</i> .....	21
2.4.3. <i>Optimización multiobjetivo con el algoritmo NSGA-II</i> .....	24
2.5 LA OPTIMIZACIÓN MULTIOBJETIVO CON ALGORITMOS GENÉTICOS EN INGENIERÍA QUÍMICA. ....	25
2.6 EL IMPACTO AMBIENTAL EN LA OPTIMIZACIÓN DE PROCESOS .....	27
<b>CAPÍTULO 3 EFICIENTIZACIÓN DEL ALGORITMO GENÉTICO</b> .....	<b>29</b>
3.1 AJUSTANDO LOS PARÁMETROS DEL ALGORITMO GENÉTICO.....	29
3.1.1. <i>Experimentos computacionales</i> .....	30
3.2 LA CONVERGENCIA EN LOS ALGORITMOS EVOLUTIVOS MULTIOBJETIVO.....	38
3.3 DEFINICIÓN DE UN NUEVO CRITERIO DE CONVERGENCIA PARA EL NSGA-II .....	40
3.4 ANÁLISIS COMPARATIVO ENTRE METODOLOGÍAS DE PARO PARA UN ALGORITMO GENÉTICO. ....	49
<b>CAPÍTULO 4 RESULTADOS NUMÉRICOS</b> .....	<b>51</b>
4.1 ESTABLECIENDO LA COMUNICACIÓN ENTRE EL ALGORITMO Y EL SIMULADOR ASPEN PLUS .....	51
4.2 DEFINICIÓN DE LAS FUNCIONES OBJETIVO .....	52
4.2.1. <i>Cálculo del costo del equipo</i> .....	52



---

4.2.2. Cuantificación del riesgo .....	53
4.3 EXPERIMENTOS Y RESULTADOS .....	55
4.3.1. Problema 1, funciones objetivo sencillas. ....	55
4.3.2. Problema 2, un sistema de separación (Pureza vs Costo). ....	57
4.3.3. Problema 3a, sistema de separación con recirculación (Costo vs Riesgo). ....	63
4.3.4. Problema 3b, sistema de separación con recirculación (Ganancias vs Riesgo). .....	65
4.4 ANÁLISIS DE RESULTADOS.....	66
<b>CAPÍTULO 5 CONCLUSIONES Y TRABAJO FUTURO .....</b>	<b>69</b>
5.1 CONCLUSIONES .....	69
5.2 TRABAJOS FUTUROS.....	69
<b>BIBLIOGRAFÍA .....</b>	<b>71</b>
<b>ANEXO A.....</b>	<b>75</b>
<b>ANEXO B.....</b>	<b>79</b>
<b>ANEXO C.....</b>	<b>81</b>
<b>ANEXO D.....</b>	<b>85</b>

---

## Índice de tablas.

Tabla 3-1 Resultados del problema 1, codificación binaria	31
Tabla 3-2 Resultados del problema 1, codificación real	32
Tabla 3-3 Resultados del problema 8, codificación binaria	33
Tabla 3-4 Resultados del problema 8, codificación real	34
Tabla 3-5 Resultados del problema 14, codificación binaria	36
Tabla 3-6 Resultados del problema 14, codificación real parte 1	37
Tabla 4-1. Subrutinas principales del NSGA-II	51
Tabla 4-2. Comparación entre las soluciones obtenidas con y sin criterio de paro problema 1.	56
Tabla 4-3 Parámetros de ajuste del criterio de paro problema 1	57
Tabla 4-4 Variables de optimización, codificadas como reales	58
Tabla 4-5 Variables de optimización, codificadas como binarias	58
Tabla 4-6 Valores de los operadores genéticos	58
Tabla 4-7. Diseños obtenidos con el algoritmo original. Problema 2	60
Tabla 4-8. Valores de los parámetros del criterio de paro	60
Tabla 4-9. Diseños obtenidos con el criterio de paro implementado	61
Tabla 4-10. Diseños obtenidos variando el parámetro $\Phi=0.60$	62
Tabla 4-11. Valores de los parámetros del criterio de paro	63
Tabla 4-12 Diseño “óptimo” para la optimización de Costo vs. Riesgo	64
Tabla 4-13 Relación pureza-precio de venta para el heptano.	65
Tabla 4-14 Relación pureza-precio de venta para el tolueno	65
Tabla 4-15 Diseños obtenidos para el problema Ganancias vs. Riesgo	66
Tabla D 1 Valores de calor de combustión	85
Tabla D 2. Factores de penalización asignados por TLV	87
Tabla D 3. Categorías del riesgo global R	89

---

---

---

## Índice de figuras.

Figura 1-1 Sistema de separación	10
Figura 1-2. Sistema de separación, segunda opción	10
Figura 2-1 Una población de 4 soluciones, para un problema multiobjetivo	19
Figura 3-1 Diagrama de flujo condensado del criterio de paro	41
Figura 3-2 Ejemplo de la división de la curva de Pareto en intervalos.	43
Figura 3-3 Diagrama de convergencia 1 de 3	46
Figura 3-4 Diagrama de convergencia 2 de 3	47
Figura 3-5 Diagrama de convergencia 3 de 3	48
Figura 4-1 Secuencia de cálculo del índice Mond	54
Figura 4-2 Resultados problema 1, sin criterio de paro después de 20 y de 100 generaciones.	55
Figura 4-3. Resultados problema 1 con criterio de paro después de 17 generaciones	56
Figura 4-4 Sistema de separación propuesto para el problema 2.	57
Figura 4-5 Soluciones obtenidas después de 200 generaciones. Problema 2	59
Figura 4-6. Soluciones obtenidas con el criterio de paro (1)	61
Figura 4-7 Soluciones obtenidas, con parámetro $\Phi=0.60$	62
Figura 4-8 Sistema de separación, segunda opción.	63
Figura 4-9 Soluciones del problema Costo vs Riesgo	64
Figura 4-10 Soluciones del problema Ganancias vs. Índice de riesgo.	66
Figura A 1 Diagrama de flujo del NSGA-II	78
Figura B 1 Diagrama de flujo de la interfase.	79

---

---

# Capítulo 1

## Introducción

### 1.1 Descripción del problema

Durante los últimos años, la industria de procesos viene enfrentando una problemática importante debido al efecto adverso que la emisión de residuos tiene sobre el medio ambiente. Presionada por el surgimiento de legislaciones más estrictas, los ingenieros de proceso se han visto en la necesidad de realizar ajustes en las condiciones de operación considerando dos aspectos, por un lado cumplir con las legislaciones gubernamentales, por otro mantener en un nivel aceptable las ganancias de la empresa.

Ante esta situación se ha considerado como una necesidad el incluir los aspectos ambientales como un criterio de optimización en la síntesis y el diseño de procesos. Tal inclusión implica un aumento en la complejidad del problema desde el punto de vista de modelación como en su solución numérica.

En el aspecto de modelación no se cuenta con alguna forma general que pueda definir y medir el impacto ambiental, por lo cual es necesario buscar en la literatura metodologías establecidas que nos permitan cuantificarlo para posteriormente incluirlo en un problema de optimización.

Desde el punto de vista numérico, minimizar el impacto ambiental del proceso y mantener las ganancias en valores óptimos representan un problema complejo dado que por lo general estos dos objetivos se contraponen, lo que hace difícil el identificar una solución óptima.

### 1.2 Ejemplo motivador

Consideremos el proceso presentado en la figura 1-1, el cual corresponde a un sistema de separación, el cual es utilizado para separar una mezcla (ALIMENT) por medio de destilación extractiva.

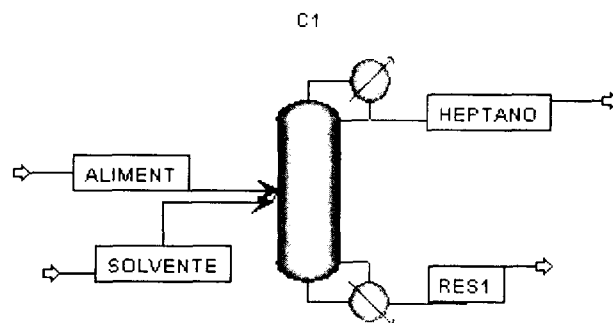


Figura 1-1 Sistema de separación

La corriente que se desea separar está compuesta de heptano y tolueno, el solvente empleado es fenol (SOLVENTE).

Si deseáramos optimizar este sistema bajo dos criterios: maximizar las ganancias y minimizar el impacto ambiental nos damos cuenta que los criterios se contraponen.

La manera más económica de operar el sistema sería desechando el solvente una vez que se logró la separación del producto, pero esto generaría como consecuencia un posible impacto al ambiente, debido a la toxicidad del fenol y a la inflamabilidad del tolueno; reflejándose esto es una penalización para la empresa por parte de las autoridades ambientales.

Una segunda opción para realizar la separación sería agregando al sistema otra columna de destilación para separar de los fondos (RES1) de la primera columna el tolueno del fenol (para posteriormente venderlo como producto) y recircular el solvente utilizado, como alimentación de la primera columna, figura 1-2.

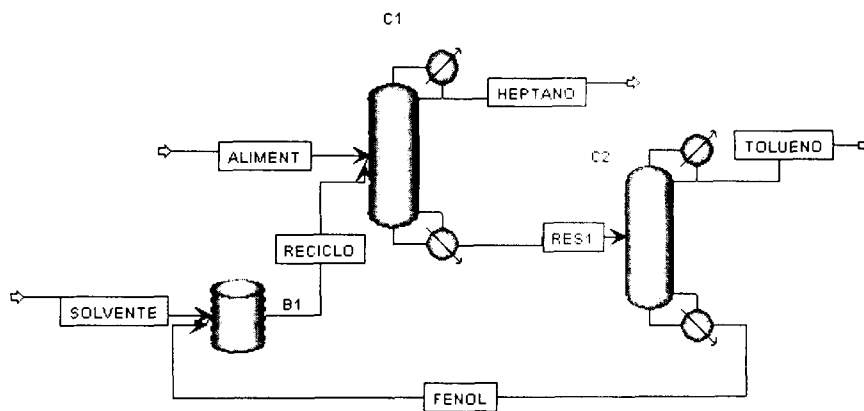


Figura 1-2. Sistema de separación. segunda opción

Esto traería como beneficios una disminución en gastos de solvente y la reducción del impacto sobre el ambiente por disposición de desechos, pero aparecerían gastos de inversión por el equipo adquirido que posiblemente pueden ser recuperados por la venta del segundo producto.

Para el ejemplo planteado, surge la pregunta de ¿cuál de las dos opciones es la más conveniente considerando el criterio económico y el ambiental?

Evidentemente no existe una solución óptima única, por lo que es necesario recurrir a una optimización multiobjetivo por medio de la cual podamos obtener un conjunto de soluciones a través de las cuales se puedan visualizar los efectos que tiene el tomar una u otra decisión.

Actualmente existen en la literatura diversas técnicas de optimización multiobjetivo que han sido implementadas para la solución de problemas multicriterio, que a menudo se presentan en ingeniería química.

Una herramienta útil en el desarrollo de las técnicas multiobjetivo son los algoritmos genéticos. Los cuales están basados en el principio de la evolución natural.

Otro recurso de gran utilidad en la optimización de procesos es la simulación. Se han desarrollado diversos simuladores comerciales que ayudan al ingeniero de procesos en el modelado de las operaciones unitarias mediante técnicas matemáticas complejas. Estos simuladores comúnmente han seguido una estrategia de solución modular secuencial es decir, cada operación unitaria se modela de manera independiente. Se han reportado trabajos en los que se propone el acoplamiento de un algoritmo genético en un simulador de este tipo, obteniéndose resultados satisfactorios.

Lo anterior nos da la pauta para acoplar una técnica multiobjetivo en un simulador de procesos y con esto, determinar las alternativas existentes en la solución de problemas de optimización en los que se desea incluir como criterios la maximización de ganancias y la minimización del impacto ambiental.

## **1.3 Objetivos**

### ***1.3.1. Objetivo general***

De acuerdo a los antecedentes el objetivo general de la tesis es:

- Integración de una técnica de optimización multiobjetivo con algoritmos genéticos con el simulador de procesos Aspen Plus para la solución de problemas de optimización en procesos de separación.

### ***1.3.2. Objetivo específico***

- Desarrollar una metodología para eficientizar el desempeño del algoritmo genético.
- Aplicar esta metodología a problemas de optimización de procesos considerando aspectos económicos y de riesgo simultáneamente.



## 1.4 Descripción de la tesis

El resto de este trabajo se describe a continuación:

En el capítulo 2 se presentan los aspectos generales de los algoritmos genéticos: mecanismo básico, la representación binaria y real, los operadores genéticos selección, cruce y mutación, se describen brevemente algunas aplicaciones de los algoritmos genéticos en ingeniería química. Posteriormente se aborda el problema multiobjetivo, conceptos importantes como frente de Pareto y “dominación”. Se incluye una revisión bibliográfica con las técnicas multiobjetivo propuestas en la literatura para dar paso a la descripción del algoritmo implementado en la realización de este trabajo. En la parte final se muestra la definición de los medios de cuantificación del impacto ambiental para su inclusión como función objetivo en un problema de optimización multicriterio.

En el capítulo 3 se presenta la solución de una serie de problemas típicos de optimización presentados en la literatura. Esto con el fin de determinar cual sería el valor adecuado (o el rango de valores adecuados) de los operadores del algoritmo genético utilizado. Posteriormente se presenta una metodología para reducir la complejidad en la solución de problemas multiobjetivo, descrita en la literatura. Seguido se propone un proceso de clasificación de los individuos en intervalos según sus aptitudes; por medio del cual se busca la definición de un criterio de convergencia (o paro) en el algoritmo utilizado.

El capítulo 4 presenta la implementación de la estrategia multiobjetivo seleccionada en el simulador de procesos Aspen Plus. Se describe el funcionamiento de la interfase y la formulación de las funciones objetivo para el problema de optimización. Posteriormente esto es aplicado a un proceso industrial con el que se hacen una serie de experimentos con el fin de observar el desempeño del algoritmo genético ante diferentes escenarios. Al final se presentan los resultados obtenidos.

En el capítulo 5 se exponen las conclusiones obtenidas después de la realización de este trabajo, así como el trabajo futuro.

## Capítulo 2

### Algoritmos genéticos y la optimización multiobjetivo

En este capítulo se describen los fundamentos teóricos de este trabajo, los cuales abarcan: los aspectos generales de un algoritmo genético, la optimización multiobjetivo, las diferentes técnicas de optimización multiobjetivo propuestas en la literatura así como también el funcionamiento del algoritmo genético implementado y por último la inclusión del aspecto ambiental como criterio en la optimización de procesos industriales.

#### 2.1 Aspectos básicos de un algoritmo genético

Durante la última década, los algoritmos genéticos han sido ampliamente utilizados como herramientas de búsqueda y optimización en una gran gama de problemas en áreas como la ingeniería, las ciencias y el comercio. La razón principal de su éxito es debido a su gran aplicabilidad, su facilidad de uso y su perspectiva global. El concepto de algoritmo genético fue propuesto por primera vez en 1975 por John Holland [22]

Holland mostró que las soluciones potenciales de un problema de búsqueda pueden ser codificadas usando representaciones simples: cadenas de bits por ejemplo, imitando así el papel que juegan los genes. Un número determinado de tales cadenas (cromosomas) *individuos* forman la población del algoritmo genético.

La calidad de un individuo se define a través de una medida llamada *aptitud*. Conforme el algoritmo genético evoluciona, un conjunto de operadores genéticos básicos se aplican iterativamente a la población entera para producir nuevas poblaciones con mejores individuos. Después de un número de generaciones, éste proceso nos llevará a tener una población con individuos altamente aptos, es decir, a estructuras que representan buenas soluciones del problema que se quiere resolver. El individuo más apto de la generación final deberá ser una solución óptima o cercana a la óptima del problema dado [11]

El mecanismo básico de un algoritmo genético es el siguiente:

- Primero: se genera una población inicial de *individuos (cromosomas)* para el algoritmo genético.
- Segundo: para cada *cromosoma* de la población se evalúa la función de aptitud.
- Tercero: se aplican los operadores genéticos de reproducción, cruce y mutación para crear la población de *cromosomas* de la nueva generación.

Los pasos segundo y tercero se repiten hasta que se satisface la condición de convergencia (o paro) establecida por el usuario.

#### ***Mecanismo de representación (codificación)***

Consiste en elegir una manera de representar una solución del problema real como una cadena de longitud fija mediante un alfabeto específico. La codificación binaria (cadenas de bits [0,1]) es la representación más utilizada. Esta cadena es lo que se denomina *cromosoma*. Cuando un algoritmo genético que utiliza codificación binaria para

representar a las variables de solución, necesita ser aplicado para resolver un problema que tiene un espacio de búsqueda continuo, surgen algunas dificultades. Una de éstas es la aparición de discontinuidades matemáticas (Hamming Cliffs) asociadas con ciertos tipos de cadenas ( como son 01111 y 10000) para las cuales un pequeño cambio en el espacio de búsqueda real, requiere de la alteración de muchos bits. Estas “discontinuidades matemáticas” presentes en la codificación binaria representan una especie de obstáculo artificial para lograr una búsqueda gradual en un espacio continuo. La otra dificultad es la incapacidad de alcanzar cierta precisión en la solución óptima. Para un algoritmo genético con codificación binaria, la longitud del cromosoma debe ser escogida con anticipación para permitir que el algoritmo alcance cierta precisión en la solución. Entre más grande sea la precisión requerida, más grande será el cromosoma. Para cromosomas grandes, es necesario que la población sea grande [69], en consecuencia la complejidad computacional del algoritmo aumenta. A partir de que es usado un tamaño de cromosoma fijo en la representación de las variables de decisión, los límites de éstas deben ser tales que abarquen los valores óptimos de las variables. Usualmente en la solución de problemas, este tipo de información no se conoce con anticipación lo cual es una desventaja.

En contraste, existe una serie de implementaciones donde se utiliza la codificación real para representar las variables de optimización en un algoritmo genético, donde los operadores de cruce y mutación son aplicados directamente a los parámetros reales (sin estar contenidos en algún tipo de cadena).

### ***La población inicial***

La selección de la población inicial de un algoritmo genético usualmente se hace de manera aleatoria, con el objetivo de seleccionar soluciones de todas partes del espacio de búsqueda. Otra manera de seleccionar la población es la inicialización directa, cuando algunos aspectos de una buena solución se conocen de antemano.

### ***El mecanismo de evaluación***

Consiste en evaluar la *función aptitud*. Esta función devuelve un valor que denota que tan buena es la solución del cromosoma evaluado para el problema en cuestión. En un problema de optimización, la función de aptitud corresponde a la función objetivo que debe ser optimizada.

## **2.2 Operadores genéticos**

### ***Selección***

Operador que simula el proceso de selección natural del más apto. Los individuos se copian de una generación a la siguiente de acuerdo a su valor de aptitud. Los métodos tradicionales de selección son *el de rueda de ruleta* y *el de selección por torneo*; éste último tiene similitud con el tipo de competencia que ocurre en la naturaleza.

### ***Cruce***

Este mecanismo trabaja con un par de individuos (cromosomas) previamente seleccionados y cuyo fin es el intercambio de información entre éstos. Existen tres tipos de cruce: el *cruce simple*, en la cual se escoge un lugar al azar a lo largo del cromosoma, luego los genes de los 2 individuos se intercambian a partir del lugar seleccionado; el *cruce de dos puntos* que

a diferencia del simple se seleccionan dos puntos al azar en el cromosoma y el intercambio de material genético de un individuo a otro se hace en la selección delimitada por los puntos seleccionados; el *cruce uniforme* el cual consiste en un intercambio gen por gen a lo largo del cromosoma con una probabilidad predefinida.

Para el caso de la implementación de parámetros reales se han desarrollado operadores de cruce, resalta el trabajo de Deb et al [13] quienes desarrollaron el operador de cruce binario simulado (SBX por sus siglas en inglés), el cual trabaja con un par de soluciones “padres” ( $x_i^{(1,t)}$  y  $x_i^{(2,t)}$ ) y crea un par de soluciones “hijas” ( $x_i^{(1,t+1)}$  y  $x_i^{(2,t+1)}$ ) [4,5].

Como su nombre lo indica, este operador simula el principio del operador de cruce de un solo punto usado para la codificación binaria de las variables de decisión. A continuación se muestra el procedimiento de la creación de las soluciones “hijas”:

**Paso 1:** Seleccionar un número aleatorio  $u_i \in [0,1)$

**Paso 2:** Calcular  $\beta_i$

$$\text{Si } u_i \leq 0.5 \quad \beta_i = (2u_i)^{\frac{1}{\eta_c+1}}$$

$$\text{SI } u_i \geq 0.5 \quad \beta_i = \left( \frac{1}{2(1-u_i)} \right)^{\frac{1}{\eta_c+1}}$$

**Paso 3:** Calcular las soluciones resultantes por medio de las siguientes ecuaciones:

$$x_i^{1,t+1} = 0.5 \left[ (1 + \beta_i) x_i^{(1,t)} + (1 - \beta_i) x_i^{(2,t)} \right]$$

$$x_i^{2,t+1} = 0.5 \left[ (1 + \beta_i) x_i^{(2,t)} + (1 - \beta_i) x_i^{(1,t)} \right]$$

donde

$i$  puede tomar cualquier valor entre 1 y el tamaño de la población.

$\eta_c$  es un índice de distribución, el cual puede tomar el valor de cualquier número real no negativo. Con un valor grande de  $\eta_c$  existe una alta probabilidad de crear una solución “hija” cercana a la solución “padre”, en caso contrario existe la probabilidad de que la solución resultante sea distante a sus “padres”

$\beta_i$  función de distribución probabilística utilizada en el cruce

$x_i^{1,t+1}$  representa a la solución hija 1 obtenida a partir de los padres  $x_i^{(1,t)}$  y  $x_i^{(2,t)}$

$x_i^{2,t+1}$  representa a la solución hija 2 obtenida a partir de los padres  $x_i^{(1,t)}$  y  $x_i^{(2,t)}$

### Mutación

Usualmente es aplicado a cada individuo después del cruce. Consiste en modificar aleatoriamente el contenido genético de un individuo, esto con la finalidad de mantener la diversidad de la población. Con relación a la aplicación de este parámetro en variables de optimización que fueron codificadas en forma real Deb propone el siguiente procedimiento para crear un nuevo individuo  $y_i^{(1,t+1)}$  a partir del espacio de búsqueda:

**Paso 1:** Seleccionar un número aleatorio  $r_i \in [0,1)$

**Paso 2:** Calcular  $\bar{\delta}_i$

632745

$$\text{Si } r_i \leq 0.5 \quad \bar{\delta}_i = (2r_i)^{\frac{1}{(\eta_m+1)}} - 1$$

$$\text{SI } r_i \geq 0.5 \quad \bar{\delta}_i = 1 - [2(1 - r_i)]^{\frac{1}{(\eta_m+1)}}$$

**Paso 3:** Calcular la solución resultante por medio de:

$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + (x_i^{(U)} - x_i^{(L)})\bar{\delta}_i$$

donde

$i$  puede tomar cualquier valor entre 1 y el tamaño de la población.

$x_i^{(U)}$  y  $x_i^{(L)}$  son los límites superior e inferior de la variable de decisión.

$\eta_m$  es un índice de distribución, el cual puede tomar el valor de cualquier número real no negativo. En el capítulo 3, se presenta un estudio, por medio del cual se busca determinar un valor adecuado para este parámetro.

$\bar{\delta}_i$  función de distribución probabilística utilizada en la mutación

$y_i^{(1,t+1)}$  es la solución obtenida después de la mutación.

Otros trabajos relacionados con estos operadores han sido presentados por Wright (linear crossover)[6], Goldberg (blend crossover )[7], Voigt et al (fuzzy recombination operator)[8], Ono y Kobayashi (unimodal normally distributed crossover)[9], Michalewicz (mutación aleatoria y mutación no-uniforme)[10]

### 2.3 Los AG's en ingeniería química

Poco a poco los ag's se han ido introduciendo en el campo de la ingeniería química. Existe una gran variedad de trabajos publicados en los cuales los algoritmos genéticos han sido utilizados como herramientas de optimización.

En la gran mayoría de éstos se busca la minimización del costo, ya sea de diseño y/o de operación, como es el caso de Androulakis y Venkatasubramanian [66] quienes utilizaron un ag para sintetizar redes de intercambio de calor minimizando su costo. Otro ejemplo es el trabajo presentado por Mukherjee et al [67] en el que utilizan un AG para la optimización de un sistema de destilación batch industrial, con lo cual lograron una reducción en el costo de operación.

Otro punto importante son los esfuerzos que se han realizado por mejorar al algoritmo genético tradicional, buscando la disminución del tiempo máquina requerido para alcanzar la solución óptima. Algunos trabajos relacionados son: el presentado por Wang et al [68] el cual propone un algoritmo genético mejorado (IGA por sus siglas en inglés) como estrategia para la optimización de sistemas de destilación usando codificación real en las variables de decisión. Dicho algoritmo divide a la población en un número de subpoblaciones, cada subpoblación evoluciona independientemente tratando de localizar un óptimo local. El cruce se realiza entre dos individuos de dos sub poblaciones; los individuos resultantes compiten por la sobre vivencia dentro de su respectiva población. En los resultados obtenidos se demuestra que el IGA trabaja más eficientemente ya que encuentra un mejor diseño con bajo costo y en menor tiempo máquina.

De forma similar Montesinos et al [referencia] proponen un algoritmo genético modificado para la optimización de una red de distribución de agua. Las modificaciones al algoritmo genético son en los procesos de selección y de mutación. Los autores reportan que resultados obtenidos al aplicar el algoritmo propuesto fueron satisfactorios ya que lograron obtener la solución factible de más bajo costo reportada en la literatura en un número menor de generaciones que un algoritmo genético tradicional.

Recientemente Leboreiro [49,50] presenta un algoritmo genético para el diseño óptimo de columnas de destilación acoplado a un simulador de procesos comercial (Aspen Plus) en el cual se implementaron algunas estrategias para mejorar su desempeño; estas consisten en la selección de valores adecuados de los parámetros del algoritmo, el uso de un algoritmo híbrido (algoritmo genético con resolvidor matemático) y un nuevo procedimiento para la definición de la convergencia del algoritmo. La implementación computacional de éstas propuestas permitió reducir el tiempo de ejecución en más del 50% en algunos problemas numéricos, manteniendo la robustez del algoritmo para encontrar la solución óptima. A través de el criterio de convergencia lograron encontrar la solución óptima de los problemas y detener el AG una vez que el óptimo ha sido encontrado. Los resultados obtenidos demuestran las ventajas de utilizar los algoritmos genéticos acoplados a simuladores de procesos para facilitar la modelación de sistemas de destilación convencionales.

## 2.4 Optimización multiobjetivo

El término de optimización multiobjetivo (también llamada optimización multicriterio) involucra la presencia de dos o más objetivos para los cuales es necesario encontrar una solución óptima.

La optimización multiobjetivo evolutiva (EMOO por sus siglas en inglés) se presenta como una herramienta adecuada para la solución problemas multiobjetivo ya que involucra simultáneamente un conjunto de posibles soluciones (llamadas población) del problema en estudio.

Existen varias versiones de este conjunto de soluciones, dependiendo de la técnica multiobjetivo que se esté aplicando, por ejemplo, en la técnica llamada Funciones de agregación, estas soluciones son el valor de los coeficientes finales (pesos) asignados a cada objetivo. Por otro lado en la técnica NSGA, estas soluciones son las llamadas *soluciones de Pareto*.

En principio, la optimización multiobjetivo es muy diferente a la optimización de un solo objetivo; en ésta la obtención del mejor diseño o de la mejor decisión es usualmente el máximo o mínimo global (dependiendo del tipo del problema)

En un problema típico de optimización multiobjetivo, existe un conjunto de soluciones las cuales son superiores al resto dentro del espacio de búsqueda cuando todas las funciones objetivo son consideradas. Estas soluciones son conocidas como *soluciones de Pareto* o *soluciones no dominadas*; el resto de las soluciones son conocidas como *soluciones dominadas*.

Ninguna de las soluciones del conjunto de las *no dominadas* es absolutamente mejor que otra, cualquiera de ellas es aceptable. La elección de alguna de ellas requiere conocimiento

del problema, por lo que una solución escogida por el ingeniero de proceso puede ser aceptada o no por otro o en circunstancias diferentes.

Por lo anterior podemos resumir los objetivos de la optimización multiobjetivo en [13]:

1. Encontrar un conjunto de soluciones tan cercanas como sea posible al frente óptimo de Pareto
2. Encontrar un conjunto de soluciones tan diversas como sea posible.

El primero es obligatorio en cualquier tarea de optimización (de un objetivo o de varios) ya que un conjunto de soluciones convergiendo lejos del óptimo no es deseable.

El segundo objetivo es específico de la optimización multiobjetivo; ya que además de que es preferible que las soluciones estén lo más cercanas posibles al óptimo de Pareto, es necesario que estén distribuidas a lo largo de éste.

Con un conjunto de soluciones diverso, podemos estar seguros de que se tiene un buen conjunto de soluciones que forman las llamadas curvas de "trade-off" o curva de Pareto entre los objetivos.

#### 2.4.1. El problema multiobjetivo

De manera general un problema multiobjetivo puede representarse como sigue:

$$\text{Minimizar/Maximizar } f_m(x) \quad m = 1, 2, \dots, M;$$

$$\text{Sujeto a } g_j(x) \geq 0, \quad j = 1, 2, \dots, J;$$

$$h_k(x) = 0, \quad k = 1, 2, \dots, K;$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, N;$$

Una solución  $x$  es un vector de  $n$  variables de decisión:  $x = (x_1, x_2, \dots, x_n)^T$ . El último conjunto de restricciones son conocidas como los límites de las variables, las cuales restringen a cada una de las variables de decisión  $x_i$  para que tomen un valor entre el límite inferior  $x_i^{(L)}$  y el límite superior  $x_i^{(U)}$ . Estos límites constituyen el espacio de variables de decisión  $D$ . Asociados al problema están las restricciones de desigualdad  $J$  y las de igualdad  $K$ . La solución  $x$  que no satisface el conjunto de restricciones  $(J + K)$  y que está fuera de los  $2N$  límites es llamada *solución no factible*, en caso contrario tenemos una solución factible.

#### **El concepto de Dominación**

La mayoría de los algoritmos de optimización multiobjetivo manejan el concepto de dominación. En estos algoritmos, dos soluciones son comparadas para determinar si una domina a la otra o no utilizando el operador definido como  $\prec$  [13].

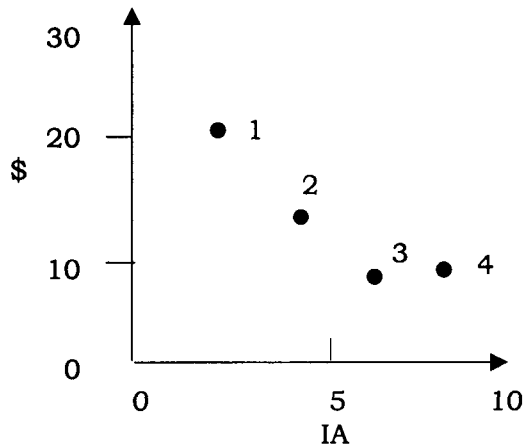
- $i \triangleleft j$  denota que la solución  $i$  es mejor que la solución  $j$  en un objetivo en particular.
- $i \triangleright j$  denota que la solución  $i$  es peor que la solución  $j$  en un objetivo en particular.

entonces se dice que la solución  $x^{(1)}$  domina a la solución  $x^{(2)}$  si se cumplen las siguientes condiciones:

1. La solución  $x^{(1)}$  no es peor que la solución  $x^{(2)}$  en todos los objetivos,  $f_j(x^{(1)}) \not\geq f_j(x^{(2)})$  para todo  $j = 1, 2, \dots, M$ , donde  $M$  es el número de funciones objetivo.
2. La solución  $x^{(1)}$  es estrictamente mejor que  $x^{(2)}$  en al menos un objetivo,  $f_j(x^{(1)}) \triangleleft f_j(x^{(2)})$  para al menos un  $j \in \{1, 2, \dots, M\}$

Si alguna de las condiciones no se cumple, la solución  $x^{(1)}$  no domina a la solución  $x^{(2)}$ .

Para un número determinado de soluciones, es posible realizar todas las posibles comparaciones y encontrar cuales son las soluciones dominadas y las no dominadas. Para ejemplificar esto supongamos que tenemos un conjunto de soluciones (figura 2.1) para un problema en el cual se desea minimizar el impacto ambiental y por otro lado mantener las ganancias en un estado óptimo.



**Figura 2-1** Una población de 4 soluciones, para un problema multiobjetivo

Observamos que la solución 1 es mejor que la solución 2 y la 3 con respecto al criterio económico, pero con respecto al criterio ambiental, la solución 3 es mejor que la 1 y la 2. En consecuencia, ninguna de las tres soluciones satisfacen las dos condiciones para ser dominadas, por lo tanto las soluciones son no-dominadas. La solución número 4 es una solución dominada por la número 3, ya que cumple con las dos condiciones antes mencionadas.



### La diversidad en la población

Un algoritmo genético tiende a converger hacia una solución única debido al tamaño finito de la población, lo que es deseable cuando se quiere alcanzar un óptimo global o cercano a él. Sin embargo, en la optimización multiobjetivo se requiere encontrar el frente de Pareto (conjunto de soluciones), por lo que es necesario mantener la diversidad en las soluciones durante el proceso evolutivo para evitar que el algoritmo genético converja hacia una solución única, para atacar este problema han surgido las siguientes propuestas [2]:

- Cavicchio [14] introdujo un mecanismo llamado *preselección*, el cual consistía en que un hijo reemplazara a su padre, si la aptitud del hijo era mayor; de esta manera se preservaba la diversidad de la población. El tamaño de la población utilizada por Cavicchio era relativamente pequeño ( $n=20$ ).
- Holland [1] propone el uso de un operador de similitud y cercanía (*crowding*) el cual fue implementado por Dejong [28]. Esta idea permitía que el descendiente de un individuo reemplace a aquél con el cual tenga mayor similitud a nivel genotípico.
- Goldberg et al [16] proponen dividir la población en subconjuntos (nichos) de acuerdo a su similitud, lo que involucra la existencia de un parámetro que establezca el grado de similitud entre los individuos. Los individuos que pertenecen al mismo nicho compartirán su aptitud por medio de una función de distribución de manera que disminuya; de esta manera se premia a los individuos que son diferentes y se limita el crecimiento de especies dentro de la población. Para determinar la aptitud de un individuo que se encuentra en un nicho determinado se divide ésta entre un contador de nicho  $m_i$ , este parámetro estima que tan poblado se encuentra el nicho  $i$ . Se calcula de la siguiente manera [2,19]:

$$m_i = \sum_{j=1}^{npopsize} sh(d_{i,j})$$

donde

- $d_{ij}$  es la distancia entre los individuos  $i$  y  $j$
- $sh(d_{i,j})$  es la función de distribución de aptitud (sharing function), la cual es una función decreciente de  $d_{ij}$  de manera que  $sh(0) = 1$  y  $sh(d \leq \sigma_{SHARE}) = 0$
- $\sigma_{SHARE}$  es el radio del nicho escogido por el usuario, los individuos que se encuentren dentro del radio degradarán su aptitud entre si. La función de distribución de aptitud mas utilizada es la triangular (triangular sharing function) [2]

$$sh(d_{i,j}) = \begin{cases} 1 - \left( \frac{d_{i,j}}{\sigma_{Share}} \right) & \text{Si } d_{i,j} < \sigma_{Share} \\ 0 & \text{Si } d_{i,j} \geq \sigma_{Share} \end{cases}$$

### 2.4.2. Técnicas para la optimización multiobjetivo propuestas

Las técnicas más populares para el manejo de la optimización multiobjetivo con ag's son [17,18]:

- Funciones de agregación
- El algoritmo genético del vector evaluado de Schaffer (VEGA)
- El algoritmo genético multiobjetivo de Fonseca y Fleming (MOGA)
- El algoritmo genético de los no dominados ordenados de Srinivas y Deb (NSGA)
- El algoritmo genético de nido compartido de Horn y Nafpliotis (NPGA)

A continuación se describen brevemente cada una de ellas:

#### **Funciones de agregación**

La forma intuitiva de manejar varios objetivos es la de combinarlos en uno solo y con ello obtener un único valor de aptitud, la manera de unirlos puede ser por medio de una operación aritmética como puede ser la suma o la multiplicación. Un ejemplo de esta técnica consiste en sumar todas las funciones objetivo utilizando diferentes pesos como coeficientes para cada una [1,6]:

$$\min \sum_{i=1}^k w_i f_i(\bar{x}) c_i$$

Donde  $w_i \geq 0$  son los coeficientes que representan la importancia de cada función objetivo y  $c_i$  son factores de escalamiento de los objetivos. Con esto se mantienen los valores numéricos de cada función objetivo similares.

Los pesos se normalizan de manera que  $\sum_{i=1}^k w_i = 1$

La principal ventaja de esta técnica es su eficiencia computacional y su utilidad para generar una solución inicial para otras técnicas. La desventaja radica en la dificultad para determinar los pesos correctos que representen adecuadamente las funciones objetivo cuando no se tiene suficiente información acerca del problema; además sólo nos proporciona un punto la curva de Pareto por lo que no se tendría información sobre el cambio en esta curva, lo cual es necesario para la toma de decisiones. Algunas aplicaciones de esta técnica se describen en los trabajos de W. Jakob [20], G. Jones [21], J. Lis [22], G. Syswerda [23].

#### **El algoritmo genético de vector evaluado de Schaffer (VEGA)**

David Schaffer[17] extiende el uso del algoritmo genético simple en su propuesta de algoritmo genético de vector evaluado, el cual se distingue por su manera de aplicar el operador de selección. En cada generación, un número de subpoblaciones es generado por medio de una selección proporcional de acuerdo a la función objetivo en turno. Así para un problema con “k” objetivos, “k” subpoblaciones de tamaño “n/k” serán generadas (donde n

es el número total de individuos de la población), esas subpoblaciones serán intercambiadas para obtener una nueva población de tamaño  $n$ , sobre la cual el algoritmo genético aplicará los operadores de cruce y mutación de manera usual. De esta manera se trata de elegir individuos que sean no dominados de manera local (en un objetivo), buscando que se combinen con otros que tengan las mismas características pero con otro objetivo, tratando con esto el obtener soluciones no dominadas a nivel global y formar el frente de Pareto correspondiente. Algunos autores que han utilizado esta técnica son: Ritzel [24], Cvetkovic [25], Tamaki [26], H. Tamaki y M. Mori [27].

### ***El algoritmo genético multiobjetivo de Fonseca y Fleming (MOGA)***

El algoritmo propuesto por estos autores difiere del algoritmo genético convencional en la manera en que es asignada la aptitud a cada solución en la población. Su funcionamiento se describe a continuación:

1. Como primer paso se lleva a cabo una revisión para determinar si las soluciones son dominadas o no. A cada solución ( $i$ ), se le asigna un rango que es igual a el número de soluciones ( $n_i$ ) que dominan a la solución  $i$  más uno, esto es:  $r_i = 1 + n_i$
2. Los individuos *no dominados* tienen un rango de 1 mientras que los individuos dominados son penalizados dependiendo de la densidad de población en la región.
3. La población es ordenada por rangos en forma ascendente.
4. Es asignada una aptitud a cada individuo interpolando a partir de la aptitud del individuo de menor rango (rank=1) hasta la aptitud del individuo de mayor rango, siguiendo la propuesta de Goldberg [11], de acuerdo a alguna función usualmente lineal.
5. Se obtiene un promedio de la aptitud de los individuos con el mismo rango. Esto mantiene constante la aptitud global de la población.

De acuerdo con lo reportado por Goldberg y Deb [28], la manera en que el *MOGA* asigna la aptitud produce una fuerte presión de selección lo que trae como consecuencia la convergencia prematura. Para evitarlo Fonseca y Fleming [29] usan la formación de *nichos* para distribuir la población a lo largo de la región óptima de Pareto, para dividir la aptitud de los individuos cercanos utilizan el valor de las funciones objetivo en lugar de los valores de las variables de decisión. Las ventajas de esta técnica son su eficiencia y sencillez para implementarse. Sus desventajas: depende en gran parte del valor que se le sea asignado al parámetro  $\sigma_{Share}$  y no pueden existir al mismo tiempo dos soluciones diferentes con los mismos valores aptitud, ya que esto puede traer como consecuencia una tendencia hacia ciertas soluciones de espacio de búsqueda.

Algunos trabajos realizados con esta técnica son: optimización de controladores [30], diseño de un sistema de control multivariable para una turbina de gas [31] y como base para desarrollar la programación genética multiobjetivo (MOGP) de Fonseca et al[32].

### ***El algoritmo genético de ordenamiento de individuos no-dominados de Srinivas y Deb (NSGA)***

Esta técnica varía del algoritmo genético usual debido a que antes de la selección genera varios niveles de clasificación mientras que los operadores de cruce y mutación se aplican de manera normal. Estos niveles de clasificación se generan bajo el criterio de *no-dominancia*, el procedimiento es el siguiente [17]:

1. Los individuos no dominados son clasificados en una categoría en donde se les asigna una aptitud proporcional al tamaño de la población, esto con el fin de igualar la capacidad de reproducción de ese conjunto de individuos.
2. Con el fin de mantener la diversidad se comparte la aptitud de ese grupo de individuos mediante el uso de *nichos*.
3. Este grupo de individuos es separado temporalmente.
4. Con los individuos restantes (de la población original), se genera un nuevo grupo de soluciones no dominadas a las cuales se les asigna un valor de aptitud temporal menor que al del primer grupo.
5. Repetir los pasos 1, 2 y 3 hasta que todos los individuos estén clasificados.
6. Realizar proceso de selección estocástica.
7. Aplicar los operadores de cruce y selección.
8. Detener cuando se haya alcanzado el criterio de paro.

Debido a que los individuos con jerarquías más altas adquieren los máximos valores de aptitud y a su vez tienen las mejores posibilidades de ser seleccionados esto conduce hacia una rápida convergencia de la población hacia regiones no dominadas.

Esta técnica ha resultado ser eficiente debido a que los objetivos múltiples se reducen a un valor temporal de aptitud el cual jerarquiza a la población en subconjuntos de individuos. En la literatura se ha reportado que esta técnica es más sensible al valor elegido del factor  $\sigma_{Share}$  que el MOGA [33].

Algunos autores que han aplicado esta técnica son: Michielssen et al [19] para el diseño de sistemas electromagnéticos, Bhaskar et al [34] en la optimización de un reactor industrial.

### ***El algoritmo genético de nicho compartido de Horn y Nafpliotis (NPGA)***

Horn y Nafpliotis [30] proponen usar la selección por torneo considerando el criterio de no-dominancia. En este caso el torneo no se limita a dos individuos como en el torneo normal, sino que se selecciona un subconjunto de individuos  $t_{dom}$  al principio de cada proceso de selección y contra éste compiten dos individuos que serán seleccionados aleatoriamente. El ganador será aquel que no sea dominado por el subconjunto seleccionado, en caso de que ambos resulten ser *no dominados* o *dominados* un contador de nicho (niche count) para cada miembro de la población será determinado.

Este se calcula por medio del conteo de los individuos que se encuentren a cierta distancia ( $\sigma_{Share}$ ) de un individuo específico; el individuo con el contador de nicho menor será seleccionado. El éxito de esta técnica depende en gran parte del parámetro  $t_{dom}$ , ya que si el tamaño elegido no es el apropiado, será difícil localizar a los individuos no dominados. Si  $t_{dom}$  es muy pequeño, podría resultar en un número pequeño de individuos no dominados; por el contrario si  $t_{dom}$  es muy grande puede presentarse el problema de convergencias prematura.

Esta técnica tiene la ventaja de ser más rápida en su desempeño ya que no clasifica a toda la población sino que utiliza un subconjunto de ella. Una desventaja es que depende en gran medida de los parámetros  $t_{dom}$  y  $\sigma_{Share}$ . Algunos trabajos realizados con esta técnica son: diseño de compuestos para cerámica laminada por Belegundu et al [35] y Poloni et al [36] reportan el uso de esta técnica en el diseño de una lamina aérea.

### 2.4.3. Optimización multiobjetivo con el algoritmo NSGA-II

En esta sección se describen las variaciones entre la primera y la segunda versión del NSGA. El motivo de su elección se debió a que en la búsqueda de algoritmos genéticos que manejaran técnicas de optimización multiobjetivo, este algoritmo permite la codificación real y binaria para las variables de decisión, además de que el código completo está disponible para propósitos de investigación [37].

A partir del algoritmo propuesto por Deb y Srinivas [17] presentado en la sección anterior, han surgido diversas críticas en varios aspectos de su funcionamiento.

El primer punto tiene que ver con la complejidad computacional que requiere en el proceso de ordenamiento de los individuos para cada frente, ya que aproximadamente requiere de  $mN^3$  comparaciones (donde M es el número de objetivos y N el número de individuos) entre los individuos para identificar a los no-dominados.

Otro aspecto es que el NSGA forma parte de los algoritmos evolutivos que no cuentan con un mecanismo explícito para conservar a los mejores individuos de cada generación, es decir es un algoritmo no elitista.

Por último el NSGA requiere de la especificación del parámetro de compartición  $\sigma_{share}$ , por medio del cual es posible mantener la diversidad en la población. El NSGA-II surge como un algoritmo evolutivo elitista propuesto por Deb et al [38] con diversas modificaciones buscando solucionar los puntos anteriores. A continuación se describen brevemente éstas modificaciones, para una explicación más detallada el lector puede referirse a [13,38]

Para disminuir la complejidad computacional relacionada con el ordenamiento de los individuos en cada frente, esta nueva versión propone un procedimiento de ordenamiento rápido donde el número de comparaciones se reduce a  $mN$ . En este nuevo procedimiento los individuos que son seleccionados en el primer frente (rango 1) son considerados como los mejores (no dominados) Tales individuos serán los primeros en ser seleccionados para formar parte de la siguiente generación. De esta manera se introduce el concepto de elitismo.

A diferencia de su antecesor el NSGA-II cuenta con un operador de selección por torneo de amontonamiento (*Crowded tournament selection operator*,  $\leq_c$ ) Asumiendo que cada individuo  $i$  tiene los siguientes atributos:

- Un rango de no- dominación  $r_i$  dentro de la población
- Una distancia de amontonamiento (*crowding distance*)  $d_i$  dentro de la población

donde la distancia de amontonamiento  $d_i$  de una solución  $i$  es una medida de cuantas soluciones rodean a la solución  $i$  en el espacio de búsqueda, entonces si dos individuos entran al torneo  $(i, j)$  el operador  $<_c$  declara ganadora a la solución  $i$  si cualquiera de las siguientes condiciones es verdadera:

1. La solución  $i$  tiene mejor rango, esto es  $r_i < r_j$
2. Ambas soluciones tienen el mismo rango pero la distancia de amontonamiento de la solución  $i$  es mejor que la de  $j$ , esto es  $r_i = r_j$  y  $d_i > d_j$

La primera condición nos asegura que la solución escogida se encuentra en un mejor rango. La segunda nos permite escoger de dos soluciones que se encuentren en el mismo rango, la que tiene a la distancia más grande, es decir la que se encuentra en un área con menos densidad de población.

Por medio del operador anteriormente descrito se mantiene la diversidad de la población; sustituyendo con esto al parámetro de compartición  $\sigma_{share}$ , utilizado por el NSGA inicialmente propuesto [68].

En el Apéndice A se describe el funcionamiento del algoritmo genético utilizado para la realización de este trabajo.

## 2.5 La optimización multiobjetivo con algoritmos genéticos en Ingeniería química.

En el campo de la ingeniería química Yee et al [56] presentan la optimización multiobjetivo de dos tipos de reactor (adiabático y de inyección de vapor) para la producción de estireno a partir de etil benceno. Las funciones objetivo involucradas en el estudio fueron la producción, selectividad y rendimiento del estireno. Los parámetros considerados como variables de optimización fueron: temperatura, flujo molar y presión de alimentación del etilbenceno, así como la relación vapor/etilbenceno para el caso del reactor de inyección de vapor; la codificación de estas variables fue binaria. Como resultado se obtuvo un conjunto de soluciones (frente de Pareto) las cuales representan el rango de condiciones de operación óptimas dentro de las cuales la planta puede operar en forma satisfactoria.

Rajesh et al [58] realizan la simulación de un reformador de vapor utilizando un modelo riguroso que incorpora aspectos cinéticos y de transferencia de calor, para después por medio de una optimización multiobjetivo encontrar las condiciones óptimas de operación de la unidad en estudio. Las funciones objetivo de este trabajo fueron la minimización del flujo de metano en la alimentación y la maximización del flujo de monóxido de carbono en el gas de síntesis de salida. Para este estudio la codificación de las 7 variables de decisión utilizadas fue binaria, resultando un cromosoma de 32 bits de longitud. Los autores reportan que el conjunto de soluciones encontradas (frente de Pareto) es sensible a valores de probabilidad de mutación entre 0.002 y 0.003, ya que al graficarse se observa una mayor dispersión entre ellas.

Oh et al [60] utilizan el algoritmo para la predicción de condiciones de operación óptimas de una planta productora de hidrógeno que es alimentada con una mezcla de metano, etano, propano y butano proveniente de una refinería. Los autores presentan dos casos: en el primero se realiza la optimización de 2 objetivos, en donde se busca la maximización del flujo de hidrógeno (producto) y del vapor de salida ya que ambos parámetros son determinantes para en el aspecto económico de esta planta. En el segundo caso se considera la maximización de las dos funciones objetivo del caso uno y la minimización del flujo de combustible utilizado para el horno.

Gupta et al [12] presentan un problema de optimización multiobjetivo que involucra procesos de separación gas-sólido. Se trata de la optimización de un conjunto de N separadores tipo ciclón para el tratamiento de una corriente de alimentación de  $165 \text{ m}^3/\text{seg}$ . Las funciones objetivo usadas fueron la maximización en la eficiencia de la colección de partículas y la minimización de la caída presión, esta última relacionada directamente con el costo de la energía; en este caso, fueron utilizadas 9 variables de decisión entre las cuales están: el número de ciclones, diámetro del ciclón, diámetro del tubo de salida, diámetro de la base del ciclón, altura total del ciclón, codificadas en forma binaria. Por medio de este estudio se logró determinar que tres de las variables de decisión tienen influencia sobre las soluciones óptimas.

Bashkar et al [11] realizaron la optimización de un reactor industrial de tres etapas para la producción de polietilén tereftalato; en este trabajo se tienen como funciones objetivo minimizar la concentración final de ácido y grupos vinilo en el producto. Este problema de optimización incorpora dos restricciones importantes para la calidad del producto: que el producto final alcance un grado de polimerización establecido y que la concentración de dietilén glicol en el producto se encuentre dentro del rango permitido. Las variables de decisión utilizadas en este estudio fueron: la presión del reactor, la temperatura, la concentración del catalizador, el tiempo de residencia y la velocidad de agitación, las cuales fueron codificadas de manera binaria.

En la búsqueda de maximizar ganancias además de respetar las restricciones de proceso Kasat et al [59] presentan la optimización de una unidad industrial de craqueo de lecho fluidizado la cual fue simulada por medio de un modelo simplificado. En este tipo de unidades la maximización en la producción de gasolina conlleva a un incremento en las ganancias, la minimización de la cantidad de monóxido de carbono en el gas de salida está ligado con la vida útil del catalizador, mientras que la disminución del flujo de aire a la entrada de la unidad se relaciona con los costos de operación. Por medio de la optimización simultánea de los tres factores antes mencionados, se obtuvo un conjunto de soluciones (de Pareto) en tres dimensiones, las cuales representan las diversas condiciones de operación de la unidad en estudio. Para este estudio, fueron utilizadas cuatro variables de decisión y dos restricciones relacionadas con la concentración de monóxido de carbono en el gas de salida, cuyos límites fueron obtenidos de datos de planta. La población utilizada para este experimento fue de 100 individuos codificados en forma binaria, la longitud total de cromosoma fue de 40 bits; mientras que el número de generaciones fue de 90. Los autores reportan que variaciones en la probabilidad de cruce, probabilidad de mutación y del número utilizado como semilla para el generador de números aleatorios, no tuvo efecto sobre el Pareto tridimensional obtenido.

## 2.6 El impacto ambiental en la optimización de procesos

La reducción del impacto ambiental en la industria de procesos ha sido objeto de atención durante los últimos años, esto debido a que la legislación se ha vuelto más estricta por lo que los ingenieros de procesos se ven en la necesidad de buscar alternativas de producción. Debido a esto la minimización del impacto ambiental ha empezado a considerarse como un criterio de optimización, en los problemas que surgen a diario en la síntesis y diseño de procesos industriales. Diversas propuestas han sido presentadas en donde las funciones objetivo están relacionadas con el aspecto económico y el impacto ambiental.

El-Halwagi y Manousiouthakis [39] utilizaron el concepto de “pinch mass” como herramienta para obtener el costo óptimo de una red de intercambio de masa con un mínimo de emisión de desechos.

Grossmann y Drabbant[63] presentan un procedimiento para la síntesis de complejos químicos el cual tiene por objetivos la maximización de las ganancias y la minimización de la toxicidad en el sistema, formulado como un problema de optimización mixta-entera.

Moncada y Acevedo [57] proponen la solución de un problema multiobjetivo que surge al buscar la optimización simultánea de los criterios económico y ambiental de un proceso relacionado con la industria petrolera por medio de programación paramétrica.

También existen algunas metodologías propuestas para la minimización del impacto ambiental como la que presentan Pistikopoulos et al [41] (MEIM por sus siglas en inglés) en la que involucran la estimación del deterioro ambiental a través de la cuantificación de efectos adversos del sistema de proceso sobre el ambiente. Esto es realizado por medio de:

1. La expansión de los límites convencionales del proceso para incluir todos los procesos asociados con la manufactura de la materia prima y la generación de energía.
2. Definición de un conjunto de medidas (índices) para la cuantificación de la contaminación en aire, agua, calentamiento global, entre otras.
3. Incorporar la cuantificación de la contaminación en la síntesis de procesos y en las herramientas de optimización.

Además el MEIM comprende los principios del análisis del ciclo de vida (ACV) el cual es una herramienta para cuantificar el impacto ambiental de un producto durante todo su ciclo de vida; comprendiendo la extracción y el procesamiento de la materia prima, manufactura, transportación y distribución, uso, reuso, mantenimiento reciclado y disposición final. Esta metodología ha sido aplicada en varios casos de estudio: la producción del VCM a partir de etileno[41], en el diseño de solventes óptimos [64], así como en la implementación de una optimización multiobjetivo en la industria de los lácteos para la obtención de un diseño y programación óptimos de procesos tipo batch con el mínimo impacto ambiental [64].

Song et al [61] abordan la situación que se presenta en los procesos de refinación de crudo, la búsqueda de una programación efectiva; con la que, además de buscar la maximización



de las ganancias, se tome en cuenta el impacto ambiental producido por la emisión de contaminantes de las unidades de destilación de crudo. La valoración del impacto es realizada por medio de una metodología llamada "Critical Surface-Time 95", la cual se compone de 5 etapas:

- 1) Clasificación del impacto
- 2) Caracterización del impacto
- 3) Clasificación del daño
- 4) Caracterización del daño
- 5) Evaluación

El resultado de la etapa de evaluación denota el impacto global que tiene el sistema en estudio sobre el ambiente, el cual se mide en unidades de  $m^2$  - año.

El *indicador ecológico* (Eco-indicator) es otra herramienta por medio de la cual los ingenieros de proceso pueden cuantificar el impacto que tiene sobre el ambiente un producto o un proceso. Fue desarrollado por Pré Consultants en colaboración con Philips, Océ, NedCar, universidades (Amsterdam, Leiden) y con el apoyo económico del gobierno Holandés. Su origen data del año 1995 y al igual que el MEIM está basado en el análisis del ciclo de vida. A la fecha existen más de 200 "eco-indicadores" para diversos productos y procesos. La versión más reciente de esta herramienta es el "Eco-indicator 99" [42].

Otra manera de cuantificar el impacto ambiental, es tomando en cuenta el efecto tóxico de las sustancias en los seres humanos, ya sea por inhalación o por contacto dérmico. Hui Chen et al [62] presentan una herramienta que acoplada al simulador de procesos comercial HYSYS, considera el destino y el transporte de los contaminantes emitidos (EFRAT por sus siglas en inglés) para después valorar la exposición potencial, así como el riesgo. Esta metodología ha sido aplicada para el análisis de un proceso de recuperación de tolueno y acetato de etilo a partir de una corriente de desecho.

Existe también otra metodología que además de considerar efectos de toxicidad, engloba la identificación y jerarquización de riesgo que puede existir en una unidad de proceso, el índice MOND [54]; el cual está basado en la asignación de penalizaciones y/o bonificaciones a las instalaciones de una planta química. Las penalizaciones son asignadas en función de las sustancias presentes y las condiciones de proceso. Las bonificaciones tienen en cuenta las instalaciones de seguridad que pueden mitigar o prevenir los efectos accidentales. La combinación de ambas lleva a la determinación del índice con el que se afecta una determinada parte de la instalación, pudiendo examinar, a la vista de estos índices, la importancia relativa de las partes estudiadas en función del riesgo asociado.

## Capítulo 3

### Eficientización del algoritmo genético

Los algoritmos genéticos han sido utilizados como una herramienta en la solución de problemas de optimización, por medio de los cuales se han obtenido resultados aceptables. Pero esta útil herramienta, presenta un inconveniente, requiere de un gran esfuerzo computacional para obtener los resultados, por lo que es de gran importancia eficientizar su desempeño para reducir el tiempo requerido en la solución de problemas.

En este capítulo se presenta la solución de una serie de problemas típicos de optimización presentados en la literatura. Esto con el fin de determinar cual sería el valor adecuado (o el rango de valores adecuados) de los operadores del algoritmo genético utilizado. Posteriormente se presenta una metodología para reducir la complejidad en la solución de problemas multiobjetivo, presentada por Reed[43]

Seguido se propone un proceso de clasificación de los individuos en intervalos según sus aptitudes; con este proceso se busca la definición de un criterio de convergencia (o paro) en el algoritmo utilizado. En la parte final se presenta un análisis comparativo entre el criterio de convergencia propuesto y la metodología presentada en la literatura.

#### 3.1 Ajustando los parámetros del algoritmo genético.

El desempeño de un algoritmo genético depende en gran medida del valor asignado a los operadores genéticos utilizados por éste. Existen en la literatura diversas estrategias propuestas por medio de las cuales se ha logrado estimar los valores adecuados para los distintos operadores que controlan el comportamiento del algoritmo genético [45]

Con respecto al tamaño de la población Reed et al [45] describen en su trabajo que si la población es pequeña se pueden necesitar de muchas generaciones para encontrar el óptimo, en cambio, si es muy grande, el óptimo es encontrado más rápido, pero puede tomar mucho tiempo el saber que ya ha sido encontrado, lo cual se reflejaría en evaluaciones de la función objetivo innecesarias. Los tamaños de población que se reportan en la literatura son 25, 50, 100 y 200 individuos [49]

Con relación a la probabilidad de cruce, De Jong concluye que el mejor valor de la probabilidad de cruce ( $pc$ ) es de 0.6, mientras que Deb sugiere un valor que se encuentre entre 0.5 y 1.

De Jong [15] demuestra que generalmente el Ag tiene buen desempeño cuando la probabilidad de mutación simple (mutación de bit,  $pmut$ ) es inversamente proporcional al tamaño de la población. Por su parte Deb[13] propone que el valor para la probabilidad de mutación, cuando se ha escogió codificación real para las variables de optimización sea un número entre cero y  $1/Nvar$ ; donde  $Nvar$  es el número de variables codificadas de manera real.

El algoritmo utilizado para la elaboración de este trabajo (NSGA-II) tiene un par de operadores que son específicos para las variables que son codificadas de manera real. Ambos propuestos por Deb [13]. El primero de ellos es llamado índice de distribución( $\eta_c$ ) el cual es utilizado en el operador de cruce binario simulado (SBX), para el cual se

proponen valores entre el rango de 0.5 hasta 100. El otro es el índice distribución utilizado por el operador para la mutación de variables, codificadas de manera real ( $\eta_m$ ). El rango de valores propuestos para este índice oscila entre 0.5 y 500.

### 3.1.1. Experimentos computacionales

El algoritmo utilizado en la realización de este trabajo, nos permite utilizar dos tipos de codificación para las variables de optimización, la binaria y la real; esto trae como consecuencia que requiera operadores genéticos diferentes a los utilizados por un algoritmo genético básico. Para la estimación de los valores óptimos de éstos operadores, recurrimos a la experimentación. La cual consistió en resolver problemas típicos de optimización publicados en la literatura. Los problemas resueltos fueron tomados del artículo presentado por Sahinidis et al [55]. Previo a este trabajo Leboireiro [49] presentó la solución de los problemas 1, 3, 9 y 14 del artículo antes mencionado, como parte de un estudio para determinar el valor apropiado de los parámetros de un algoritmo genético desarrollado por Carroll, el cual tiene características diferentes al presentado en este trabajo: es de un solo objetivo, maneja solo codificación binaria para las variables de decisión, cuenta con mutación de bit y de variable, entre otras.

Se realizaron 500 experimentos en los cuales se variaron los siguientes parámetros:

- Tamaño de la población (npopsiz): 50, 100, 200
- Número de generaciones (ngener): 500, 1000, 2000, 3000
- Probabilidad de cruce (pcross): 0.5, 0.53, 0.644, 0.65, 0.755, 0.80, 0.85, 0.90, 0.95
- Semilla para la generación de los números aleatorios (seed): 0.223, 0.586, 0.749
- Probabilidad de mutación para variables con codificación real (pmut\_r): 0.0010, 0.0100, 0.035, 0.046, 0.055, 0.065, 0.100, 0.333
- Índice de distribución para el cruce ( $\eta_c$ ): 0.5, 1, 5, 10, 15, 20
- Índice de distribución para la mutación ( $\eta_m$ ): 5, 10, 15, 30, 40, 45, 50
- Probabilidad de mutación (pmut\_b) para variables con codificación binaria: 0.030, 0.033, 0.035, 0.040, 0.046, 0.050, 0.055, 0.065, 0.10, 0.150, 0.250
- Tipo de cruce (optype) para variables con codificación binaria: sencillo y uniforme.
- Número de bits para la representación de las variables binarias (nbits): 10, 11, 12

Primero se realizaron los experimentos codificando las variables como binarias y posteriormente como reales; esto con la finalidad de comparar el desempeño del algoritmo. Después de realizar las corridas resultó que los resultados más cercanos al óptimo global fueron obtenidos por medio de codificaciones reales. Los resultados se muestran a continuación.

**Problema 1**

**Función objetivo:**                    minimizar         $f[0] = -(x[0] + x[1])$

**Sujeto a:**                             $restricción[0] = -(x[0] * x[1]) + 4 \quad 0 \leq x \leq (6,4)$

**Óptimo global en:**                 $x[0] = 6.0 \quad x[1] = 0.6666667 \quad f = -6.666667$

**Resultado**

**Codificación binaria**

Codificando ambas variables como binarias, de los individuos obtenidos, fueron seleccionados 77, los cuales alcanzaron una aptitud igual  $-6.6666$ , los detalles de estos individuos se presentan en la tabla 3.1.

tamaño población (npobsz)	número de individuos
50	10
100	35
200	32
número de generaciones (ngener)	número de individuos
500	22
1000	28
2000	27
prob. cruce (pcross)	número de individuos
0.533	20
0.644	33
0.755	24
tipo de cruce (optype)	número de individuos
1(sencillo)	34
2(uniforme)	43

prob. mutación (pmut_b)	número de individuos
0.035	19
0.046	18
0.055	20
0.065	20
número de bits (nbits)	número de individuos
10	0
11	77
12	0
semilla (seed)	número de individuos
0.223	30
0.586	25
0.749	22

**Tabla 3-1 Resultados del problema 1, codificación binaria**

Para este tipo de codificación, los tamaños de población con los que se obtuvieron los mejores resultados fueron de 100 y 200 individuos, mientras que los mejores valores para la probabilidad de mutación son de 0.035, 0.046, 0.055, 0.065 para todos los tamaños de población. El tipo de cruce pierde importancia ya que ninguno de los dos tipos fue

predominante. De la probabilidad de cruce podemos decir que valores mayores de 0.5 producen buenos resultados.

Por su parte, Leboreiro [49] reportó que para poblaciones de 50, 100 y 200 individuos, las probabilidades de mutación pierden importancia ya que no hay ningún valor predominante. Con respecto a la probabilidad de cruce obtuvo que 0.6 produce los mejores resultados y que el cruce uniforme resultó ser el mejor.

### **Codificación real**

Al codificar a las variables como reales fueron considerados como mejores 58 individuos: 10 con aptitud  $-6.666667$ , 26 con aptitud  $-6.666666$ , 3 con aptitud  $-6.666665$ , 5 con aptitud  $-6.666664$ , 3 con aptitud  $-6.666663$ , 2 con aptitud  $-6.666664$ , 3 con aptitud  $-6.666661$  y 6 con aptitud  $-6.666666$ . Los detalles de estos individuos se muestran en la tabla 3.2.

tamaño población (npoysize)	número de individuos
50	3
100	18
200	29
número de generaciones (ngener)	número de individuos
500	26
1000	16
2000	27
prob. Cruce (pcross)	número de individuos
0.533	17
0.644	18
0.755	24
índice distrib. cruce real (di)	número de individuos
5	28
10	16
15	6
20	6

índice distrib. mutación real (dim)	número de individuos
30	13
40	16
45	12
50	17
prob. mutación (pmut_real)	número de individuos
0.0035	15
0.0046	12
0.0055	16
0.0065	14
semilla (seed)	número de individuos
0.223	28
0.586	9
0.749	21

**Tabla 3-2 Resultados del problema 1, codificación real**

Los tamaños de población que reportaron los mejores resultados fueron de 200 y 100, así como probabilidades de cruce arriba de 0.6, para el índice de distribución para el cruce, los mejores valores se reportan entre 5 y 10 y para el índice de distribución para el cruce entre 30 y 50. Probabilidades de mutación a bajo de 0.0010 con poblaciones de más de 100 individuos no dieron buenos resultados; así como combinaciones de ambos índices con menos de 25 unidades de diferencia entre ellos.

**Problema 8**

**Función objetivo:**  $minimizar \quad f[x] = 2 * x[0] + x[1]$

**Sujeto a:**  $restricci3n[0] = -(x[0] * x[1]) + 4 \leq 0 \quad 0 \leq x \leq 1$   
 $restricci3n[1] = -4 * (x[0])^2 - 4 * (x[1])^2 + 1 \leq 0$

**3ptimo global en:**  $x[0] = 0.129409 \quad x[1] = 0.48963 \quad f = 0.741782$

**Resultado**

**Codificaci3n binaria**

Fueron seleccionados como mejores, 17 individuos con aptitud de 0.74188, los detalles se muestran en la tabla 3.3

tama3o poblaci3n (npobsiz)	n3mero de individuos
50	1
100	7
200	9
n3mero de generaciones (ngener)	n3mero de individuos
500	5
1000	6
2000	6
prob. Cruce (pcross)	n3mero de individuos
0.65	5
0.8	5
0.85	7
tipo de cruce (optype)	n3mero de individuos
1(sencillo)	13
2(uniforme)	4

prob. mutaci3n (pmut)	n3mero de individuos
0.03	5
0.033	5
0.035	1
0.04	6
n3mero de bits (nbits)	n3mero de individuos
10	0
11	0
12	12
semilla (seed)	n3mero de individuos
0.333	4
0.586	8
0.749	5

**Tabla 3-3Resultados del problema 8, codificaci3n binaria**

Para este tipo de codificaci3n la probabilidad de cruce que mostr3 los mejores resultados fueron valores de 0.65, 0.8 y 0.85; para todos los tama3os de poblaci3n el cruce sencillo fue el que predomin3. Con respecto a la probabilidad de mutaci3n, los mejores valores fueron 0.03, 0.033 y 0.04.

**Codificación real**

Fueron considerados como mejores 47 individuos: 2 con aptitud 0.741781, 5 con aptitud 0.741782, 11 con aptitud 0.741783, 9 con aptitud 0.741784, 9 con aptitud 0.741785, 9 con aptitud 0.741786, 1 con aptitud 0.741787, 3 con aptitud 0.741788 y 1 con aptitud 0.741789. Los detalles se muestran en la tabla 3.4.

tamaño población (npopsize)	número de individuos
100	9
200	15
300	23
número de generaciones (ngener)	número de individuos
500	15
1000	12
2000	20
prob. Cruce (pcross)	número de individuos
0.67	14
0.7	14
0.75	19
índice distrib. cruce real (di)	número de individuos
5	21
10	8
20	9
25	9

índice distrib. mutación real (dim)	número de individuos
15	11
20	9
40	11
45	16
prob. mutación (pmut_real)	número de individuos
0.3	13
0.33	14
0.35	10
0.37	10
semilla (seed)	número de individuos
0.333	20
0.586	13
0.749	14

**Tabla 3-4 Resultados del problema 8. codificación real**

En este ejemplo, los tamaños de población de 200 individuos con 1000 generaciones arrojaron los mejores resultados, lo mismo se dio con la combinación de índices de distribución para el cruce de 5 con índices de distribución para la mutación con valores arriba de 20. En cuestiones de probabilidad de mutación se observó que los mejores valores están entre 0.3 y 0.37.

**Problema 14****Función** *minimizar***objetivo:**  $f[0] = (x[3]-1)^2 + (x[4]-2)^2 + (x[5]-1)^2 - \ln(x[6]+1) + (x[0]-1)^2 + (x[1]-2)^2 + (x[2]-3)^2$ 

$$\text{restricción}[0] = -(x[3] - x[4] - x[5] - x[0] - x[1] - x[2] + 5)$$

$$\text{restricción}[1] = -(x[5])^2 - (x[0])^2 - (x[1])^2 - (x[2])^2 + 5.5$$

$$\text{restricción}[2] = -x[3] - x[0] + 1.2$$

$$\text{restricción}[3] = -x[4] - x[1] + 1.8$$

$$\text{restricción}[4] = -x[5] - x[2] + 2.5$$

$$\text{restricción}[5] = -x[6] - x[0] + 1.2$$

$$\text{restricción}[6] = -(x[4])^2 - (x[1])^2 + 1.64$$

$$\text{restricción}[7] = -(x[5])^2 - (x[2])^2 + 4.25$$

$$\text{restricción}[8] = -(x[4])^2 - (x[2])^2 + 4.64$$

$$0 \leq x \leq (1.2, 1.8, 2.5) \text{ para } x[0], x[1], x[2]$$

$$x = \{0,1\}, i = 3,4,5,6$$

**Sujeto a:****Óptimo**

$$x[0] = 0.2 \quad x[1] = 0.8 \quad x[2] = 1.907878$$

**global en:**

$$x[3] = 1 \quad x[4] = 1 \quad x[5] = 0 \quad x[6] = 1 \quad f = 4.579582$$

**Resultado****Codificación binaria**

Para este experimento fueron codificadas como binarias todas las variables. Se consideraron 8 individuos con aptitud 4.57958. Los detalles se muestran en la tabla 3.5



tamaño población (npobsiz)	número de individuos
50	1
100	3
200	4
número de generaciones (ngener)	número de individuos
500	0
1000	1
3000	7
prob. Cruce (pcross)	número de individuos
0.5	1
0.9	5
0.95	2
tipo de cruce (optype)	número de individuos
1(sencillo)	6
2(uniforme)	2

prob. mutación (pmut)	número de individuos
0.05	3
0.1	2
0.15	3
0.2	0
número de bits (nbits)	número de individuos
10	0
11	0
12	12
semilla (seed)	número de individuos
0.223	2
0.586	2
0.749	4

**Tabla 3-5 Resultados del problema 14, codificación binaria**

En este caso, probabilidades altas de cruce (entre 0.9 y 0.95) predominaron entre los mejores individuos, el tipo de cruce no es relevante para ningún tamaño de población. Las probabilidades de mutación con las que se observaron buenos resultados fueron 0.05, 0.1 y 0.15.

Por su parte, Leboreiro reporta [49] que los mejores valores de probabilidad de mutación fueron 0.005, 0.01 y 0.05. El cruce uniforme fue el que dio mejor resultado con tamaños de población de 50, con poblaciones de 100, no favorece ningún tipo de cruce. La probabilidades cruce con mejor desempeño fue de 0.6 para todos los tamaños de la población.

### **Codificación real**

Únicamente se codificaron como reales a las variables continuas ( $x[0], x[1], x[2]$ ). Fueron considerados como mejores 71 individuos con aptitud de 4.579583. Los detalles se muestran en la tabla 3.6

tamaño población (npobsz)	número de individuos
50	0
100	12
200	59
número de generaciones (ngener)	número de individuos
500	26
1000	20
3000	25
prob. cruce (pcross)	número de individuos
0.5	15
0.9	27
0.95	29
índice distrib. cruce real (di)	número de individuos
0.5	35
1	34
5	2
10	0

índice distrib. mutación real (dim)	número de individuos
5	10
10	22
15	18
30	21
prob. mutación (pmut_real)	número de individuos
0.001	18
0.01	17
0.1	19
0.333	17
semilla (seed)	número de individuos
0.223	28
0.586	18
0.749	25

**Tabla 3-6 Resultados del problema 14, codificación real parte 1**

Los mejores resultados se obtuvieron con tamaños de población de 100 y 200 individuos; con probabilidad de mutación alta (entre 0.9 y 0.95), con combinaciones de índices de distribución para el cruce y la mutación de 0.5 y 1.0 y de 10 y 30. Se observó que combinaciones de valores cercanos ó iguales no son favorables, así como probabilidades de mutación abajo de 0.001

Después de observar los resultados obtenidos, podemos concluir que, cuando las variables de optimización son codificadas en formato binario, el tamaño de población que obtuvo los mejores resultados fue de 100 y 200 individuos; en relación a la probabilidad de cruce, se obtuvieron buenos resultados con valores arriba de 0.50. Con respecto al tipo de cruce, el sencillo predominó entre los mejores individuos; de la probabilidad de mutación, resultados favorables se obtuvieron con valores entre 0.03 y 0.3

Por otro lado, cuando las variables son codificadas en formato real, los mejores resultados se obtuvieron con tamaños de población de 200 y 300 individuos. Al igual que en la codificación binaria, la probabilidad de cruce dio buenos resultados con valores arriba de 0.5. Para la probabilidad de mutación los valores entre 0.001 y 0.3 mostraron buen desempeño. Los mejores valores para el índice de distribución para el cruce fueron entre 5 y 20 y para la mutación entre 10 y 50.

### 3.2 La convergencia en los algoritmos evolutivos multiobjetivo.

El desarrollo y la aplicación de los algoritmos multiobjetivo evolutivos datan desde los años ochenta. Los más recientes presentan mejoras en el sentido de que es más fácil entender su funcionamiento; sin embargo, no existe novedad con respecto a teorías de convergencia relacionadas a éstos. Este hecho se debe en parte a que las teorías de convergencia para algoritmos evolutivos de un solo objetivo son de reciente creación [49]. El desarrollo de una teoría para la optimización multiobjetivo es más complicada al compararla con el desarrollo de una teoría para el problema de optimización de un solo objetivo. Esto debido a que el procedimiento multiobjetivo tiene dos tareas: la convergencia hacia el frente óptimo de Pareto y el mantenimiento de la diversidad entre los individuos que forman ese frente.

Reed [43] en un estudio desarrollado para el monitoreo de sitios contaminados utilizando el NSGA-II, presenta una metodología para reducir la complejidad en la solución de problemas multiobjetivo.

Esta metodología consiste de 3 etapas:

- 1) Análisis preliminar del problema.
- 2) Balance entre la ruptura y la innovación.
- 3) Corridas de prueba con poblaciones diversas

A continuación se describe cada una de ellas.

#### *Etapa 1: Análisis preliminar del problema.*

Para esta es necesario que el usuario tenga en mente lo siguiente:

- ¿Cuántos individuos no-dominados ( $R_{ND}$ ) son aceptables?
- ¿Cuál es el porcentaje mínimo de cambio en el número de individuos no-dominados ( $\Delta_{ND}$ ) para que dos corridas sucesivas sean consideradas como idénticas?
- ¿Cuál es el tiempo promedio requerido para evaluar un diseño,  $T_{eval}$  ?
- ¿Cuál es el tiempo máximo de corrida aceptable,  $T_{tot}$  ?

El número de individuos no-dominados ( $R_{ND}$ ) es usado para calcular el límite inferior de la población  $N_0$  para el NSGA-II usando la siguiente expresión:

$$N_0 \approx 2R_{ND} \quad (3.1)$$

La segunda pregunta esta relacionada con la exactitud de la solución requerida por el usuario para especificar el mínimo porcentaje de cambio en el número de individuos no-dominados  $\Delta_{ND}$  para que dos corridas sucesivas sean consideradas como idénticas.

Por ejemplo considere dos experimentos sucesivos del NSGA-II de los cuales el primero utiliza una población de  $N$  individuos que evoluciona a un conjunto de  $A$  individuos no-dominados, mientras que el segundo experimento, utiliza una población de tamaño  $2N$  para evolucionar hacia un conjunto de  $K$  individuos no-dominados. El resultado de esas corridas se introduce en la ecuación 2 para definir que camino tomar:

- el tamaño de la población será duplicado nuevamente

- el algoritmo se detendrá debido a que el conjunto de los no-dominados ha alcanzado la exactitud suficiente.

$$\text{Si } \Delta_{ND} < \left( \frac{|K-A|}{A} \right) 100 \text{ entonces } N = 2N \text{ y continuar la búsqueda} \quad (3.2)$$

Caso contrario: finalizar

Esto es, si una corrida que tiene un tamaño de población  $2N$ , no logra incrementar el número de individuos no-dominados al menos  $\Delta_{ND}$  quiere decir que la búsqueda ha llegado a un punto en el que duplicar la población no significará mejora alguna. En el estudio presentado por Reed, el valor de  $\Delta_{ND}$  se estableció en 10%

Para las preguntas 3 y 4 se requiere que el usuario especifique ¿cuanto tiempo esta dispuesto a esperar por una solución ( $T_{tot}$ )? Para el estudio realizado por Reed  $T_{tot}$  fue igual a 4 horas donde el tiempo promedio para la evaluación de una función objetivo fue de 0.044 segundos ( $T_{eval}$ ) por lo que el máximo número de soluciones,  $E_{tot}$ , que pueden ser evaluados están dados por la siguiente ecuación que involucra a  $T_{tot}$  y  $T_{eval}$

$$E_{tot} = \frac{T_{tot}}{T_{eval}} \quad (3)$$

$$E_{tot} = \frac{(4 * 3600)}{0.044} = 327,272 \text{ evaluaciones}$$

El resultado de esta ecuación, es usado en la ecuación 3.4 para calcular el número máximo de veces ( $\beta_{MAX}$ ) en que la población puede ser duplicada a partir de su límite inferior calculado anteriormente:

$$\sum_{\beta=0}^{\beta_{MAX}} 2^{\beta} N_0 t \leq E_{tot} \quad (3.4)$$

En esta ecuación  $t$  representa el número total de generaciones usadas en cada experimento del NSGA-II para la búsqueda del conjunto de individuos No- dominados. La duración de  $t$  se estima que sea aproximadamente igual a  $2l$ , donde  $l$  es la longitud del cromosoma binario que representa al individuo Reed et al [2000b]. En este estudio, la longitud del cromosoma fue de 20, lo que dio como resultado 40 generaciones. El término  $N_0 t$  representa al número de individuos que son evaluados cuando el tamaño de la población es igual a  $N_0$  en  $t$  generaciones (40 en este caso); para este estudio, este término tiene valor de 2400. Así que dado el máximo de individuos que pueden ser evaluados ( $E_{tot}= 327,272$ ), la ecuación 3.4 nos muestra que la población inicial ( $N_0$ ) puede ser duplicada un total de seis veces ( $\beta_{max}= 6$ ), dando como limite superior para el tamaño de la población 3840 individuos.

**Etapa 2: Balance entre la ruptura y la innovación.**

En esta etapa se explica cómo los parámetros de probabilidad de cruce y de mutación son establecidos para esta metodología. Existen dos “fuerzas” que deben ser balanceadas cuando son establecidos éstos parámetros: la ruptura y la innovación. La ruptura ocurre cuando debido a un exceso en el cruce o en la mutación hace imposible que el AG agrupe soluciones óptimas. La ecuación 3.5 presenta el límite de ruptura desarrollado por Thierens [44] y usado en la metodología del diseño de un AG simple desarrollado por Reed et al. [45]

$$P_c \leq \frac{s-1}{s} \quad (3.5)$$

El parámetro  $s$  representa el total de individuos que compiten en la selección por torneo. Esta ecuación intenta proteger a los individuos del cruce excesivo. La probabilidad de mutación  $P_m$  se establece usando la ecuación (3.6) basada en los estudios de De Jong [15] y Schaffer et al. [46].

$$P_m \approx \frac{1}{N} \quad (3.6)$$

Donde  $N$  es el tamaño de la población; esta ecuación asegura que la mutación tenga el mínimo efecto destructivo.

**Etapa 3: experimentos de prueba con poblaciones diversas.**

Las etapas 1 y 2 de esta metodología proporcionan todos los parámetros requeridos para llevar a cabo los experimentos con el NSGA-II, en la parte final de esta metodología el algoritmo incrementa de manera automática el tamaño de la población en experimentos sucesivos hasta que se obtiene un número suficiente de individuos no dominados o hasta que el tiempo de cómputo establecido por el usuario ha sido alcanzado. Para el problema abordado por Reed, es necesario un mínimo de dos experimentos usando dos dobles tamaños de población sucesivos para determinar si el número de individuos no-dominados es el suficiente.

Para el estudio realizado por Reed, la aplicación de esta metodología significó un ahorro del 80% en costos de cómputo ya que la evaluación de las funciones objetivo disminuyó de 160,000 a sólo 38,000, lo cual representa una gran ventaja.

**3.3 Definición de un nuevo criterio de convergencia para el NSGA-II**

Al igual que otros algoritmos genéticos propuestos en la literatura para la optimización multiobjetivo, el NSGA-II no cuenta con un criterio de convergencia o paro; si no que se ejecuta un número fijo de generaciones, donde en la última generación se espera obtener un conjunto de soluciones óptimas. Motivados por lo anterior se propone el siguiente criterio de paro, el cual se basa en la clasificación de los individuos en intervalos según sus aptitudes; la condición que determinará si el experimento debe detenerse ó seguir, es cuando exista cierta “cercanía” en la aptitud de las soluciones (individuos) que han sido clasificados como *dominados* con la aptitud de las soluciones que han sido clasificadas por

el algoritmo como *no-dominadas*. En la Figura 3-1, se presenta un diagrama de flujo condensado del criterio propuesto; posteriormente se describe el proceso con más detalle.

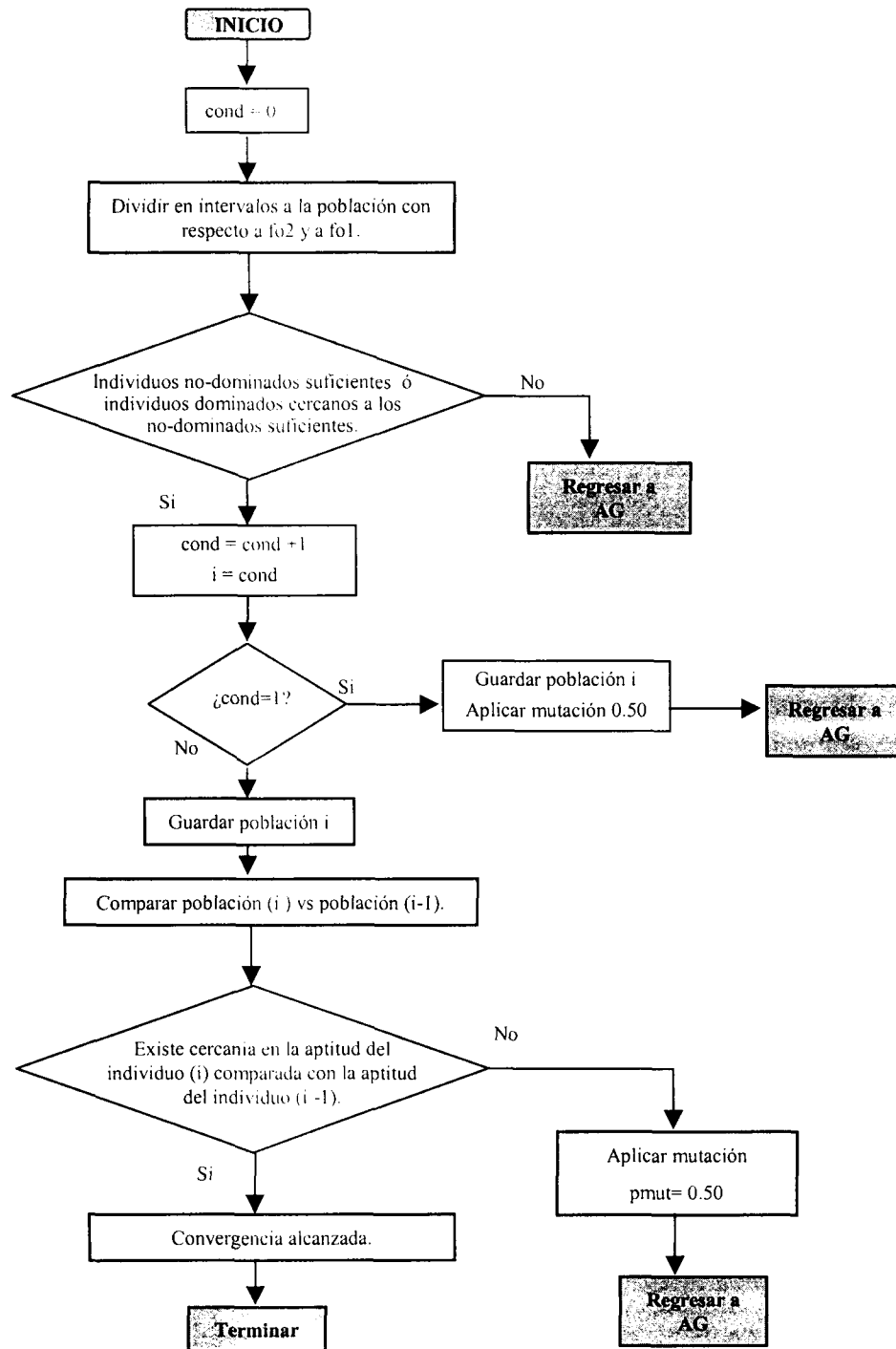


Figura 3-1 Diagrama de flujo condensado del criterio de paro

Al término de la generación  $i$  (cuando los operadores genéticos: selección, cruce, mutación, elitismo han sido aplicados y se tiene la nueva población que pasará a la siguiente generación), el primer paso es hacer un conteo del número de individuos ( $\theta$ ) cuyo error sea menor a una tolerancia propuesta ( $\epsilon$ ). Una vez realizado el conteo, se calcula la siguiente relación:

$$\frac{\theta}{npopsiz} \geq \Phi \quad (3.7)$$

donde  $\Phi$  representa un porcentaje de individuos bien adaptados. Si la condición anterior no es alcanzada, el proceso termina. De otro modo los individuos son ordenados en forma descendente con respecto al valor de la función objetivo 2 ( $f_2$ ). Con esto obtenemos el rango total de ésta función objetivo. Posteriormente, este rango será dividido en un número de intervalos; el valor obtenido ( $deltaint$ ) representa la amplitud del primer intervalo a formar. Teniendo entonces el límite inferior y la amplitud de intervalo, procedemos a establecer los límites del primer intervalo.

El siguiente paso es cuantificar el número de individuos presentes en el intervalo; pueden presentarse los siguientes casos:

- 1) Que el intervalo esté vacío: cuando esto sucede el límite superior del intervalo es aumentado, en una cantidad igual a  $deltaint$ . Realizado esto, se repite el proceso de conteo de individuos presentes en el intervalo, si nuevamente el intervalo queda vacío, se da por terminado el proceso. Si existen individuos, se verifica que al menos uno de ellos sea *no-dominado*, en caso contrario, el proceso termina.
- 2) Sólo un individuo en el intervalo: para este caso se verifica que el individuo sea No-Dominado, en caso contrario, el proceso termina.
- 3) Más de un individuo en el intervalo: se verifica que al menos un individuo sea No-Dominado, en caso contrario el proceso termina.

Una vez que estamos seguros de que hay al menos un individuo *no-dominado* en el intervalo (casos 1 y 3), el siguiente paso es ordenar a los individuos en forma ascendente pero ahora con respecto a los valores de la función objetivo 1 ( $f_1$ ).

Esta etapa tiene la finalidad de determinar la diferencia entre el individuo que obtuvo el mayor valor en esa función objetivo y el menor. Una vez identificados éstos se procede a calcular el porcentaje de separación entre ambos individuos (respecto a  $f_1$ ). Si esta diferencia supera al porcentaje establecido como límite ( $\Omega$ ), es necesario subdividir el intervalo hasta que la condición sea cumplida. Para después, verificar la existencia de al menos un individuo *no-dominado* en el intervalo. Si el intervalo quedase sin individuos *no-dominados* el procedimiento termina.

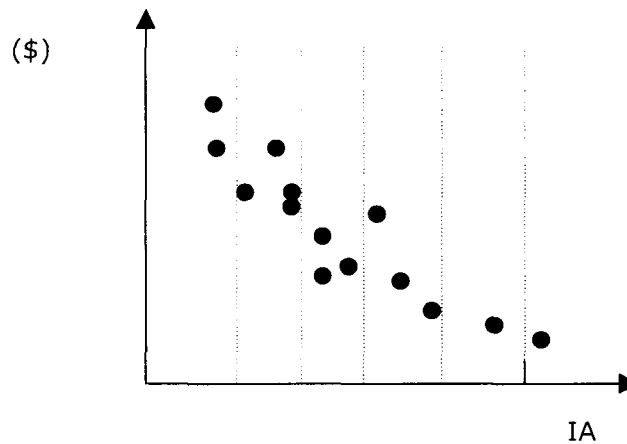


Figura 3-2 Ejemplo de la división de la curva de Pareto en intervalos.

Después de que se comprobó que hay al menos un individuo no-dominado, en el intervalo, lo que sigue es contabilizar los individuos dominados y calcular que tan alejados están (en cuestión de aptitud) de los individuos *no-dominados*, por lo que se propone el cálculo de “la distancia” entre las  $n$  funciones objetivo de cada individuo, la ecuación propuesta es:

$$\sqrt{\frac{(FO_i^D - FO_i^{ND})^2}{(FO_i^{ND})^2}} \quad (3.8)$$

donde

$FO_i^D$  representa el valor de la función objetivo  $i$  de un individuo dominado

$FO_i^{ND}$  representa el valor de la función objetivo  $i$  de un individuo no-dominado

En este caso, en el que solo se desea optimizar dos funciones objetivo, se busca que el resultado de 3.8 se encuentre dentro de una tolerancia denominada en este caso como  $\epsilon_2$ . Si esto sucede, el contador de individuos dominados cercanos a los *no-nominados* es incrementado. Dicho contador está representado por la variable *dom\_ok*.

El procedimiento anterior es necesario repetirlo hasta que el límite superior del intervalo  $i$  es igual al límite superior total de la función objetivo 2, cuando esta condición es alcanzada, es necesario hacer los conteos de:

- 1) Número de individuos no-dominados en los  $i$  intervalos formados ( $\Sigma_{ndom}$ )
- 2) Número de individuos dominados en los  $i$  intervalos formados ( $\Sigma_{dom\_tot}$ )
- 3) Número de individuos dominados cercanos a un no-Dominado ( $\Sigma_{dom\_ok}$ )



A partir de lo anterior podemos obtener las siguientes relaciones:

$$\frac{\Sigma dom\_ok}{\Sigma dom\_tot} \quad (3.9)$$

$$\frac{\Sigma ndom}{\Sigma individuos\ por\ int\ ervalo} \quad (3.10)$$

Si alguna de las dos resultara ser mayor o igual a una tolerancia establecida como  $(\lambda)$ , el conjunto de individuos es considerado como el primer conjunto de “soluciones óptimas”. Posteriormente se aplica una probabilidad de mutación grande (0.50, tanto para las variables codificadas como reales y para las codificadas como binarias), se deja correr una generación el algoritmo genético para en la siguiente, regresar la probabilidad de mutación a su valor original.

El paso anterior puede ser comparado con el procedimiento realizado por Reed (sección 3.2), en donde para determinar si el experimento debe seguir o continuar, es necesario que exista un mínimo número de soluciones no-dominadas, Reed requiere de 2 corridas sucesivas; en nuestro caso, esto es realizado dentro de la misma corrida y el porcentaje mínimo de individuos no-dominados es un parámetro que puede ser ajustado

Cuando se presenta por segunda ocasión que la relación 3.9 ó la relación 3.10, sean mayor ó igual a  $(\lambda)$ , se considera al conjunto de soluciones obtenido como “óptimo”. Para comprobarlo, se realiza un proceso de comparación de aptitudes de la función objetivo 1, entre los individuos recién clasificados y de los que fueron clasificados como el primer conjunto de “soluciones óptimas”. La ecuación a utilizar es:

$$\frac{FO1_{i,anterior} - FO1_{i,nueva}}{FO1_{i,nueva}} \leq \psi \quad (3.11)$$

donde

$FO1_{i,anterior}$  representa el valor de la función objetivo 1, de la solución  $i$  del primer conjunto de soluciones óptimas

$FO1_{i,nueva}$  representa el valor de la función objetivo 1, de la solución  $i$  del segundo conjunto de soluciones óptimas clasificadas.

$\psi$  representa un parámetro de diferencia entre las soluciones comparadas.

Si esta relación permanece en todos los individuos de los  $i$  intervalos definidos, podemos decir que el programa ha convergido y que hemos encontrado un conjunto de soluciones “óptimas”. Si en algún individuo no se mantuviera esa relación, se da por terminado el proceso.

El procedimiento anteriormente descrito se ejemplifica en las figuras 3-3 , 3-4 y 3-5, considerando como valores de los parámetros (tolerancias) descritos en los párrafos anteriores en la tabla 3.7. Cabe resaltar que estos valores no son definitivos, ya que con el propósito de mejorar el desempeño del criterio de paro, en los casos presentados en el capítulo 4 algunos de estos parámetros son modificados.

Parámetro	Valor
$\varepsilon$	0.000001
$\Phi$	0.40
$\Omega$	0.30
$\varepsilon_2$	0.10
$\lambda$	0.80
$\Psi$	0.01

**Tabla 3.7 Valores para los parámetros del criterio de paro**

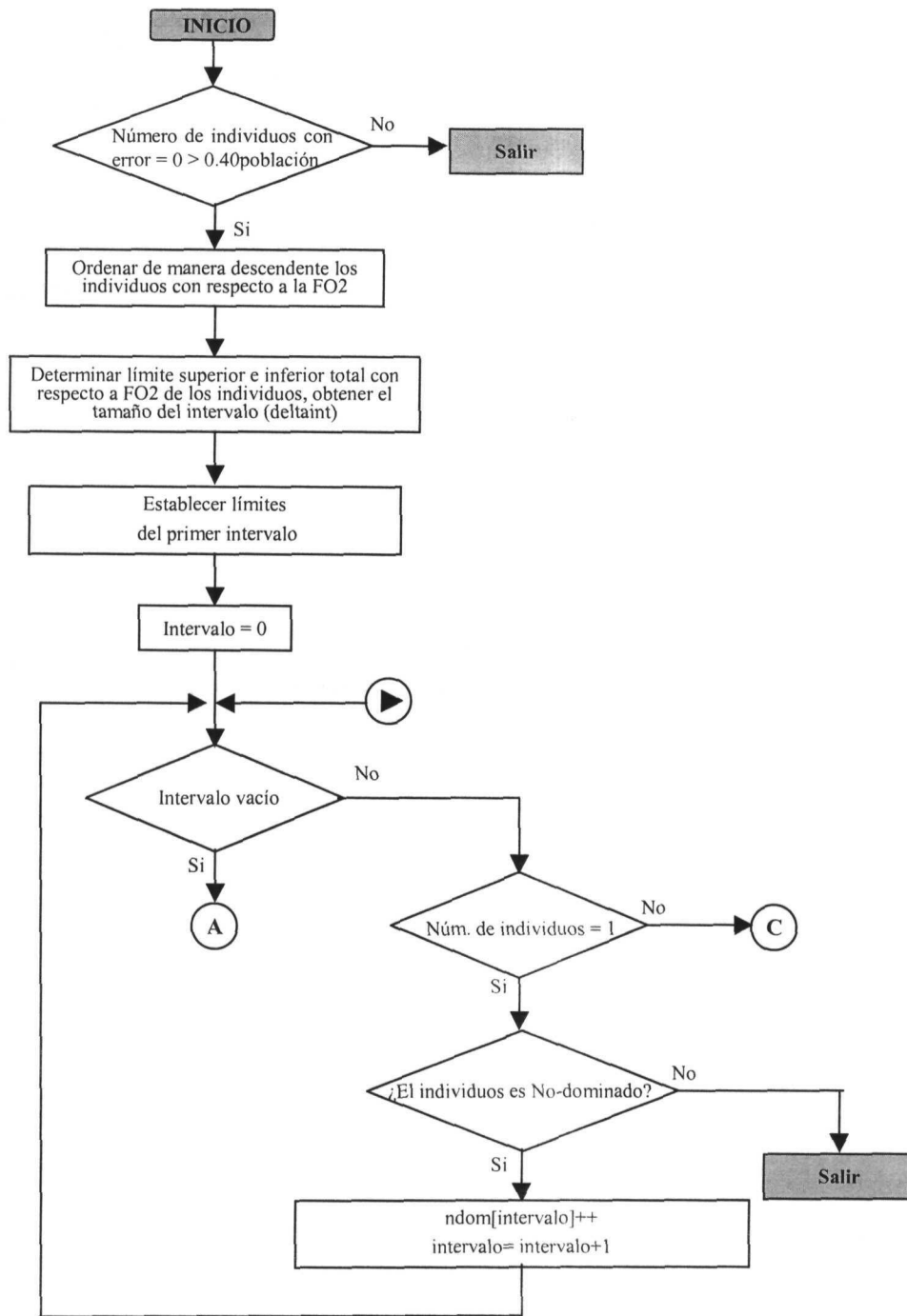


Figura 3-3 Diagrama de convergencia 1 de 3

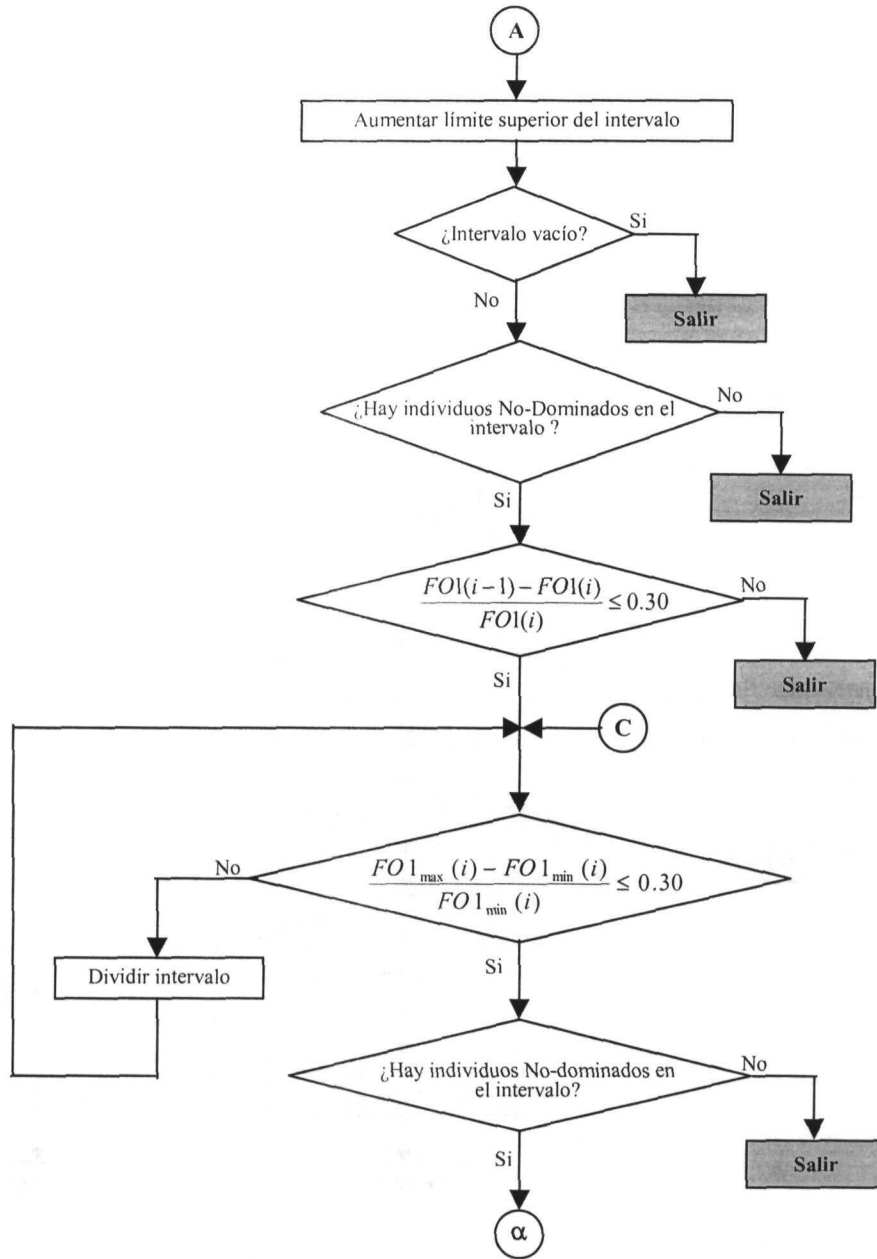


Figura 3-4 Diagrama de convergencia 2 de 3

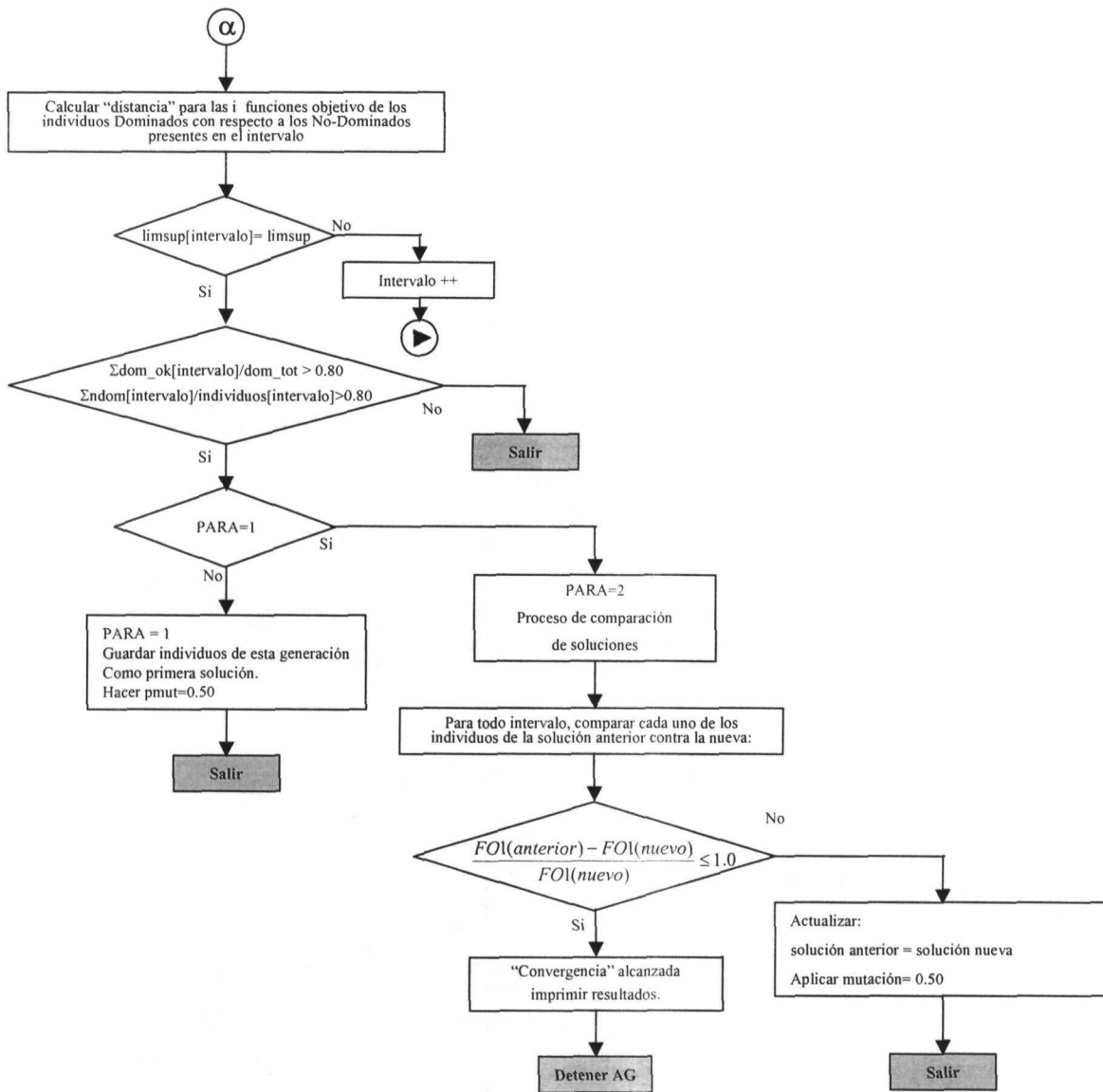


Figura 3-5 Diagrama de convergencia 3 de 3

### 3.4 Análisis comparativo entre metodologías de paro para un algoritmo genético.

La metodología propuesta por Reed (revisada en la sección tal de éste trabajo) introduce un método multi-poblacional que de manera automática ajusta el valor de los operadores genéticos (probabilidad de cruce y mutación), lo cual resulta una ventaja, ya que de éstos depende en gran parte que la población evolucione hacia un conjunto de soluciones óptimas.

Leboreiro et al [49] reporta (para la optimización de un solo objetivo) que valores altos de convergencia de la aptitud solo se logran con bajas probabilidades de mutación, por el contrario valores altos de mutación logran encontrar con mayor rapidez la solución óptima. También observó que el AG algunas veces no encontraba mejores soluciones una vez cerca del óptimo durante muchas generaciones y después saltaba a una mejor solución. La causa de estos periodos es que la población ya perdió su capacidad de búsqueda debido a que todos los individuos son muy similares entre sí por lo que el cruce pierde eficiencia en el proceso de búsqueda, en estos periodos la mutación juega un papel importante.

Si recordamos que para la optimización multiobjetivo el procedimiento de convergencia tiene dos tareas llevar a la población hacia el frente óptimo de Pareto y mantener la diversidad entre los individuos que forman ese frente, un cambio en la probabilidad de mutación puede favorecer al cumplimiento de estas tareas.

El duplicar la población como lo hace Reed, favorece en la búsqueda de individuos para formar el frente de Pareto, ya que como se describió en párrafos anteriores, la probabilidad de mutación se ajusta automáticamente al cambiar el tamaño de la población, todo esto resulta en la minimización del tiempo máquina requerido para encontrar un conjunto de soluciones.

A diferencia del NSGA-II que permanece con el mismo tamaño de la población, pero que por medio del criterio propuesto en varias situaciones sufre cambios drásticos en la probabilidad de mutación, (lo cual resulta favorable en la búsqueda de individuos no-dominados) también ha logrado disminuir el número de generaciones que debe ejecutarse el algoritmo genético.



## Capítulo 4

### Resultados numéricos

Uno de los objetivos de este trabajo es la integración de herramientas de simulación existentes para la síntesis y el diseño óptimo de procesos industriales. Actualmente existen una gran variedad de simuladores que permiten la modelación rigurosa de las operaciones unitarias que son comúnmente utilizadas en la industria. Tal es el caso del simulador Aspen Plus, el cual fue utilizado en la realización de este trabajo, ya que cuenta con modelos rigurosos de diseño y modelos termodinámicos para el cálculo de las propiedades fisicoquímicas de los compuestos involucrados en el proceso. En este caso el simulador es utilizado para evaluar el desempeño de los individuos generados por el algoritmo genético y a partir de los resultados obtenidos de las simulaciones se calcula la aptitud de cada uno de estos. Para realizar lo anterior es necesario contar con un medio de comunicación entre el ag y el simulador, es decir con una interfase.

En este capítulo se describe el diseño de la interfase utilizada para la comunicación entre el algoritmo genético y el simulador. Posteriormente se presenta una metodología para la definición de las funciones objetivo que fueron utilizadas para la realización de este trabajo. En la parte final se presentan los resultados obtenidos después del acoplamiento del ag con el simulador de procesos para la optimización de un sistema de separación en el que se busca maximizar las ganancias y minimizar el impacto ambiental.

#### 4.1 Estableciendo la comunicación entre el algoritmo y el simulador

##### Aspen Plus

El algoritmo utilizado para la realización de este trabajo está compuesto de un programa maestro (nsga2.c) y un conjunto de subrutinas, las cuales se describen brevemente en la Tabla 4-1:

Subrutina	Función
input	Guarda los datos de entrada
warmup_random	Inicializa el generador de números aleatorios.
init	Inicializa a las variables codificadas como binarias.
realinit	Inicializa a las variables codificadas como reales
decode	Decodifica a las variables codificadas como binarias.
func	Evalúa las funciones objetivo y las restricciones.
nselect	Realiza el proceso de selección.
crossover	Realiza el proceso de cruce de un solo punto para variables codificadas como reales.
unicross	Realiza el proceso de cruce uniforme para variables codificadas como reales.
realcross	Realiza el proceso de cruce para variables codificadas como reales.
mutate	Realiza el proceso de mutación para variables codificadas como binarias.
real_mutate	Realiza el proceso de mutación para variables codificadas como reales.
keepalive	Aplica el elitismo

Tabla 4-1. Subrutinas principales del NSGA-II



Como podemos apreciar en la tabla 4-1, en la subrutina *func.h* es en la que el usuario debe definir las funciones objetivo y las restricciones a evaluar. El llamado a la interfase se realiza desde esta subrutina por medio de *\_spawnl*; que es una función definida en lenguaje C, la cual nos permite ejecutar un programa externo. Para mayor información sobre el manejo de esta función, el lector puede consultar los temas de ayuda de lenguaje C.

Esta interfase fue desarrollada por Leboeiro [49] en el paquete computacional Visual Basic versión 5.0; ya que por medio de este lenguaje es posible trabajar con los controles Active X que maneja el simulador de procesos. Estos controles nos permiten modificar las propiedades de objetos o ejecutar métodos que están definidos en el simulador. El funcionamiento de la interfase se presenta en el anexo B.

## 4.2 Definición de las funciones objetivo

La formulación de la función objetivo constituye una etapa clave en la solución de problemas de optimización. Es de suma importancia que las funciones que se desean optimizar se reflejen en una expresión matemática. Como se ha descrito anteriormente, uno de los objetivos de este trabajo, es realizar una optimización multiobjetivo donde las funciones objetivo sean la maximización de las ganancias y la minimización del impacto ambiental. Si retomamos el segundo sistema de separación planteado en el capítulo 1 (Figura 1-2) las ganancias pueden ser obtenidas a través de la cuantificación de los ingresos por la venta de los productos (los cuales dependen de la pureza) y del costo del equipo utilizado para realizar la separación.

El cálculo del costo del equipo fue tomado del trabajo presentado por Leboeiro[49] ya que constituye un buen antecedente en la definición del costo como función objetivo. Debido a esto, en la sección 4.2.1 se la metodología utilizada para el cálculo.

La otra función objetivo considerada fue la cuantificación del riesgo asociado a una planta química o partes de la misma por medio del Índice Mond. En la sección 4.2.2 se describe la secuencia de cálculo utilizada.

Los cálculos necesarios para determinar el valor de las funciones objetivo definidas son realizados en la interfase.

### 4.2.1. Cálculo del costo del equipo

La estimación de los costos de los equipos se realizó mediante la técnica de Costeo por Módulo que involucra un modelo riguroso de costeo [52] Esta técnica de costeo relaciona todos los costos posteriores a la compra del equipo evaluado a ciertas condiciones base, está basado en los siguientes cinco pasos para estimar el costo de cada equipo [53].

1. Calcular el costo base por módulo del equipo ( $C_p$ ), lo que representa el costo de compra del equipo para el caso base. El caso base considera que el equipo esta construido con acero al carbón y opera a presión ambiente.
2. Definir el factor de material ( $F_M$ ) y el factor de presión ( $F_p$ ) para las condiciones deseadas de construcción del equipo y de operación.

3. Calcular el factor de costo del módulo base ( $F_{BM}^{\circ}$ ), a partir de los valores de los factores de material y de presión.
4. Calcular el costo por módulo del equipo para las condiciones reales ( $C_{BM}^{\circ}$ ).
5. Actualizar el costo a valor presente del año del cual son los datos.

En el anexo C se presentan las ecuaciones para calcular cada uno de los factores involucrados en la metodología.

#### **4.2.2. Cuantificación del riesgo**

Para este apartado se utilizó un método de evaluación de riesgo semi-cuantitativo, el Índice Mond. Por medio del cual obtuvimos una clasificación relativa del riesgo asociado con el sistema de separación planteado en la figura 1-2 de este escrito.

Este método fue desarrollado por técnicos de Imperial Chemical Industries PLC (ICI) a partir del índice DOW[54]. La principal diferencia entre estos índices, es que el Índice MOND considera la toxicidad de las sustancias presentes introduciendo este parámetro como un factor independiente, considerando los efectos de las sustancias tóxicas por contacto cutáneo o por inhalación. La secuencia de cálculo de este índice se muestra en la Figura 4-1. El sistema de separación fue dividido en cuatro unidades, dos columnas y dos tanques para el almacenamiento de los productos (heptano y tolueno) El índice de cada unidad fue calculado por separado. En el apéndice B se describen las consideraciones que se tomaron en cuenta para el cálculo del índice de riesgo global.

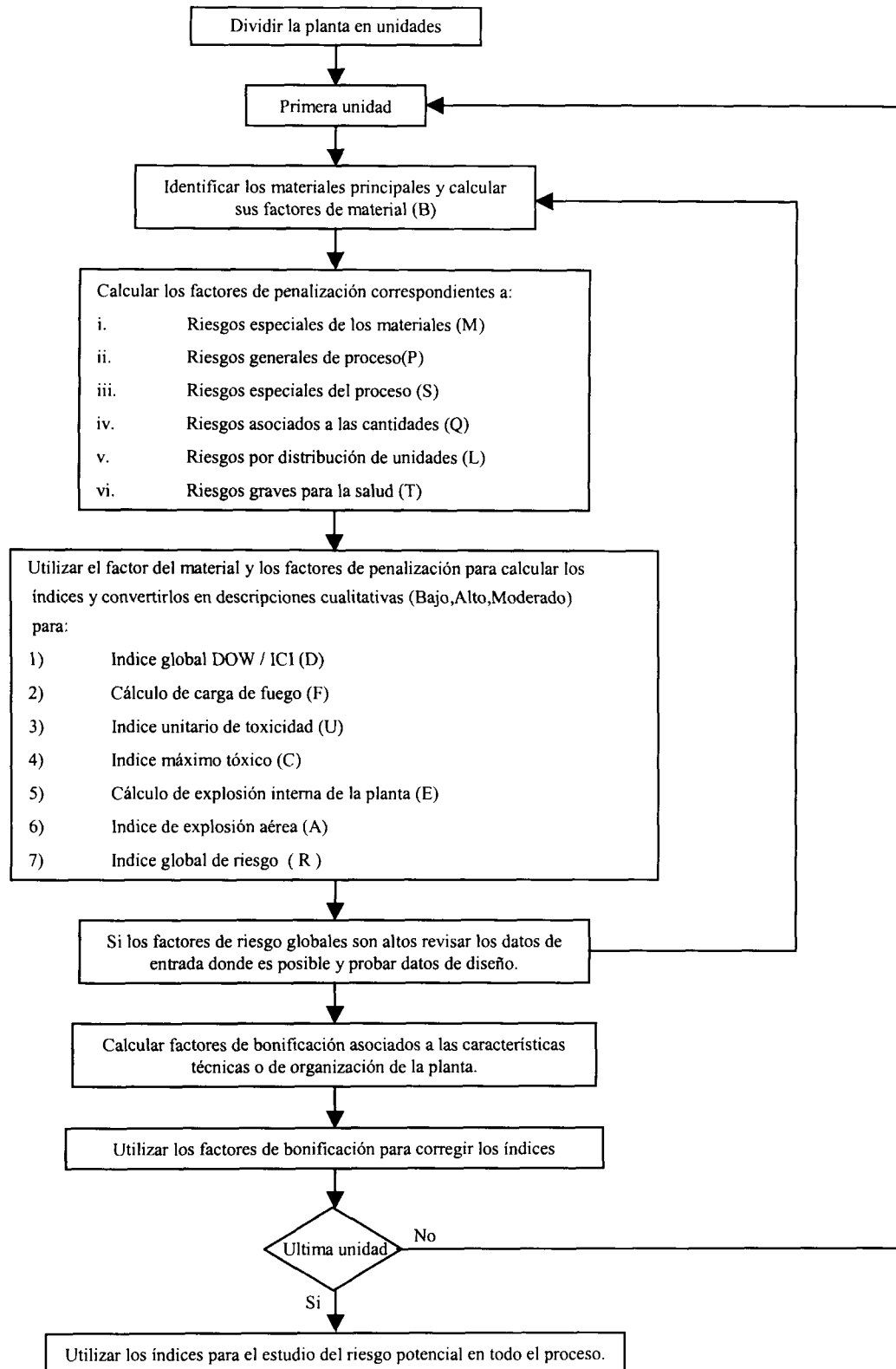


Figura 4-1 Secuencia de cálculo del índice Mond

### 4.3 Experimentos y resultados

#### 4.3.1. Problema 1, funciones objetivo sencillas.

El primer problema consiste en la optimización de 2 funciones objetivo sencillas:

$$f[0] = x[0] \quad f[1] = \frac{(1 + x[1])}{x[0]}$$

sujeto a las restricciones:

$$\text{restricción}[0] = -1 + \frac{(x[1] + 9x[0])}{6} \quad \text{restricción}[1] = 9 - f[1]$$

Ambas variables fueron codificadas como reales; primero se realizaron dos experimentos con el programa original, con 20 y 100 generaciones, utilizando una población de 50 individuos. Posteriormente se hizo el experimento con el código que tiene implementado el criterio de paro propuesto. Los resultados se presentan a continuación:

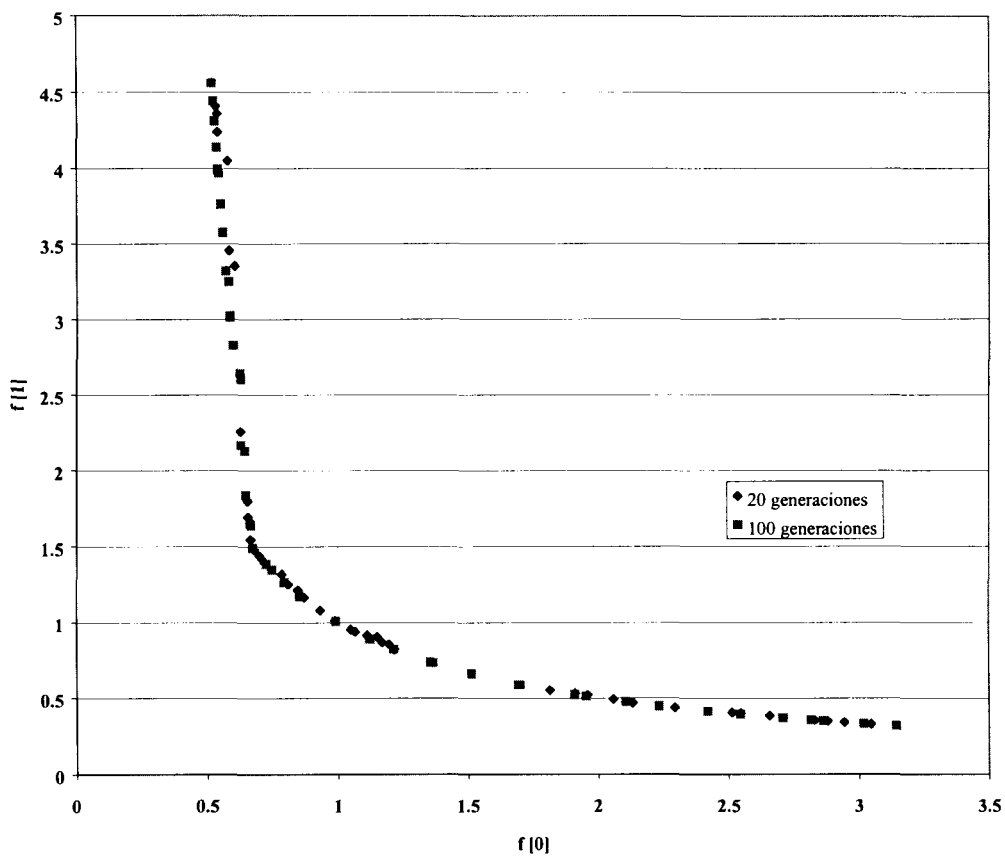
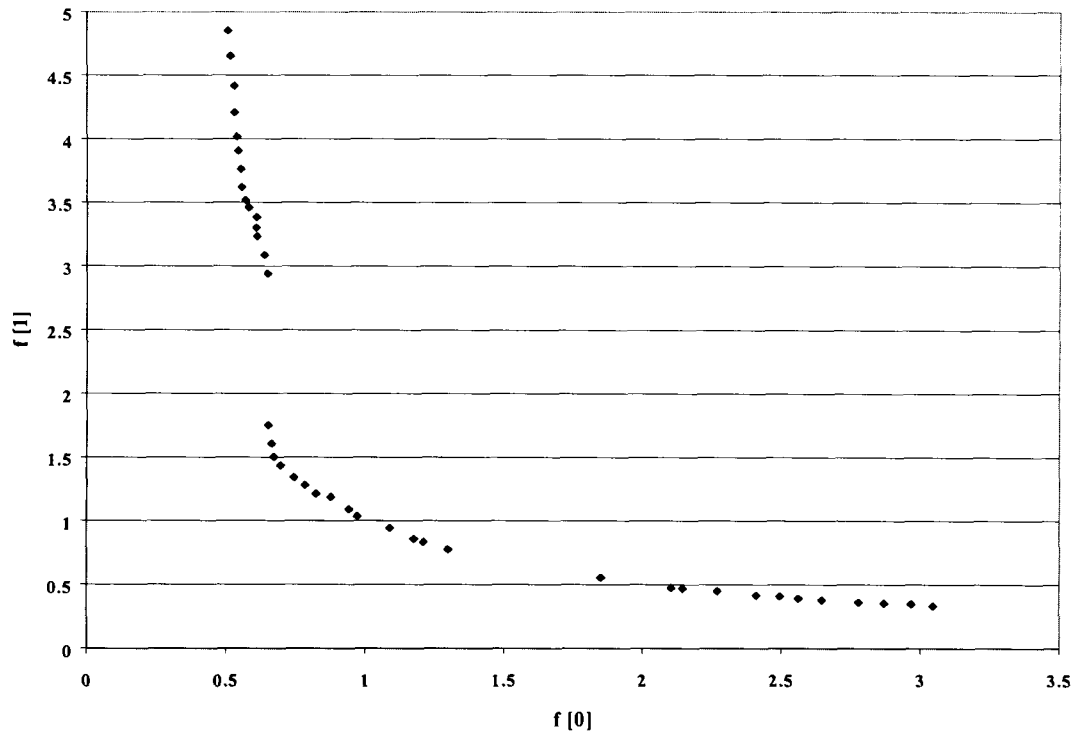


Figura 4-2 Resultados problema 1, sin criterio de paro después de 20 y de 100 generaciones.



**Figura 4-3. Resultados problema 1 con criterio de paro después de 17 generaciones**

Podemos apreciar en la figura 4-2 que el sistema a las 20 generaciones de evolución, había logrado un buen grado de convergencia; el conjunto de soluciones obtenidas es muy parecido al que se obtuvo después de dejarlo evolucionar 80 generaciones más. En la Figura 4-3 podemos observar que el criterio de paro tuvo un buen desempeño, ya que en un

Sin criterio de paro		Con criterio de paro	
f [0]	f [1]	f [0]	f [1]
0.516	4.563	0.516	4.651
0.551	3.717	0.556	3.620
0.649	1.777	0.651	1.749
1.063	0.941	1.089	0.943
2.114	0.474	2.105	0.475
2.488	0.401	2.495	0.408
2.994	0.333	3.047	0.333

**Tabla 4-2. Comparación entre las soluciones obtenidas con y sin criterio de paro problema 1.**

menor número de generaciones logró obtener un conjunto de soluciones muy similares a las obtenidas por el algoritmo original en 20 generaciones; evitando con esto evaluaciones

innecesarias una vez que el sistema alcanzó un cierto número de individuos no dominados. Un factor que aceleró este proceso de convergencia fue la probabilidad de mutación de 0.50, aplicada a la población, en este problema esa condición se presentó 7 veces.

Para este experimento, los parámetros de ajuste del criterio de paro, quedaron con los siguientes valores:

Parámetro	Valor
$\varepsilon$	0.000001
$\Phi$	0.40
$\Omega$	0.30
$\varepsilon_2$	0.10
$\lambda$	0.80
$\psi$	0.01

Tabla 4-3 Parámetros de ajuste del criterio de paro problema 1

#### 4.3.2. Problema 2, un sistema de separación (Pureza vs Costo).

Consiste en un problema que ya involucra el acoplamiento del algoritmo genético con el simulador. Se trata de maximizar la pureza del fenol que sale por la corriente del destilado de la segunda columna del sistema planteado en la Figura 4-4.

Hay que aclarar que la versión original del problema seleccionado para realizar la experimentación, fue tomada de Leboeiro [49]

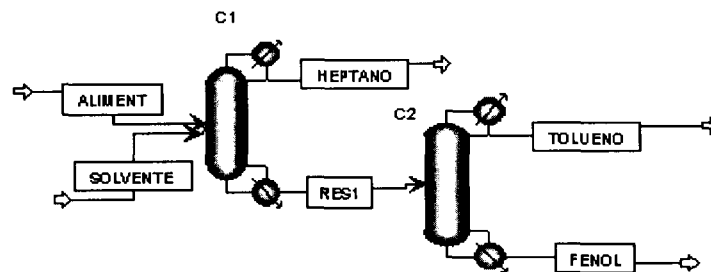


Figura 4-4 Sistema de separación propuesto para el problema 2.

Las condiciones de la corriente a separar (ALIMENT), son temperatura 200°F, presión 20 psia, y contiene 200 lbmol/hr de n-heptano y 200 lbmol/hr de tolueno. La corriente que va a ser utilizado como solvente se encuentra a 220°F y 20 psia, y tiene una composición de 1188 lbmol/hr de fenol y 12 lbmol/hr de tolueno. La presión de la columna permanece constante en 20 psia. Ambas columnas fueron simuladas con el bloque RADFRAC, que es un modelo riguroso de destilación con el que cuenta el Aspen Plus. El cálculo de las propiedades termodinámicas de la fase vapor se realizó mediante la ecuación de estado de Redlich-Kwong-Soave. Para calcular la hidráulica de las columnas se utilizó el método de

Fair y se consideró el uso de platos perforados, fracción de superficie perforada de 0.12, espacio entre platos de 24 pulgadas, y una fracción de inundación de 0.8  
 Las variables utilizadas para la optimización se muestran en la Tabla 4-4 y Tabla 4-5.

Variable	Tipo de codificación	Rango
Reflujo columna 1	real	0-10
Reflujo columna 2	real	0-5
Flujo de solvente (lbmol/hr)	real	190-330

Tabla 4-4 Variables de optimización, codificadas como reales

Variable	Tipo de codificación	Número de bits	Rango
Número de platos columna 1	Binaria	6	0-63
Plato de alimentación de la mezcla a separar, columna 1	Binaria	6	0-63
Plato de alimentación del solvente, columna 1	Binaria	6	0-63
Número de platos, columna 2	Binaria	5	0-31
Plato de alimentación de la mezcla a separar, columna 2	Binaria	5	0-31

Tabla 4-5 Variables de optimización, codificadas como binarias

Con el fin de limitar configuraciones no factibles, Leboreiro y Acevedo [49] utilizaron la *Codificación de Rango Resultante*, por medio de la cual es posible reflejar las siguientes consideraciones: el mínimo número de platos fue fijado en 20, no se alimentará ninguna corriente en los primeros ni en los últimos 5 platos de la columna y para la primera columna, debe haber por lo menos 3 platos de separación entre ambas alimentaciones. El cálculo del número de platos y el plato de alimentación para cada columna queda:

$$N_{platos} = iN_{platos} + 20$$

$$Plata\ lim = Int\left( (N_{platos} - 11) * \frac{iPlata\ lim}{63} + 6 \right)$$

$$Platsolv = Plata\ lim - 3 - Int\left( (Plata\ lim - 6 - 3) * \frac{iPlatsolv}{63} \right)$$

Los valores de los operadores genéticos utilizados fueron

Tamaño de la población	50
Probabilidad de cruce	0.5
Probabilidad de mutación (binaria)	0.035
Probabilidad de mutación (real)	0.065
Índice de distribución para el cruce de variables reales	35
Índice de distribución para la mutación de variables reales	55

Tabla 4-6 Valores de los operadores genéticos

Primero se realizó el experimento con el algoritmo original, el cual se dejó evolucionar durante 200 generaciones, la duración fue de aproximadamente 6 horas 32 minutos. Los resultados obtenidos se presentan a continuación:

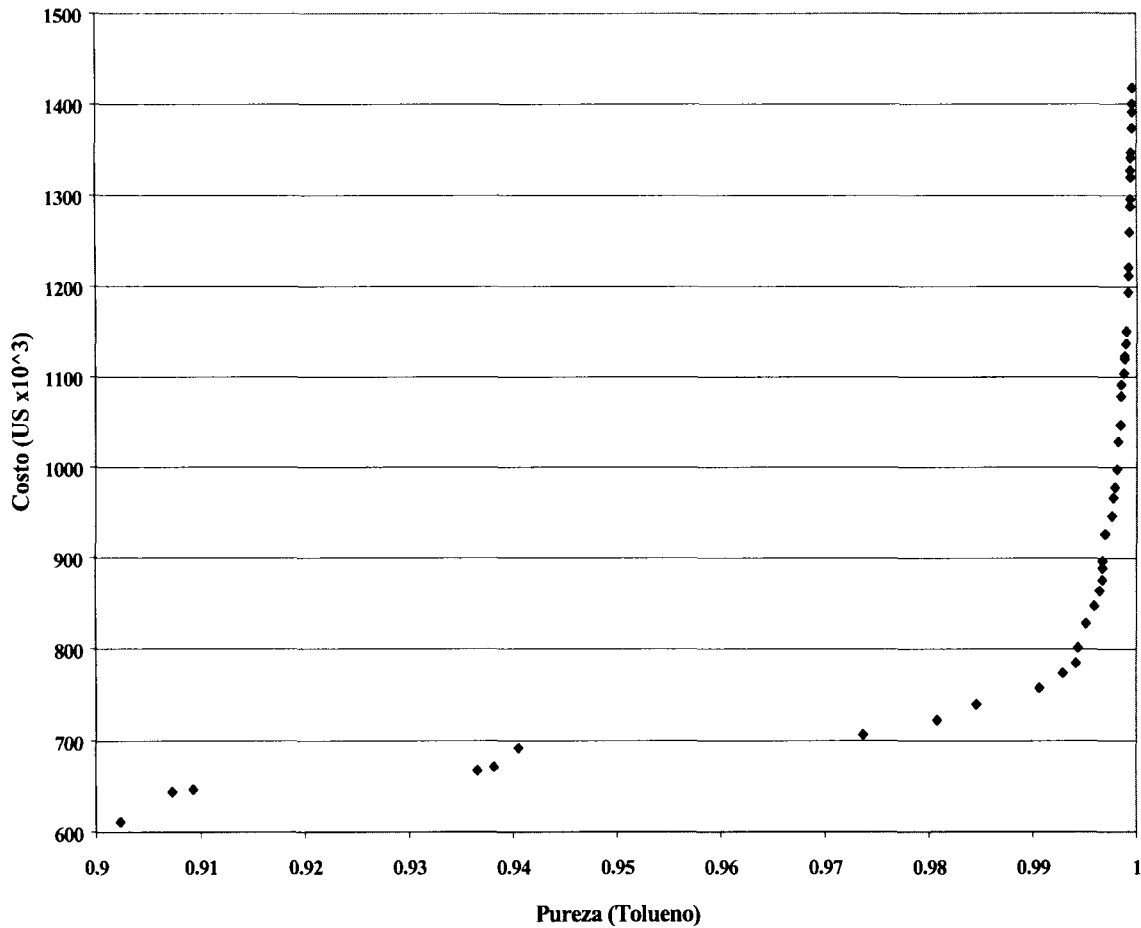


Figura 4-5 Soluciones obtenidas después de 200 generaciones. Problema 2

La razón por la cual existan pocos individuos en el rango de pureza de 0.90 y 0.98 es debido a que el algoritmo genético implementado en este trabajo considera como mejor solución a aquella que se encuentre en un área con menor densidad de población (crowding distance).



De estas soluciones, los diseños de equipo son:

Costo (USD)	Pureza (%)	Nplatos C1	Reflujo C1	Fsolvente (lbmol/hr)	Reflujo C2	Nplatos C2
611,368.94	0.902312	40	2.835479	212.20985	0.140252	20
692,221.56	0.940523	48	2.769701	211.49011	0.272537	20
706,620.75	0.973671	41	2.953743	206.77046	0.494116	20
739,997.88	0.984565	42	2.953743	214.72165	0.654672	22
757,677.25	0.990649	47	2.994827	231.50964	0.536105	24
863,967.13	0.996420	64	3.397457	206.76051	0.575353	27
997,595.31	0.998095	66	5.08176	212.37343	0.629635	28
1,346,801.00	0.999487	83	8.720819	212.96967	0.672674	36

**Tabla 4-7. Diseños obtenidos con el algoritmo original. Problema 2**

Podemos apreciar que el algoritmo se comportó adecuadamente, ya que al aumentar la pureza del producto, aumenta el costo del equipo necesario.

Después se repitió el experimento, con el criterio de paro implementado en el algoritmo genético, donde los parámetros de ajuste quedaron como se muestra en la tabla Tabla 4-8. Los resultados se muestran en la Figura 4-6 y en la Tabla 4-9. La duración del experimento fue aproximadamente de 37 minutos durante los cuales el algoritmo aplicó la probabilidad de mutación de 0.50 en tres ocasiones.

Parámetro	Valor
$\epsilon$	0.000001
$\Phi$	0.40
$\Omega$	0.30
$\epsilon_2$	0.10
$\psi$	0.01

**Tabla 4-8. Valores de los parámetros del criterio de paro**

En este experimento, el valor del factor  $\lambda$  fue diferente, ya que para la ecuación 3.9 fue establecido como menor a 0.30 y para la ecuación 3.10 fue establecido como mayor que 0.50.

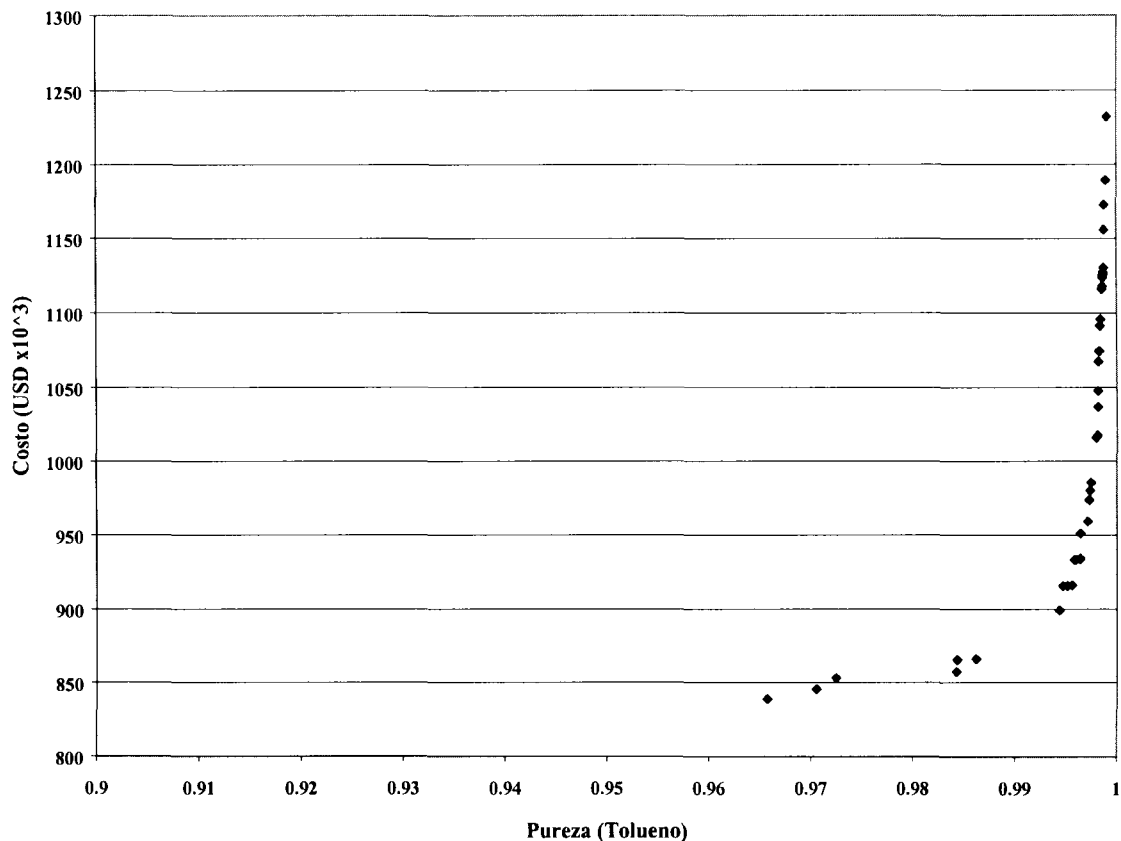


Figura 4-6. Soluciones obtenidas con el criterio de paro (1)

Los diseños obtenidos fueron

Costo (USD)	Pureza	Nplatos C1	Reflujo C1	Fsolvente lbmol/hr	Nplatos C2	Reflujo C2
838,811.69	0.965720	29	4.5773	214.8212	30	0.8318
852,878.81	0.972460	32	4.8322	214.3261	25	0.8254
866,012.44	0.986160	36	4.8974	212.1719	22	0.8054
915,655.81	0.994710	49	4.8820	198.0167	21	0.8006
933,337.31	0.995840	49	4.8398	214.3259	26	0.8254
951,079.13	0.996420	61	4.5778	195.5123	21	0.8318
1,172,719.63	0.998820	76	6.5942	214.8153	33	0.8269
1,232,027.13	0.999070	80	7.0277	214.2472	37	0.8255

Tabla 4-9. Diseños obtenidos con el criterio de paro implementado

Con propósitos de comparación, el experimento anterior fue repetido, variando un solo parámetro del criterio de paro  $\Phi=0.60$ , los resultados se muestran en la Tabla 4-10 y la

Figura 4-7 La duración de este experimento fue de 24 minutos durante los cuales el algoritmo aplicó 3 veces la probabilidad de mutación de 0.50.

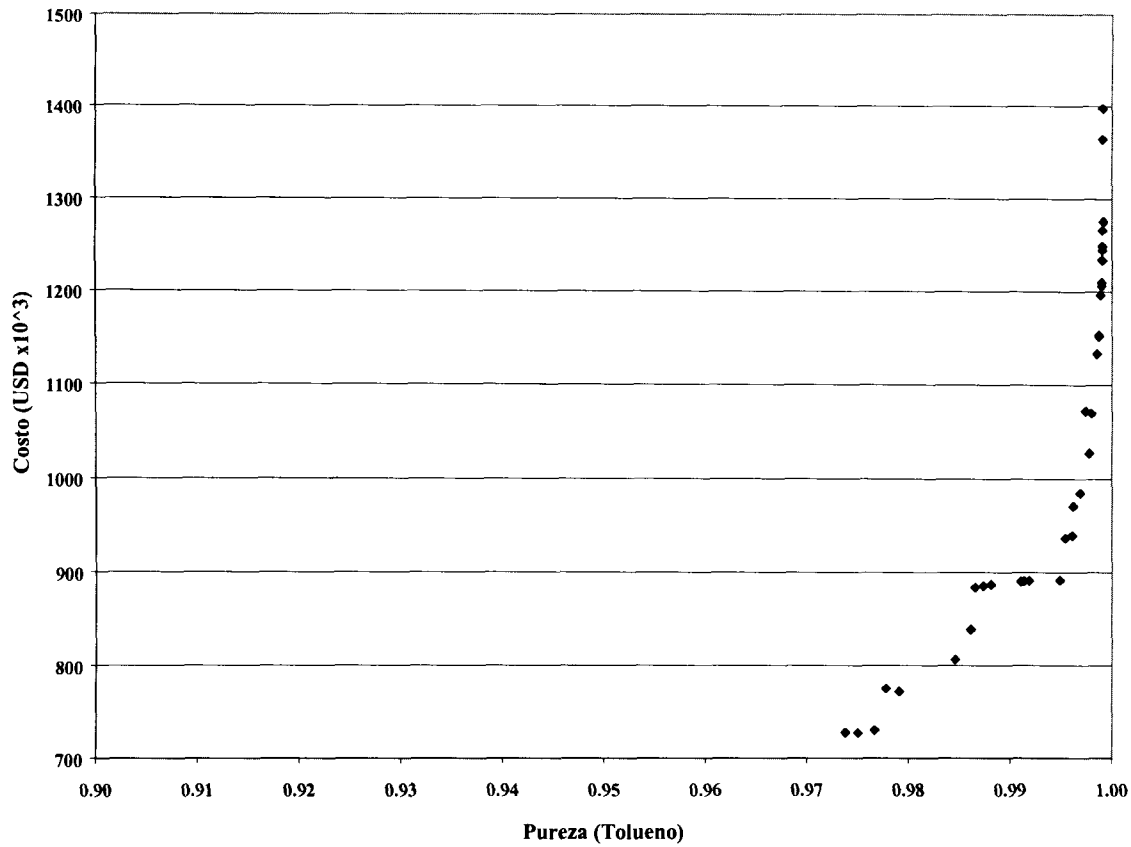


Figura 4-7 Soluciones obtenidas, con parámetro  $\Phi=0.60$

Costo (USD)	Pureza	Nplatos C1	Reflujo C1	Fsolvente lbmol/hr	Nplatos C2	Reflujo C2
727437.63	0.97379	47	2.7920	211.7009	22	0.5307
774871.38	0.97779	37	3.2046	211.6919	42	0.5308
806273.13	0.98459	51	2.7275	232.7607	22	0.9640
883833.56	0.98656	66	2.8065	212.4049	49	0.5326
890499.44	0.99104	66	2.8065	229.7565	49	0.5326
936855.69	0.99546	81	2.8241	234.2591	22	0.9563
1070126.88	0.99798	81	4.4007	235.1803	48	0.5207
1275956.75	0.99916	83	7.5269	211.5672	49	0.5308

Tabla 4-10. Diseños obtenidos variando el parámetro  $\Phi=0.60$

Podemos apreciar que en ambos casos el algoritmo de paro tuvo buen desempeño, ya que el número de generaciones requeridas por el algoritmo para encontrar un conjunto de soluciones óptimas, fue menor en comparación al requerido por el algoritmo original; para el ejemplo donde  $\Phi=0.40$  el algoritmo evolucionó durante 23 generaciones y para  $\Phi=0.60$  fueron 13 generaciones.

**4.3.3. Problema 3a, sistema de separación con recirculación (Costo vs Riesgo).**

Este problema involucra la cuantificación del riesgo asociado al sistema de separación mostrado en la Figura 4-8. como criterio de optimización. Este sistema considera el retorno de la corriente de salida de la segunda columna al sistema, lo cual se refleja en una disminución en la demanda de solvente. Los parámetros del criterio de paro se muestran en la Tabla 4-11

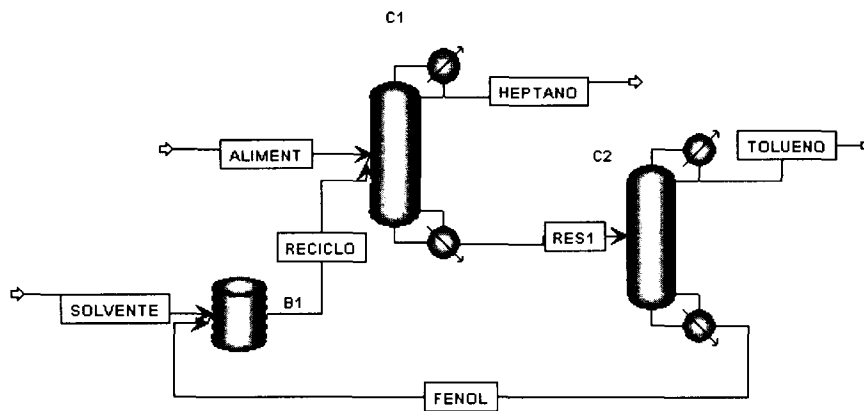
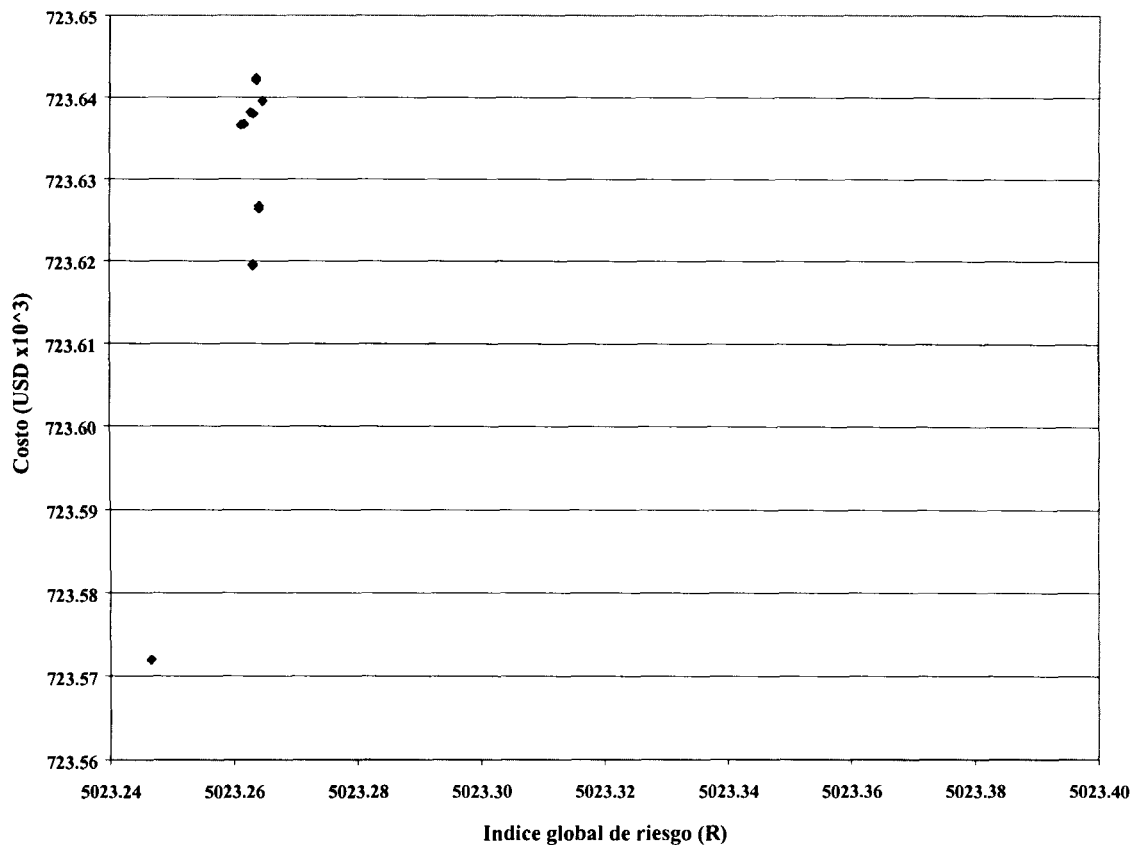


Figura 4-8 Sistema de separación, segunda opción.

Parámetro	Valor
$\epsilon$	0.000001
$\Phi$	0.40
$\Omega$	0.30
$\epsilon_2$	0.10
$\psi$	0.01

Tabla 4-11. Valores de los parámetros del criterio de paro

La duración aproximada de este experimento fue de aproximadamente 8 horas y 40 minutos, el algoritmo evolucionó un total de 150 generaciones y la probabilidad de mutación de 0.50 fue aplicada dos veces, los resultados se presentan en la figura 4-9



**Figura 4-9 Soluciones del problema Costo vs Riesgo**

La Figura 4-9 nos muestra que las funciones objetivo involucradas no se contraponen, sino que las dos van en el mismo sentido, un aumento en el costo conlleva a un aumento en el riesgo y viceversa. La razón es debido a que el costo es función del número de platos y del volumen y el riesgo está en función de la cantidad de masa que se encuentre contenida en los equipos que forman el sistema de separación. De la gráfica se distingue una solución como “óptima”. El diseño “óptimo” del sistema se presenta en la Tabla 4-12. Esto nos llevó a buscar la definición de otra función objetivo, la cual se presenta en la siguiente sección.

Costo (USD)	Indice global de riesgo ( R )	Nplatos C1	Reflujo C1	Nplatos C2	Reflujo C2	Flujo de solvente (lbmol/hr)
723,572	5023.2466	20	1.40604	20	1.9626	162.5474

**Tabla 4-12 Diseño “óptimo” para la optimización de Costo vs. Riesgo**

De la gráfica podemos apreciar la reducción en el flujo de solvente utilizado (comparado con los diseños presentados en la sección anterior), esto debido a que este sistema incluye la recirculación de la corriente FENOL (Figura 4-8) al sistema.

#### 4.3.4. Problema 3b, sistema de separación con recirculación (Ganancias vs Riesgo).

En esta segunda versión del problema planteado en la sección anterior, se utilizaron como funciones objetivo, la minimización del índice global de riesgo y la maximización de las ganancias totales. Las ganancias totales fueron definidas como la diferencia entre el costo del sistema y las ganancias por ventas del heptano y tolueno. El costo de venta por libra de producto varió dependiendo del grado de pureza con el que fueran obtenidos. Cabe aclarar que dichos costos de venta no son los reales, si no que fueron ajustados, esto con la finalidad de que las ganancias por la venta de los productos, tuvieran un orden de magnitud similar al del costo del equipo. En la tablas Tabla 4-13 y Tabla 4-14 se muestran la relaciones pureza-precio utilizada para este experimento.

Heptano	
Pureza	Precio (US)/lb
$\geq 0.9$ y $<0.95$	0.005
$\geq 0.95$ y $<0.975$	0.006
$\geq 0.975$	0.0065

Tabla 4-13 Relación pureza-precio de venta para el heptano.

Tolueno	
Pureza	Precio (US)/lb
$\geq 0.9$ y $<0.95$	0.0045
$\geq 0.95$ y $<0.97$	0.0055
$\geq 0.97$	0.006

Tabla 4-14 Relación pureza-precio de venta para el tolueno

La duración aproximada del experimento fue de 7 horas y 32 minutos, el algoritmo evolucionó un total de 133 generaciones; durante este tiempo la probabilidad de mutación de 0.50 fue aplicada 2 veces. Los parámetros utilizados en el criterio de paro fueron los mismos que en el problema de la sección 4.3.3.

Los resultados se muestran en la Figura 4-10 y en la Tabla 4-15.

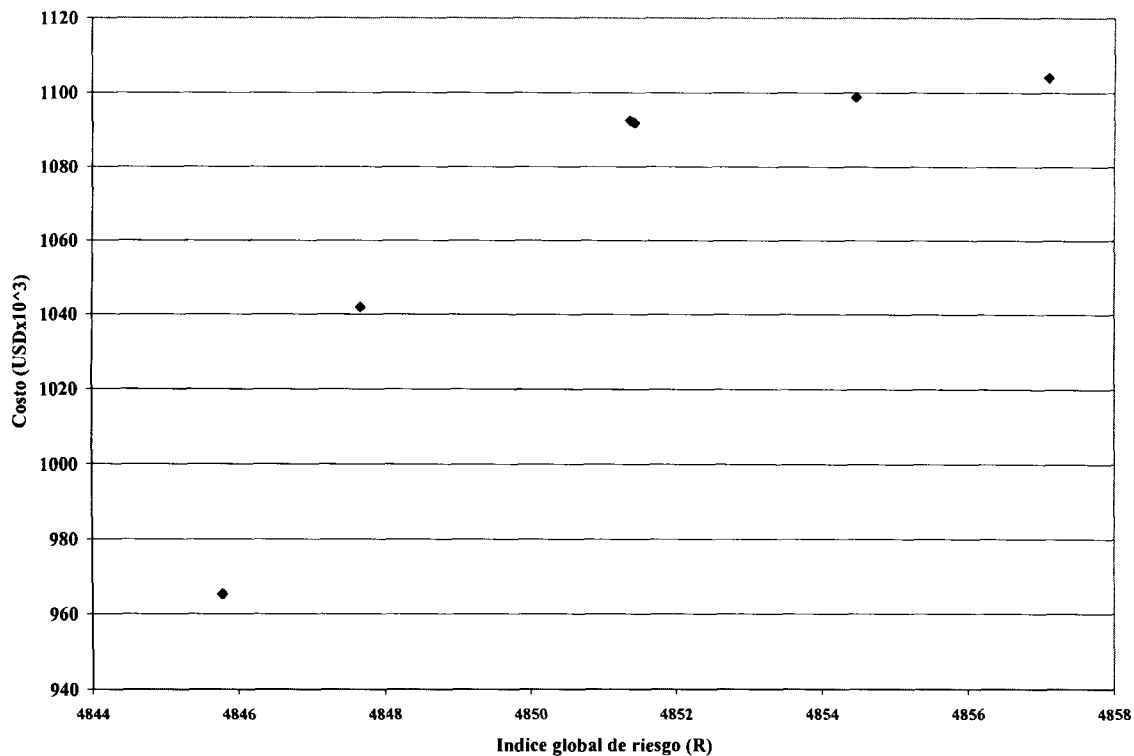


Figura 4-10 Soluciones del problema Ganancias vs. Índice de riesgo.

Los diseños resultantes se presentan en la Tabla 4-15

Ganancias (USD)	Índice de riesgo global (R)	Nplatos C1	Palim C1	Fsolvente (lbmol/hr)	Nplatos C2	Palim C2	Pureza (%) Heptano	Pureza (%) Tolueno
965,331.44	4845.76953	28	14	182.0865	22	11	0.97443	0.96942
1,091,927.50	4851.42285	28	15	182.0865	22	11	0.97897	0.97487
1,098,798.75	4854.46143	28	15	182.0865	26	12	0.97897	0.97487
1,104,110.50	4857.09717	30	15	182.0865	24	12	0.97926	0.97522

Tabla 4-15 Diseños obtenidos para el problema Ganancias vs. Riesgo

De los resultados obtenidos podemos apreciar que el algoritmo se desempeñó satisfactoriamente, ya que un aumento en la pureza del producto corresponde a un aumento en las ganancias, aunque esto se ve reflejado en un discreto aumento del índice de riesgo global del sistema.

## 4.4 Análisis de resultados

A continuación se muestra un análisis de los resultados obtenidos en los tres casos resueltos.

El primer caso consistió en la minimización de 2 sencillas funciones. El criterio de paro tuvo un buen desempeño ya que en 17 generaciones logró obtener un conjunto de soluciones muy parecidas a las obtenidas con el algoritmo original después de 100 generaciones.

En el segundo caso resuelto, ya fue acoplado el algoritmo genético con el simulador de procesos. En este ejemplo las funciones objetivo son la minimización del costo del equipo y la maximización de la pureza del tolueno que sale como destilado de la segunda columna. El criterio de paro tuvo buen desempeño ya que en un menor número de generaciones alcanzó un conjunto de soluciones muy similares a las obtenidas por el algoritmo original después de 200 generaciones. Con el factor  $\Phi=0.4$  el algoritmo solo requirió de 23 generaciones (37 minutos) para obtener un conjunto de soluciones óptimas. Modificando el factor  $\Phi$  a un valor de 0.60, se obtiene una reducción en el tiempo requerido para el experimento, ya que únicamente requirió de 24 minutos (13 generaciones). En ambos casos, el algoritmo aplicó una probabilidad de mutación de 0.50.

El tercer problema ya involucra como función objetivo la minimización del impacto ambiental. Primero se realizó el experimento utilizando como funciones objetivo la cuantificación del riesgo asociado al sistema de separación planteado y el costo del equipo. En ese ejemplo, se distingue sólo una solución como óptima. Por lo que fue necesario formular otra función objetivo por medio de la cual obtuviéramos un conjunto de soluciones. Esta alternativa consistió en considerar las ganancias totales del proceso como función objetivo, representadas como el costo del equipo menos las ventas de los productos (heptano, tolueno) cuyo precio depende de la pureza con la que hayan sido obtenidos. El algoritmo presentó un buen desempeño, aunque con respecto al índice de riesgo global, el rango que presentan el conjunto de soluciones obtenidas es pequeño.





## Capítulo 5

### Conclusiones y Trabajo futuro

#### 5.1 Conclusiones

Después de la realización de este trabajo podemos concluir que:

- Se logró el acoplamiento de un algoritmo genético con el simulador Aspen Plus para la solución de problemas de optimización multiobjetivo, utilizando dos tipos de codificación para las variables de decisión la real y la binaria.
- Por medio de una serie de corridas experimentales se logró determinar valores adecuados para los operadores genéticos que rigen el desempeño del algoritmo genético utilizado. La probabilidad de mutación fue el parámetro que resultó tener más influencia sobre el desempeño del algoritmo. Se observó que para ambos tipos de codificación con valores de probabilidad de cruce arriba de 0.5 se obtuvieron los mejores resultados. Para la codificación binaria, el rango de valores para la probabilidad de mutación que tuvo buen desempeño está entre 0.03 y 0.3. Con respecto a la codificación real este rango se encuentra entre 0.001 y 0.3.
- El criterio de paro implementado, se desempeñó satisfactoriamente ya que logró encontrar un conjunto de soluciones óptimas y detener el algoritmo genético para evitar evaluaciones innecesarias; el caso más relevante fue cuando el criterio de paro logró reducir el tiempo requerido para obtener el conjunto de soluciones en un 80%. Debido a esto podemos considerar al criterio de paro como una herramienta que mejora significativamente el desempeño del algoritmo genético.
- Se obtuvieron resultados satisfactorios de la optimización simultánea de dos aspectos importantes en el área de ingeniería de procesos: el económico y el riesgo asociado a un proceso de separación.

#### 5.2 Trabajos futuros

Este trabajo de investigación representa el primer acoplamiento (del cual el autor tiene conocimiento) de una técnica de optimización multiobjetivo (con algoritmos genéticos) con el simulador de procesos Aspen Plus, esto puede considerarse como una línea importante de investigación para trabajos futuros. Los objetivos propuestos fueron alcanzados, obteniéndose resultados satisfactorios, sin embargo, durante la realización de esta tesis surgieron algunos aspectos, los cuales pueden ser considerados como áreas de oportunidad para la continuación de esta investigación:

- Realizar los experimentos propuestos en la sección 4 de este documento con poblaciones más grandes, esto con la finalidad de obtener conjuntos de soluciones óptimas mejor definidas.

- Realizar un estudio más completo de la influencia que tiene el valor de los parámetros de ajuste del criterio de paro propuesto sobre su desempeño, esto con la finalidad de encontrar valores más adecuados.
- Durante la realización de los experimentos se observó que aplicar una probabilidad de mutación grande cada determinado número de generaciones ayudaba a encontrar las condiciones requeridas para que el criterio de paro se cumpliera. Pero en algunos otros la probabilidad de mutación era demasiado grande, por lo que las condiciones del criterio de paro tardaban en aparecer; por lo que sería conveniente incluir un proceso de mutación variable dependiendo del número de individuos no dominados presentes en la población, esto con el fin de mejorar el desempeño del criterio de paro propuesto.
- Realizar la optimización de un sistema de separación más complejo, en el cual se incluyan para la cuantificación del riesgo los factores de bonificación y en el caso del criterio económico, los precios reales de productos.
- Implementación del sistema AG-Interfase-Aspen Plus para la solución de problemas de optimización y diseño de procesos, en los que se incluya como criterio de optimización la minimización del impacto ambiental, considerando alguna de las metodologías presentadas en la sección 2 de este documento.
- Durante el acoplamiento del algoritmo genético con el simulador de procesos, se detectó un problema relacionado con las direcciones de memoria de la computadora, el cual detiene la ejecución del algoritmo (sin afectar el resultado del experimento que se está llevando a cabo) y hace necesaria la intervención del usuario para que la corrida se reanude. Se recomienda consultar con un experto en lenguajes de programación para revisar las interacciones entre el AG, la interfase y el simulador de procesos.

## Bibliografía

- [1] **Holland J.** "Adaptation in natural and artificial systems". Ann Arbor: The University of Michigan Press, USA, 1975.
- [2] **Goldberg D. E.** "Genetic algorithms in search, optimization and machine learning". Addison – Wesley Publishing Company, Inc. 1989
- [3] **Goldberg D. E, Deb K., and Clark J. H.** "Genetic algorithms, noise, and the sizing of populations". *Complex Systems* 6(4), pp 333-362.
- [4] **Deb K. and Agrawal R.B.** "Simulated binary crossover for continuous search space". *Complex Systems* 9 (2), pp 115-148, 1985.
- [5] **Deb K. and Kumar A.** "Real-coded genetic algorithms with simulated binary crossover: Studies on multi-modal and multi-objective problems". *Complex Systems* 9(6), pp 431-454, 1995.
- [6] **Wright, A.** "Genetic algorithms for real parameter optimization". In *foundations of Genetic Algorithms 1 (FOGA-1)* pp. 205-218, 1991.
- [7] **Goldberg D. E.** "Real-coded genetic algorithm, virtual alphabets, and blocking". *Complex Systems* 5(2), pp 139-168, 1991.
- [8] **Voight H.M., Muhlenbein, H.** "Fuzzy recombination for the Breeder Genetic Algorithm". In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp. 104-111, 1995.
- [9] **Ono I. And Kobayashi S.** "A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover". In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp. 246-253, 1997
- [10] **Michalewicz Z.** Genetic Algorithms + Data Structures= Evolution Programs. Berlin: Springer- Verlag, 1992.
- [11] **Bhaskar V, Gupta S K, Ray A K.** "Multiobjective optimization of an industrial wiped film poly(ethylene terephthalate) reactor: some further insights" *Computers and Chemical Engineering* 25, pp 391-407. 2001.
- [12] **Ravi G, Gupta S K, Ray M B** "Multiobjective optimization of cyclone separator using genetic algorithm" *Industrial Engineering Chemical Research*, 39, pp 4272-4286, 2000.
- [13] **Deb K.** "Multi- Objective Optimization using Evolutionary Algorithms" Jonh Wiley & Sons, 2001.
- [14] **D. Cavicchio.** "Adaptive search using simulated evolution". Unpublished doctoral dissertation, University of Michigan, Ann Arbor. 1970.
- [15] **Dejong A. K.** "An analysis of the behavior of a class of genetic adaptative systems". PhD thesis, University of Michigan, 1975.
- [16] **Goldberg D. E. and Richardson J.** "Genetic algorithm with sharing for multimodal function optimization". In J.J. Grefenstette editor. Genetic algorithm and their applications: Proceeding of the second international conference on genetic algorithm. 1987.
- [17] **Srinivas N. and Deb K.** "Multiobjctive Optimization using nondominated sorting in genetic algorithms". *Journal of Evolutionary Computation*, Vol. 2.

- [18] **Coello C. A.** "An updated of evolutionary multiobjective optimization techniques: state of the art and future trends". Laboratorio nacional de informatica avanzada.
- [19] **Mahfoud S. W.** "Niching methods for genetic algorithms". IlliGAL report No. 95001. PhD thesis, University of Michigan, 1995.
- [20] **Jakob, Gorges and Blume.** "Application of genetic algorithms to task planning and learning". Amsterdam, 1992.
- [21] **Jones, Brown and Clark.** "Searching databases of two dimensional and three dimensional chemical structures using genetic algorithms". San Mateo, California. 1993.
- [22] **Lis and Eiben.** "A multisexual genetic algorithms for multiobjective optimization", Nagoya, Japan 1996.
- [23] **Syswerda G. and Palmucci J** "The application of genetic algorithms to resource scheduling". San Mateo California 1991.
- [24] **Ritzel B.J, Eheart J. W. and Runjithan S.** "Using genetic algorithms to solve a multiobjective groundwater pollution containment problem" Water resources research, May 1994.
- [25] **Cvetkovic, Parmee and Webb.** "Multiobjective optimisation and preliminary airframe design". April 1998.
- [26] **Tamaki H. , Kita and Kobayashi.** "Multiobjective optimization by genetic algorithms: a review". *Proceedings of the 1996 International Conference on Evolutionary Computation.* Nagoya, Japan. IEEE, 1996.
- [27] **Tamaki H, Mori M and Ogai H.** "Multicriteria optimization by genetic algorithms. A case of scheduling in hot rolling process". *In proceedings of the 3<sup>rd</sup>. APORS,* 1995.
- [28] **Goldberg D. E. and Deb K.** "A comparison of selection schemes used in genetic algorithms, foundations of genetic algorithms" 1991.
- [29] **Fonseca C. M and Fleming P.J.** "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization". *Proceedings of the fifth international conference on genetic algorithms.* San Mateo, CA. 1993.
- [30] **Chen Tan K. and Li. Y.** "Multiobjective genetic algorithm based time and frequency domain design unification of linear control system". Technical report CSC-97007. University of Glasglow, Glasglow Scotland 1997.
- [31] **Chipperfield A. J. and Fleming P.J.** "Gas turbine engine controller design using multiobjective optimization genetic algorithm". Proceeding of the first IEE/IEEE international conference on genetic algorithm in engineering systems: innovations and applications. Galesia 95. IEEE.
- [32] **Rodriguez-Vazquez K, Fonseca C.M and Fleming P.J** "Multiobjective optimization genetic programming: A nonlinear system identification application". Stanford University California, July 1997.
- [33] **Coello C.A.** "An empirical study of evolutionary techniques for multiobjective optimization in engineering design". PhD thesis, department of computer science. Tulane University. New Orleans, Louisiana. April 1996.
- [34] **Michielsens E. and Weile D.S.** "Electromagnetic system design using genetic algorithms". In genetic algorithms and evolution strategies in engineering and computer science. John Wiley and Sons. England 1995.
- [35] **Belegundu A. D, Murthy D. V and E. W. Constants.** "Multiobjective optimization of laminated ceramic composites using genetic algorithms". In fifth AIAA / USAF / NASA symposium of multidisciplinary analysis and optimization. Panama City, Florida. 1994. AIAA.

- [36] **Poloni C. and Pediroda V.** "Genetic algorithm coupled with computationally expensive simulations: tools to improve efficiency". In D. Quagliarella, J. Periaux, M. Sefrioui and B. Mantel. Genetic algorithms and evolution strategies in engineering and computer science. Recent advances and industrial applications, chapter 11. John Wiley and Sons, West Sussex, England 1997.
- [37] Website del *Kanpur Genetic Algorithms Laboratory (KanGAL)* <http://www.iitk.ac.in/kangal>
- [38] **Deb K, Agrawal S, Pratap A, Meyarivan T.** "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II" *Kanpur Genetic Algorithms Laboratory (KanGAL)* KanGAL Report No. 200001
- [39] **El-Halwagi M, Manousiouthakis V.** "Synthesis of mass exchange networks" *American Institute of Chemical Engineering Journal*, 8, pp 1233-1244,
- [40] **Wang Y, Smith R.** "Wastewater minimisation". *Chemical Engineering Science*, 49, pp 3127-3145,
- [41] **Stefanis, S.K., Livingston, A.G. and Pistikopoulos, E.N** "A Methodology for Minimum Environmental Impact Analysis", *AIChE Symp.Series* 90, No.303, pp 139-150, 1994.
- [42] Website Eco-indicator 99 <http://www.pre.nl/eco-indicator99>
- [43] **Reed P.M.** "Striking the balance: Long- term groundwater monitoring design for multiple conflicting objectives" PhD. Thesis, *University of Illinois at Urbana- Champaign*, 2002.
- [44] **Thierens D.** "Analysis and design of genetic algorithms" Doctoral dissertation, Katholieke Universiteit Leuven, Leuven, Belgium, 1995.
- [45] **Reed P, Minsker B, Goldberg D.E.** "Designing a competent simple genetic algorithm for resaerch and optimization" *Water Resources Research*, 36(12) pp 3737-3761, 2000.
- [46] **Schaffer J, Caruana R. A, Eshelman J and Das R.** "A study of control parameters affecting online performance of genetic algorithms for function optimization", In Schaffer, J.D.(Ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, p. 51-60, Morgan Kaufmann, San Mateo, CA.
- [47] **Rudolph G.** "Convergence analysis of canonical genetic algorithms". *IEEE Transactions on Neural Network* 5(1), pp 96-101, 1994.
- [48] **Rudolph G.** "Evolutionary search for minimal elements in partially ordered finite sets. In *Proceedings of the 7<sup>th</sup> Annual Conference on Evolutionary Programming*, pp. 345-353. Berlin Springer, 1998a
- [49] **Leboreiro J, Acevedo, J.** "An Improved Infeasible Path Strategy for Process Optimisation using Genetic Algorithms in ASPEN Plus", 6th World Congress of Chemical Engineering, Melbourne, Australia, 2001
- [50] **Leboreiro H, J.** "Eficientizacion De Un Algoritmo Genetico Para La Optimizacion De Secuencias De Destilación Utilizando Un Simulador", Tesis de maestría, ITESM, diciembre 2001.
- [51] **Edgar, T. F. & Himmelblau, D. M.** "Optimization of Chemical Processes", McGraw Hill. 1988
- [52] **Guthrie, K. M.** "Capital Cost Estimating", *Chemical Engineering*, 76 (3), pp. 114-475 1969.
- [53] **Turton, R., Bailie, R. C., Whiting, W. B. & Shaeiwitz ,** "Analysis, Synthesis, and Design of Chemical Processes", Prentice Hall, 1998.

- [54] **Celanese Mexicana S.A** , Manual *Análisis de Seguridad en Procesos (ASP)*, Enero 1990.
- [55] **Ryoo H. S, Sahinidis N. V.** “Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design”, *Computers and Chemical Engineering*, 19 (5), pp. 551-566, 1995.
- [56] **Yee A. K. Y, Ray A. K, Rangaiah G. P.** “Multiobjective optimization of an industrial styrene reactor”, *Computers and Chemical Engineering*, 27, pp. 111-130, 2003.
- [57] **Moncada A. E, Acevedo J.** “A methodology for the minimisations of environmental impact of chemical processes under the presence of uncertainty”, 6th World Congress of Chemical Engineering, Melbourne, Australia, 2001.
- [58] **Rajesh J. K, Gupta S. K, Rangaiah G. P and Ray A. K.** “Multiobjective optimization of steam reformer performance using genetic algorithm”, *Ind Eng Chem Res*, 39, pp. 706-717, 2000.
- [59] **Kasat R. B, Kunzru D, Saraf D. N and Gupta S. K** “Multiobjective optimization of industrial fcc units using elitist nondominated sorting genetic algorithm”, *Ind Eng Chem Res*, 41, pp. 4765-4776, 2002.
- [60] **Oh P. P, Rangaiah G. P and Ray A. K** “Simulation and multiobjective optimization of an industrial hydrogen plant based on refinery off-gas”, *Ind Eng Chem Res*, 41, pp. 2248-2261, 2002.
- [61] **Song J, Park H, Lee D and Park S** “Scheduling of actual size refinery processes considering environmental impacts with multiobjective optimization”, *Ind Eng Chem Res*, 41, pp. 4794-4806, 2002.
- [62] **Chen H, Badenschier S. M and Shonnard D. R** “Uncertainty analysis for toxicity assessment of chemical process designs”, *Ind Eng Chem Res*, 41, pp. 4440-4450, 2002.
- [63] **Grossmann I. E, Drabbant R and Jain R. K** “Incorporating toxicology in the synthesis of industrial chemical complexes”, *Chem. Eng. Commun*, 17, pp. 151-170, 1982.
- [64] **Pistikopoulos, E.N and Stefanis, S.K** “Optimal solvent design for environmental impact minimization”, *Computers and Chemical Engineering*, 22(6), pp 717- 733, 1998.
- [65] **Stefanis, S.K., Livingston, A.G. and Pistikopoulos, E.N** “Environmental impact considerations in the optimal design and scheduling of batch processes” *Computers and Chemical Engineering*, 21(10), pp 1073- 1094, 1997.
- [66] **Androulakis I. P and Venkatasubramanian V.** “A genetic algorithmic framework for process design and optimization”, *Computers and Chemical Engineering*, 15(4), pp 217-228, 1991
- [67] **Mukherjee S, Dahule R.K, Tambe S. S, Ravetkar D. D and Kulkarni B. D** “Consider genetic algorithms to optimize batch distillation”, *Hydrocarbon Processing* 20(9), pp 59-66, 2001
- [68] **Deb K.** “Multi- objective genetic algorithms: Problem difficulties and construction of test functions” *Evolutionary Computation*, 7(3), pp. 205-230, 1999.
- [69] **Goldberg D. E, Deb K and Clark J. H** “Genetic algorithms, noise and the sizing of populations.” *Complex Systems* 6(4), pp. 333-362, 1992.

## Anexo A

### *El funcionamiento del NSGA-II*

1. Obtención de los datos de entrada. Por medio de la subrutina input el usuario provee al programa de los parámetros para realizar la corrida:

- Número de variables que serán codificadas como reales o como binarias.
- Número de funciones objetivo y el número de restricciones.
- Tamaño de la población (debe ser un número par de individuos), número de generaciones.
- Probabilidad de cruce.
- Un número entre 0 y 1 que servirá como semilla del generador de números aleatorios.

En caso de que las variables del problema sean codificadas como reales, los parámetros necesarios son:

- Probabilidad de mutación
- Índice de distribución para cruce e índice de distribución para mutación
- Límites superiores e inferiores de las variables.

En caso de que las variables sean codificadas como binarias los parámetros necesarios son:

- Tipo de cruce (1 para simple, 2 para uniforme)
- Número de bits asignados a cada variable
- Límites de las variables
- Probabilidad de mutación

2. Iniciación de las variables. Después de que fueron especificados los parámetros de entrada, el siguiente paso es inicializar a las variables de los individuos que formarán a la primera población (*oldpop*) Para las variables codificadas como reales se genera un número entre los límites de la variable en forma aleatoria (*realinit.h*) En el caso de las variables codificadas como binarias de manera aleatoria se van llenando con ceros y unos los  $n$  espacios (donde  $n$  es igual al número de bits asignados a cada variable, *init.h*); posteriormente estas variables serán decodificadas, es decir su valor en binario será convertido en decimal(*decode.h*)

3. Evaluación de la función objetivo(1). Una vez formada la primera población de individuos (*oldpop*), el paso siguiente es evaluar las funciones objetivo (*func.h*) y las



restricciones para obtener la aptitud (*fitness*) de cada uno de los individuos y el error. El siguiente paso es determinar a los individuos *no-dominados* para clasificarlos en rangos. El procedimiento consiste en comparar uno a uno los individuos de la población donde los criterios para determinar quien es el dominado y quien es el dominante son: el error y la aptitud. Es decir que los individuos con menor error y menor aptitud (para el caso de un problema de minimización) serán individuos no-dominados. Este proceso continúa hasta que todos los individuos son clasificados en cierto rango.

4. Selección. Los 3 puntos anteriores fueron para crear la población inicial a partir de esta se generan las poblaciones que evolucionarán durante un número  $n$  de generaciones. La siguiente población que surge es la llamada *matepop*, la cual es el resultado de someter a la población anterior (*oldpop*) a un proceso de selección (*nselect.h*) donde los individuos mejor adaptados serán los que formen parte de ella. El proceso es como sigue: dos individuos seleccionados de manera aleatoria son comparados, el primer criterio de comparación es el rango, resultará vencedor aquél que esté clasificado en el rango menor. El segundo criterio de comparación es la distancia de amontonamiento (*crowding factor*), el ganador será aquel que posea el factor más alto.
5. Cruce. Seleccionando parejas de individuos de la población recién formada (*matepop*) se realiza el intercambio genético entre ellos para obtener a otra población llamada (*newpop*) Para el caso de las variables codificadas como reales el proceso es el siguiente: seleccionados ya los dos individuos, los hijos se calculan a partir de un promedio entre los valores de los padres ( $x_{real}[i]$ ), los límites de las variables y el índice de distribución para cruce (*di*) En caso de las variables codificadas como binarias existen 2 opciones: el cruce de un punto y el cruce uniforme. Para el primer caso se determina un punto de cruce, antes de ese punto los genes del hijo 1 (*gen [i]*) serán los mismos que los del padre 1, lo mismo sucede con el hijo 2, a partir del punto de cruce, el contenido genético del padre 1 se copiará al hijo 2 y el del padre 2 se copiará al hijo 1. En el caso del cruce uniforme, a partir de un número aleatorio se determina a que individuo hijo se heredará el contenido genético, es decir si el número aleatorio es menor o igual a la probabilidad de cruce, entonces el *gen [j]* del padre 2 será copiado al hijo 1 y el *gen[j]* del padre 1 será copiado al hijo 2 en caso contrario el contenido genético del padre 1 será copiado al hijo 1 y el del padre 2 al del hijo 2.
6. Mutación. Operador que se aplica a cada individuo de la nueva población (*newpop*) Para las variables codificadas como reales, es un procedimiento parecido al del cruce solo que esta vez el parámetro que se considera en el cálculo es el índice de distribución para mutación (*dim*) En el caso de las variables codificadas como binarias consiste en cambiar el contenido si es cero por un uno y viceversa.
7. Evaluación de la función objetivo(2). La población recién formada evalúa la función objetivo el procedimiento es el mismo que el del punto 3.
8. Formación de la población global (*globalpop*), cálculo de la distancia de amontonamiento (*crowding distance*) y elitismo. La población global es el resultado de la unión de *oldpop* y de *newpop*, por lo que ahora el tamaño de la población es dos veces el tamaño de la población inicial. El paso siguiente es la clasificación de la

población recién formada en rangos tal como se explica en el paso 3 (después de la evaluación de la función objetivo) Al finalizar la clasificación se tiene el número total de rangos y cuantos y cuales individuos quedaron en cada uno de ellos. Lo siguiente es calcular la distancia de amontonamiento (*crowding distance*) para cada uno de los rangos en los que fue clasificada la población, esta operación se hace por rangos. El primer paso es ordenar de mayor a menor aptitud los individuos presentes en el rango, una vez ordenados se identifica al individuo con mayor aptitud (max). El cálculo de la distancia (cub\_len) es como sigue para los individuos que están en los extremos del rango (es decir el que tiene menor y el que tiene mayor aptitud) es igual al producto de un factor (100.0) por la máxima aptitud localizada en ese rango (max); para el resto de los individuos el cálculo es igual al valor absoluto de la diferencia entre la aptitud del individuo [i+1] y el individuo [i-1]. Para determinar cuales individuos serán los escogidos para formar parte de la siguiente generación se aplica el elitismo. Este operador consiste en seleccionar a los mejores individuos de la población global, por lo que para seleccionarlos el proceso se realiza por rangos, generalmente los mejores son los ubicados en los primeros rangos de la clasificación. La selección continúa mientras el número de clasificados sea menor o igual al número de la población.

9. La nueva población. Una vez aplicado el elitismo ya se tienen identificados los individuos que formarán parte de la nueva población que será copiada a la estructura matepop.
10. Cálculo del Rank ratio. Después de creada la nueva población se determina una relación de rangos entre la población inicial (oldpop) y el número del rango de donde fue copiado el último individuo seleccionado por el elitismo. Esto es se calcula el número total de individuos que hay en cada uno de los “n” rangos de ambas poblaciones (tot) y después se divide el número de individuos de cada rango de la población recién formada entre el total.
11. Repetir ciclo “n” generaciones. Por último, los individuos de la estructura matepop son copiados a la estructura oldpop que es la que ahora repetirá nuevamente el ciclo (regresar al punto 4) “n” generaciones.

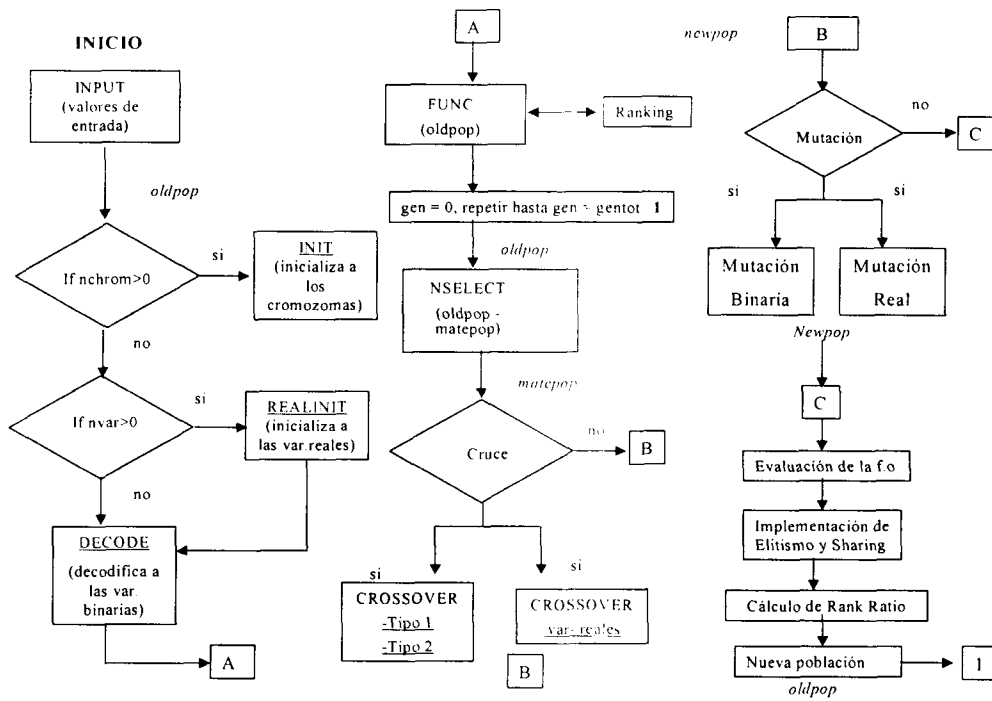


Figura A 1 Diagrama de flujo del NSGA-II

## Anexo B

El funcionamiento de la interfase se describe a continuación: una vez que la función *\_spawnl* se ejecuta, la interfase abre un archivo en donde el programa optimizador (el algoritmo genético) guarda la información del individuo que será evaluado por el simulador; una vez que el simulador obtiene los datos de entrada, realiza la corrida y guarda los resultados de ésta en un archivo que genera la interfase; para que después el optimizador abra el archivo, lea los resultados y evalúe la aptitud de dicho individuo. Un diagrama de flujo de la interfase, se presenta en la Figura B 1:

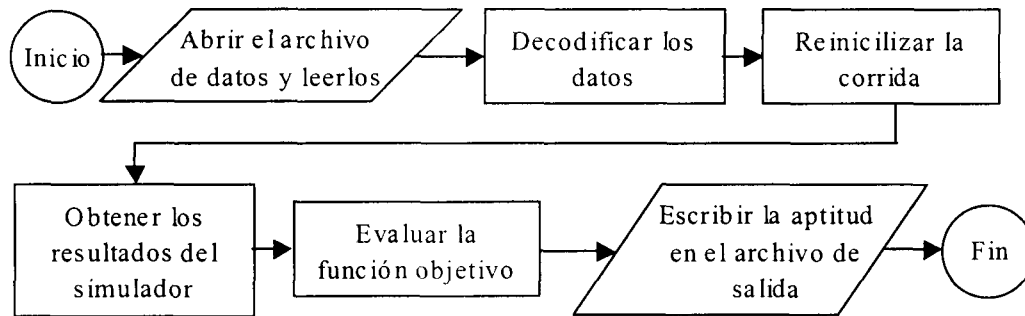


Figura B 1 Diagrama de flujo de la interfase.

Por medio de los controles Active X es posible manipular el simulador ya que pueden modificarse propiedades o ejecutar métodos de los objetos definidos en éste. El objeto cliente, o principal, es el HappIP con el cual se puede abrir, cerrar, o guardar archivos de Aspen Plus. Los datos de entrada y salida en una simulación están organizados en una estructura de árbol dentro del objeto IHNode. Para acceder cualquier dato es necesario contar con la ruta en el árbol, la cual se puede obtener del Explorador de Variables.

Para ejemplificar lo anterior se presenta un caso en el cual desea obtenerse la composición de la corriente de destilado de una columna, que es alimentada con una mezcla de heptano-tolueno y que pretende ser separada por medio de fenol. Los datos proporcionados al simulador (generados por el algoritmo genético y almacenados en el archivo *entradas.inp*) son el número de platos de la columna (*nplatos*), el plato de alimentación de la mezcla a separar (*platalim*), el plato de alimentación del fenol (*platfenol*) y la relación de reflujo (*rrf*). Una vez que el simulador obtiene los datos de entrada, realiza la simulación y obtiene la composición de la corriente de destilado accediendo al explorador de variables; para posteriormente escribir los resultados en un archivo (*salidas.res*). Como referencia, el lector puede remitirse al Manual del usuario del Simulador Aspen Plus versión 11.1

---

```
Sub Main()
'Declaración de variables
Dim sim As HappIP
Dim ColInput, ColOutputXM1, ihSTGXNode2, ihSTGXNode2 As IHNode
Dim nplatos, platalim, platfenol As Integer
Dim rrf, xheptano, xtolueno, xfenol As Double

'Abrir el archivo de datos
Open "entradas.inp" For Input As #1
'Leer los datos de entrada
Input #1, nplatos, platalim, platfenol, rrf
Close #1
'Tomar control del Aspen
Set sim = GetObject("Simulación.apw")
'Asignar los valores deseados a las variables de la simulación
Set ColInput = sim.Tree.Data.Blocks.COLUMNNA.Input
ColInput.NSTAGE.Value = nplatos
ColInput.FEED_STAGE.ALIM.Value = platalim
ColInput.FEED_STAGE.SOLV.Value = platfenol
ColInput.BASIS_RR.Value = rrf

'Correr la simulación de Aspen
Call sim.Engine.Run2(False)
'Obtener los datos de salida de ASPEN
Set ColOutputXM = sim.Tree.Data.Blocks.COLUMNNA.Output.Elements("X_MS")
For Each ihSTGXNode2 In ColOutputXM2.Elements
  For Each ihCompNode2 In ihSTGXNode2.Elements
    contador= contador + 1
    Select Case contador
      Case Is = 1      'fraccion de heptano
        xheptano = ihCompNode2.Value
      Case Is = 2      'fraccion de tolueno
        xtolueno = ihCompNode2.Value
      Case Is = 3      'fraccion de fenol
        xfenol= ihCompNode2.Value
    End Select
  Next ihCompNode2
Next ihSTGXNode2
'Abrir el archivo de resultados
Open "Resultado.out" For Output As #2
'Escribir el resultado
Write #2,xheptano,xtolueno,xfenol
Close #2
End Sub
```

---

## Anexo C

Las ecuaciones utilizadas para calcular el costo base por módulo de los intercambiadores y de las columnas tienen la siguiente forma:

$$\log_{10} C_p = K_1 + K_2 \log_{10} A + K_3 (\log_{10} A)^2 \quad (C.1)$$

donde  $A$  es el parámetro de capacidad del equipo correspondiente. En el caso de los intercambiadores el parámetro de capacidad es el área de transferencia ( $m^2$ ) mientras que para las columnas de destilación es la altura (m). Todas las constantes utilizadas de la técnica de Costeo por Módulo [(Turton, Bailie, Whiting y Shaeiwitz, 1998)] en este trabajo se encuentran en el anexo C.

Una excepción a estas fórmulas para el cálculo del costo base por módulo se necesitó utilizar cuando el área del rehervidor tipo canasta excedía los  $100 m^2$ , ya que la fórmula arriba presentada tiene como máximo ese límite, en estos casos se empleó la siguiente fórmula [(Corripio, Chrien y Evans, 1982)]

$$C_p = 1.25 \exp(8.202 + 0.01506 \ln A + 0.06811 * (\ln A)^2) \quad (C.2)$$

Las conducciones base para este caso son acero al carbón y una presión de  $2100 \text{ kN/m}^2$ . Para el cálculo del costo de los platos se utiliza la siguiente fórmula.

$$C_p = 235 + 19.80A + 75.07A^2 \quad (C.3)$$

En este caso el parámetro de capacidad es el diámetro de la columna.

Para calcular el valor de los equipos es necesario conocer los materiales de construcción y la presión de operación. Con ello se puede conocer el valor del factor de material y el factor de presión para cada equipo. El factor de presión, para los intercambiadores de calor es calculado a partir de la siguiente fórmula:

$$\log_{10} F_p = C_1 + C_2 \log_{10} P + C_3 (\log_{10} P)^2 \quad (C.4)$$

Las unidades de la presión ( $P$ ) son en bar manométrico o barg ( $1 \text{ bar} = 0.0 \text{ barg}$ ). Esta ecuación no debe ser utilizada fuera del rango especificado, si se desea utilizar otras presiones fuera de las de los rangos indicados existen otras funciones.

El factor de presión de las columnas de destilación es calculado de la siguiente forma; su valor es de 1.25 si la presión es menor a  $-0.5 \text{ barg}$ , 1.00 si la presión esta entre  $0.5$  y  $3.7 \text{ barg}$  y se emplea la siguiente fórmula si la presión esta entre  $3.7$  y  $100 \text{ barg}$ :

$$F_p = 0.5146 + 0.6838 \log_{10} P + 0.2970 (\log_{10} P)^2 + 0.0235 (\log_{10} P)^6 + 0.0020 (\log_{10} P)^8 \quad (C.5)$$

Los valores del factor de costo del módulo base se encuentran graficados en función de los factores de presión y de materiales, esto facilita la obtención de dicho factor. Este factor no es linealmente proporcional a los factores de material y de presión. Esto se debe a que no

todos los costos indirectos incrementan en proporción directa con el incremento del costo de material.

Algunos costos fijos como fletes, aislamiento, cimientos y estructuras muestran cambios ligeros y en algunos casos no presentan cambio con los costos de material, mientras que otros costos como los materiales y la mano de obra son alterados en forma significativa.

El método de costeo de equipo por módulo toma en cuenta estas variaciones en el factor de costo del módulo base.

El factor de costo del módulo base de los equipos se puede calcular a partir de la siguiente ecuación en caso de no contar con las gráficas necesarias

$$F_{BM}^{\circ} = B_1 + B_2 F_M F_p \quad (C.6)$$

donde  $B_1$  y  $B_2$  las constantes para el cálculo del factor de costo del módulo base, las cuales dependen del equipo.

La siguiente ecuación se utiliza para calcular el costo del módulo de los equipos a partir del costo base del módulo y el factor de costo del módulo base.

$$C_{BM}^{\circ} = C_p F_{BM}^{\circ} \quad (C.7)$$

En el caso de las columnas de destilación es necesario calcular el costo de los platos por separado y sumarlo al costo de la columna. En el presente trabajo se consideró que los platos son perforados. El costo de los platos se puede calcular de la siguiente fórmula:

$$C_{BM}^{\circ} = C_p N_p F_{BM}^{\circ} F_q \quad (C.8)$$

donde  $N_p$  es el número de platos de la columna y  $F_q$  es el factor de cantidad de platos. Este último contabiliza el efecto de escala de economía, ya que al comprar una mayor cantidad el precio por unidad disminuye.

Una vez calculado el costo de todos los equipos involucrados en el diseño, se puede calcular el costo de capital ( $C_{Cap}$ ) como la sumatoria de los costos mediante la siguiente fórmula.

$$C_{Cap} = \sum_{i=1}^n C_{BM}^{\circ} i \quad (C.9)$$

El costo del servicio de enfriamiento ( $C_{SE}$ ) de los condensadores se consideró como el costo de agua, la cual está disponible a 30°C y se utilizaría con un cambio de temperatura de 10°C para no exceder los 40°C como temperatura de salida de los equipos de enfriamiento. El costo de este servicio es de 0.16 \$/GJ. En el de servicio de calentamiento se consideraron tres niveles de vapor por lo que el costo ( $C_{SC}$ ) varía según el vapor requerido.

Es importante mencionar que es necesario tomar en cuenta los diferentes niveles de servicios, porque en la realidad los servicios con los que se dispone se encuentran a una temperatura fija y no se puede disponer de servicios a cualquier temperatura. El vapor de baja presión tiene un costo de 3.17 \$/GJ y se encuentra a 5 barg y a una temperatura de 160°C. El vapor de media presión tiene un costo de 3.66 \$/GJ y se encuentra a 10 barg y a una temperatura de 184°C. El vapor de alta presión tiene un costo de 3.17 \$/GJ y se encuentra a 41 barg y a una temperatura de 254°C. El costo total de los servicios de una columna ( $C_{Ser}$ ) se calcula a partir de:

$$C_{Ser} = Q_E C_{SE} + Q_C C_{SC} \quad (C.10)$$

donde  $Q_E$  es la carga térmica del condensador y  $Q_C$  es la carga térmica del rehervidor de la columna correspondiente.

Finalmente, el costo total anualizado ( $C_{Tot}$ ) del proceso se puede obtener de la siguiente fórmula.

$$C_{Tot} = \frac{C_{Cap}}{T_{Ret}} + \sum_{i=1}^n C_{Ser} i T_{Ope} \quad (C.11)$$

donde  $T_{Ret}$  es el tiempo esperado de retorno de la inversión y  $T_{Ope}$  es el tiempo de operación por año. En este trabajo se fijó el tiempo de retorno de la inversión en 2.5 años y se consideró 8,000 horas de operación al año, los cuales son valores comúnmente utilizados.

En muchas ocasiones se cuenta con datos o correlaciones de costos de equipo del pasado para poder utilizar esta información es necesario calcular el valor presente tomando en cuenta el efecto del tiempo en el costo de la compra de los equipos. El uso de índices es común para actualizar estos costos tomando en cuenta las condiciones económicas actuales y otros factores tal como la inflación. Con la siguiente fórmula se puede estimar el costo presente de un equipo ( $C_2$ ) utilizando datos de costos de años anteriores.

$$C_2 = C_1 \left( \frac{I_2}{I_1} \right) \quad (C.12)$$

donde  $C_1$  es el costo del equipo estimado con datos de años anteriores,  $I_1$  es el valor del índice correspondiente al año del cual son los datos y  $I_2$  es el valor presente del índice.

Existen varios índices utilizados por la industria química para ajustar los costos debido a la inflación con base en datos pasados. Cabe mencionar que todos los índices muestran la misma tendencia inflacionaria a través del tiempo, por lo que al utilizar cualquiera se obtiene el mismo resultado. Este tipo de índices son compuestos, esto quiere decir que no solo toman en cuenta los costos de adquisición, además contabilizan otros gastos adicionales como los de instalación, los de los accesorios, los de equipo secundario, entre otros. Los índices más utilizados en la industria química reportados en el Chemical Engineering son el Marshall and Swift Equipment Cost Index y el Chemical Engineering Plant Cost Index (CECPI).





## Anexo D

En este apéndice se detalla la secuencia de cálculo del índice Mond mostrada en la figura 4.2 de este escrito.

**Factor de material (B)** este se define como una medida del fuego, explosión o energía potencial liberada por el material clave. Este factor se calculó como una suma ponderada de los factores de material de los compuestos involucrados. La ponderación se consideró como la fracción másica de cada componente ( $xm_i$ ). En el caso de las columnas, se utilizó la fracción másica promedio. Para obtener la fracción másica promedio de cada compuesto, fue necesario calcular el volumen promedio de líquido contenido en los platos así como un promedio de la densidad de la mezcla líquida, esta propiedad es calculada por el simulador en cada una de las etapas (así como otras propiedades como la temperatura, la composición, flujo másico, entre otras); es posible acceder a estos valores por medio del explorador de variables. Cabe mencionar que para este cálculo no se consideró la fase gaseosa. Una vez calculados el volumen y la densidad promedio, obtenemos la masa de líquido promedio que se encuentra en la columna. Después es necesario obtener un promedio de la fracción másica de cada compuesto en la columna, con esto, ya es posible estimar la masa promedio de cada compuesto en la columna ( $mprom_i$ ). La expresión final para el cálculo del factor del material de la mezcla es:

$$B = \sum_{i=1}^3 \Delta Hc * xm_i / 1000 \quad (D.1)$$

donde

$\Delta Hc$  es el calor neto de combustión del material a 25°C en Btu/lb

Los datos de calor de combustión utilizados en este estudio se presentan en la Tabla D 1

Heptano	20,825	btu/lb
Tolueno	18,422	btu/lb
Fenol	13,400	btu/lb

**Tabla D 1 Valores de calor de combustión**

**Riesgos especiales del material (M)** Este punto tiene por objeto tomar en cuenta las propiedades específicas del material clave o cuando se mezcle con otros materiales tales como catalizador. Para nuestro estudio, este factor no se tomó en cuenta.

**Riesgos generales del proceso (P)** Este factor considera riesgos en las operaciones de proceso que incluyan manejo y cambio físico. El valor asignado de este factor fue 10.

**Riesgos especiales del proceso (S)** En esta sección los factores se asignan con respecto a las características de operación del proceso, almacenamiento, transporte, que intensifiquen el riesgo total. Los factores considerados para calcular los riesgos especiales del proceso fueron: alta presión, alta temperatura y riesgo de explosión mayor del promedio. El factor por alta presión se obtuvo a partir de la ecuación D.2 la cual se aproxima a la curva presentada en la página 103 de [54]

$$ap = -0.0001 * (pres^2) + (0.1824 * pres) + 3.8285 \quad (D.2)$$

donde

*pres* representa la presión de operación de la unidad en psi.

A los factores de alta temperatura y riesgo de explosión mayor que el peligro se les asignó un valor de 25 y 40 respectivamente.

**Riesgos asociados a cantidades (Q)** En este punto se asigna un factor para los riesgos relacionados con el uso de grandes cantidades de combustibles inflamables, explosivos o materiales que puedan descomponerse. Debe calcularse la cantidad total del material en la unidad del proceso estudiada. La cantidad del material se registra como cantidad total en peso. El factor de cantidad se calculó por medio de la ecuación B.3, la cual se aproxima a la curva presentada en la página 104 de [54]

$$Q = 22.4 * \left(\frac{mliq}{1000}\right)^{0.2271} \quad (D.3)$$

donde

*mliq* representa la masa total del material presente en la unidad.

**Riesgos por distribución de unidades (L)** Cuando las unidades de proceso o edificios se localizan juntos, un incidente en una unidad puede involucrar unidades adyacentes por el *efecto dominó*. En este apartado se consideran el debilitamiento de estructuras por fuego, explosión, colapso de los cimientos, etc; lo principal es asegurarse de que haya suficiente espacio de manera que las unidades que estén cayendo no lo hagan en las unidades vecinas. Para este punto, el factor de penalización es de 20. en caso de que la unidad tenga entre 20 y 30 metros de altura. Esto no aplica para unidades de almacenamiento, por lo que para nuestro cálculo se aplicó únicamente a las 2 columnas de destilación.

**Riesgos graves para la salud (T)** Los riesgos para la salud pueden variar tanto en el grado como en la forma en que se presentan. Algunos son identificables en condiciones anormales de proceso, como mantenimiento a procesos fuera de control o en incendios, mientras que otros están presentes continuamente como resultado de pequeñas fugas en juntas, empaques, venteos de gases de proceso, etc. La toxicidad de gases, vapores y polvos se clasifica con base en los Time Weighted Threshold Limits Values (TLV) Para determinar el factor de penalización correspondiente a este apartado, es necesario tener la concentración del material en partes por millón (ppm) Debido a que en este estudio se manejan tres sustancias con valores de TLV diferentes (fenol 5 ppm, tolueno 53 ppm, heptano 400 ppm), se definió una metodología para asignar el factor de penalización por el valor del TLV.

Primero se calcula un factor denominado “gama” por medio de la ecuación (D.4) para cada uno de los compuestos presentes en la mezcla:

$$gama_i = \left( \frac{\rho_i}{TLV_i} \right) * fp_i \tag{D.4}$$

donde

$\rho_i$  = es la concentración del compuesto i en partes por millón

$TLV_i$  = es el Time Weighted Threshold Limit Value del compuesto i

$fp_i$  = es el factor de penalización asignado que depende de la magnitud del TLV

El compuesto que resulte con el factor más alto, es el que será considerado como el más peligroso por efectos de toxicidad y sobre el valor de su TLV será aplicado el factor de penalización. Los factores de penalización utilizados fueron los que se muestran en la Tabla D 2.

Rango del TLV	Factor de penalización asignado
TLV >1 ppm y hasta 10 ppm	75
TLV > 10 ppm y hasta 100 ppm	50
TLV > 100 ppm y hasta 1000 ppm	30

**Tabla D 2. Factores de penalización asignados por TLV**

Otros factores que se tomaron en cuenta para determinar los riesgos graves a la salud fueron:

*Forma del material:* se asignó una penalización de 50, ya que el material usado en el proceso, se encuentra como un líquido.

*Riesgo de exposición corta:* está relacionado con el TLV; la penalización asignada fue 100.

*Posible absorción por la piel:* este factor se consideró ya que las sustancias que están presentes en la mezcla pueden ser absorbidas por la piel.

**Cálculo del índice global DOW/ICI (D)** Se calcula a partir de la ecuación D.5

$$D = B * \left( 1 + \frac{M}{100} \right) * \left( 1 + \frac{P}{100} \right) * \left( 1 + \frac{(S + Q + L)}{100} + \frac{T}{400} \right) \tag{D.5}$$

donde

B representa el Factor del material

M representa el Factor por riesgos especiales del material

P representa el Factor por riesgos generales del proceso

S representa el Factor por riesgos especiales del proceso

Q representa el Factor por riesgos de cantidad

L representa el Factor de riesgos por distribución de unidades

T representa el Factor por riesgos graves para la salud

**Cálculo de carga de fuego (F)** Por medio de este factor, es posible tener una idea de la duración de un incendio en la unidad en caso de un incidente. El cálculo se basa en los  $\frac{BTU}{pie^2}$  de área plana de la unidad.

$$F = \left( \frac{BK}{N} \right) \quad (D.6)$$

donde

B representa el Factor del material

K representa la cantidad del material

N representa el área normal de trabajo

**Índice de explosión interna de la planta (E)** Este índice refleja una medida del riesgo de explosión interna en la planta, se calcula a partir de la ecuación D.7

$$E = 1 + \frac{m + ap + S}{100} \quad (D.7)$$

donde

m representa el factor por mezclado y dispersión

H representa la altura de la unidad

t Representa la temperatura del proceso

ap Representa el factor por alta presión

La altura de las columnas se calcula a partir del número de platos y del espacio entre éstos. La altura de los tanques de almacenamiento se calculó a partir del volumen de producto (heptano y tolueno) almacenado durante 8 horas. El valor estimado fue de 6.15 metros para ambos tanques.

Para las columnas, la temperatura del proceso se manejó como una temperatura promedio entre los platos. Para los tanques, se consideró la temperatura de la corriente a almacenar reportada en la simulación.

**Índice de explosión aérea (A)** Debido a que el índice de explosión interna de la planta (E) no representa el único potencial de explosión de la sección. De un estudio de un gran número de escapes de sustancias inflamables que han dado lugar ya sea a explosiones aéreas o a nubes que han causado únicamente fuego por ignición, ha sido posible derivar este índice, el cual se calcula a partir de la ecuación D.8.

$$A = B * \left( 1 + \frac{m}{100} \right) * Q * H * E * \left( \frac{t}{300} \right) * \left( \frac{1 + ap}{1000} \right) \quad (D.8)$$

donde

B representa el factor del material

m representa el factor por mezclado y dispersión

E representa el índice de explosión interna de la planta

H representa la altura de la unidad  
 Q representa el factor por riesgos de cantidad  
 ap representa el factor por alta presión

**Índice unitario de toxicidad (U)** Se calcula de manera que represente la influencia de la toxicidad de la sección en estudio. Por medio de la ecuación

$$U = \frac{T}{100} * \left( 1 + \frac{M + P + S}{100} \right) \quad (D.9)$$

donde

M representa el factor de riesgos especiales del material  
 P representa el factor de riesgos generales del proceso  
 S representa el factor de riesgos especiales del proceso  
 T representa el factor de riesgos por toxicidad

**Índice del máximo accidente tóxico (C)** Este se calcula a partir de la combinación del índice unitario de toxicidad U y el factor de cantidad Q, la ecuación utilizada es

$$C = Q * U \quad (D.10)$$

**Índice global de riesgo (R)** La ecuación que define el factor global de riesgo es:

$$R = D * \left( 1 + \frac{\sqrt{FUEA}}{1000} \right) \quad (D.11)$$

La clasificación del riesgo según el factor obtenido se presenta en la Tabla D 3

<b><i>Factor global de riesgo R</i></b>	<b><i>Categoría del riesgo global</i></b>
0-20	Suave
20-100	Bajo
100-500	Moderado
500-1,100	Alto (grupo 1)
1,100-2,500	Alto (grupo 2)
2,500-12,500	Muy alto
12,500-65,000	Extremo
65,000	Muy extremo

**Tabla D 3. Categorías del riesgo global R**

Los factores de bonificación asociados a las características técnicas o de organización, **no fueron considerados.**



Centro de Información-Biblioteca



30002006327456