

**Departamento de Informática – Facultad de Ingeniería
Universidad Nacional de la Patagonia San Juan Bosco
Comodoro Rivadavia, Argentina**

**TESINA DE LICENCIATURA EN
INFORMÁTICA**

**ALGORITMOS GENÉTICOS PARALELOS
APLICADOS A LA RESOLUCIÓN DE
PROBLEMAS DE ASIGNACIÓN DE
FRECUENCIAS EN REDES CELULARES**

**CRISTIAN PERFUMO
GERARDO MORA
LUCAS ROJAS**

ABRIL 2006

**Tutores de Tesina:
Sergio Nesmachnow
José Gallardo**

ALGORITMOS GENÉTICOS PARALELOS APLICADOS A LA RESOLUCIÓN DEL PROBLEMA DE ASIGNACIÓN DE FRECUENCIAS EN REDES DE TELEFONÍA CELULAR

RESUMEN

En este trabajo se presenta una investigación sobre algoritmos genéticos, considerando sus variantes secuenciales y paralelas. En base a las características que presentan los algoritmos genéticos se decide su aplicación a la resolución de un problema en el ámbito de las redes de comunicación inalámbrica: la asignación de frecuencias. En particular se enfoca en las redes de telefonía celular, donde la meta es asegurar el servicio y su calidad, minimizando los costos. Estos objetivos se logran maximizando la reutilización de frecuencias y minimizando la suma total de interferencias. Esta situación se modela mediante el problema de optimización denominado MI-FAP (*minimun interference frequency assignment problem*), que en los últimos años ha sido investigado por varias organizaciones vinculadas a las telecomunicaciones a nivel mundial. En la etapa de experimentación con los algoritmos genéticos se utilizaron las instancias COST, escenarios reales que representan abstracciones de redes de telefonía celular instaladas en distintas ciudades, cada una con características propias. Para lograr mejoras en la calidad de resultados, se utilizó una variante elitista de algoritmos genéticos llamada CHC. Adicionalmente, se buscó mejorar la eficiencia computacional de los algoritmos utilizando técnicas de procesamiento paralelo, utilizando versiones paralelas de las implementaciones desarrolladas. Los resultados obtenidos permiten afirmar que las técnicas evolutivas son aplicables a este tipo de problemas. En particular, se logra establecer que los algoritmos paralelos presentan mejores resultados que sus pares secuenciales, tanto en calidad de las soluciones halladas como en eficiencia computacional.

Palabras clave: Algoritmos genéticos, paralelismo, FAP, asignación de frecuencias, COST259.

ÍNDICE

RESUMEN	I
ÍNDICE	II
CAPÍTULO 1. INTRODUCCIÓN	1
CAPÍTULO 2. ALGORITMOS GENÉTICOS	3
2.1. INTRODUCCIÓN	3
2.2. COMPUTACIÓN EVOLUTIVA	4
2.2.1. FUNDAMENTOS BIOLÓGICOS DE LA COMPUTACIÓN EVOLUTIVA...	4
2.2.2. CORRESPONDENCIA ENTRE LA EVOLUCIÓN BIOLÓGICA Y LA COMPUTACIONAL.....	5
2.3. CLASIFICACIÓN DE ALGORITMOS EVOLUTIVOS	5
2.3.1. ESTRATEGIAS DE EVOLUCIÓN	6
2.3.2. PROGRAMACIÓN EVOLUTIVA	6
2.4. ALGORITMOS GENÉTICOS	7
2.4.1. CODIFICACIÓN DE SOLUCIONES	7
2.4.2. ESTRUCTURA BÁSICA DE UN ALGORITMO GENÉTICO.....	8
2.4.2.1. GENERACIÓN DE LA POBLACIÓN INICIAL.....	9
2.4.2.2. EVALUACIÓN.....	9
2.4.2.3. SELECCIÓN.....	11
2.4.2.4. CRUZAMIENTO.....	13
2.4.2.5. MUTACIÓN.....	15
2.4.2.6. REEMPLAZO.....	16
2.4.2.6.1. MODELO EVOLUCIONARIO.....	17
2.4.3. FUNDAMENTOS DEL MECANISMO DE EXPLORACIÓN DE LOS ALGORITMOS GENÉTICOS.....	18
2.4.4. APLICACIONES DE LOS ALGORITMOS GENÉTICOS.....	18
2.5. ALGORITMOS GENÉTICOS PARALELOS.....	19
2.5.1. CLASIFICACIÓN DE ALGORITMOS GENÉTICOS PARALELOS.....	20
2.5.1.1. PARALELISMO MAESTRO-ESCLAVO.....	20
2.5.1.2. PARALELISMO MEDIANTE SUBPOBLACIONES ESTÁTICAS CON MIGRACIÓN.....	21
2.5.1.3. PARALELISMO MEDIANTE SUBPOBLACIONES ESTÁTICAS SOLAPADAS SIN MIGRACIÓN.....	22

2.5.1.4. ALGORITMOS GENÉTICOS MASIVAMENTE PARALELOS....	23
2.5.1.5. OTROS MODELOS DE PARALELISMO.....	23
2.5.2. EVALUACIÓN DEL DESEMPEÑO DE LOS ALGORITMOS PARALELOS..	24
2.6. CONCLUSIONES.....	24
CAPÍTULO 3. ASIGNACIÓN DE FRECUENCIAS EN REDES INALÁMBRICAS.....	26
3.1. INTRODUCCIÓN.....	26
3.2. COMUNICACIONES INALÁMBRICAS.....	26
3.3. PROBLEMAS DE ASIGNACIÓN DE FRECUENCIAS.....	28
3.4. CLASIFICACIÓN DE LOS PROBLEMAS DE ASIGNACIÓN DE FRECUENCIAS.....	30
3.5. PROBLEMAS DE MÍNIMA INTERFERENCIA EN ASIGNACIÓN DE FRECUENCIAS.	31
3.6. ESCENARIOS COST.....	32
3.6.1. PARÁMETROS DE INFORMACIÓN GENERAL.....	33
3.6.2. PARÁMETROS REFERENTES A CELDAS.....	34
3.6.3. PARÁMETROS REFERENTES A RELACIONES ENTRE DOS CELDAS....	35
3.7. EL PROBLEMA DE MÍNIMA INTERFERENCIA EN EL MARCO DE LOS ESCENARIOS COST259.....	35
3.8. UN EJEMPLO DEL PROBLEMA.....	36
3.8.1. DATOS GENERALES.....	36
3.8.2. RESTRICCIONES.....	37
3.8.3. EVALUACIÓN DE UNA SOLUCIÓN.....	40
3.9. CONCLUSIONES.....	40
CAPÍTULO 4. UN ALGORITMO GENÉTICO PARA EL PROBLEMA DE MÍNIMA INTERFERENCIA EN LA ASIGNACIÓN DE FRECUENCIAS EN REDES DE TELEFONÍA CELULAR.....	41
4.1. INTRODUCCIÓN.....	41
4.2. BIBLIOTECA MALLBA.....	41
4.3. MPI.....	43
4.4. ARQUITECTURA, IMPLEMENTACIÓN Y PLATAFORMA DE EJECUCIÓN.....	44
4.5. CODIFICACIÓN DE SOLUCIONES.....	44
4.5.1. CODIFICACIONES ALTERNATIVAS.....	45
4.6. FUNCIÓN DE <i>FITNESS</i>	45
4.7. ALGORITMOS GENÉTICOS UTILIZADOS.....	46
4.8. ANÁLISIS PARAMÉTRICOS DE RESULTADOS EXPERIMENTALES.....	49
4.9. ALGORITMOS GENÉTICOS SECUENCIALES Y PARALELOS	50
4.9.1. RESULTADOS PARA EL ESCENARIO <i>TINY</i>	50
4.9.2. RESULTADOS PARA EL ESCENARIO <i>K</i>	51
4.9.3. RESULTADOS PARA EL ESCENARIO <i>SWISSCOM</i>	52

4.9.4. ANÁLISIS DE RESULTADOS	53
4.10. PARTICULARIDADES DE DIFERENTES ESCENARIOS.....	56
4.11. OTROS ENFOQUES PROPUESTOS PARA ESCENARIOS COST259.....	57
4.12. CONCLUSIONES.....	57
CAPÍTULO 5. CONCLUSIONES Y TRABAJOS FUTUROS.....	58
5.1. CONCLUSIONES.....	58
5.2. TRABAJOS FUTUROS.....	59
ANEXO I. AGRADECIMIENTOS	60
AI.1. AGRADECIMIENTOS GENERALES.....	60
AI.2. AGRADECIMIENTOS PERSONALES DE GERARDO MORA.....	60
AI.3. AGRADECIMIENTOS PERSONALES DE CRISTIAN PERFUMO.....	61
AI.4. AGRADECIMIENTOS PERSONALES DE LUCAS ROJAS.....	61
REFERENCIAS BIBLIOGRÁFICAS.....	62

CAPÍTULO 1

INTRODUCCIÓN

*“Proyecta lo difícil, partiendo de donde aún es fácil. Realiza lo grande, partiendo de donde aún es pequeño. Todo lo difícil comienza siendo fácil. Todo lo grande comienza siendo pequeño.”*¹

LAO TSE
Dao De Jing, 600 AC

Los algoritmos evolutivos son una de las técnicas de optimización más utilizadas en los últimos 20 años para resolver problemas complejos. Conforme avanza la investigación sobre el mecanismo algorítmico evolutivo y se incrementa el poder de cómputo disponible en los laboratorios, los científicos intentan aplicarlos como herramienta de resolución para problemas de optimización combinatoria cada vez más difíciles.

Con la masificación de las redes inalámbricas, los entes reguladores del espectro de radiofrecuencias se han encontrado con el problema de asignar una frecuencia a cada comunicación sin que esto conlleve a interferencias. Dado que las frecuencias son un recurso finito, el objetivo es maximizar la reutilización de las frecuencias sin disminuir la calidad de la transmisión.

En este trabajo se investiga sobre la aplicación de los algoritmos genéticos como herramienta de resolución del problema de asignación de frecuencias en redes inalámbricas. En particular se consideran las redes de telefonía celular presentadas como escenarios de prueba en el proyecto COST 259 de la Unión Europea [COST259, 1996].

Los objetivos fundamentales de esta tesina son: la comprensión del mecanismo de los algoritmos genéticos como técnica de resolución de problemas de optimización, evaluar la aplicabilidad de los algoritmos genéticos a la resolución del problema de asignación de frecuencias en redes inalámbricas, diseñar versiones secuenciales y paralelas de algoritmos genéticos para resolver el problema considerado, realizar una comparación entre la calidad de resultados obtenidos y la eficiencia computacional de las versiones desarrolladas, y comparar los resultados obtenidos con los mejores resultados hallados con otras técnicas para el problema.

La principal motivación de este trabajo radica en la inexistencia de resultados publicados para el conjunto de escenarios COST259 utilizando algoritmos evolutivos.

Esta tesis es la consecuencia de un extenso trabajo de investigación en el cual se adquirió un respaldo teórico en los principales tópicos involucrados, utilizando como

¹

故	猶	是	多	夫	故	是	天	天	為	圖	大	為
終	難	以	易	經	能	以	下	下	大	難	小	無
無	之	聖	必	諾	成	聖	大	難	於	於	多	為
難		人	多	必	其	人	事	事	其	易	報	事
矣			難	寡	大	終	必	必	易	怨	以	味
				信		不	作	作		德		無
						為	於	於				味
						大	細	易				

referencias a reconocidos investigadores en el área de los algoritmos evolutivos. Así, se consultaron libros y artículos de referentes como D. Goldberg, Z. Michalewicz, E. Alba, C. Cotta, J. J. Merelo y M. Mitchell. Adicionalmente se utilizaron artículos de M. Nowostawski, S. Thede y S. Nasmachnow entre otros. Respecto del problema de asignación de frecuencias, se consultaron fuentes tales como organizaciones nacionales e internacionales relacionadas a las comunicaciones inalámbricas y la organización oficial encargada del tema. Además, debido a que los algoritmos genéticos están inspirados, como su nombre lo indica, en aspectos biológicos se han consultado publicaciones clásicas de naturalistas como por ejemplo Charles Darwin.

Para poder formar un juicio responsable sobre el tema de investigación se realizaron pruebas experimentales utilizando datos reales obtenidos de fuentes oficiales del proyecto que trata esta problemática.

Este documento está compuesto de 5 capítulos. La presente introducción describe la motivación y objetivos del trabajo. El capítulo 2 versa sobre algoritmos genéticos, explicando su estructura, sus características, así como las variantes, sus aplicaciones y finalmente describe las diferentes versiones paralelas de algoritmos genéticos.

El problema para el cual este trabajo intenta hallar solución se presenta detalladamente en el capítulo 3, describiendo conceptos básicos de comunicaciones inalámbricas, el problema de asignación de frecuencias en general y una enumeración de sus diferentes variantes. Este problema cobra un aspecto más concreto al presentarse un conjunto de escenarios reales, con resultados publicados obtenidos utilizando otras técnicas de optimización combinatoria.

La técnica y el problema descritos en los capítulos 2 y 3 respectivamente se conjugan en el capítulo 4, donde se propone un algoritmo evolutivo para resolver el problema de asignación de frecuencia de los escenarios COST 259. Con el objetivo que el lector conozca profundamente el entorno de trabajo que se utilizó para esta investigación, este capítulo describe inicialmente la arquitectura del hardware, el sistema operativo y demás software utilizados. Finalmente se presentan los detalles de los experimentos realizados y se presentan y analizan los resultados obtenidos.

El capítulo 5 presenta las conclusiones del trabajo, así como las líneas abiertas a abordar en el futuro que se desprenden de esta investigación.

CAPÍTULO 2

ALGORITMOS GENÉTICOS

“Si la naturaleza es la respuesta ¿cuál era la pregunta?”

JORGE WAGENSBERG

Si la naturaleza es la respuesta ¿cuál era la pregunta?, 2002

2.1 – Introducción

En ambientes tan dispares como el industrial y el académico se presentan problemas de optimización de distinto grado de dificultad. Algunos de ellos forman parte de un conjunto especial denominado NP. Un problema perteneciente a éste grupo es aquel para el cual no existe un algoritmo conocido de resolución cuyo costo computacional guarde una relación polinomial respecto al tamaño de la entrada del problema.

Resolver dichos problemas de manera clásica (mediante métodos exactos ó exhaustivos), asegura encontrar la solución óptima, pero demanda un gran esfuerzo computacional y puede tornarse impracticable. En otras palabras, no podrán obtenerse resultados en tiempos razonables.

Una alternativa a los métodos exhaustivos son las técnicas heurísticas, que intentan encontrar el óptimo explorando el espacio de soluciones sin barrerlo completamente. Este recorrido puede ser guiado de manera aleatoria, determinística o mediante una combinación de ambas en la cual la componente aleatoria ayuda a explorar diferentes sectores del espacio de soluciones y la componente determinística guía la búsqueda hacia soluciones de calidad.

La ventaja de estas técnicas es que encuentran buenas soluciones en tiempos razonables. Como contrapartida, no se puede tener la seguridad de haber encontrado el óptimo ni saber cuál es el error respecto del mismo.

Ciertos métodos heurísticos que no están fuertemente acoplados al problema que resuelven, definen un nivel mayor de abstracción y son conocidos como metaheurísticas.

Ejemplos de estas técnicas son:

- Recocido simulado (*simulated annealing*): está inspirado en el proceso de templado (*annealing*) en metalurgia, una técnica consistente en calentar y luego enfriar controladamente un material para aumentar el tamaño de sus cristales y reducir sus defectos. El calor causa que los átomos se salgan de sus posiciones iniciales (un mínimo local de energía) y se muevan aleatoriamente mientras que el enfriamiento lento les da mayores probabilidades de encontrar configuraciones con menor energía que la inicial.
- Búsqueda tabú (*tabu search*): combina una búsqueda local (*local search*) con una heurística para evitar detenerse en mínimos locales y entrar en ciclos. La idea básica es continuar buscando cuando se llega a un mínimo local al permitir movimientos que no mejoran la solución. Para evitar regresar a soluciones pasadas e iterar indefinidamente, se utiliza una memoria temporal, llamada lista tabú, que registra la historia reciente de la búsqueda. En resumen, es una combinación de búsqueda local con un mecanismo de memoria a corto plazo.

- Algoritmos evolutivos (*evolutive algorithms*): son métodos que emulan la evolución de las especies en la naturaleza, basándose en conceptos de genética. Se trata de metaheurísticas muy difundidas y serán analizadas en detalle en las secciones subsiguientes, pues son tópicos centrales de este trabajo.

Este capítulo inicia presentando la relación analógica existente entre la computación evolutiva y la evolución natural. También brinda una clasificación de algoritmos evolutivos profundizando en los algoritmos genéticos, de los cuales se presenta su estructura básica y luego una descripción detallada de cada uno de sus pasos. Luego se exponen elementos de su fundamentación matemática así como una lista de las aplicaciones más difundidas. Por último, se introduce el concepto de algoritmos genéticos paralelos brindando también una taxonomía.

2.2 - Computación Evolutiva

La computación evolutiva es la rama de la inteligencia artificial que toma como base las ideas de evolución y herencia genética propuestas por Charles Darwin y Gregor Mendel respectivamente. Un algoritmo evolutivo, entonces, es aquella técnica que incorpora dentro de su estructura y funcionamiento conceptos como selección natural, competencia, reproducción y mutación.

El propósito de estos algoritmos es buscar dentro de un conjunto finito de posibles soluciones al problema a resolver, intentando hallar la óptima, sin garantía de lograrlo. Esta búsqueda se lleva a cabo emulando a la evolución natural de una población de individuos de una determinada especie.

2.2.1 - Fundamentos biológicos de la computación evolutiva

El concepto de “selección natural” o “supervivencia del más adecuado” fue introducido por Charles Darwin en su libro “El origen de las especies” [Darwin, 1859] para referirse a la relación directa que existe entre el nivel de aptitud de los individuos en el medio en el que viven y las posibilidades de sobrevivir (y por ende reproducirse) con las que cuentan.

Un ejemplo simplificado ayudará a profundizar en el entendimiento de este concepto: supongamos que observamos a las palomas en dos lugares diferentes del planeta. Uno es una pradera blanca inmaculada a causa de la nieve y el otro, un monte de color cobrizo debido a la vegetación característica de la zona. Supongamos también que en ambos lugares existen aves rapaces predatoras cuyo sentido más utilizado a la hora de divisar sus presas es la vista. ¿Qué palomas tendrán menos posibilidades de ser capturadas en la blanca pradera? Sin duda aquellas cuyo plumaje sea más claro y les permita pasar desapercibidas a los ojos de los gavilanes. Sin embargo, en el otro escenario planteado, las favorecidas serán, con seguridad, aquellas de plumaje cobrizo.

Mayores probabilidades de sobrevivir implican mayores chances de reproducirse, pues se reproducen más aquellos individuos que viven más, y por lo tanto legan sus cualidades genéticas a las generaciones subsiguientes.

Volviendo al ejemplo anterior, toda variación en la estructura genética de un individuo respecto de sus antepasados, ya sea por combinaciones entre las composiciones genéticas de los progenitores o una mutación que introduzca una característica inexistente no heredada, que involucre que una paloma nazca más blanca, tendrá más posibilidades de formar parte de la herencia genética de la comunidad de las palomas de la nieve, pues mientras más pálidas sean las mismas, más desapercibidas pasan en este entorno y (como consecuencia) tienen más posibilidades de reproducirse.

Es decir, la especie *evoluciona* hacia individuos más blancos y por ende más aptos para las zonas nevadas.

Como contrapartida, si naciera una paloma albina en el monte, seguramente tendría muy pocas chances de reproducirse, pues sería muy divisable a la distancia. Esto invita a apoyar la idea que la evolución biológica no es un optimizador de los seres que la experimentan, sino un proceso de adaptación que involucra cambios y donde se favorece la supervivencia de los individuos más idóneos. Esta aptitud es relativa al tiempo y espacio que transitan dichos seres [Merelo, 1997].

En síntesis, la selección natural es un mecanismo ejercido por la naturaleza sobre todos los individuos de una especie por el cual el más adaptado tiene más posibilidades de sobrevivir y de propagar las cualidades genéticas que los hacen ser los más adecuados a sus descendientes. ¿Esto quiere decir que existe la seguridad de que un individuo mejor que otro siempre viva y se reproduzca más? No, es simplemente una cuestión probabilística.

2.2.2 - Correspondencias entre la evolución biológica y la computacional

Los algoritmos evolutivos intentan resolver problemas emulando la capacidad que tiene la naturaleza para producir individuos cada vez más adaptados dentro del ambiente donde viven. Este mecanismo se llama evolución.

A continuación se establece una analogía entre la evolución natural y la artificial, simplificando al máximo los conceptos de la primera, puesto que los autores de este documento no son (ni pretenden ser) expertos en biología.

Así como en la naturaleza existen muchos individuos de una especie, unos más aptos que otros, que se reproducen y evolucionan hacia un óptimo; en la evolución computacional también se dispone de una población formada por un conjunto de individuos, donde cada uno es una codificación de una posible solución (cuya calidad puede ser desde excelente a pésima) al problema que se quiere resolver. Dicha calidad es perfectamente cuantificable en los algoritmos evolutivos y se calcula mediante una función llamada *fitness*. Otra analogía presente entre el mundo natural y el computacional es que en ambos interviene la selección natural, mecanismo que da más posibilidades de reproducirse a aquellos individuos más aptos. La reproducción en ambos mundos también tiene correspondencia directa ya que las codificaciones de los individuos seleccionados como padres se combinan para que los descendientes cuenten con el material genético que hizo que sus ancestros fueran elegidos para reproducirse. Para no converger rápidamente a un mínimo local (en la analogía biológica, que todas las palomas de la nieve lleguen a ser bastante claras, pero nunca blancas totalmente) se introducen variaciones genéticas aleatorias, conocidas como mutación (también presentes en la naturaleza), para explorar con material genético no presente entre los reproductores disponibles. Por último, una vez realizado el proceso evolutivo de evaluación de individuos, selección de reproductores, recombinación (reproducción) y mutación se procede al reemplazo de la nueva generación por la anterior.

2.3 - Clasificación de algoritmos evolutivos (AE)

Una taxonomía de estos algoritmos propuesta por Alba y Cota [2003] se esquematiza mediante la “Ecuación Evolucionaria”, presentada en la figura .2.1, donde se muestran los algoritmos evolutivos como la conjunción de los algoritmos genéticos, las estrategias de evolución y la programación evolutiva.

$\mathbf{CE = AG + EE + PE}$	<p>CE = Computación Evolutiva AG = Algoritmos Genéticos EE = Estrategias de Evolución PE = Programación Evolutiva</p>
------------------------------	--

Figura 2.1 – “Ecuación Evolucionaria”

A continuación se describen brevemente los últimos dos tipos de algoritmos del segundo miembro de la ecuación evolucionaria. Los algoritmos genéticos son tratados en detalle en la sección 4.

2.3.1 - Estrategias de evolución

Las estrategias de evolución (EE) son métodos de optimización paramétricos, que trabajan sobre un conjunto de vectores de valores reales (población de cromosomas) y utilizan la mutación como operador evolutivo fundamental. Se comenzaron a desarrollar por I. Rechenberg en la Universidad Técnica de Berlín en 1964 y su propuesta inicial fue luego secundada por variantes introducidas por Schwefel. La primer variante fue llamada $EE - (1+1)$. Esta alternativa trabaja con dos individuos, un padre y un descendiente por generación. El mejor de ambos pasa a ser padre de la siguiente generación.

Luego de la $EE - (1+1)$, Schwefel propuso las variantes $EE - (\mu, \lambda)$, $EE - (\mu + \lambda)$ con $\mu > 1$ padres y $\lambda > 1$ descendientes por generación.

En el modelo $EE - (\mu, \lambda)$, inicialmente se seleccionan ρ ($1 \leq \rho \leq \mu$) padres de la población con probabilidad uniforme, que generan λ descendientes aplicando el operador de mutación. Luego se seleccionarán entre éstos a los μ padres de la próxima generación.

En el modelo $EE - (\mu + \lambda)$ lo que se modifica es la selección, pues los μ mejores individuos del conjunto $\mu + \lambda$ son los padres de la siguiente generación.

2.3.2 - Programación evolutiva

La programación evolutiva PE fue propuesta por Fogel [1966], y nació con el fin de utilizar a la evolución para crear máquinas inteligentes, que pudieran prever su entorno y reaccionar adecuadamente a él.

Como modelo de estas máquinas fueron usados los autómatas celulares, definidos por un conjunto de estados y reglas de transición entre ellos, que al recibir una entrada, cambian o no de estado y producen una salida. Fogel trataba de hacer aprender a estos autómatas a encontrar regularidades en los símbolos que recibían.

Retomando estas ideas, John Koza [1990] dio formalmente el nombre de programación evolutiva y precisó que la misma aplica el enfoque de los algoritmos genéticos en el espacio de soluciones de los programas posibles para resolver determinado problema. En otras palabras, un individuo de la población consiste en un algoritmo que resuelve un determinado problema. Para lograr esta tarea, pueden utilizarse diferentes representaciones como reglas *if-then*, árboles de decisión, redes neuronales, lógica proposicional, gramáticas formales, etc.

Las principales líneas de trabajo sobre programación evolutiva en la actualidad se encuentran centradas en el entrenamiento de redes neuronales y evolución de sistemas difusos.

Las estrategias de evolución y la programación evolutiva tienen la misma base teórica con analogía biológica que los algoritmos genéticos, descritos a continuación.

2.4 - Algoritmos Genéticos

Los algoritmos genéticos (AG) constituyen la técnica de computación evolutiva más difundida y se destacan por utilizar como operador evolutivo principal la recombinación, también llamada cruzamiento (*crossover*), consistente en combinar material genético de dos o más individuos llamados progenitores o padres para producir otros llamados descendientes o hijos.

En cualquier algoritmo genético existen dos características estrictamente dependientes del problema a resolver: la codificación de las soluciones como individuos de la población y la función de evaluación del valor de aptitud (*fitness*) de cada una de las soluciones generadas. Las otras características de un AG (recombinación, selección, reemplazo, mutación, etc.) pueden abordarse independientemente del problema considerado.

2.4.1 - Codificación de las soluciones

Los AG no operan directamente con las soluciones al problema que intentan resolver sino que cada una de éstas es codificada en una estructura de datos conocida como cromosoma o individuo.

En biología, se conoce como genotipo al conjunto de cromosomas que definen las características de un individuo. La manifestación visible del genotipo en un determinado ambiente es llamada fenotipo.

En los AG, el genotipo también está formado por cromosomas, pero como generalmente un cromosoma se corresponde con una solución al problema, se habla indistintamente de cromosoma, genotipo e individuo. El fenotipo es la materialización de un punto del espacio de soluciones del problema. Por ejemplo si un AG busca el mejor carácter ASCII para resolver determinado problema, el genotipo 1000001 representa a la letra "A" (fenotipo).

Otros términos que tienen correspondencia con la naturaleza, y presentan sutiles diferencias entre ellos se detallan a continuación:

- **Cromosoma:** individuo de la población en el que se encuentran codificados los parámetros relevantes del problema. Un cromosoma está formado por un conjunto de uno o más genes.
- **Gen:** parte de un cromosoma que usualmente representa una característica o parámetro del problema a resolver.
- **Alelo:** valor que toma un determinado gen.

En la figura 2.2 se grafica la relación entre estos conceptos.

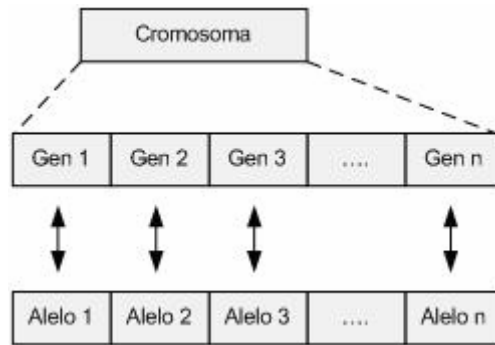


Figura 2.2 – Estructura de un cromosoma

La codificación de las soluciones en cromosomas consiste en representarlos como cadenas de símbolos pertenecientes a alfabetos de variada cardinalidad. Generalmente en los AG esta cardinalidad es 2, es decir se utiliza codificación binaria $\{0,1\}$ y se habla de cadenas de bits (*bitstrings*). Dependiendo del problema a resolver puede ser más conveniente utilizar elementos no binarios como números reales, enteros y otros. De la misma manera, las estructuras que los contengan pueden diferir de simples *strings*, utilizándose matrices, árboles, grafos, etc..

2.4.2 - Estructura básica de un algoritmo genético

Un algoritmo genético tiene como *modus operandi* la creación de una población inicial de individuos que evolucionará mediante un procedimiento iterativo, donde cada iteración representa el paso de una generación a otra. Un ciclo consiste en evaluar qué tan bueno es cada uno de los individuos y seleccionar una población intermedia $P(n')$ de padres dándole mayor probabilidad de ser elegidos a aquellos individuos mejores. A esta población intermedia se le aplican los operadores evolutivos cruzamiento y mutación y luego se reemplaza la población $P(n')$ por $P(n+1)$. El pasaje de una generación a la siguiente es esquematizado en la figura 2.3.

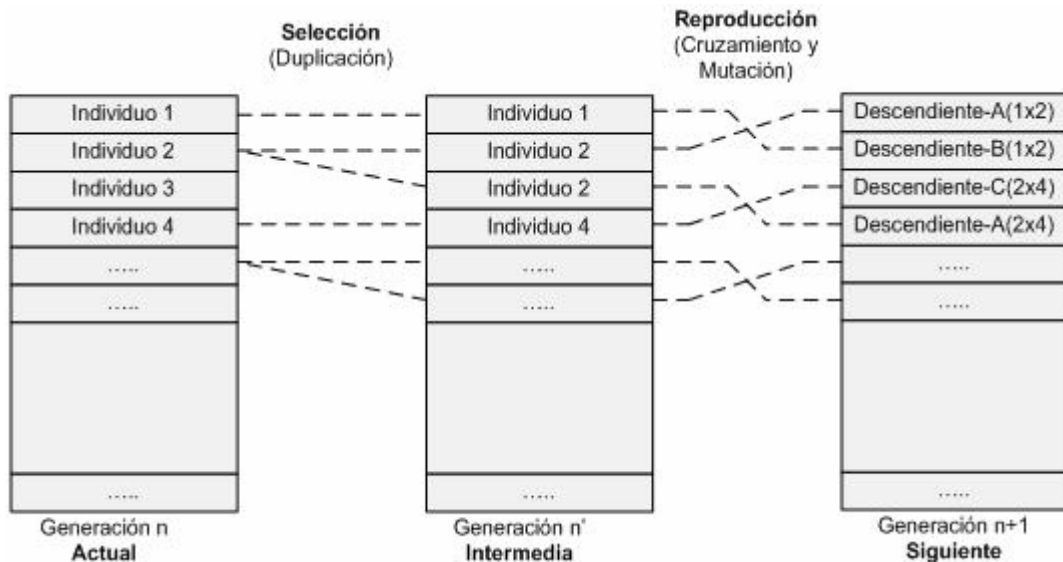


Figura 2.3 – Transición entre las generaciones n y $n+1$, a través de la generación intermedia n'

Finalizado el proceso evolutivo, se retorna el mejor individuo de la última generación. La condición de fin de la iteración suele ser el alcance de un determinado número de generaciones o valor de *fitness*. La Figura 2.4 presenta el esquema genérico de un AG.

```

(1) Generar P(0);
(2) Generación ← 0;
(3) Mientras No (criterio de parada) hacer
    Comienza
        (4) Evaluar P(Generación)
        (5) Población Intermedia ← Seleccionar (P(Generación))
        (6) Población Intermedia ← Aplicar Crossover (Población
Intermedia)
        (7) Población Intermedia ← Aplicar Mutación (Población
Intermedia)
        (8) P(Generación +1) ← Reemplazar (P(Generación), Población
Intermedia)
        (9) Generación ← Generación +1
    Termina
(10) Retornar (Mejor Individuo Encontrado)

```

Figura 2.4 – Estructura genérica de un algoritmo genético.

En las secciones siguientes se describirán cada una de las etapas de un AG, presentando algunas de las variantes más difundidas.

2.4.2.1 - Generación de la población inicial

El paso (1) presentado en la figura 2.4 (*Generar P(0);*) representa el inicio de la ejecución de un algoritmo genético, consistente en generar la población inicial, que servirá como base para evolucionar hacia mejores individuos conforme avanzan las generaciones. Téngase en cuenta que la población es un conjunto de individuos y cada uno de éstos representa una solución al problema a resolver.

El proceso de generar la población inicial puede ser aleatorio o arbitrario, es decir el algoritmo podría partir de un conjunto de soluciones aceptables, conocidas, producto de otro algoritmo de optimización o cualquier otra técnica que pueda servir como base.

2.4.2.2 - Evaluación

El primer paso de cada iteración (4) presentado en la figura 2.4 (*Evaluar P(Generación)*) consiste en la evaluación de la calidad de los individuos de la población. Los términos “función de evaluación” y “función de *fitness*” suelen utilizarse indistintamente en muchos textos. Sin embargo, la diferencia entre ellos amerita ser aclarada: la primera devuelve un valor absoluto asociado a la calidad de un individuo determinado, en cambio la segunda evalúa la bondad del individuo relativa al resto de la población y se traduce de alguna manera en probabilidades de supervivencia [Whitley, 1994] [Thede, 2004].

Clarifiquemos esta diferencia con el siguiente ejemplo: supongamos que intentamos maximizar a la función de evaluación $FE(x, y) = 6xy - y^2 + 2x$ cuyas variables independientes pueden tomar valores enteros desde 0 a 15. Una codificación posible es la siguiente: cada individuo i será un *string* de 8 *bits* donde los cuatro primeros representen el valor de x y los restantes el de y . De esta manera, el *string* 00010010 representa aquella solución al problema que tiene como parámetros $x=1$ e $y=2$. En este caso el resultado de la función de evaluación se obtendrá calculando la función a maximizar utilizando los parámetros dados. Pensemos ahora en la función de *fitness*. ¿Existe una única función de *fitness* posible? No. Podría ser el resultado de FE dividido el promedio de FE de toda la población, o sustraído al valor de FE del individuo *poor*, o simplemente coincidir con la función de evaluación de cada individuo.

En la tabla 2.1 se muestra una población de 5 individuos, los valores de los parámetros para cada uno de ellos y la función de evaluación. La función de *fitness* elegida en este caso es la primera de las tres mencionadas anteriormente y la definimos formalmente como $f(i) = \frac{FE(x_i, y_i)}{FE_{prom}}$ donde i es el i -ésimo individuo de la población cuyos parámetros al decodificarlo son x_i e y_i , y FE_{prom} es el promedio de la función de evaluación para todos los individuos de la población.

Cromosoma (genotipo)	Individuo (fenotipo)		$FE(x_i, y_i)$	$f(i)$ ($FE_{prom} = 511.8$)
	x_i	y_i		
11001101	12	13	791	1.545525595935910
11010100	13	4	322	0.629152012504885
11111001	15	9	759	1.483001172332940
11101010	14	10	768	1.500586166471280
00001001	0	9	-81	-0.158264947245018

Tabla 2.1 – Comparativa genotipo vs fenotipo y función de evaluación vs función de *fitness*.

La entrada para la función de *fitness* es el fenotipo del individuo a evaluar y su salida es un valor llamado "valor de *fitness*" con el cual se conoce a ciencia cierta qué tan buena es esa solución.

De acuerdo a Alba y Cota [2003], al momento de la construcción de la función de *fitness* existen ciertos aspectos a tener en cuenta. Dichos aspectos son:

- La función de *fitness* no debe estar ajena al criterio del problema de optimización, es decir, si se trata de un problema de minimización o maximización de un objetivo. También se deben contemplar las restricciones existentes en el problema. Por lo tanto si surgen soluciones no factibles debe asignárseles un valor de *fitness* adecuado que garantice que la misma no perdure en el proceso evolutivo (en otras palabras, "que no sobreviva").
- Deben contemplarse aquellos casos en los cuales el entorno presente problemas para la evaluación de la función de *fitness*, utilizando evaluaciones parciales cuando existen valores de *fitness* no definidos. Asimismo, debe considerarse el uso de funciones de *fitness* multivaluadas que asignen diferentes valores a un mismo individuo para simplificar la labor del operador de selección.
- Se debe contemplar si la función de *fitness* varía dinámicamente durante la evolución del algoritmo genético. En este caso mecanismos complejos como las representaciones múltiples son útiles para introducir memoria en la operativa del algoritmo genético.
- Como el cálculo de la función de *fitness* posiblemente será complejo, esto demandará un alto esfuerzo computacional. Se debería pensar en utilizar técnicas de computación paralela para obtener mayor eficiencia en el algoritmo genético.

Para resolver problemas de dominancia de soluciones en generaciones tempranas de la evolución, y para evitar poblaciones similares al final de la evolución, debe considerarse el *escalado* de los valores de *fitness* [Michalewicz, 1992]. Esto se debe a que cualquier selección proporcional al *fitness*, de acuerdo a las características inherentes del algoritmo genético, padecen los siguientes problemas:

1. En generaciones tempranas, los valores de la función de *fitness* de cada uno de los individuos son muy dispares (algunos individuos son muy buenos en relación a la población en general y otros son muy malos). Esto indefectiblemente lleva a seleccionar muchas veces a individuos pertenecientes al subconjunto de los más adaptados, lo que genera mayor cantidad de descendientes de éstos en la población. Como consecuencia dicha población se encontraría dominada por descendientes de características genéticas similares, provocando el estancamiento de la búsqueda en óptimos locales (convergencia prematura).
2. En generaciones avanzadas, los valores de los *fitness* son muy similares y entonces “la pelea por ser mejor es muy dura”, es decir, la diferencia de probabilidad de selección entre individuos es despreciable. Esta situación se traduce en un estancamiento de la evolución del algoritmo (*random walk*).

2.4.2.3 - Selección

Una vez evaluada la función de *fitness* para cada cromosoma de la generación n , debe seleccionarse una población intermedia (generación n') de individuos. Ésta se convertirá en la población de la generación $n+1$ una vez aplicados los operadores evolutivos (cruzamiento y mutación). En la figura 2.4 se identifica la selección con el número (5) (Población Intermedia \leftarrow Seleccionar (P(Generación))).

El proceso de selección es, sin duda, la característica fundamental que hace que un algoritmo genético sea una técnica de búsqueda guiada (y no aleatoria), pues se trata del elemento que establece una tendencia hacia soluciones mejores conforme avanzan las generaciones. Esto se logra dándole mayores posibilidades de sobrevivir (formar parte de la población de la generación n' , y por ende de la población de la generación $n+1$, ya sea participando en ésta última directa o indirectamente²) a aquellos individuos más adecuados.

El término “más adecuado” (*fitter*) refiere a aquellas soluciones que mejor resuelven el problema respecto de los parámetros y estructura que se haya considerado al construir la función de *fitness*. En definitiva, son más adecuadas aquellas soluciones cuyo valor de *fitness* sea mayor.

El diseño del criterio de selección en un AG es un punto clave para la calidad del mecanismo de búsqueda, ya que se necesita mejorar la aptitud de los individuos sin perder la siempre deseable diversidad. Esto se logra dándole mayor probabilidad de sobrevivir a aquellos individuos más adecuados sin impedir la proliferación de aquellos no tan buenos.

Para ejemplificar la importancia de la característica descrita anteriormente, a continuación se presenta un ejemplo en el cual el criterio de selección sea elegir a los k mejores individuos. Esto es, en principio, injusto para los más aptos, pues tienen la misma participación en la generación n' que aquellos no tan buenos. Suponiendo que los k individuos que forman la generación n' son I_1, I_2, \dots, I_n y están ordenados de tal forma que I_i es mejor que I_{i+1} el peor, vemos aquí claramente la igualdad de condiciones en las que se encuentran, pues ya fue aplicado el mecanismo de selección y este conjunto es la resultante del mismo. Por otra parte, este primer intento de selección no

² Un individuo seleccionado en la generación n forma parte directa o indirectamente en la generación $n+1$ ya que cada uno es sometido a *crossover* y mutación con probabilidades p_c y p_m respectivamente. En el caso de que un individuo no sea cruzado ni mutado, forma directamente parte de la generación siguiente, en cambio si sufre alguna variación, interviene indirectamente, aportando su material genético para formar nuevos individuos que sí serán parte de la generación posterior.

da posibilidad de reproducción a los peores individuos, característica que es deseable (sobre todo en generaciones tempranas) para mantener la diversidad y no converger prematuramente a un óptimo local.

Varios mecanismos de selección han sido desarrollados para los algoritmos genéticos, a continuación se describen los más difundidos [Mitchell, 1999].

- **Selección proporcional al *fitness*.**

Esta selección, propuesta por Holland [1975] en su algoritmo genético original, se basa en el concepto estadístico de esperanza matemática. Por esto, se *espera* que haya f_x/f_p copias del individuo x en la generación intermedia, siendo f_x el valor del *fitness* del individuo en cuestión y f_p el promedio del *fitness* de todos los individuos de la generación.

Una posible implementación de esta técnica de muestreo, conocida como “Rueda de Ruleta” (*Roulette Wheel*) [Goldberg, 1989], consiste en simular el giro de una ruleta con n compartimentos, cada uno con tamaño proporcional al *fitness* del individuo que representa. Realizando n giros se seleccionan n individuos.

Otra solución posible, que acerca más el valor real de la cantidad de veces que un individuo es seleccionado a su valor esperado, es la selección universal estocástica (*stochastic universal*), donde ya no se considera una ruleta con una aguja que se gira n veces, sino n agujas y un solo giro.

- **Selección por ranking**

Este mecanismo categoriza a los individuos por su posición respecto del resto de la población de acuerdo al valor de la función de *fitness*. Es decir el individuo con mejor valor de *fitness* se ubica en la primera posición y el individuo con el peor valor se sitúa en la última posición del *ranking*.

La imposibilidad de saber si el individuo de la posición k es ampliamente superior al de la posición $k-1$ o la diferencia entre ellos es casi insignificante, elimina la convergencia prematura a la que puede llevar la selección proporcional respecto del valor de *fitness* absoluto de cada individuo.

Utilizar este tipo de selección es adecuado para explorar el espacio de soluciones en generaciones tempranas, sin embargo su velocidad de convergencia no es tan alta como la de otros métodos de selección.

- **Selección por torneo o competición.**

Es similar a la selección por *ranking* en cuanto a presión selectiva, pero mucho más eficiente en términos de costos computacionales y funciona de la siguiente manera: dos individuos de la población son seleccionados al azar y luego, también aleatoriamente, se elige un número $r(0 \leq r \leq 1)$. Si $r < k$, donde k es un parámetro del mecanismo de selección que identifica el rigor de la presión selectiva, entonces el mejor individuo del par es el elegido. De lo contrario, será seleccionado el peor.

Nótese que esta técnica debe su nombre a que dos individuos “compiten” entre sí, teniendo más probabilidades de “ganar” aquel individuo mejor adaptado. Esto es lo que Darwin llamaba selección sexual, donde, por ejemplo,

dos leones machos pelean entre sí, siendo el vencedor quien puede aparearse con las hembras.

- **Selección de Boltzmann**

Además de los clásicos mecanismos descritos en las secciones anteriores, existen otras variantes que suelen presentar buenos resultados. Un ejemplo es este tipo de selección cuya característica fundamental es que la presión selectiva es más rigurosa conforme avanzan las generaciones (iteraciones de nuestro algoritmo). Esto se logra introduciendo al cálculo del valor esperado de un individuo, el parámetro t que identifica la generación en la cual se encuentra el algoritmo.

La idea básica radica en que al principio es necesario explorar más que explotar. En otras palabras: podría ser deseable que el algoritmo realice una exploración del espacio de soluciones en etapas tempranas y en sus generaciones finales, cuando es de suponer que ya se encuentra cerca del óptimo, converja más rápidamente. La presión selectiva juega un papel fundamental en este comportamiento, pues mientras más relajada sea, la búsqueda de nuestro algoritmo abarcará gran parte del espacio de soluciones, y conforme ésta aumente, la búsqueda será más localizada, es decir, sólo en una determinada porción del mismo [Whitley, 1994].

2.4.2.4 - Cruzamiento

En el paso (6) de la figura 2.4 (Población Intermedia \leftarrow Aplicar Crossover (Población Intermedia)) se representa la aplicación del principal operador evolutivo y consiste en un intercambio de material genético entre varios individuos. Según Holland [1975] no hay algoritmo genético sin cruzamiento, sin embargo sí es posible prescindir de la mutación.

La lógica de este operador es seleccionar varios individuos de la población (generalmente dos) llamados padres o progenitores y cruzarlos de alguna manera para obtener individuos descendientes u *offprings* (también generalmente dos).

A continuación se mencionan las clases de cruzamiento más difundidas.

- **Cruzamiento simple, o de un punto**

Consiste en elegir aleatoriamente un punto de corte que servirá de referencia para dividir cada uno de los cromosomas padres en dos partes. A partir de este punto se intercambia el material genético de cada uno de los individuos.

Este cruzamiento se muestra gráficamente en la figura 2.5

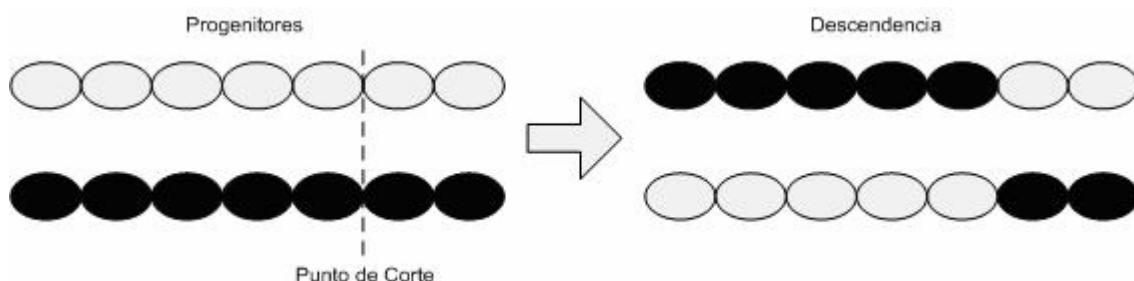


Figura 2.5 – Cruzamiento simple o de un punto

- **Cruzamiento multipunto**

Esta variante es la generalización del cruzamiento simple, considerando $n > 1$ puntos de corte que generan $n + 1$ segmentos que son alternados para formar los descendientes.

En la figura 2.6 se presenta un ejemplo de cruzamiento multipunto con $n = 2$.

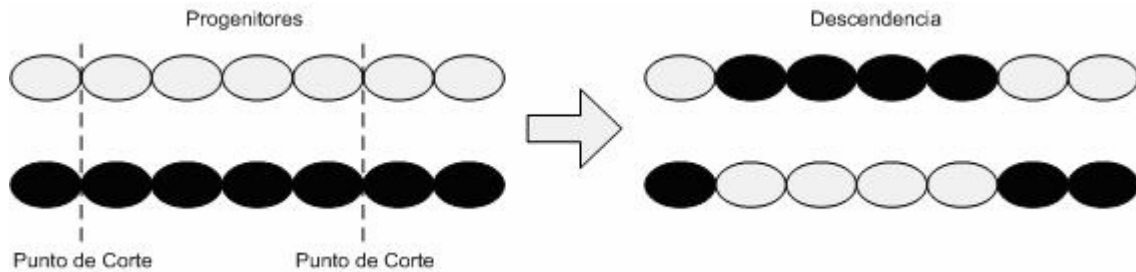


Figura 2.6 – Cruzamiento de dos puntos

- **Cruzamiento segmentado**

Es una variante del anterior donde la existencia de un punto de corte entre cualesquiera dos genes está determinada por una probabilidad p_s (probabilidad “de segmentación”). Esta probabilidad se evalúa a medida que se recorre la cadena. Suponiendo que p_s está establecida en 0.6, al generar un número aleatorio $s(0 \leq s \leq 1)$ entre los genes de las posiciones k y $k+1$, existirá un punto de corte en dicha posición si $s < 0.6$. Esto se presenta gráficamente en la figura 2.7.

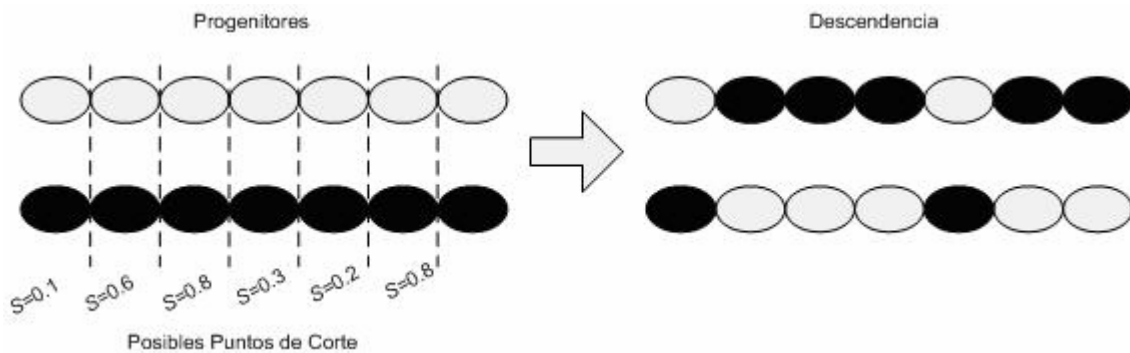


Figura 2.7 – Cruzamiento segmentado ($p=0.6$)

- **Cruzamiento uniforme**

Este tipo de cruzamiento genera dos descendientes dados dos padres. El primero es generado asignando a cada posición material genético seleccionado aleatoriamente (de la misma posición) perteneciente a uno u otro padre. El segundo descendiente se forma con el material complementario que no se asignó al primero.

En la figura 2.8 se esquematiza el cruzamiento uniforme.

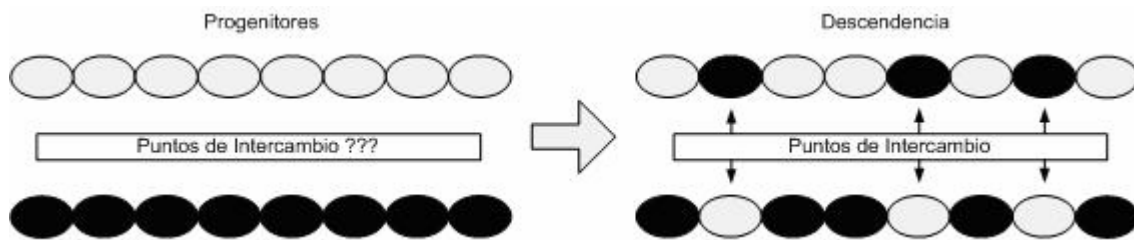


Figura 2.8 – Cruzamiento uniforme.

- **Cruzamiento Aritmético**

Cuando los individuos se codifican mediante valores continuos (números reales) puede utilizarse este tipo de cruzamiento en el cual los descendientes se obtienen a partir de una combinación lineal de sus progenitores o padres, es decir que surgen de una función lineal que está conformada por los progenitores y por un parámetro que puede ser constante (*cruzamiento aritmético estacionario*) o variable (*cruzamiento aritmético no estacionario*). De este modo al cruzar los cromosomas x_t e y_t , se obtienen los descendientes x_{t+1} e y_{t+1} según las formulas de la figura 2.9.

<p>a. $x_{t+1} = ax_t + (1 - a)y_t$</p> <p>b. $y_{t+1} = ay_t + (1 - a)x_t$</p>

Figura 2.9 – Fórmulas de cálculo de descendientes mediante cruzamiento aritmético.

- **Cruzamiento Especializado**

Este tipo de cruzamiento tiene como objetivo evitar la generación de individuos no válidos, teniendo en cuenta para esto información específica del problema (por ejemplo sus restricciones). La idea es guiar la búsqueda dentro de los límites del espacio de soluciones al problema a resolver [Merelo, 2005].

Algunos ejemplos de éstos son el cruzamiento geométrico (tiene la particularidad de producir un solo hijo como resultado del cruzamiento de dos padres) y el cruzamiento heurístico (se caracteriza por producir uno o ningún hijo al cruzar dos padres) [Michalewicz, 1999].

Cabe destacar que esta clase de operadores se aplica generalmente cuando las codificaciones son no binarias, por ejemplo de punto flotante.

2.4.2.5 - Mutación

A pesar de que el cruzamiento es la fuente de variación y exploración más importante en los AG, existe un operador secundario llamado mutación cuyo objetivo es introducir diversidad genética y así poder “escapar” de la convergencia prematura a óptimos locales. Esto se representa en el paso (7) de la figura 2.4 (Población Intermedia \leftarrow Aplicar Mutación (Población Intermedia)).

Mutar un individuo consiste en alterarlo genéticamente de alguna manera. Esta variación se realiza con muy poca probabilidad (entre el 1% [Moore, 2001] y 0.1% [Goldberg, 1989]).

Generalmente se implementa modificando aleatoriamente alguno o algunos de los genes de los individuos a los que se les aplica.

En la mayoría de los casos el individuo mutado no sobrevive (mutación letal), pero la minoría restante contribuye a la diversidad de la población, permitiendo así escapar (quizás) de mínimos locales.

Mutar un individuo implica realizar un salto a un punto diferente del espacio de soluciones. Para clarificar esta idea, supóngase que codificamos la solución a nuestro problema con una cadena de 6 bits que representan dos parámetros enteros sin signo de tres bits cada uno. Así, el individuo 000111 es la codificación de aquella solución a nuestro problema cuyos parámetros son $x_1=0$ y $x_2=7$ respectivamente. Si la mutación consiste en cambiar el valor de un bit (alelo), entonces mutar un individuo modifica la solución que representa. Por ejemplo, si el individuo 101011 sufre una mutación en su tercer bit menos significativo, el resultante será 101111 y se ha transformado una solución al problema con parámetros $x_1=5$ y $x_2=3$ en una con parámetros $x_1=5$ y $x_2=7$. Este salto en el espacio de soluciones se muestra gráficamente en la figura 2.10.

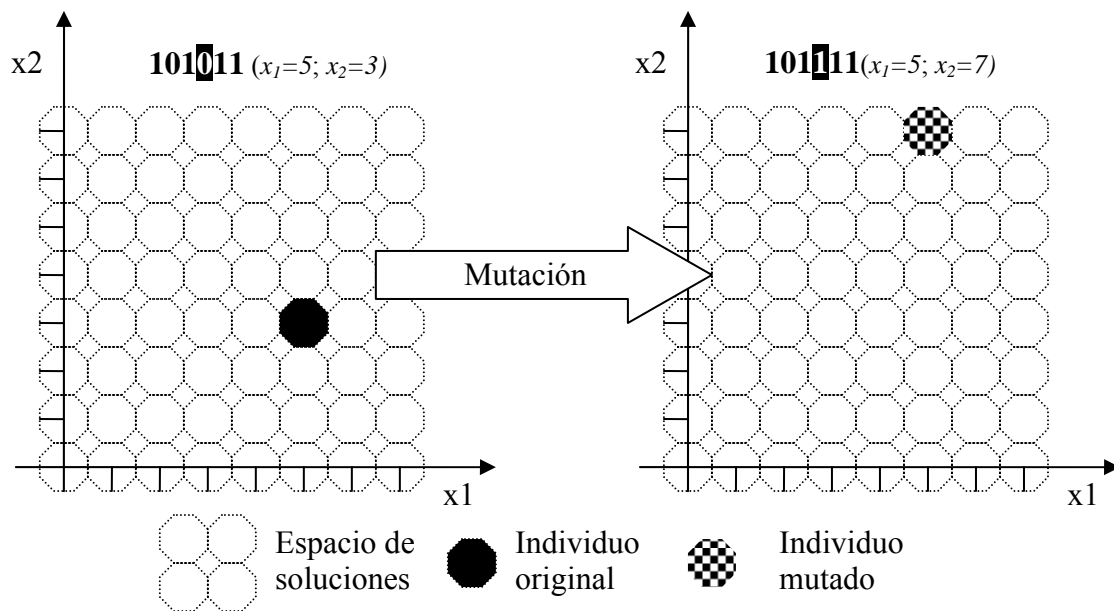


Figura 2.10 – Salto en el espacio de soluciones tras la mutación.

2.4.2.6 - Reemplazo

Luego de aplicar los operadores genéticos, deben reemplazarse los padres por los hijos como culminación de la iteración, produciendo así el cambio generacional. En la figura 2.4 esto se representa en el paso (8) ($P(\text{Generación} + 1) \leftarrow \text{Reemplazar}(P(\text{Generación}), \text{Población Intermedia})$).

Debido a la constante búsqueda de la calidad de las soluciones, elegir la mejor estrategia de reemplazo no es una tarea trivial. Esta actividad es análoga a la de selección, con la diferencia que en lugar de seleccionar los individuos que formarán parte de la próxima generación, se seleccionan aquellos que serán reemplazados por los nuevos cromosomas. Es importante destacar que no existe una relación de dependencia directa entre las estrategias de selección y de reemplazo del algoritmo, de manera que por ejemplo, un algoritmo genético puede realizar la selección por ranking y el reemplazo aleatorio. Por esto a lo largo de los años se han desarrollado diversas alternativas para mejorarlo. Las más conocidas se detallan a continuación:

- **Reemplazo elitista**

Todo mecanismo de selección asigna mayores posibilidades de sobrevivir a aquellos individuos más aptos. Sin embargo esta asignación es absolutamente probabilística y por ende es factible que uno o más de los mejores individuos de la población no resulten seleccionados.

Un AG es “elitista” cuando implementa un mecanismo para que los p mejores individuos formen parte de la generación de reproductores independientemente de los resultados probabilísticos del mecanismo de selección.

- **Reemplazo aleatorio**

Los individuos de la nueva generación que se reemplazarán por los seleccionados de la generación anterior son elegidos al azar. Esto puede disminuir la presión selectiva debido a que un individuo de buen *fitness* en la población y otro de peor *fitness* tienen las mismas posibilidades de ser reemplazados.

- **Reemplazo gradual o por torneo**

Generalmente la actualización gradual utiliza un método basado en el orden tanto para la selección como para el reemplazo, usualmente un método por torneo. El reemplazo por torneo es exactamente igual a la selección por torneo, excepto que las peores soluciones se eligen con mayor frecuencia que las mejores.

- **Reemplazo por rango**

A cada individuo de la población se le asigna un rango numérico basado en su aptitud, y la selección para el reemplazo se basa en este ranking, en lugar de las diferencias absolutas en aptitud.

- **Reemplazo por nichos**

Este tipo de reemplazo tiende a mantener la diversidad genética de la población, de forma que cromosomas similares sustituyan sólo a cromosomas similares, de esta manera se relaja la presión selectiva conforme avanzan las generaciones.

- **Reemplazo proporcional**

En esta variante, la probabilidad de reemplazo es inversamente proporcional a la aptitud del individuo. Utilizar este esquema favorece el reemplazo de aquellos con menor valor de *fitness*.

2.4.2.6.1 – Modelo Evolucionario

En el algoritmo genético simple propuesto por Goldberg [1989] todos los individuos de la población son reemplazados (*reemplazo generacional*), es decir la generación $n+1$ sustituye por completo a la generación n .

A continuación se enumeran otros modelos evolucionarios:

- **Gap Generacional**

Esta estrategia consiste en reemplazar sólo una porción de la población en cada iteración del algoritmo. Para esto se ordena la misma según el valor de

fitness de cada individuo y se eliminan los m peores, introduciendo m nuevos descendientes.

El factor m , que simboliza un porcentaje de la población, es conocido como *gap generacional*. Mientras menor sea m , más lento evoluciona el algoritmo, pero menores son las posibilidades de convergencia prematura.

- **Estado estacionario**

Si el factor m mencionado en la sección anterior es reducido a tal punto que solo muy pocos (incluso un único) individuos son reemplazados, entonces el algoritmo genético se dice “de estado estacionario” (*steady state*).

Este tipo de algoritmo presenta como ventaja la garantía de permanencia de los mejores individuos encontrados pero como contrapartida la evolución es más lenta.

2.4.3 - Fundamentos del mecanismo de exploración de los AG

El éxito de los algoritmos genéticos al explorar el espacio de soluciones fue explicado por Holland en 1975 mediante el concepto fundamental de esquema. Un esquema representa un hiperplano (subconjunto del espacio de soluciones) y puede definirse en términos de AG como un conjunto de cromosomas con características idénticas en determinados puntos de su codificación. Por ejemplo, si se codifican individuos como *bitstrings* de longitud 5, el esquema denotado por (101**) representa al conjunto de todos los individuos cuyos primeros tres bits son 1, 0 y 1 respectivamente, sin importar qué valor tengan los últimos dos (es por esto que el carácter * es conocido como “símbolo *don't care*”).

Dos propiedades importantes de los esquemas son su orden (cantidad de valores 0 y 1, también llamados fijos, del esquema) y la distancia entre la primera y última posición fija, conocida como largo de definición (*defining length*). Así, el esquema (**1**1*101**) tiene orden 5 y largo de definición 7.

Se ha descrito en secciones anteriores que la probabilidad de supervivencia de los individuos está directamente relacionada con su aptitud. Generalizando esta idea se demuestra que aquellos esquemas más aptos (conocidos también como esquemas aventajados), de bajo orden y de *defining length* pequeño, verán incrementada exponencialmente su presencia en la población conforme ésta evoluciona. Esto es conocido como el “teorema de los esquemas”.

Como consecuencia de este teorema surge la conocida hipótesis de bloques constructivos (*building blocks hypothesis*) que consiste en yuxtaponer durante el proceso de búsqueda esquemas cortos, de bajo orden y de alta aptitud (*building blocks*). De esta manera, si la codificación y el cruzamiento favorecen la existencia y permanencia de bloques constructivos, la eficacia del algoritmo se verá incrementada.

Nótese que en una población hay más esquemas que individuos representados. Holland demostró que la cantidad de esquemas procesados útilmente es del orden del cubo del tamaño de la población. Esta propiedad es conocida como paralelismo intrínseco (*implicit parallelism*).

2.4.4 - Aplicaciones de los algoritmos genéticos

Los algoritmos genéticos han sido y son aplicados con éxito a una amplia gama de problemas de índole comercial, académico y científico. Entre ellos se pueden mencionar [Obitko, 1998] [Alba y Cota, 2003]:

- Sistemas de predicción y análisis de datos, usando AG para aprender reglas simples basadas en datos numéricos, cantidad de mensajes por días y el número total de palabras publicadas en foros de mercados de finanzas.
- Diseño de redes neuronales, utilizando AG para la codificación de la estructura de redes neuronales sobre cromosomas binarios.
- Trayectoria de robots, para las tareas de planeamiento del camino que debe recorrer el brazo de un robot para desplazarse de un lugar a otro para cumplir con una tarea determinada.
- Estrategias de planeamiento, trabajos recientes han mostrado que búsqueda del conocimiento necesario para la toma de decisiones estratégicas de planeamiento pueden ser aprendidas mediante la utilización de técnicas de AG.
- El problema del viajante de comercio (*travelling salesman problem*), cuyo objetivo es encontrar una ruta para recorrer un conjunto de ciudades unidas mediante caminos, minimizando los costos del viaje.
- Programación de actividades (*scheduling*), para encontrar asignaciones eficientes de actividades en el tiempo. Un ejemplo son los horarios de las enfermeras en un hospital.
- Diseño paramétrico de aviones, donde lo que se propone es representar las tareas de diseño como un listado de parámetros a optimizar utilizando AG.
- Ruteo en redes telefónicas, aplicado en redes telefónicas del oeste de los Estados Unidos con el objetivo de minimizar los costos de comunicaciones mediante la optimización de las comunicaciones.
- Análisis de estructura de ADN, donde se utilizan AGs para determinar la estructura de una muestra de ADN basado en datos recolectados de la misma.

Además, en la actualidad, los algoritmos genéticos son aplicados a la resolución de problemas de asignación de frecuencias en redes inalámbricas, tema que es abordado en este trabajo.

2.5 – Algoritmos Genéticos Paralelos

Los AG secuenciales suelen presentar las siguientes tres desventajas [Nowostawski y Poli, 1999][Alba y Cotta, 2003]:

- Para mantener grandes poblaciones de individuos se necesita mucha memoria y puede tornarse ineficiente abordar problemas con estos requerimientos de manera secuencial.
- Al afrontar problemas complejos, la evaluación de la función de *fitness* puede ser muy costosa en tiempo de cómputo y el lapso demandado por una ejecución completa de un AG podría ser inaceptable (en situaciones extremadamente complejas, se han reportado casos de hasta un año).
- Los AG secuenciales pueden converger prematuramente hacia valores subóptimos.

Una solución para contrarrestar estos tres problemas consiste en paralelizar la ejecución de los AG. Un algoritmo genético que utiliza más de un procesador para llevar a cabo su ejecución, es llamado algoritmo genético paralelo (AGP).

Además de las mejoras en el rendimiento de los sistemas paralelos respecto de sus pares secuenciales, pueden lograrse beneficios adicionales referentes a la calidad de las soluciones.

En la naturaleza, las especies evolucionan en paralelo en diferentes poblaciones realizando de algún modo intercambio genético entre ellas (más aún en este mundo globalizado). Esta característica permite ampliar la cantidad de focos de búsqueda dentro del espacio de soluciones aumentando la diversidad y por ende minimizando la probabilidad de converger prematuramente.

Varios modelos de paralelización han sido propuestos a lo largo de la historia de los AGP, en la sección siguiente se describen algunos de ellos.

2.5.1 - Clasificación de Algoritmos Genéticos Paralelos

Debido a que el acto de clasificar es inherentemente subjetivo, existen diferentes taxonomías de AGP. Sin embargo intentaremos ofrecer una clasificación coherente de acuerdo a los siguientes aspectos:

- La manera en que se evalúa el *fitness*.
- La existencia de una única población o varias subpoblaciones.
- El tipo de intercambio de individuos (en el caso de haber varias subpoblaciones).
- El modo en que se aplica la selección (global o localmente).

Cada una de las clases que se desprenden de estos criterios se describe en las secciones subsiguientes.

2.5.1.1 Paralelismo Maestro-Esclavo

El tipo de algoritmo genético paralelo maestro-esclavo utiliza una única población centralizada controlada por un procesador llamado maestro. Éste está encargado de la selección y la asignación a cada uno de los esclavos, quienes reciben uno o varios individuos a los cuales evalúan, cruzan (en caso de ser posible) y mutan para luego retornar los resultados al maestro.

La motivación de este modelo es el hecho ya conocido que la evaluación de *fitness* es el proceso computacional más costoso. Por ende distribuir este cómputo entre varios esclavos reduce los tiempos de ejecución y obtención de resultados.

Se denomina sincrónico a aquel modelo en el cual el maestro espera a que el más lento de los procesadores devuelva los resultados de la tarea asignada antes de comenzar con la siguiente generación. Nótese que esta variante se comporta exactamente como un AG secuencial, salvo por la reducción de los tiempos de ejecución.

Como contrapartida, en un AGP asincrónico el maestro avanza conforme los procesadores esclavos terminan sin esperar que todos completen su tarea.

En la figura 2.11 se esquematiza un AGP maestro-esclavo.

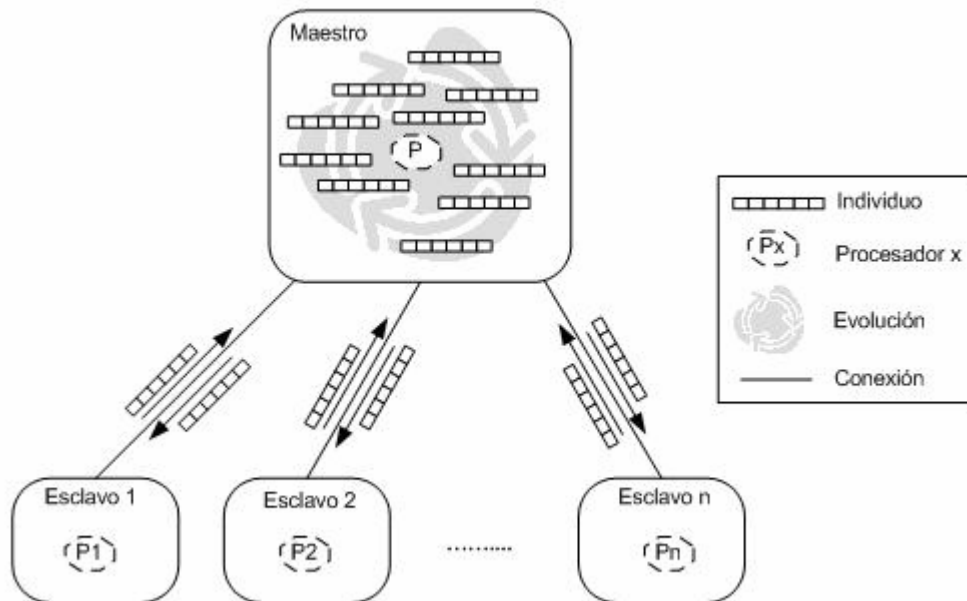


Figura 2.11 – AGP maestro-esclavo.

2.5.1.2 Paralelismo mediante subpoblaciones estáticas con migración

En el esquema de paralelismo mediante subpoblaciones estáticas con migración existen varias subpoblaciones (*demes*) que evolucionan independientemente intercambiando de manera periódica algunos individuos. Este intercambio es conocido como migración y consiste en copiar el o los individuos a migrar a la población destino (aunque acordaremos en el término “migración” pues es el común en la literatura, consideramos esto una clonación más que cualquier otra cosa).

Si un individuo puede migrar a cualquier subpoblación del conjunto de subpoblaciones existentes, decimos que el AGP adopta un “modelo de islas” (*island model*). En cambio si la migración está restringida a las subpoblaciones vecinas, se habla de un modelo “*stepping stone*”³.

La migración de un individuo depende de los siguientes parámetros:

- La topología de conexión entre las subpoblaciones (las más comunes son matricial, toroide e hipercúbica).
- La tasa de migración, que controla cuántos individuos migran cada vez.
- El esquema de migración, que determina cuáles individuos de la subpoblación origen migran hacia otra (los mejores, los peores, elegidos al azar), y cuáles son reemplazados en la subpoblación destino (los mejores, los peores, elegidos al azar).
- El intervalo de migración, que determina la frecuencia con la que uno o más individuos emigran hacia otra subpoblación.

La relación existente entre la cantidad de subpoblaciones y la cantidad de individuos que integran cada una suele utilizarse como parámetro para clasificar a este tipo de AGP. Un AGP de grano grueso (*coarse grained*) es aquel que trabaja con pocas subpoblaciones en cada una de las cuales coexiste un gran número de individuos. Por el

³ Este término inglés se utiliza cuando se quiere cruzar un río pisando las piedras que asoman en la superficie. La analogía aparece porque sólo es posible saltar de una piedra a cualquiera de sus vecinas.

contrario, aquellos algoritmos que manejen muchas subpoblaciones conformadas cada una por pocos individuos (al extremo de poder llegar a ser uno sólo), reciben el nombre de grano fino (*fine grained*).

En la figura 2.12 se presenta el modelo de islas, un esquema de paralelismo con subpoblaciones estáticas y migración.

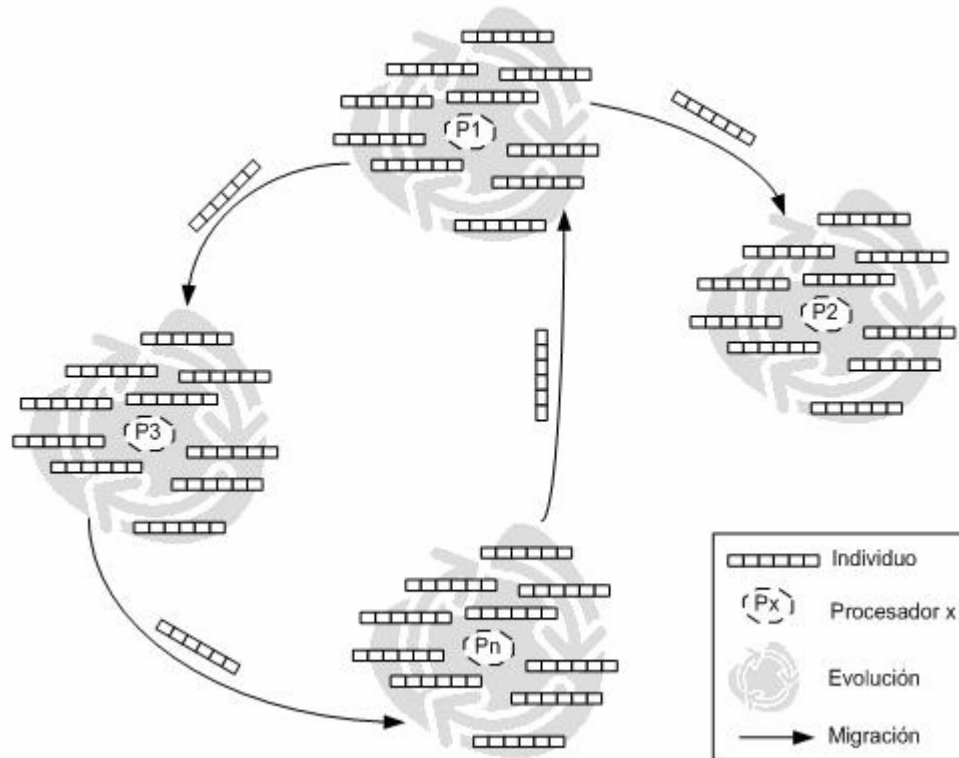


Figura 2.12 – AGP modelo de islas.

2.5.1.3 Paralelismo mediante subpoblaciones estáticas solapadas sin migración

El modelo de paralelismo mediante subpoblaciones estáticas solapadas sin migración es muy similar al anteriormente detallado y su principal diferencia radica en la carencia del operador de migración. En su lugar se utiliza el concepto de solapamiento para compartir material genético entre las subpoblaciones. De esta manera un individuo que se encuentra en una zona solapada (*overlapping area*) pertenece a más de una subpoblación y por lo tanto puede participar en numerosas operaciones de selección y cruzamiento de distintos algoritmos genéticos.

Este esquema es fácilmente aplicable en sistemas de memoria compartida donde la población entera se ubica en el área común y los algoritmos genéticos se ejecutan en cada subpoblación concurrentemente aplicando los mecanismos de sincronización necesarios. En la figura 2.13 se esquematiza el modelo de subpoblaciones estáticas solapadas.

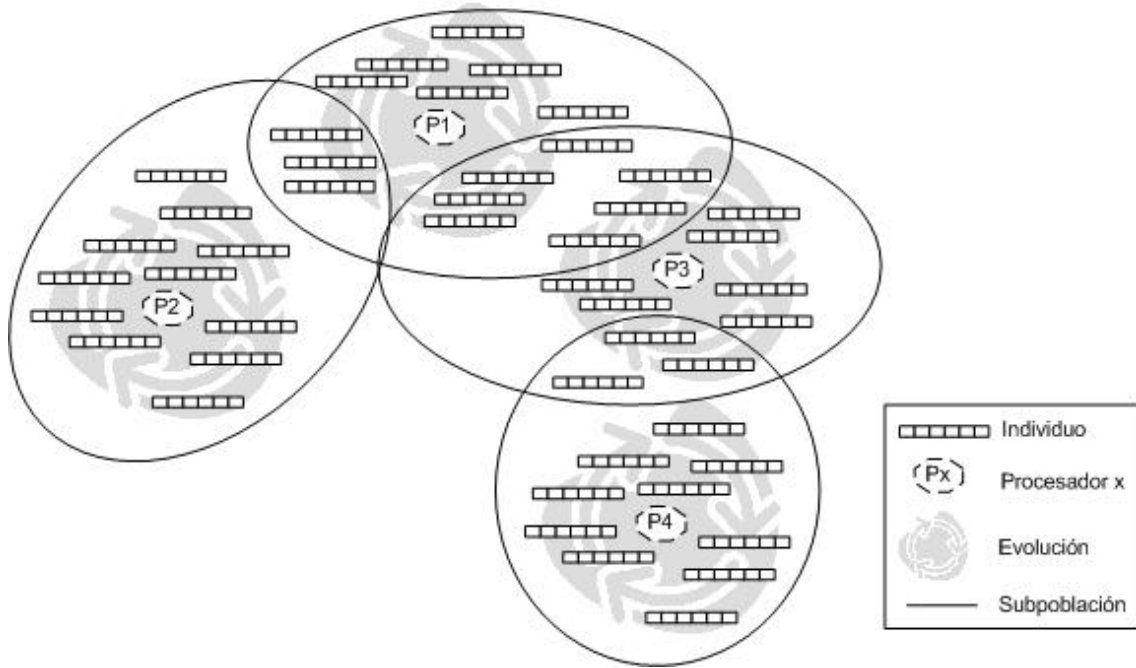


Figura 2.13 – AGP modelo con subpoblaciones estáticas solapadas.

2.5.1.4 Algoritmos genéticos masivamente paralelos

Los algoritmos genéticos masivamente paralelos se caracterizan por asignar, en el caso ideal, un procesador a cada individuo. Es por esto que en general se utiliza en arquitecturas con un gran número de procesadores y alguna topología de interconexión entre ellos (tal como anillo, grilla, hipercubo, etc.) [Michalewicz, 1999].

Durante la ejecución del algoritmo un individuo puede solamente competir y cruzarse con sus vecinos (que estarán definidos según la topología de interconexión).

En la figura 2.14 se esquematiza un AG masivamente paralelo, también conocido como AG celular.

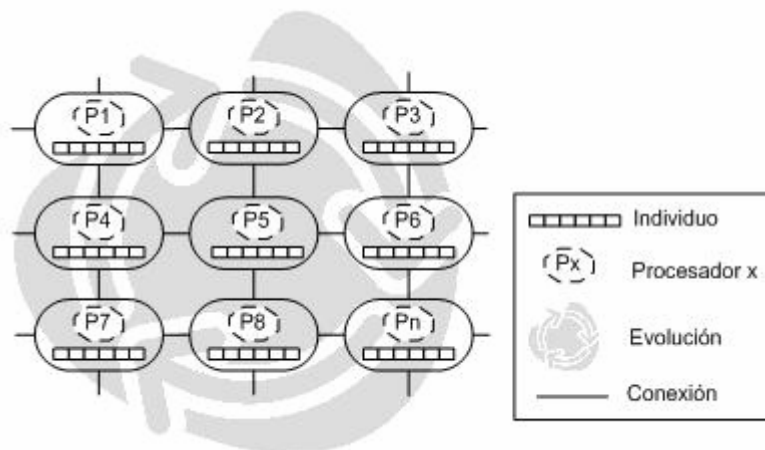


Figura 2.14 – AGP modelo masivamente paralelo

2.5.1.5 Otros modelos de paralelismo

Además de los esquemas clásicos descritos en las secciones anteriores, se pueden encontrar otras variantes como algoritmos híbridos que se logran combinando las características de más de un enfoque.

Nowostawski y Poli [1999] introdujeron el modelo de subpoblaciones dinámicas en el cual la población es tratada como un único conjunto de individuos formado por subpoblaciones que se reorganizan dinámicamente para permitir el intercambio genético. Su principal característica es que permite la combinación entre un AG maestro-esclavo y uno de grano grueso.

Los AG de estado estacionario (*steady state*) también tienen su versión paralela. Ésta consiste básicamente en paralelizar los operadores genéticos debido a que constantemente se actualiza la población. Es importante utilizar mecanismos de sincronización para asegurar la integridad de los individuos.

Otra de las alternativas existentes son los AG adaptativos como es el caso de “*messy ga*” [Goldberg et al. 1988], donde los cromosomas o individuos pueden ser de longitud variable, pudiendo contener muy pocos o muchos *bits* para la representación de los mismos. Así el algoritmo tiene la capacidad de adaptarse a la instancia particular del problema que se está resolviendo.

Muchos de los modelos mencionados anteriormente pueden combinarse para dar lugar a AGP híbridos. Por ejemplo los modelos de islas pueden tener problemas de escalabilidad que pueden solucionarse si se utilizan pocas islas y un AG maestro-esclavo en cada una de ellas.

2.5.2 – Evaluación del desempeño de los algoritmos paralelos

El concepto de paralelismo surge con la necesidad de acelerar los tiempos de cómputo de cualquier proceso. El caso de los algoritmos genéticos no es la excepción a esta afirmación. Para cuantificar la mejora representada por el paralelismo, se utilizan dos conceptos: *speedup* y *eficiencia* [Nesmachnow, 2004].

El *speedup* es la razón entre los tiempos de ejecución de la versión secuencial y paralela del mismo algoritmo. Se realiza la distinción entre *speedup* absoluto y algorítmico. En el primer caso, el cociente se plantea entre el tiempo de ejecución del algoritmo secuencial conocido más veloz y el tiempo del algoritmo paralelo ejecutado sobre una cantidad n de procesadores. El *speedup* algorítmico surge como alternativa cuando no se dispone del algoritmo secuencial más veloz, y la diferencia respecto del *speedup* absoluto radica en que se considera como numerador al tiempo de ejecución de la versión paralela utilizando un único procesador.

La eficiencia computacional es el cociente entre el *speedup* y la cantidad de procesadores utilizados. Un valor de eficiencia mayor a 1 denota sinergia entre los elementos de procesamiento, pues en conjunto logran mejores resultados que por separado.

2.6 – Conclusiones

En este capítulo se han presentado a los algoritmos genéticos como técnicas de búsqueda basadas en el concepto biológico de selección natural. También se analizó cada parte de la estructura básica de cualquier algoritmo genético, mostrando las variantes más conocidas de cada una.

Como propuesta de solución a múltiples problemas de los algoritmos genéticos secuenciales, se detallan varios enfoques de paralelización junto con los criterios que rigen a la taxonomía.

Las múltiples aplicaciones de estas técnicas de computación evolutiva sugieren que es posible aplicarlos con éxito al problema de asignación de frecuencias en redes

inalámbricas, cuyo origen, variantes y otros datos de interés se presentan en el capítulo siguiente.

CAPÍTULO 3

ASIGNACIÓN DE FRECUENCIAS EN REDES INALÁMBRICAS

“La formulación de un problema suele ser más importante que su solución...”⁴

ALBERT EINSTEIN Y LEOPOLD INFELD
The Evolution of Physics, 1938.

3.1 – Introducción

Hace ya más de 100 años que Guglielmo Marconi realizó la primera transmisión inalámbrica desde un barco a tierra. El avance tecnológico ha posibilitado el crecimiento en variedad y densidad de uso de las diferentes aplicaciones, que se han diversificado y van desde el uso del control remoto de la TV y el teléfono celular hasta comunicaciones espaciales interplanetarias, incluyendo radio, televisión, sistemas de navegación y (lamentablemente) aplicaciones militares.

La creciente demanda de estas comunicaciones, sobre todo para telefonía celular, obliga a quienes administran el espectro de radiofrecuencias, que es el recurso finito necesario para llevarlas a cabo, a pensar en esquemas de optimización para lograr un balance entre reutilización y calidad.

Este capítulo comienza presentando conceptos básicos de comunicaciones inalámbricas que servirán de base para introducir los problemas de asignación de frecuencias (*Frequency Assignment Problem* - FAP) a los que se enfrenta el diseñador de una red inalámbrica. Posteriormente se brinda una taxonomía de este tipo de problemas y se profundiza en la variante que busca minimizar la suma total de interferencias (MI-FAP), presentando su formulación matemática. Además en la sección 3.6 se describen los escenarios COST259, que servirán como casos de prueba para el algoritmo de asignación de frecuencias propuesto. Finalmente se proporciona un ejemplo de un escenario pequeño que permitirá lograr un mayor entendimiento de las características de las instancias COST259.

3.2 – Comunicaciones inalámbricas

Una comunicación es un movimiento de materia o energía capaz de transportar información, entre dos partes del universo. Para que exista una comunicación deben estar presentes, como se muestra en la figura 3.1, cuatro elementos básicos: el emisor (extremo que transmite la información), el mensaje (información transmitida), el medio de transmisión (vía por la cual se propaga el mensaje) y el receptor (extremo destino de la información).

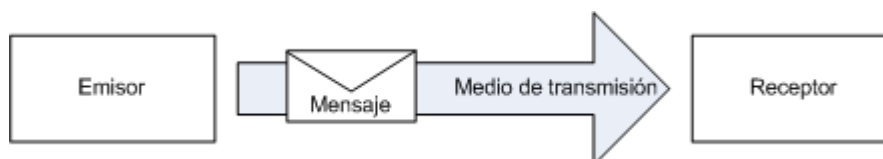


Figura 3.1 – Un emisor envía un mensaje a un receptor a través de un medio de transmisión.

⁴ "The formulation of a problem is often more essential than its solution..."

La gran ventaja de las comunicaciones inalámbricas sobre cualquier otra que utilice un medio guiado es la independencia física que existe entre transmisor y receptor y la consecuente movilidad que esta característica permite a uno o ambos agentes. Esta ventaja radica en que los enlaces inalámbricos prescinden de un medio físico tangible para propagar las ondas electromagnéticas, a diferencia de sus competidores cableados que se valen de algún tipo de material (cobre, fibra óptica) para permitir la propagación. Esta independencia es posible gracias a la propiedad que poseen las ondas electromagnéticas de propagarse en el vacío. Como prueba de este hecho basta con mencionar las comunicaciones satelitales, en las cuales los satélites orbitan la tierra fuera de la atmósfera enviando y recibiendo información permanentemente.

Se puede decir entonces que una comunicación inalámbrica utiliza como medio para la transmisión de información el espacio libre (*free space*) a través de ondas electromagnéticas. Estas últimas pertenecen al espectro de las radiofrecuencias, que abarca el rango comprendido entre 3 KHz y 300 GHz. El uso de ciertos rangos de frecuencias para determinadas aplicaciones es regulado a nivel mundial por la Unión Internacional de Telecomunicaciones (ITU) y a nivel nacional por entes gubernamentales de cada país. En Argentina es la Comisión Nacional de Comunicaciones (CNC) quien tiene a cargo esta tarea. La figura 3.2 muestra la distribución del espectro de frecuencias entre las distintas aplicaciones.

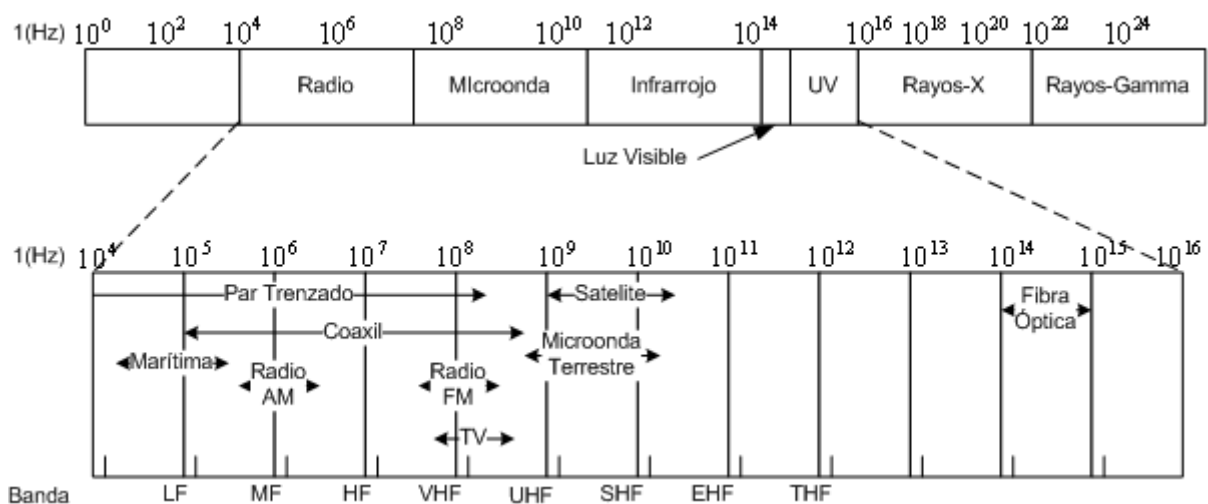


Figura 3.2 – Distribución del espectro de radiofrecuencias entre las diferentes aplicaciones [Tanenbaum, 2003].

El espectro de radiofrecuencias es un recurso limitado, es decir, para establecer una comunicación se requerirá un determinado rango de frecuencia que no debería ser utilizado simultáneamente por otra comunicación en las proximidades geográficas. De no respetarse esta restricción, las comunicaciones se interfieren y, dependiendo de la intensidad con la que lo hagan, corren el riesgo de que los mensajes no sean inteligibles en los extremos receptores.

Lo expresado anteriormente permite inferir que es necesario tomar medidas para maximizar el aprovechamiento del espectro de frecuencias disponibles. Para lograr este objetivo existen dos enfoques absolutamente complementarios: mejorar el hardware de comunicaciones para que la tasa de información transmitida sea la mayor posible (se logra construyendo transmisores y receptores más precisos que minimicen el ruido) y optimizar la asignación de frecuencias para un conjunto de comunicaciones, reutilizando canales en la mayor cantidad de casos posibles [Nolan, 2004].

3.3 – Problemas de asignación de frecuencias

Al diseñar una red inalámbrica se debe tener en cuenta que las frecuencias tienen asociado un costo y por tanto es deseable reutilizarlas, es decir, asignar a distintas comunicaciones simultáneas la misma frecuencia. Además, es necesario considerar que la reutilización puede causar interferencias inaceptables entre las señales. Entonces, el problema de la asignación de frecuencias (más conocido como FAP, siglas de *Frequency Assignment Problem*) consiste lograr un balance entre la reutilización y la pérdida de calidad en la red debido a la interferencia [Koster, 1999].

Los problemas de asignación de frecuencias poseen dos características básicas:

1. Deben asignarse frecuencias a cada una de las comunicaciones inalámbricas de un determinado conjunto de tal manera que la transmisión de datos entre transmisor y receptor sea posible para cada comunicación. Las frecuencias deben seleccionarse de un conjunto dado que puede diferir entre conexiones. (Nótese que existen muchas conexiones que son bidireccionales. En estos casos deben seleccionarse dos frecuencias, una en cada dirección).
2. Las frecuencias asignadas a dos conexiones pueden interferirse, y como consecuencia provocar una pérdida de calidad de la señal. Para que exista interferencia deben cumplirse simultáneamente las siguientes condiciones:
 - a. Las dos frecuencias o sus armónicos deben situarse próximas en el rango electromagnético.
 - b. Las conexiones deben estar cerca geográficamente. Las señales pueden interferirse si tienen un nivel de energía similar en el lugar donde se produce la interferencia.

La interferencia se mide como la relación entre la señal deseada y la señal indeseada, equivalente a una relación señal/ruido. Existe un umbral en esta relación a partir del cual el receptor no puede discriminar la señal útil del ruido. Esta situación se denomina interferencia inaceptable.

Considerando lo expuesto anteriormente, es deseable reutilizar las frecuencias disponibles en la medida de lo posible. Por ejemplo, supongamos que tenemos tres transmisores, una frecuencia disponible y tres receptores. Si un transmisor A no tiene la potencia para alcanzar a ninguno de los receptores de otro transmisor B y viceversa, es absolutamente factible que A y B transmitan a la misma frecuencia (frecuencia 1), sin que se produzcan interferencias. Ahora bien, si un transmisor B tiene la suficiente potencia como para llegar al área de cobertura de otro transmisor C, es muy probable que se produzcan interferencias y por ende ambos transmisores no podrán utilizar la misma frecuencia (frecuencia 1). La situación comentada en este ejemplo se grafica en la figura 3.3. La manera de lograr que los transmisores B y C no se interfieran es asignándoles distintas frecuencias para comunicarse con sus receptores; en este caso debería incorporarse otra frecuencia (frecuencia 2) al conjunto de frecuencias disponibles.

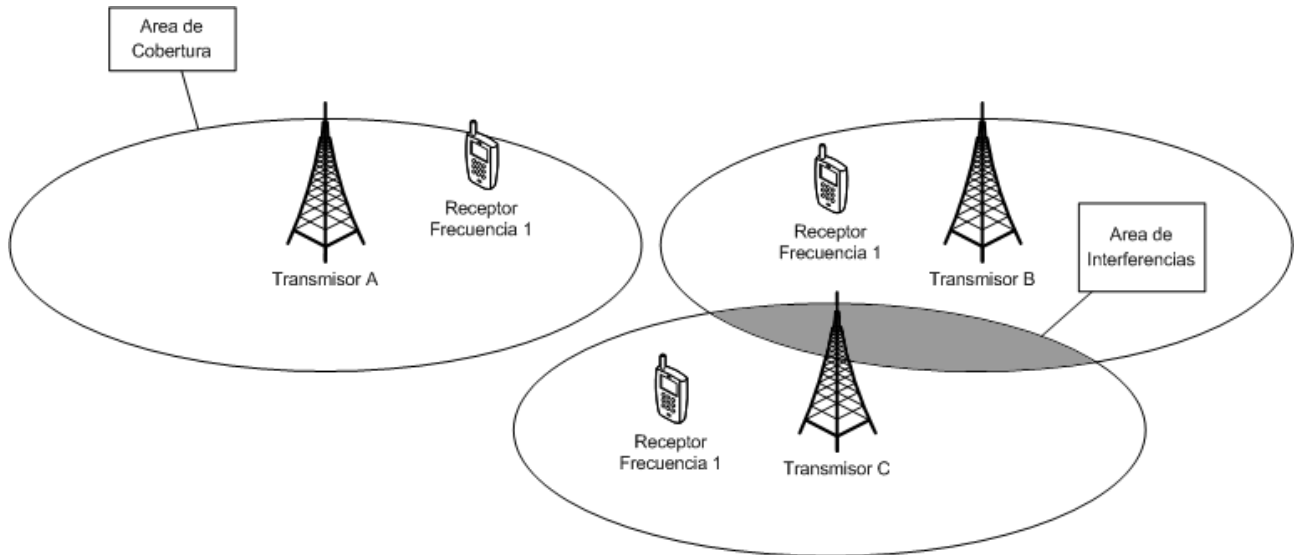


Figura 3.3 – Interferencia entre transmisores.

Para generalizar lo expuesto anteriormente y extrapolarlo al diseño de una red inalámbrica se define el problema en los siguientes términos:

El ancho de banda disponible será $(f_{max} - f_{min})$, donde f_{min} es la frecuencia mínima disponible y f_{max} es la frecuencia máxima disponible. Este ancho de banda se divide en N canales de un ancho Δ , entonces $N = \frac{(f_{max} - f_{min})}{\Delta}$. El conjunto de todos estos canales conforma el dominio del problema $D = \{1..N\}$. Es posible que existan restricciones que determinen que ciertos canales del dominio no están disponibles para una determinada conexión v (esto puede deberse por ejemplo a la proximidad física con otro país), por lo tanto los canales realmente disponibles para v forman un subconjunto de D . Dicho subconjunto se denota como $D_v (D_v \subseteq D)$.

En general son necesarias comunicaciones bidireccionales, es decir dos frecuencias por cada conexión (una para cada sentido de transmisión). El enfoque adoptado en la literatura suele ser tener en cuenta una sola de estas frecuencias para la resolución de problemas, pues una vez hallada una asignación, basta con desplazar cada frecuencia f , sumando s canales a ésta y obtener así su par en sentido contrario $f + s$. Nótese que s debe ser lo suficientemente grande como para evitar interferencias ($s \gg N$).

El nivel de interferencia es medido por la relación señal-ruido en el extremo receptor de la conexión. Calcular efectivamente este nivel es una tarea ardua ya que depende no sólo de la señal elegida y de la potencia de la misma, sino también de las condiciones del ambiente. Sin tener en cuenta las condiciones del ambiente es posible calcular la interferencia mediante la fórmula $\frac{P}{d^\gamma}$ donde P es la potencia del transmisor que está interfiriendo, d es la distancia al receptor interferido y γ ; ($2 < \gamma < 4$) es un factor de desvanecimiento (*fading factor*⁵). Es importante destacar que en general en la literatura se consideran sólo interferencias entre pares y se ignora la posibilidad de que una transmisión pueda ser interferida por muchas señales. Otra simplificación que suele encontrarse en la mayoría de los textos es no tener en cuenta la asimetría de la

⁵ *Fading factor*: es un coeficiente que representa la medida en la cual una señal va disminuyendo su potencia conforme aumenta la distancia de la misma respecto del transmisor.

interferencia, es decir, que dado un par de transmisores interferentes, cada uno tiene un determinado valor de interferencia que no tiene porque ser el mismo. Como ya se mencionó, una de las condiciones para que exista la interferencia es que la potencia de alguno de los transmisores sea lo suficientemente grande como para interceptar el área de cobertura del otro. En la figura 3.4 se ejemplifica esta situación.

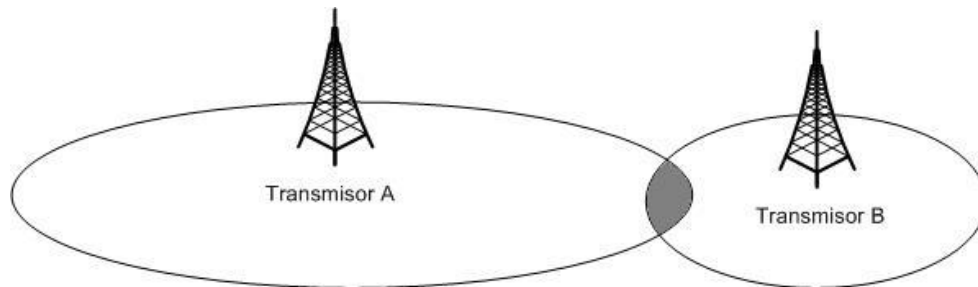


Figura 3.4 – Asimetría de interferencia.

Es importante destacar que, a simple vista, la potencia del transmisor A es mucho mayor que la del transmisor B. En la práctica no se realiza esta distinción, tomándose como valor de referencia el mayor de ambos.

Dependiendo de la aplicación de la que se trate, una conexión v requiere una cantidad $c_v \in \mathbb{Z}^+$ de frecuencias. La interferencia entre dos frecuencias $f, g \in D_v$ puede ser evitada asegurando una distancia mínima necesaria entre ambas. Si c_v varía a lo largo del tiempo, se habla de una asignación de canal dinámica DCA (*dynamic channel assignment*), por el contrario si la cantidad de frecuencias demandadas por una conexión permanece fija, el término utilizado será FCA (*fixed channel assignment*). Como en toda situación donde existen dos enfoques antagónicos, surge naturalmente una asignación de canal híbrida (HCA) en la cual un número de frecuencias es asignado a cada conexión de antemano mientras que la otra parte del espectro se asigna bajo demanda.

Antes de continuar con la siguiente sección, se considera necesario aclarar que en el ámbito de FAP, los términos “canal” y “frecuencia” se creen sinónimos, por lo que en este documento se hablará de ellos indistintamente así como se considerarían equivalentes las expresiones “asignación de canales” y “asignación de frecuencias”.

3.4 – Clasificación de los problemas de asignación de frecuencias

Los problemas de asignación de frecuencias pueden clasificarse de diversas maneras. A continuación se presenta la taxonomía propuesta por Koster [1999]. En particular se profundiza en la variante que procura minimizar las interferencias (MI-FAP), pues es la variante abordada en este trabajo.

- **Minimum Order Frequency Assignment Problem (MO-FAP)**

Consiste en hallar una asignación de frecuencias de tal modo que no se produzcan interferencias inaceptables y que la cantidad de frecuencias diferentes utilizadas sea la menor posible. Este último objetivo data de los comienzos de la telefonía móvil a principios de la década del 70, donde las frecuencias eran muy costosas y se vendían por unidad.

- **Minimum Span Frequency Assignment Problem (MS-FAP)**

Consiste en hallar una asignación de frecuencias de tal modo que no exista interferencia inaceptable y que la diferencia entre las frecuencias máxima y mínima utilizadas (*span*) sea minimizada. Este problema guarda una relación más estrecha

con la situación actual, donde el espectro radioeléctrico se licencia por ancho de banda y no por frecuencias individuales.

- **Minimum Blocking Frequency Assignment Problem (MB-FAP)**

En el caso que no puedan asignarse a cada nodo todas las frecuencias que requiere sin incurrir en interferencias inaceptables, puede optarse por realizar una asignación parcial (sin proporcionarle a todos los nodos la totalidad de las frecuencias que demandan). En definitiva esta variante consiste en asignar frecuencias de tal modo que no se produzcan interferencias inaceptables y la probabilidad de bloqueo global sea minimizada.

- **Minimum Interference Frequency Assignment Problem (MI-FAP)**

Consiste en asignar frecuencias de un conjunto finito de frecuencias disponibles de tal manera de no incurrir en interferencias inaceptables y que la suma total de interferencias sea minimizada. Como esta variante es la elegida para su resolución en este trabajo será ampliada en la sección siguiente.

3.5– Problema de mínima interferencia en asignación de frecuencias (MI-FAP)

Cualquier problema FAP se representa mediante un grafo $G = (V, E)$ conocido como grafo de interferencia o grafo de restricciones donde:

- V es el conjunto de vértices del grafo, que se corresponden con los nodos del escenario inalámbrico.
- E es el conjunto de aristas del grafo. Para dos nodos $v, w \in V$, $(v, w) \in E$ si y sólo si las señales de los nodos v y w pueden interferirse en al menos un par de frecuencias.
- $\forall v \in V; D_v \subseteq D$ es el conjunto de frecuencias disponibles para el nodo v .
- c_v es el número de frecuencias requeridas por $v \in V$.
- p_{vwfg} representa la penalización asociada a elegir las frecuencias $f \in D_v$ y $g \in D_w$. Depende de la interferencia entre las mismas.
- p_{vfg} denota la penalización asociada a la interferencia entre las frecuencias f y g asignadas al mismo nodo v .

En el caso particular de MI-FAP, la descripción formal se presenta en el cuadro 3.1.

Dados los siguientes elementos:

- Un grafo no dirigido $G = (V, E)$; $\{v, v\} \in E; \forall v \in V$.
- Los conjuntos $T_{vw} \subset Z, \{v, w\} \in E, 0 \in T_{vw}$, de las distancias relevantes entre las frecuencias de $f(v)$ y $g(w)$.
- Un conjunto de demandas $c_v \in Z^+$ para todo $v \in E$.
- Un conjunto de frecuencias válidas disponibles $D_v \subseteq Z^+, \forall v \in V$. $D_v = \{f/f \text{ es una frecuencia disponible para } v\}$.
- El conjunto $D = \bigcup_{v \in V} D_v$ de todas las frecuencias disponibles para algún nodo del escenario inalámbrico.
- Un conjunto de valores de penalidad $p_{vwfg} \in Z^+$ para todos los $\{v, w\} \in E, f \in D_v, g \in D_w$.
- Un entero positivo K que especifica el valor máximo aceptable de la suma de todas las penalizaciones del escenario inalámbrico.

Pregunta: ¿Existe una asignación de subconjuntos $f: V \rightarrow 2^D$ tal que satisfaga las siguientes condiciones?

1. $|f(v)| = c_v$
2. $f(v) \subseteq D_v$
3.
$$\left(\sum_{\{v,w\} \in E} \sum_{\substack{\bar{f} \in f(v), \bar{g} \in g(w) \\ (v \neq w) \vee (\bar{f} \neq \bar{g})}} p_{vw\bar{f}\bar{g}} \delta(|\bar{f} - \bar{g}| \in T_{vw}) \right) \leq K$$

Donde la función $\delta(A)$ es igual a 1 si la condición A es verdadera y 0 en caso contrario.

Cuadro 3.1 – Descripción formal de MI-FAP

3.6 – Escenarios COST

COST (*European COoperation in the field of Scientific and Technical Research*) es una reconocida red de investigación en Europa, basada en un marco de trabajo intergubernamental que involucra 34 estados miembros e Israel como estado cooperador. COST abarca 13 dominios científicos entre los cuales se encuentra el de interés para este proyecto: telecomunicaciones, ciencia de la información y tecnología (TIST).

Dentro del grupo TIST se desarrollan diferentes proyectos entre los cuales se encuentra COST 259⁶, el cual es descrito someramente a continuación.

COST 259 [COST259, 1996] está compuesto por varios grupos: “*Radio System Aspects*”, “*Propagation & Antenas*” y “*Network Aspects*”. Dentro de este último se encuentra el subgrupo “*Standard Scenarios for Frequency Planning*” que tiene como objetivo brindar un conjunto de escenarios de planificación de frecuencias (reales y artificiales) que sirvan como base estándar para ejecutar algoritmos de planeamiento de asignación de frecuencias y determinar la calidad de sus resultados, permitiendo realizar dos tipos de comparaciones: el comportamiento de un algoritmo sobre diferentes escenarios y el de diferentes algoritmos en un mismo escenario.

⁶ El proyecto COST259 se encuentra concluido y su reporte final se encuentra publicado con el título “*Wireless Flexible Personalised Communications*” [Wiley, 2001].

Cada escenario consta de varios archivos disponibles en Internet, con un formato estándar descrito en el trabajo de Eisenblätter [1998]. Este formato común facilita el intercambio de escenarios entre investigadores.

Un escenario es una descripción de la estructura subyacente de una red celular telefónica, que detalla los siguientes componentes intervinientes:

- Sitio: Conjunto de sectores de una red.
- Sector: Da servicio a una celda y tiene instalados varios transmisores.
- Celda: Región física cubierta por un sector.
- Transmisor: Elemento de comunicación que utiliza una determinada frecuencia.
- Canal o *Carrier*: Es equivalente a una frecuencia.
- Restricciones: Cada celda puede tener restricciones que implican que haya canales localmente bloqueados, es decir, que dicha celda no tiene la posibilidad de utilizar estos canales para ninguna comunicación. Estas restricciones pueden catalogarse en técnicas y regulatorias. Las primeras se deben a aspectos físicos de las comunicaciones inalámbricas mencionados anteriormente en este capítulo, mientras que las segundas se corresponden con cuestiones gubernamentales originadas por la proximidad física entre países⁷.
- *Handover*: Cuando un teléfono móvil que esta manteniendo una comunicación, se desplaza de una celda a otra es necesario cambiar de la frecuencia que se esta utilizando a una frecuencia perteneciente a la celda que se comenzará a utilizar. Este proceso, también denominado *handoff* o transferencia de celda, es muy complejo y delicado pues tiene que ser transparente para quien este manteniendo dicha comunicación.

Para definir completamente un escenario deben especificarse un conjunto de parámetros que se dividen en tres grupos: que se presentan en las subsecciones siguientes.

3.6.1 - Parámetros de información general

- Identificador de escenario (*scenario identifier*): nombre del escenario que lo distingue unívocamente.
- Comentarios del escenario (*scenario annotation*): información extra del escenario (opcional).
- Tipo de red (*network type*): los tipos posibles son GSM900 y GSM1800.
- Espectro (*spectrum*): Intervalo de frecuencias disponibles. Si hubiera frecuencias no disponibles dentro de ese intervalo, se listan como “canales globalmente bloqueados”.
- Canales globalmente bloqueados (*globally blocked channels*): Un canal está globalmente bloqueado si se encuentra dentro del espectro pero no puede utilizarse en la asignación de frecuencias por alguna razón (opcional).

⁷ Recordemos que COST es una institución concebida en Europa, continente en el cual la poca superficie de los países aumenta considerablemente la posibilidad de interferencias internacionales.

- Separación co-sitio (*co-site separation*): Mínima separación necesaria entre canales pertenecientes al mismo sitio. Se asume que esta separación es la misma para todos los sitios.
- Separación co-celda por defecto (*default co-cell separation*): Mínima separación necesaria entre canales usados en la misma celda. Debido a las diferentes tecnologías de transmisión, la separación co-celda puede variar en diferentes celdas. La separación aquí definida es la que se adopta en el caso de no especificarse lo contrario.
- Separación de handover (*handover separation*): Especifica la separación necesaria entre canales intervinientes en el *handover*.
- Interferencia mínima significativa (*minimal significant interference*): Todas las relaciones de interferencia conocidas entre celdas deberían ser especificadas en la descripción del escenario para lograr el mayor entendimiento posible de la estructura del mismo. Este parámetro será usado como límite para descartar las interferencias muy pequeñas (opcional).
- Interferencia máxima tolerable (*maximal tolerable interference*): Este parámetro es utilizado como cota superior para la máxima interferencia admisible. Las tasas de interferencia que excedan este límite son descartadas introduciendo un requerimiento de separación con un valor suficientemente alto (opcional).
- Modelo de demanda (*demand model*): Un modelo de demanda absoluta especifica la cantidad de canales necesarios por cada celda. Dado que en general esta información suele ser confidencial, surge el modelo de tráfico normalizado, en el cual para cada celda se especifica el tráfico esperado, que se utiliza para derivar el número de canales por celda.
- Ubicaciones de sitios (*site locations*): Para una representación gráfica del escenario, las coordenadas cartesianas son extremadamente útiles. Los valores de estos parámetros indican dónde están ubicados los sitios. Si esta información fuera confidencial, pueden utilizarse coordenadas aproximadas y aún así serán de gran utilidad para el diseño del algoritmo de asignación de frecuencias y para el análisis de estructura de escenarios.

3.6.2 - Parámetros referentes a celdas

- Nombre del sitio (*site name*): Este parámetro indica el nombre del sitio al que pertenece la celda.
- Sector (*sector*): Indica el número de sector que cubre a la celda.
- Demanda (*demand*): Depende del modelo de demanda usado. Puede estar determinado por el número de canales usados en la celda ó el tráfico de celda normalizado.
- Ubicación (*location*): Par (x,y) que determina las coordenadas cartesianas del sitio (opcional).
- Canales localmente bloqueados (*locally blocked channels*): Listado de canales que la celda no puede utilizar (opcional).
- Separación co-celda (*co-cell separation*): En este parámetro se especifica la separación entre los canales de una determinada celda en el caso de diferir de la separación co-celda por defecto (opcional).

3.6.3 - Parámetros referentes a relaciones entre dos celdas

Estos parámetros se especifican para un par ordenado de celdas (x, y) . Es importante destacar que las relaciones para (x, y) difieren de las de (y, x) .

- Handover: El valor de este parámetro especifica si es posible el *handover* desde la primera celda a la segunda (opcional).
- Separación (*separation*): Especifica la separación mínima necesaria entre los canales asignados a los pares de celdas involucrados. Si este parámetro es provisto no se tienen en cuenta las posibles mayores mínimas separaciones debido a las relaciones de *handover* u otras (opcional).
- Interferencia (*interference*): La interferencia de una celda a otra puede ser especificada en términos del área o tráfico afectado. La tasa de interferencia es un valor entre 0 y 1 y la interferencia se compone de dos valores que varían entre 0 y 1. Uno de los valores de interferencia es llamado interferencia co-canal (*co-channel interference*), que es la que puede ocurrir si dos transmisores utilizan el mismo canal, el otro es conocido como interferencia de canal adyacente (*adjacent-channel interference*) que es la que puede producirse cuando los canales asignados a dos transmisores difieren en una unidad (opcional).

3.7 - El problema de mínima interferencia en el marco de los escenarios COST259

De un escenario COST 259 pueden extraerse todos los datos necesarios para resolver el problema de mínima interferencia expresado a continuación:

Sean:

- s la cantidad de sitios del escenario.
- c_m la cantidad de celdas del sitio m ($m = 1..s$).
- c la cantidad de celdas del escenario ($c = \sum_{m=1}^s c_m$).
- t_n la cantidad de transmisores de la celda n ($n = 1..c$).
- t la cantidad de transmisores del escenario ($t = \sum_{n=1}^c t_n$).
- D el conjunto de frecuencias disponibles.
- d la cantidad de frecuencias disponibles ($d = |D|$).
- una matriz de separación $S(t \times t)$ tal que S_{ij} representa la separación mínima necesaria entre las frecuencias asignadas a los transmisores i y j .
- una matriz binaria $B(d \times c)$ de canales bloqueados tal que $B_{ij} = 1$ si el canal i está bloqueado para la celda j y 0 en cualquier otro caso.
- Una matriz de interferencias $I(c \times c)$ tal que I_{ij} representa el nivel de interferencia entre un transmisor de la celda i y uno de la celda j ($0 \leq I_{ij} \leq 1$).

Entonces, el problema puede formularse literalmente como: “asignar a cada transmisor una frecuencia f ($f \in D$) de tal manera de satisfacer las restricciones expresadas por las matrices S y B , minimizando la interferencia total calculada en base a I .”

3.8 - Un ejemplo del problema

Para poder diseñar una solución al problema de asignación de frecuencias es necesario analizar la información brindada para un escenario. Mediante un ejemplo se intentará mostrar cual es la información que puede extraerse de un escenario. En particular se utilizará para esto un escenario COST 259 llamado *Tiny*, el cual como su nombre lo indica, posee un tamaño ínfimo. Éste y otros escenarios están disponibles en la página oficial de FAP en Internet [FAP Web, 2006].

En este trabajo se construyó un programa Java que toma como entrada la descripción de un escenario y genera un conjunto de archivos de texto que contienen la información necesaria para poder realizar la evaluación de una solución en un formato predefinido. En las siguientes secciones se describe la información contenida en éstos archivos y su utilidad.

3.8.1 – Datos generales

El escenario *Tiny* está compuesto por tres sitios llamados A, B y C cada uno con 3, 2 y 2 sectores (y por ende celdas) respectivamente, como se muestra en la figura 3.5.

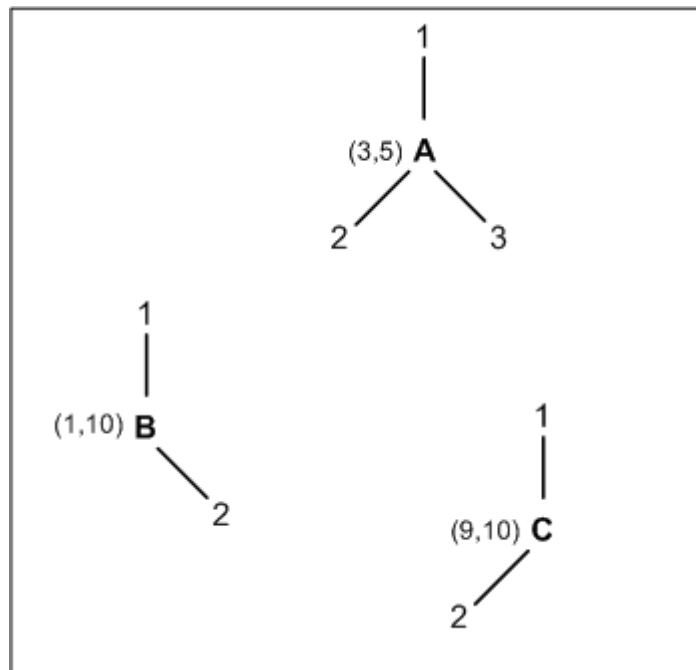


Figura 3.5 – Descripción gráfica del escenario *Tiny* [Eisenblätter, 1998].

Los valores encerrados entre paréntesis en la figura 3.5 se corresponden con las coordenadas cartesianas de las celdas, datos que para este trabajo en particular no son relevantes.

Cada celda tiene instalado un determinado número de transmisores que son presentados en la tabla 3.1. En este caso la celda A1 tiene instalado un transmisor, la celda A2 tiene instalados tres transmisores y así sucesivamente.

Celda	A1	A2	A3	B1	B2	C1	C2
Cant. Transmisores	1	3	2	2	1	1	2

Tabla 3.1 – Cantidad de transmisores por celda para el escenario Tiny.

El espectro de frecuencias proporcionado para el escenario determina el conjunto de canales disponibles. A cada uno de los transmisores debe asignársele una frecuencia de este conjunto. Un ejemplo de una asignación de frecuencias se muestra en la tabla 3.2, donde por ejemplo, el transmisor 0 de la celda A1 tiene asignada la frecuencia 16.

Celda	A1	A2	A3	B1	B2	C1	C2	
Transmisor	0	0	1	2	0	1	0	1
Canal	16	5	9	13	15	7	11	16

Tabla 3.2 – Asignación de frecuencias para el escenario Tiny.

El lector perspicaz habrá notado que la tabla anterior representa una de las soluciones al problema. La calidad de la misma quedará determinada en base al grado con el que puede lograr el objetivo de minimizar la interferencia, satisfaciendo las restricciones inherentes al problema.

3.8.2 – Restricciones

El problema que se intenta resolver consiste en hallar una asignación de frecuencias de manera tal que se minimice la suma total de interferencias y se asegure el servicio. Esto no es una tarea sencilla y, como ya se explicó en secciones anteriores, existen condicionantes físicos y políticos que implican restricciones del problema. Una buena solución será aquella que satisfaga de mejor manera las restricciones del escenario, presentadas a continuación:

- **Separación**

Una separación es la distancia física entre canales de transmisión que se obtiene de calcular la diferencia entre el mayor y el menor. Por ejemplo, los canales 7 y 5 tienen entre ellos una separación de 2. Mientras más próximas sean las frecuencias o canales, mayor es el riesgo de que se produzcan interferencias. Es por esto que el escenario impone determinadas separaciones mediante sus parámetros “separación co-sitio”, “separación co-celda” y “separación de *handover*”, explicados en secciones anteriores.

Si se consideran estos tres tipos de separaciones, puede obtenerse la separación mínima necesaria entre cada uno de los pares de transmisores del escenario, dato que puede ser de mucha utilidad a la hora de decidir una asignación de frecuencias para no incurrir en interferencias inaceptables.

La propuesta presentada en este trabajo para representar la separación requerida consiste en una matriz S , cuadrada, simétrica donde las filas y columnas representan canales de cada una de las celdas del escenario. Así S_{ij} representa la separación mínima necesaria entre las frecuencias asignadas a los transmisores i y j .

En la tabla 3.3 se representan los requerimientos de separación del escenario *Tiny*, en donde por ejemplo la separación necesaria entre el transmisor 2 de la celda A2 y el transmisor 1 de la celda B1 es 1.

		A1		A2		A3		B1		B2	C1		C2	
		0	0	1	2	0	1	0	1	0	0	0	0	1
A1	0	0	2	2	2	2	2	0	0	0	0	0	0	0
A2	0	2	0	3	3	2	2	2	1	0	0	0	0	0
	1	2	3	0	3	2	2	2	1	0	0	0	0	0
	2	2	3	3	0	2	2	2	1	0	0	0	0	0
A3	0	2	2	2	2	0	3	0	0	0	2	2	2	2
	1	2	2	2	2	3	0	0	0	0	2	2	2	1
B1	0	0	2	2	2	0	0	0	3	2	0	2	2	1
	1	0	1	1	1	0	0	3	0	2	0	2	2	1
B2	0	0	0	0	0	0	0	2	2	0	0	2	2	1
C1	0	0	0	0	0	2	2	0	0	0	0	2	2	2
C2	0	0	0	0	0	2	2	2	2	2	2	0	3	3
	1	0	0	0	0	2	1	1	1	1	2	3	0	0

Tabla 3.3 – Matriz de separaciones S para el escenario Tiny.

Los datos de la tabla de separaciones permiten evaluar si una asignación de frecuencias para un par de transmisores del escenario cumple o no con los requisitos de separación.

• **Canales bloqueados**

Existen en el escenario canales bloqueados local o globalmente. Un canal bloqueado implica que no puede ser utilizado por ningún transmisor del escenario o de la celda, dependiendo de si se trata de un bloqueo global o local respectivamente.

En el presente trabajo se propone representar en una matriz de canales bloqueados (Tabla 3.4) donde las filas representan los canales del espectro y las columnas cada uno de los transmisores de las celdas.

		A1		A2		A3		B1		B2	C1		C2	
		0	0	1	2	0	1	0	1	0	0	0	0	1
5	0	0	0	0	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	1	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 3.4 – Matriz de canales bloqueados para el escenario Tiny.

De existir un canal globalmente bloqueado, toda la fila del canal estará en 1 (no es el caso del escenario *Tiny*). La matriz de canales bloqueados permite saber si un canal puede ser utilizado o no en una celda determinada.

- **Interferencia**

Otro aspecto fundamental a tener en cuenta para resolver este problema es la interferencia entre celdas, ya que como se ha mencionado antes, el objetivo es minimizar la suma total de las mismas. Hay dos tipos de interferencia que son relevantes al problema: la interferencia co-canal y la interferencia de canal adyacente. La matriz con los valores de interferencia co-canal se presenta en la tabla 3.5.

		A1	A2			A3		B1		B2	C1	C2	
		0	0	1	2	0	1	0	1	0	0	0	1
A1	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	0	0	0	0	0	0	0.3	0.3	0.1	0	0	0
	1	0	0	0	0	0	0	0.3	0.3	0.1	0	0	0
	2	0	0	0	0	0	0	0.3	0.3	0.1	0	0	0
A3	0	0	0	0	0	0	0	0	0	0	0.05	0.2	0.2
	1	0	0	0	0	0	0	0	0	0	0.05	0.2	0.2
B1	0	0.01	0.25	0.25	0.25	0	0	0	0	0	0	0.25	0.25
	1	0.01	0.25	0.25	0.25	0	0	0	0	0	0	0.25	0.25
B2	0	0	0	0	0	0	0	0	0	0	0	0.15	0.15
C1	0	0	0	0	0	0.01	0.01	0	0	0	0	0	0
C2	0	0	0.06	0.06	0.06	0.12	0.12	0	0	0.25	0	0	0
	1	0	0.06	0.06	0.06	0.12	0.12	0	0	0.25	0	0	0

Tabla 3.5 – Interferencia co-canal para el escenario *Tiny*.

La matriz con los valores de interferencia de canal adyacente se presenta en la tabla 3.6.

		A1	A2			A3		B1		B2	C1	C2	
		0	0	1	2	0	1	0	1	0	0	0	1
A1	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	0	0	0	0	0	0	0.1	0.1	0.02	0	0	0
	1	0	0	0	0	0	0	0.1	0.1	0.02	0	0	0
	2	0	0	0	0	0	0	0.1	0.1	0.02	0	0	0
A3	0	0	0	0	0	0	0	0	0	0	0	0.06	0.06
	1	0	0	0	0	0	0	0	0	0	0	0.06	0.06
B1	0	0	0.09	0.09	0.09	0	0	0	0	0	0	0.08	0.08
	1	0	0.09	0.09	0.09	0	0	0	0	0	0	0.08	0.08
B2	0	0	0	0	0	0	0	0	0	0	0	0.04	0.04
C1	0	0	0	0	0	0	0	0	0	0	0	0	0
C2	0	0	0.01	0.01	0.01	0.03	0.03	0	0	0.08	0	0	0
	1	0	0.01	0.01	0.01	0.03	0.03	0	0	0.08	0	0	0

Tabla 3.6 – Interferencia canal adyacente para el escenario *Tiny*.

La información presentada en las matrices de interferencia es de vital importancia a la hora de evaluar la calidad de una solución. Además cabe destacar que el problema tratado en este trabajo es el de lograr la mínima interferencia en una asignación de frecuencias.

3.8.3 – Evaluación de una solución

Una vez obtenida toda la información de un escenario es posible determinar la calidad de una solución propuesta. Como se mencionó anteriormente, una posible solución a este problema consiste en un conjunto de frecuencias asignadas cada una a un transmisor.

La evaluación de la solución podría realizarse recorriendo secuencialmente la solución y comparando cada transmisor con los datos obtenidos del escenario, entre los cuales pueden encontrarse canales bloqueados que determinan una penalidad, niveles de interferencia que se van sumando a la totalidad de las interferencias y las restricciones de separación que determinan otra penalidad. Mientras menor sea el número de penalidades y menores los valores de interferencia total, mejor será la calidad de la solución propuesta.

3.9 – Conclusiones

En este capítulo se analizaron las características del problema FAP, y se describió la variante MI-FAP que constituye el tópico principal de este trabajo. Posteriormente se presentaron los escenarios COST259 que son fuentes de información para poder realizar planeamientos y asignaciones de frecuencia que se presentan en la realidad y finalmente se presentó un ejemplo de cual es la información útil que puede ser extraída de un escenario.

En este último apartado se menciona como puede realizarse una evaluación de una solución para determinar su calidad. Si se intentara determinar cual es la solución óptima al problema podrían presentarse muchas alternativas para un escenario determinado, las que deberán ser evaluadas y comparadas para encontrar la mejor. Realizar esta tarea de forma exhaustiva es impracticable en términos de tiempo de ejecución del algoritmo cuando se trata con escenarios realistas con cientos o miles de sitios. Por este motivo, la aplicación de técnicas heurísticas o metaheurísticas surge como una línea interesante para la obtención en tiempos razonables de soluciones de buena calidad para el problema de asignación de frecuencias.

En el capítulo siguiente se aborda el problema MI-FAP utilizando algoritmos genéticos en versiones secuenciales y paralelas.

CAPÍTULO 4

UN ALGORITMO GENÉTICO PARA EL PROBLEMA DE MÍNIMA INTERFERENCIA EN LA ASIGNACIÓN DE FRECUENCIAS EN REDES DE TELEFONÍA CELULAR

“La ciencia no es poderosa a pesar de la abstracción, sino justamente por ella.”

ERNESTO SABATO
Uno y el Universo, 1945

4.1 – Introducción

Como se mencionó en el capítulo anterior, en este trabajo se aborda una de las variantes del problema de asignación de frecuencias en redes inalámbricas, llamada MI-FAP (*Minimum Interference Frequency Assignment Problem*). Particularmente se intentará encontrar una asignación de frecuencias que minimice la suma total de interferencias en una red de telefonía móvil. Debido a la complejidad inherente al problema se buscarán soluciones aplicando algoritmos genéticos, una de las metaheurísticas más difundidas para la resolución de problemas de optimización combinatoria. Para afrontar los altos costos computacionales demandados por las características del problema se propone utilizar variantes paralelas de algoritmos genéticos.

En este capítulo inicia con aspectos de implementación del algoritmo, mencionando los componentes de *hardware* y *software* utilizados. Posteriormente se presentarán la codificación y la función de *fitness* utilizada. Como consecuencia de la experimentación se describe una variante elitista del algoritmo genético, que será la utilizada para las ejecuciones finales. A continuación se presenta un análisis paramétrico para determinar cual es la mejor combinación de parámetros del algoritmo. Luego, como una de las componentes más importantes del capítulo, se presentan los resultados de las ejecuciones finales del algoritmo para varios escenarios. Finalmente se proporcionan las conclusiones correspondientes al capítulo.

4.2 – Biblioteca MALLBA

Codificar la resolución de cualquier problema de optimización combinatoria implica implementar tanto el algoritmo descrito por el método como también los objetos dependientes del problema. El esfuerzo asociado a la primera de las dos tareas mencionadas no es trivial y puede reducirse considerablemente mediante la utilización de algún *framework* o biblioteca que implemente cada uno de los componentes básicos del método elegido. Además, la biblioteca debe ser lo suficientemente parametrizable como para brindar al usuario la flexibilidad requerida.

La abstracción que permite lograr este tipo de alternativas tiene como contrapartida que quién las utiliza no logra adquirir un entendimiento detallado de la implementación computacional del método elegido, y por ende podría verse limitado a la hora de extenderlo más allá de los límites planteados por sus desarrolladores originales.

La biblioteca elegida por los autores de esta tesis es denominada MALLBA [Alba et al., 2001], sigla correspondiente a las universidades españolas de Málaga, La Laguna y Barcelona.

MALLBA ofrece tres tipos de métodos genéricos de resolución: exactos, heurísticos e híbridos. Además para cada uno de ellos provee tres diferentes implementaciones: secuencial, paralela para redes de área local (*LAN*) y paralela para redes de área amplia (*WAN*).

Las estructuras provistas por MALLBA están basadas en la separación de dos conceptos: el problema concreto a resolver y el método de resolución general que se va a utilizar (en nuestro caso FAP y AG respectivamente). Estas estructuras pueden ser vistas como plantillas (*templates*) a parametrizar y se denominan *skeletons*.

Como beneficio adicional, MALLBA permite que sus usuarios se despreocupen de cuestiones relacionadas con el paralelismo, puesto que implementa de manera automática un modelo de anillo unidimensional de islas con migración cuyos parámetros pueden ajustarse mediante un archivo de configuración.

Cada solución consistirá en un conjunto de clases C++ denominadas “provistas” y “requeridas”. Las primeras implementan aspectos internos del esqueleto elegido que son independientes del problema, mientras que las segundas especifican información y comportamiento específicos del problema.

Específicamente, cada estructura requiere las clases *Problem* y *Solution*. La primera es una abstracción de las características relevantes del problema para el método de resolución escogido, mientras que la segunda representa una abstracción de una solución factible a dicho problema.

Por otra parte, se proveen las clases *Setup* y *Solver*. La primera contiene los parámetros de inicialización necesarios para llevar a cabo la ejecución (cantidad de iteraciones, número de ejecuciones independientes, parámetros que guían la búsqueda, etc.). La segunda provee métodos para ejecutar el esquema de resolución y consultar su progreso o cambio de estado. En la figura 4.1, basada en *Mallba Library* [2004], se muestra la arquitectura de MALLBA.

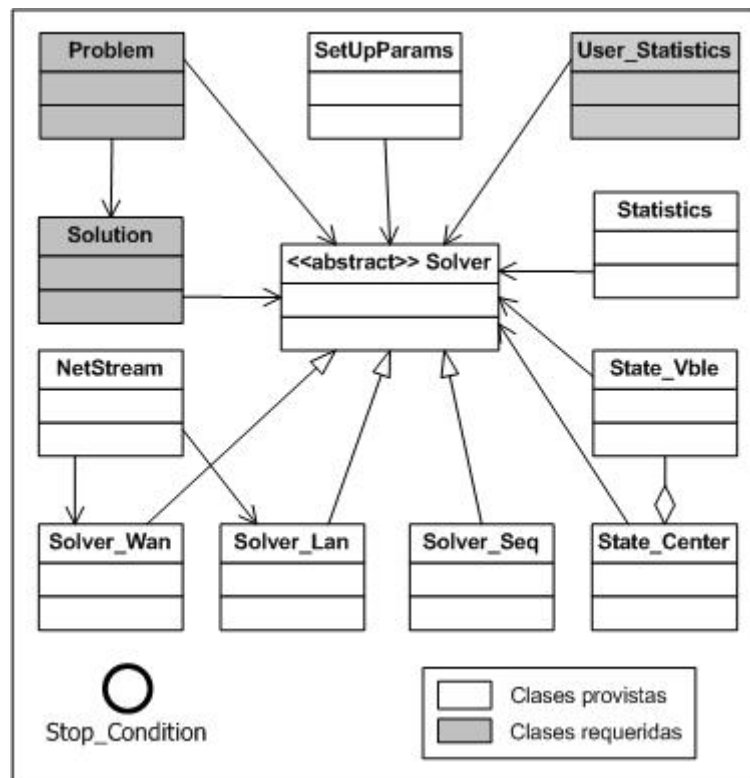


Figura 4.1 – Estructura de clases de MALLBA

En el caso particular de los algoritmos genéticos, además de las clases *Problem* y *Solution*, se requieren las clases *Crossover* y *Mutation* que representan los operadores evolutivos propios de esta metaheurística.

4.3 – MPI

Como se mencionó anteriormente, MALLBA proporciona un *framework* en el cual se pueden desarrollar aplicaciones paralelas sin preocuparse por los mecanismos de comunicación de bajo nivel. Para esto MALLBA se vale de MPI (*Message Passing Interface*) que es una interfaz estandarizada para el desarrollo de aplicaciones paralelas basadas en pasaje de mensajes. El estándar no define una implementación particular de la interfaz, brindando al programador la posibilidad de utilizar un conjunto de funciones para diseñar su aplicación sin que éste deba conocer el *hardware* sobre el cual se va a ejecutar ni la forma en que se han implementado las funciones que utiliza.

MPI se basa en una arquitectura SPMD (*Single Program, Multiple Data*), en el que todos los procesos ejecutan el mismo programa aunque no necesariamente la misma instrucción al mismo tiempo. Estas copias del programa no avanzan sincronizadas instrucción a instrucción sino que la sincronización, de ser necesaria, tiene que ser explícita. El intercambio de información, así como la sincronización, se hacen mediante paso de mensajes [Alonso, 1997].

Las funciones de MPI soportan la comunicación proceso-a-proceso, comunicación por grupos, la administración de estos grupos y su interacción con el ambiente.

Esta interfaz ha sido desarrollada por el MPI *Forum*, que es un grupo de investigadores de universidades, laboratorios y empresas involucrados en la computación de altas prestaciones. Sus objetivos fundamentales son:

- Definir un entorno de programación único que garantice la portabilidad de aplicaciones paralelas.
- Definir totalmente el interfaz de programación, sin especificar como debe ser implementado.
- Ofrecer implementaciones de calidad, de dominio público para favorecer la extensión del estándar.
- Convencer a los fabricantes de computadoras para que ofrezcan versiones MPI optimizadas para sus máquinas.

Actualmente existen varias implementaciones del estándar MPI entre las cuales se pueden citar LAM, MPICH, CHIMP, etc..

4.4 - Arquitectura, implementación y plataforma de ejecución

La arquitectura de la solución tiene como base al sistema operativo Linux. Para la comunicación entre procesos se utilizará la versión 1.2.6 de MPICH [MPI, 2005], una implementación gratuita de la interfaz MPI. Para el algoritmo genético se utilizará la biblioteca MALLBA, proveyendo las clases C++ requeridas que definen características específicas del problema. En la figura 4.2 se esquematiza la arquitectura en cuestión.

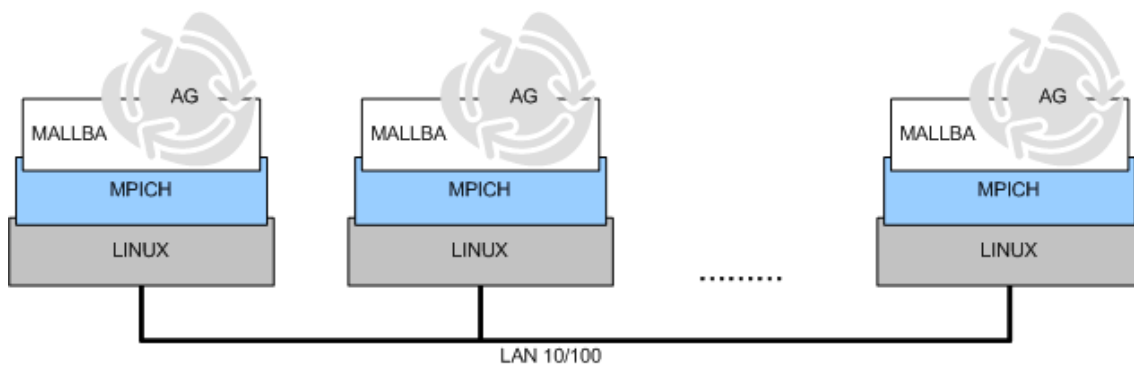


Figura 4.2 – Arquitectura de la solución propuesta.

Nótese que en la figura 4.2 se ha presentado a una parte de los AG sobre MALLBA y a otra parte por fuera, para indicar que algunos de los componentes de un AG son provistos por MALLBA y otros deben ser provistos por el programador. Para las ejecuciones experimentales se utilizó un cluster de tres equipos Athlon AMD 3000+ de 2 GHz y 64 bits, 1 GB RAM, interconectados a través de una LAN Ethernet a 100 Mbps. El sistema operativo utilizado es OpenSuse Linux 10. Además se utilizó para algunas pruebas secuenciales una computadora Pentium III de 933 MHz, 512 MB RAM con Sistema Operativo Suse Linux 9.3.

4.5 – Codificación de soluciones

Como se expresó en el capítulo que trata el tópico algoritmos genéticos, la codificación de las soluciones es una característica absolutamente dependiente del problema. En el caso de este trabajo, los individuos representan asignaciones de frecuencias para una red de telefonía móvil compuesta por un conjunto de transmisores que pertenecen cada uno a una determinada celda.

Se representará a un individuo como un vector de números enteros de longitud t (cantidad de transmisores del escenario) de tal manera que el i -ésimo elemento del vector corresponde a la frecuencia asignada al i -ésimo transmisor del escenario. Esta

codificación es presentada como R1 por Aardal [2001] y fue seleccionada por su sencillez, y consecuente simplicidad de los operadores genéticos que pueden aplicarse sobre ella.

En la figura 4.3 se presenta un ejemplo de un individuo que representa una solución para el problema de asignación de frecuencias en el escenario *Tiny* [Eisenblätter, 1998].

16	5	9	13	15	7	11	16	7	12	6	10
----	---	---	----	----	---	----	----	---	----	---	----

Figura 4.3 – Codificación de un individuo.

De esta manera, el primer transmisor del escenario tendrá asignada la frecuencia 16, el segundo la frecuencia 5, el tercero la frecuencia 9, y así sucesivamente.

4.5.1 – Codificaciones alternativas.

En la bibliografía sobre el problema de asignación de frecuencias en redes inalámbricas se mencionan las siguientes alternativas para representar una solución [Aardal et al., 2001]:

- R2: Cada cromosoma es una división de subconjuntos $S_1, S_2, \dots, S_{F_{max}}$, llamados genes. S_f es el conjunto de vértices a los cuales es asignada la frecuencia f .
- R3: Cada cromosoma es una permutación de todos los transmisores del escenario. Al transmisor de la posición i se le asigna la menor frecuencia que no cause interferencias inaceptables con aquellos transmisores de las posiciones 1 hasta $i-1$.

4.6 – Función de *Fitness*

La función de *fitness* expresa la aptitud del individuo respecto de la población. En este caso serán mejores aquellos que tengan menor penalidad e interferencia, por esto se plantea un cociente donde estas dos variables influyen en el denominador. La penalidad está determinada por la cantidad de restricciones de canales bloqueados y de separaciones violadas por la asignación, mientras que la interferencia es la suma de todas las interferencias co-canal y de canal adyacente en las que incurre la solución.

Para la selección de una función de *fitness* que realice un buen balance entre penalidades por restricciones violadas e interferencia incurrida se analizaron las tres versiones presentadas en la figura 4.4.

$f_1 = \frac{1}{\sum i + (\sum p)^2}$	$\sum p$: Sumatoria de las restricciones de separación y de canales bloqueados violadas por la solución. $\sum i$: Sumatoria de las interferencias co-canal y de canal adyacente en las que incurre la solución.
$f_2 = \frac{1}{\sum i + \log(\sum p + 1)}$	
$f_3 = \frac{1}{\sum p + \sum i + 1}$	

Figura 4.4 – Diferentes versiones para la función de *fitness*

La función f_1 enfatiza en perjudicar a aquellos individuos que violan muchas restricciones de canales bloqueados y de separación. Por otra parte, f_2 no penaliza tanto a las violaciones en cuestión puesto que les da una ponderación logarítmica. En tercer

lugar, en f_3 se penaliza de manera lineal con la adición de una unidad en el denominador para lograr resultados normalizados que caigan en el intervalo $(0,1]$.

Del análisis de estas funciones podría interpretarse que el peso de cada uno de los términos variables del denominador dependen exclusivamente del operador que se le apliquen (cuadrado, logaritmo ó identidad). Sin embargo debe tenerse en cuenta que $\sum p$ es una sumatoria de valores enteros mientras que $\sum i$ acumula valores reales entre 0 y 1, teniendo entonces menor importancia para el cómputo final.

Se compararon estas tres versiones utilizando 20 ejecuciones independientes de un algoritmo genético simple con probabilidad de cruce y mutación de 0.6 y 0.1 respectivamente. El número de generaciones fue 500 y la población estuvo conformada por 100 individuos.

Para determinar la calidad de cada una de las versiones, se tomó como parámetro el porcentaje de ocurrencias del mejor individuo conocido en las 20 ejecuciones.

Los resultados obtenidos se muestran en el gráfico 4.1.

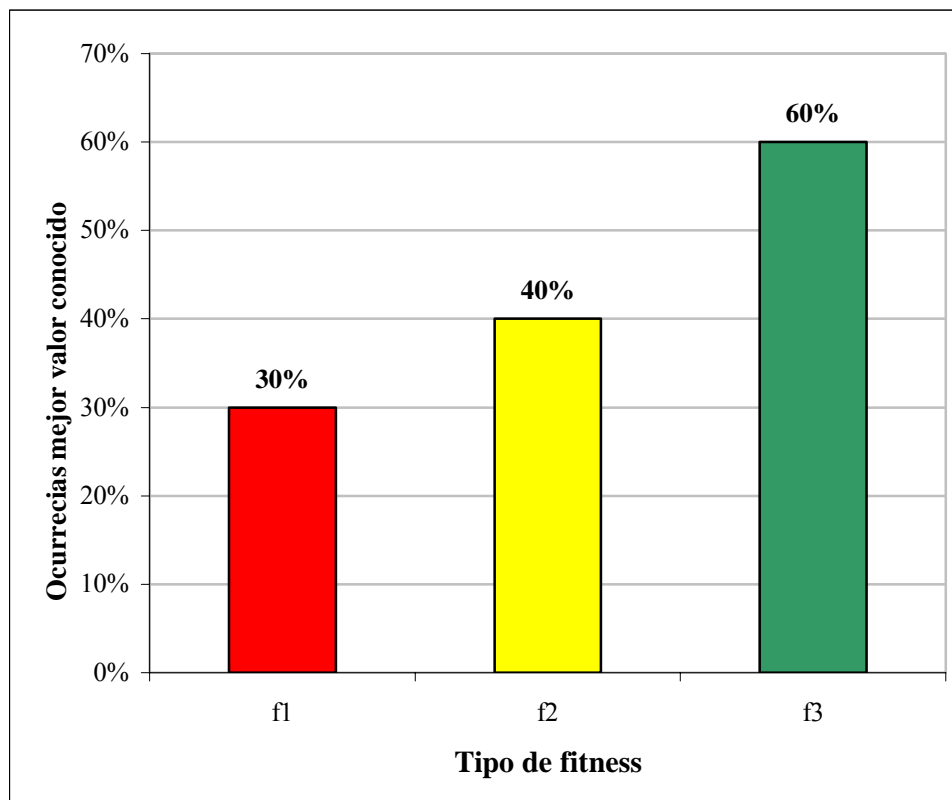


Gráfico 4.1 – Comparación de versiones de la función de fitness

Teniendo en cuenta estos resultados, se decidió utilizar f_3 para las siguientes comparaciones algorítmicas.

4.7 - Algoritmos genéticos utilizados

Las primeras pruebas se realizaron utilizando el algoritmo genético clásico. Sin embargo, la calidad de los resultados obtenidos no fue satisfactoria al compararlos con resultados obtenidos utilizando otras técnicas metaheurísticas publicados en la página web de FAP [FAP Web, 2006]. El algoritmo genético mostraba un comportamiento de

convergencia prematura, estancándose en valores lejanos a los mejores resultados publicados para el problema. En la tabla 4.1 se evidencia esta situación ya que sobre un total de 2000 generaciones, muchas ejecuciones se estancan en generaciones tempranas sin alcanzar el mejor valor conocido (0,9803). El caso más evidente es el de la primera fila.

Mejor valor de <i>fitness</i> encontrado	Generación en la cual se halló
0,9433	90
0,9615	707
0,9615	1293
0,9708	113
0,9708	137
0,9708	199
0,9708	209
0,9708	278
0,9708	487
0,9708	551
0,9708	1166
0,9708	1819
0,9803	77
0,9803	123
0,9803	134
0,9803	157
0,9803	194
0,9803	232
0,9803	262
0,9803	339
0,9803	459
0,9803	675
0,9803	763
0,9803	797
0,9803	1014
0,9803	1115
0,9803	1117
0,9803	1707
0,9803	1838
0,9803	1967

Tabla 4.1 – Comparación de versiones de la función de *fitness*

Como consecuencia, se tomó la decisión de utilizar una variante elitista del AG clásico llamada CHC (*Cross generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation*). CHC combina una estrategia de selección que conserva a los mejores individuos encontrados con un operador de *crossover* heterogéneo uniforme (*HUX*) que genera alta diversidad en la población al producir descendientes tan diferentes de los padres como sea posible. Este algoritmo introduce junto con el cruzamiento un mecanismo llamado prevención de incesto (*incest prevention*) que evita que se reproduzcan dos individuos muy similares. Además, se aplica un proceso de reinicialización de la población que introduce diversidad cuando la convergencia es detectada. En la figura 4.5 se visualiza la estructura genérica del algoritmo CHC.

```

Inicializar(P(0))
generacion = 0
cota_cruzamiento = TamañoIndividuo/4
Evaluar(P(0))
Mientras (no CriterioParada) hacer
    Padres = Seleccion(P(generacion))
    Hijos = HUX(Padres)
    Evaluar(Hijos)
    NuevaPop = Reemplazar(Hijos,P(generacion))
    Si (NuevaPop == P(generacion)) entonces
        cota_cruzamiento --
    fin Si
    generacion ++
    P(generacion) = NuevaPop
    Si (cota_cruzamiento == 0) entonces
        Reinicializacion(P(generacion))
        cota_cruzamiento = LargoCromosoma/4
    fin Si
fin Mientras
retornar Mejor Solucion Hallada

```

Figura 4.5 – Estructura genérica del algoritmo CHC.

Utilizando el algoritmo CHC se logró evitar la convergencia prematura, pues el operador de reinicialización demostró ser una muy buena herramienta para mantener la diversidad aumentando la desviación estándar del *fitness* de la población. En la figura 4.6 queda en evidencia el éxito de aplicar este operador, pues se nota claramente el aumento abrupto de la diversidad en la población (picos), luego de detectarse estancamientos (mesetas).

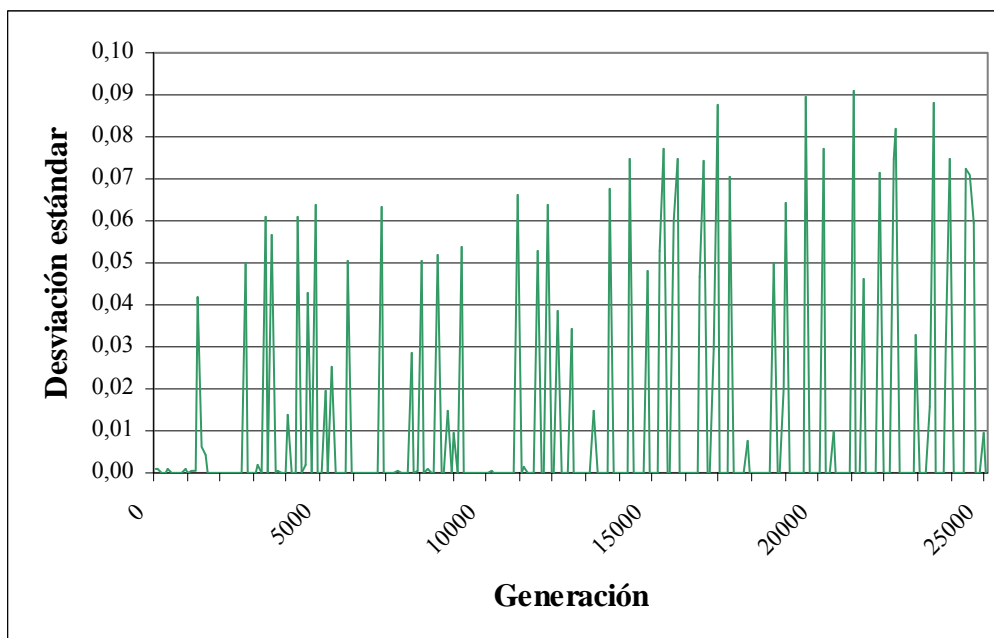


Figura 4.6 – Evolución de la desviación estándar del fitness de la población.

4.8 - Análisis paramétrico de resultados experimentales

El algoritmo CHC posee un conjunto de parámetros a cada uno de los cuales debe asignárseles un determinado valor. Esta asignación puede ser un factor determinante de la calidad de las soluciones halladas, y por ende su selección es de suma importancia para el éxito del algoritmo. Para poder decidir cuales son los mejores valores candidatos de estos parámetros es necesario realizar una comparación de ejecuciones con distintas configuraciones. Por restricciones de recurso computacional, debió optarse por realizar el análisis paramétrico utilizando un único escenario. El elegido fue el escenario *K*, ya que por sus características demanda menor tiempo de cómputo de todos a excepción del *Tiny*.

Para realizar el análisis se consideraron los parámetros: tamaño de la población (*pp*), probabilidad de cruzamiento (*pc*) y probabilidad de reinicialización (*pr*). Los valores candidatos son listados en la tabla 4.1.

Parámetro	Valores
<i>pp</i>	{90; 150}
<i>pc</i>	{0.7; 0.9}
<i>pr</i>	{0.2 ;0.4}

Tabla 4.1 - Valores candidatos para los parámetros del AG.

Se realizaron tres ejecuciones independientes para cada una de las 8 combinaciones paramétricas. El criterio de parada se fijó en 25000 generaciones (*pg*), pues es un valor que arroja resultados aceptables para el escenario *K* según experimentos preliminares.

En la figura 4.7 se presenta gráficamente la evolución del *fitness* promedio de las tres ejecuciones de cada combinación de parámetros (*pg*, *pp*, *pc*, *pr*) conforme avanzan las generaciones.

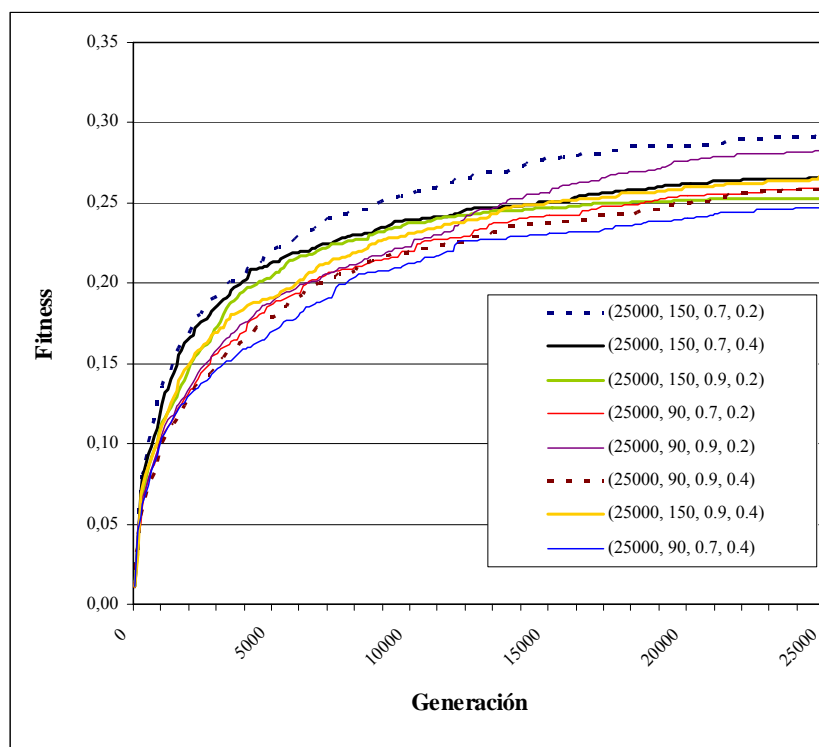


Figura 4.7 – Comparativa de evolución del *fitness* en el análisis paramétrico.

La figura 4.7 muestra que con una población de 150 individuos, probabilidad de cruzamiento de 0.7 y una probabilidad de reinicialización igual a 0.2, se logran los mayores (mejores) valores de *fitness* a lo largo de las 25000 generaciones.

4.9 - Algoritmos genéticos secuenciales y paralelos

Durante la etapa de pruebas se llevaron a cabo distintas ejecuciones de los algoritmos utilizando una versión secuencial y una paralela con 3 procesadores. Estas ejecuciones se realizaron para tres escenarios disponibles. Uno de ellos es *Tiny* que no es un escenario real, y los otros dos son *K* y *Swisscom*, escenarios reales cuyo tamaño y complejidad crecen considerablemente respecto del primero. En las siguientes subsecciones se detallan los resultados obtenidos luego de las ejecuciones de los algoritmos y su comparación en cuanto a mejor valor de *fitness* obtenido y tiempos de ejecución.

4.9.1 – Resultados para el escenario *Tiny*

Tiny se caracteriza por su simplicidad y tamaño reducido, pues no representa un escenario real sino que ha sido creado con fines didácticos [Eisenblätter, 1998]. Las tablas 4.2 y 4.3 presentan los resultados de las ejecuciones realizadas para este escenario.

Ejecución	Mejor <i>fitness</i> encontrado	Interferencia	Canales bloqueados	Separaciones violadas	Generación de encuentro del mejor	Tiempo (en microsegundos)	Tiempo (en horas)
1	0,9803920	0,0200000	0	0	21341	5,04E+08	0,14
2	0,9803920	0,0200000	0	0	14418	4,66E+08	0,13
3	0,9803920	0,0200000	0	0	18193	5,18E+08	0,14
4	0,9803920	0,0200000	0	0	1188	4,48E+08	0,12
5	0,9803920	0,0200000	0	0	13404	4,72E+08	0,13
6	0,9803920	0,0200000	0	0	7261	4,89E+08	0,14
7	0,9803920	0,0200000	0	0	45666	5,08E+08	0,14
8	0,9803920	0,0200000	0	0	3335	4,61E+08	0,13
9	0,9803920	0,0200000	0	0	68432	5,91E+08	0,16
10	0,9803920	0,0200000	0	0	12480	4,48E+08	0,12
Promedio	0,9803920	0,0200000	0	0	20572	4,91E+08	0,14
Desvío estándar	0,0000000	0,0000000	0	0	20900	4,30E+07	0,01

Tabla 4.2 – Detalle ejecuciones secuenciales escenario *Tiny*.

Ejecución	Mejor <i>fitness</i> encontrado	Interferencia	Canales bloqueados	Separaciones violadas	Generación de encuentro del mejor	Tiempo (en microsegundos)	Tiempo (en horas)
1	0,9803920	0,0200000	0	0	21705	1,65E+08	0,05
2	0,9803920	0,0200000	0	0	29937	1,56E+08	0,04
3	0,9803920	0,0200000	0	0	23645	1,68E+08	0,05
4	0,9803920	0,0200000	0	0	27570	1,74E+08	0,05
5	0,9803920	0,0200000	0	0	20896	1,70E+08	0,05
6	0,9803920	0,0200000	0	0	18481	1,68E+08	0,05
7	0,9803920	0,0200000	0	0	24086	1,89E+08	0,05
8	0,9708740	0,0300000	0	0	7883	1,74E+08	0,05
9	0,9803920	0,0200000	0	0	33108	1,82E+08	0,05
10	0,9803920	0,0200000	0	0	12155	1,77E+08	0,05
Promedio	0,9794402	0,0210000	0	0	21947	1,72E+08	0,05
Desvío estándar	0,0030099	0,0031623	0	0	7705	9,08E+06	0,00

Tabla 4.3 – Detalle ejecuciones paralelas escenario Tiny.

4.9.2 – Resultados para el escenario K

La instancia K, provista por la compañía alemana E-Plus Mobilfunk, contiene datos de una red urbana densa, con 264 celdas, 267 transmisores en total y un número promedio de relaciones (interferencia potencial) por transmisor de 151 [COST259, 1996].

A continuación se presentan las tablas 4.4 y 4.5, correspondientes al detalle de las ejecuciones secuenciales y paralelas del algoritmo respectivamente.

Ejecución	Mejor <i>fitness</i> encontrado	Interferencia	Canales bloqueados	Separaciones violadas	Generación de encuentro del mejor	Tiempo (en microsegundos)	Tiempo (en horas)
1	0,3112077	2,2043400	0	0	30740	5,99E+10	16,65
2	0,3369780	1,9675600	0	0	65153	5,93E+10	16,47
3	0,3273140	2,0551703	0	0	64562	5,96E+10	16,56
4	0,3165600	2,1589588	0	0	55125	5,95E+10	16,54
5	0,3100420	2,2253695	0	0	67839	6,04E+10	16,77
6	0,2551140	2,9198162	0	0	64402	5,99E+10	16,65
7	0,2975990	2,3602263	0	0	68078	6,02E+10	16,72
8	0,3240740	2,0857150	0	0	58217	5,84E+10	16,23
9	0,2944150	2,3965661	0	0	57594	5,88E+10	16,33
10	0,2957440	2,3813027	0	0	59499	5,90E+10	16,39
Promedio	0,3069048	2,2755025	0	0	59121	5,95E+10	16,53
Desviación estándar	0,0230621	0,2684777	0	0	10928	6,28E+08	0,17

Tabla 4.4 – Detalle ejecuciones secuenciales escenario K.

Ejecución	Mejor <i>fitness</i> encontrado	Interferencia	Canales bloqueados	Separaciones violadas	Generación de encuentro del mejor	Tiempo (en microsegundos)	Tiempo (en horas)
1	0,2928100	2,4151839	0	0	65079	1,68E+10	4,65
2	0,3041630	2,2877109	0	0	66615	1,67E+10	4,64
3	0,3035950	2,2938619	0	0	65020	1,67E+10	4,64
4	0,2863220	2,4925713	0	0	68614	1,68E+10	4,66
5	0,3051690	2,2768728	0	0	64177	1,69E+10	4,71
6	0,3251640	2,0753712	0	0	54514	1,67E+10	4,64
7	0,2818590	2,5478732	0	0	56736	1,68E+10	4,67
8	0,3062010	2,2658287	0	0	65895	1,67E+10	4,63
9	0,3043110	2,2861119	0	0	60647	1,68E+10	4,66
10	0,2756620	2,6276309	0	0	61801	1,65E+10	4,59
Promedio	0,2985256	2,3569017	0	0	62910	1,67E+10	4,65
Desviación estándar	0,0144749	0,1630605	0	0	4483	1,08E+08	0,03

Tabla 4.5 – Detalle ejecuciones paralelas escenario K.

4.9.3 – Resultados para el escenario Swisscom

Las características de este escenario son brindadas por Swisscom Ltd., compañía suiza de telecomunicaciones, y representan datos de una red celular GSM 900 para una ciudad en la cual existe gran cantidad de canales localmente bloqueados. El número promedio de transmisores por celda es de 2.09. Además hay 148 celdas, cada una con entre 1 y 4 transmisores y 707 relaciones entre vecinos. En general se dispone de 47 frecuencias, pero 136 celdas sufren de restricciones. En promedio, 19 frecuencias están disponibles por celda aunque en el peor de los casos, sólo se cuenta con una [COST259, 1996].

Las tablas 4.6 y 4.7 presentan, respectivamente, los resultados de las ejecuciones secuenciales y paralelas realizadas para este escenario.

Ejecución	Mejor <i>fitness</i> encontrado	Interferencia	Canales bloqueados	Separaciones violadas	Generación de encuentro del mejor	Tiempo (en microsegundos)	Tiempo (en horas)
1	0,0006293	28,1890000	6	4	66176	9,37E+10	26,02
2	0,0003308	21,7080000	8	6	69992	9,40E+10	26,10
3	0,0006545	26,8160000	5	5	50962	9,56E+10	26,55
4	0,0007935	29,2750000	5	4	48068	9,62E+10	26,73
5	0,0006299	26,6320000	6	4	49456	9,72E+10	26,99
6	0,0010138	25,4120000	4	4	68876	9,51E+10	26,40
7	0,0003370	25,9830000	7	7	60598	9,50E+10	26,38
8	0,0012857	26,8020000	4	3	69541	9,38E+10	26,06
9	0,0002930	22,4560000	8	7	43608	9,37E+10	26,03
10	0,0006319	21,5140000	6	4	68065	9,43E+10	26,20
Promedio	0,0006599	25,4787000	6	5	59534	9,49E+10	26,35
Desvío estándar	0,0003135	2,7071864	1	1	10408	1,19E+09	0,33

Tabla 4.6 – Detalle ejecuciones secuenciales escenario Swisscom.

Ejecución	Mejor <i>fitness</i> encontrado	Interferencia	Canales bloqueados	Separaciones violadas	Generación de encuentro del mejor	Tiempo (en microsegundos)	Tiempo (en horas)
1	0,0004579	22,8860000	6	6	56332	3,04E+10	8,45
2	0,0007995	19,8510000	5	4	67620	3,02E+10	8,38
3	0,0007251	28,1930000	6	3	63772	3,06E+10	8,49
4	0,0006538	28,5410000	5	5	61538	3,04E+10	8,43
5	0,0006557	24,0950000	5	5	57406	3,01E+10	8,35
6	0,0006296	27,2600000	6	4	63363	3,01E+10	8,35
7	0,0007954	26,2250000	5	4	56118	3,02E+10	8,39
8	0,0005407	18,4290000	6	5	61500	3,02E+10	8,38
9	0,0006536	29,0380000	5	5	63517	3,04E+10	8,45
10	0,0005395	22,7140000	5	6	64081	3,03E+10	8,41
Promedio	0,0006451	24,7232000	5	5	61525	3,03E+10	8,41
Desvío estándar	0,0001108	3,7280539	1	1	3790	1,70E+08	0,05

Tabla 4.7 – Detalle ejecuciones paralelas escenario Swisscom.

4.9.4 – Análisis de resultados

Esta sección tiene como objetivo comparar la versión secuencial y paralela del algoritmo a través de dos criterios: calidad promedio de soluciones obtenidas y tiempo promedio de ejecución.

Considerando el primer criterio de comparación, las figuras 4.8, 4.9 y 4.10 presentan gráficamente los niveles de calidad promedio de las soluciones obtenidos mediante las versiones secuenciales y paralelas.

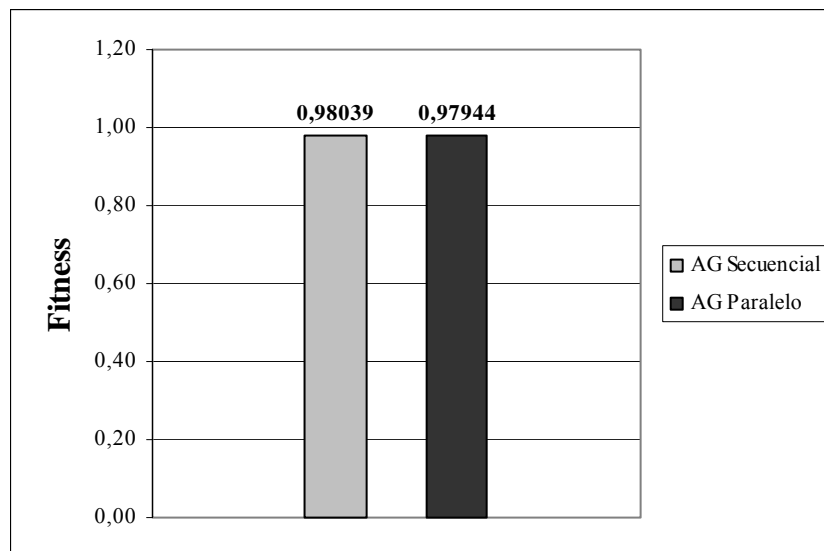


Figura 4.8 – Comparación de valores promedio de mejores fitness alcanzados para el escenario Tiny.

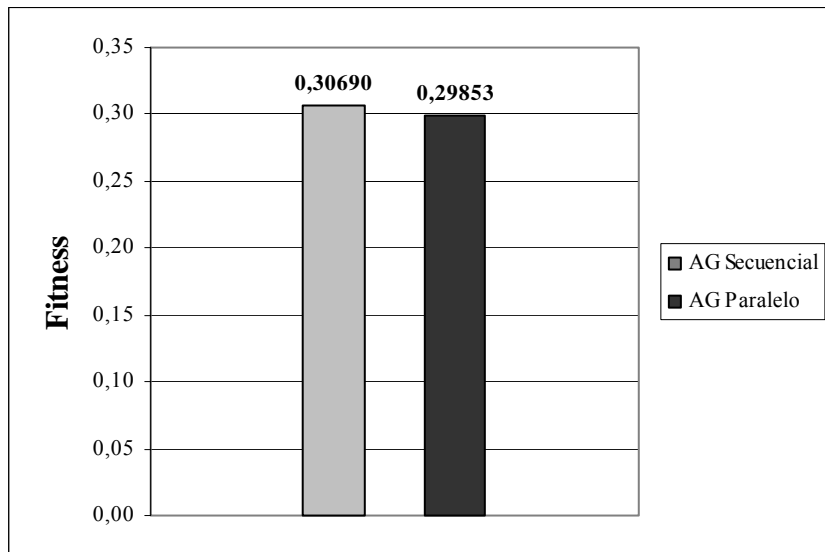


Figura 4.9 – Comparación de valores promedio de mejores fitness alcanzados para el escenario K.

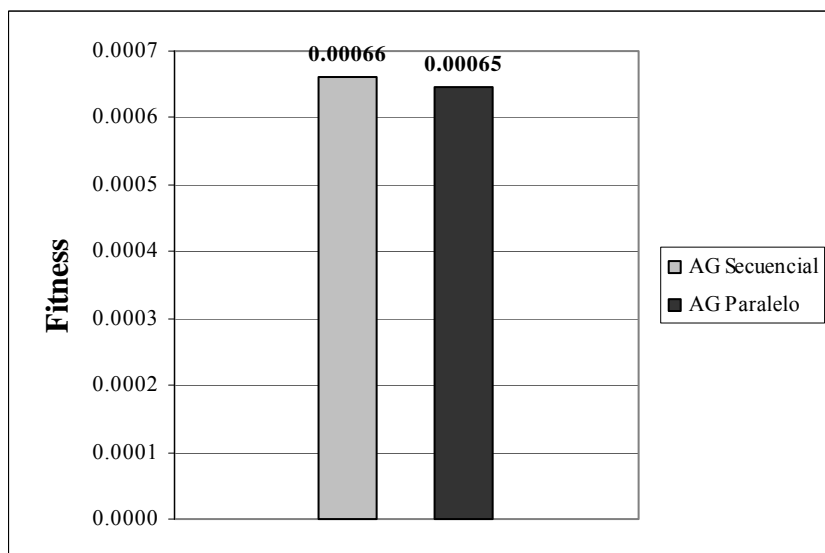


Figura 4.10 – Comparación de valores promedio de mejores fitness alcanzados para el escenario Swisscom.

En todas las comparaciones puede notarse que el valor de *fitness* promedio obtenido por el AG secuencial es superior al del hallado para el AG paralelo. Sin embargo la diferencia no representa una mejora sustancial en la calidad de la solución, ya que la misma es menor a la desviación estándar de los mejores valores de *fitness* obtenidos en las diez ejecuciones. Puede afirmarse entonces que la versión paralela alcanzó resultados de igual calidad que su par serial.

Contrastar los tiempos de ejecución promedio de la versión paralela y secuencial en cada escenario resulta de interés para determinar los beneficios de la distribución de la carga computacional. Las figuras 4.11, 4.12 y 4.13 presentan este contraste.

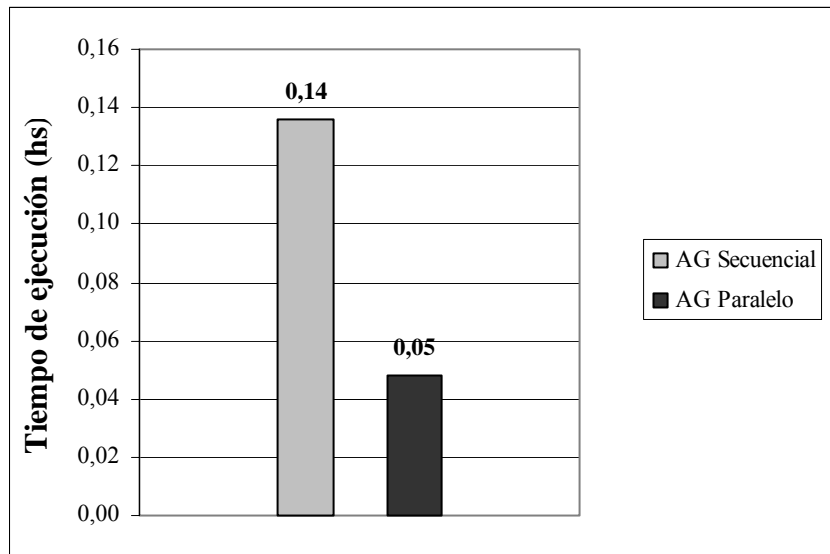


Figura 4.11 – Comparación de tiempos de ejecución promedio alcanzados para el escenario Tiny.

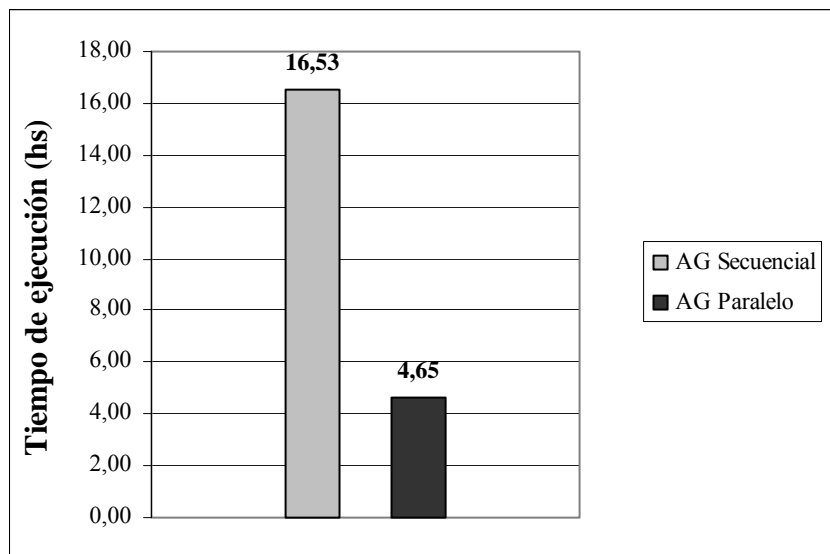


Figura 4.12 – Comparación de tiempos de ejecución promedio alcanzados para el escenario K.

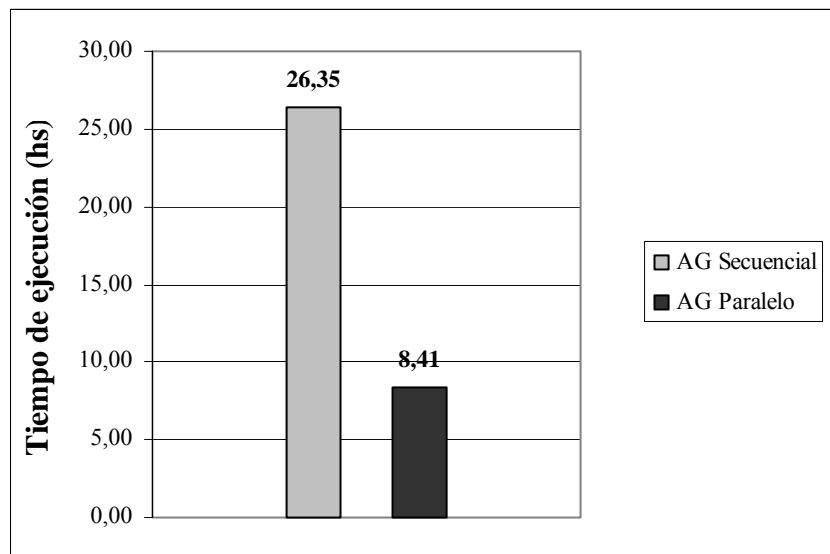


Figura 4.13 – Comparación de tiempos de ejecución promedio alcanzados para el escenario *Swisscom*.

Al analizar los tiempos de ejecución promedio para una y otra versión del AG, se detectó un *speedup* superlineal para los dos escenarios reales, pues al dividir la tarea en tres procesadores, se reduce en un factor mayor a 3 el tiempo de ejecución. Esta característica otorga crédito a la paralelización de algoritmos como recurso para disminuir los tiempos de cómputo.

4.10 - Particularidades de diferentes escenarios.

El algoritmo genético propuesto tiene la capacidad de resolver el problema de asignación de frecuencias (*MI-FAP*) para cualquiera de los escenarios existentes, sin embargo la calidad de las soluciones halladas depende, en parte, de las características propias de cada escenario. Estas condiciones dependen del contexto de la red de comunicaciones inalámbricas, por ejemplo, en algunos casos estas redes están inmersas en densas poblaciones con un alto nivel de comunicaciones y con una pequeña extensión territorial a cubrir. Alternativamente, existen redes con bajo volumen de comunicaciones, pero altas cantidades de restricciones y limitaciones. En etapas de pruebas pudo evidenciarse que, para determinadas instancias, pequeños ajustes en la codificación del algoritmo propiciaban la obtención de soluciones de mayor calidad.

Por ejemplo en el caso del escenario *Swisscom*, la linealidad del denominador de la función de *fitness* presentó aspectos negativos, pues los niveles de interferencia promedio del escenario son altos. Para paliar esta situación, debió intensificarse el modo de darles prioridad de supervivencia a los individuos con menor cantidad de penalidades violadas antes que a aquellos con alta interferencia. Para lograr este objetivo, inicialmente se sustituyeron los términos correspondientes a las penalidades por sus cuadrados. Los resultados obtenidos fueron positivos durante generaciones tempranas, donde había gran cantidad de restricciones violadas pero no se lograron mejoras significativas cuando estas penalidades alcanzaban el orden de la raíz cuadrada de la interferencia. Finalmente, se decidió multiplicar estos cuadrados por el valor aproximado de la interferencia promedio del escenario, obteniendo así los resultados presentados con anterioridad.

4.11 - Otros enfoques propuestos para escenarios COST 259

Un relevamiento bibliográfico permite encontrar diversos métodos heurísticos que han sido utilizados para abordar el problema FAP, entre ellos se pueden mencionar:

- "DSATUR with Costs" (5%) followed by "Iterated 1-Opt": Andreas Eisenblätter, ZIB, Alemania [DC5_IM, 1998].
- Simulated Annealing: Luis de Urries, Telefónica, España [SA, 2000].
- Simulated Annealing: Dirk Beckmann, TU Hamburg-Harburg, Alemania [SA, 1999].
- Threshold Accepting: Martin Hellebrandt, RWTH Aachen, Alemania [TA, 2000].
- Threshold Accepting: Hans Heller, Siemens AG, Alemania [TA – 2000b].
- Dynamic Tabu Search: D.H. Smith, U Glamorgan, Reino Unido [DTS, 2001].
- Simulated Annealing combined with Dynamic Programming to compute local optima: C. Mannino, G. Oriolo, F. Ricci (University of Rome La Sapienza), Italia [K-THIN, 2002].

Los resultados de cada una de estas alternativas están publicados el sitio de COST259 [1996], donde se presentan en forma de tablas los valores alcanzados para cada una de las variables relevantes del problema. Entre otras se pueden mencionar las violaciones de las restricciones de separación de canales y de canales bloqueados, la suma total de interferencias, la suma total de interferencias co-canal y canal adyacente, etc..

Los resultados presentados en este trabajo no superan a los publicados, sin embargo es importante destacar que la propuesta de este trabajo es la primera que utiliza técnicas evolutivas para la resolución de MI-FAP.

4.12 – Conclusiones

Al momento de la publicación de esta tesina no se conocen referencias sobre la resolución del problema MI-FAP para escenarios COST259 usando algún tipo de algoritmo genético. Este capítulo demuestra que es posible el enfoque mencionado tanto para versiones secuenciales como paralelas de estos algoritmos.

En todos los casos se utilizaron operadores genéticos simples que no incorporan información específica del problema. Introducir operadores *ad hoc* sugiere la posibilidad de mejorar la calidad de las soluciones encontradas, pues disminuiría considerablemente la cantidad de individuos inaceptables generados.

Por razones de tiempos y disponibilidad de poder de cálculo, no se han podido mejorar los resultados publicados [FAP Web, 2006], sin embargo los autores de este trabajo están convencidos de que pueden alcanzarse, e incluso superarse, debido a la potencialidad de la técnica. Algunas de las alternativas para mejorar los resultados podría ser alternar o modificar los operadores genéticos, cambiar la codificación de las soluciones, aumentar el número de subpoblaciones, utilizar otra función de *fitness*, etc. Este argumento se ve reforzado con el logro de un *speedup* superlineal al utilizar algoritmos genéticos paralelos, sin que esto vaya en detrimento considerable de la calidad de las soluciones encontradas.

CAPÍTULO 5

CONCLUSIONES GENERALES

*"Hay hombres que luchan un día
Y son buenos,
Hay otros que luchan un año
Y son mejores,
Hay quienes luchan muchos años
Y son muy buenos,
Pero hay los que luchan toda la vida,
Esos son los imprescindibles."*⁸

BERTOLT BRECHT
Die Mutter, 1930

5.1 - Conclusiones

Gracias al intensivo trabajo de investigación sobre los problemas de asignación de frecuencias se pudieron identificar claramente sus características, en particular las del MI-FAP (minimum interference frequency assignment problem). De este análisis se infirió que se trata de un problema de optimización combinatoria ubicado dentro de la categoría de aquellos conocidos como NP-difíciles. Además, se logró un alto nivel de entendimiento de los algoritmos genéticos, comprobando que por sus características particulares, son una herramienta que responde perfectamente al desafío que plantea el problema elegido. Después de las correspondientes experimentaciones se pudo afirmar que los algoritmos genéticos constituyen una técnica aplicable para la resolución del problema de asignación de frecuencias.

Se realizaron comparaciones entre algoritmos secuenciales y paralelos, proporcionando resultados de relevancia principalmente en dos aspectos: la calidad de las soluciones halladas y el tiempo empleado para realizar la búsqueda evolutiva. De este análisis se infiere que los algoritmos paralelos presentan soluciones de calidad similar en tiempos mucho menores, obteniéndose en algunos casos, niveles de eficiencia computacional mayores a uno.

Durante el desarrollo de esta tesis, en particular en la etapa de implementación de los algoritmos, la principal motivación fue tratar de obtener resultados de similar calidad a los reportados con otras técnicas en la resolución de este problema. Si bien los resultados obtenidos utilizando operadores evolutivos simples indican que no fue posible lograr este objetivo en esta etapa del trabajo, los autores de esta tesis consideran que disponiendo de mayor capacidad de cómputo e introduciendo mejoras al mecanismo de búsqueda (a través de operadores específicos para el problema), existen alentadoras perspectivas para la mejora de los resultados. Dicha consideración espera ser comprobada con el desarrollo de los trabajos futuros propuestos en el presente documento.

⁸ *"Die Schwachen kämpfen nicht. Die Stärkeren kämpfen vielleicht eine Stunde lang. Die noch stärker sind, kämpfen viele Jahre. Aber die Stärksten kämpfen ihr Leben lang. Diese sind unentbehrlich."*

5.2 - Trabajos futuros

Este trabajo deja abiertas ciertas líneas de investigación que al abordarse permitirán mejorar la calidad de las soluciones apuntando a superar los resultados publicados.

En primer lugar, debe mencionarse que los operadores evolutivos utilizados en esta tesis han sido genéricos. Esto deja planteada la duda acerca de la posibilidad de lograr una mejora en la calidad de las soluciones implementando operadores diseñados específicamente para este problema, como se planteó en las conclusiones del capítulo 4. Una posibilidad es modificar el operador de reinicialización para evitar asignar a un transmisor un canal bloqueado.

La codificación utilizada fue la que resultó más intuitiva a un grupo de personas que se iniciaban en el tópico. Como se planteó en este trabajo, existen para este problema otros dos esquemas de codificación publicados. Los diferentes tres tipos podrían compararse para evaluar cuál arroja mejores resultados. Indefectiblemente cambiar la codificación repercutirá en la estructura de los operadores. También es posible incluir, en la comparación, un esquema original que represente las soluciones de una manera nunca antes publicada.

Conforme avanza la evolución, el balance entre exploración y explotación puede modificarse. Utilizar una función de *fitness* dependiente del grado de avance del algoritmo podría ser útil para considerar un entorno dinámico que permita una amplia cobertura de exploración en etapas tempranas de la evolución y una explotación de las mejores soluciones a medida que aumenta el número de generaciones.

Analizar las modificaciones que debieron realizarse a la función de *fitness* cuando se abordaron diferentes escenarios sugiere enfocar el problema de diseñar una función versátil que pueda ser aplicada con éxito a escenarios con características tan dispares como sea posible.

Como se ha mencionado en la sección correspondiente a paralelismo del capítulo 2, existen varias alternativas para ejecutar un algoritmo en más de un procesador. Profundizar la aplicación de esta técnica proponiendo diferentes modelos de paralelismo tales como un esquema celular ó uno híbrido es una tarea que queda latente en este trabajo. Asimismo, investigar la escalabilidad de la propuesta de subpoblaciones distribuidas al disponer de mayores recursos de cómputo es una tarea a realizar en un futuro. En el caso de disponer de grandes *clusters*, podrían abordarse escenarios reales de mayor tamaño.

Por último, considerando las penalidades de separación, los canales bloqueados y la interferencia como objetivos independientes a minimizar, es absolutamente factible plantear un algoritmo evolutivo multiobjetivo para el problema de asignación de frecuencias.

Como continuación del proyecto de investigación reportado, se propone abordar los tópicos comentados en esta sección en el futuro cercano.

ANEXO I

AGRADECIMIENTOS

“Ningún hombre es una isla, algo completo en sí mismo; todo hombre es un fragmento del continente, una parte de un conjunto.”⁹

JOHN DONNE
Devotions Upon Emergent Occasions, 1624.

AI.1 - Agradecimientos generales

Luego de experimentar largas jornadas laborales y sortear innumerables obstáculos que naturalmente surgen en el camino de la investigación, los autores de este trabajo reconocen gratamente la colaboración directa o indirecta, explícita o implícita de un gran número de personas que de una manera u otra aportaron a la causa.

En primer lugar se agradece enormemente a Sergio Nesmachnow, quien actuó de guía durante todo el proceso de elaboración de esta tesina, desde la hermana República del Uruguay. Al profesor José Gallardo por su importante apoyo desde el área de las telecomunicaciones. También es importante mencionar la participación de Mariusz Nowostawski y Ricardo Poli a través del correo electrónico.

Es oportuno extender el sincero agradecimiento a todos los profesores con los cuales los autores han compartido estos años de estudio que, de alguna manera cimentaron este trabajo y muchos que vendrán. Tampoco es posible olvidar a todos aquellos compañeros de estudio que desde otro lugar influyeron en la formación personal y académica de quienes desarrollan este trabajo.

El agradecimiento debe ampliarse a todos los investigadores, profesores y estudiantes que publican su conocimiento, pues es, sin duda, el pilar fundamental de este y muchos otros trabajos. Asimismo, no puede dejar de agradecerse a la comunidad de software libre, que con su filosofía permite que proyectos como este salgan adelante.

Finalmente resulta en extremo difícil ponderar el aporte de la familia Mora (Julio, Mabel y Cecilia) quienes han albergado y soportado a los autores durante todo este tiempo.

Antes de comenzar con los agradecimientos personales se desea dedicar este trabajo a todo el alumnado de la Universidad Nacional de la Patagonia San Juan Bosco.

AI.2 - Agradecimientos personales de Gerardo Mora

Este trabajo se lo dedico a mis papás (Julio y Mabel), a mis hermanas (Cecilia y Fernanda), a mis tíos (Gastón y Ana), a mis abuelos (Gerardo, Lila, Julio, Ani), a mis primos (Gaby, Fredy, Daniela, Gabriela), a mis sobrinos (Agustín, Valentina y Julián) y a mi novia (Vanesa).

Visite <http://www.greenpeace.org/>.

⁹ “No man is an island, entire of itself; every man is a piece of the continent, a part of the main”.

AI.3 - Agradecimientos personales de Cristian Perfumo

Quiero agradecer a mis amigos Lucas y Gerardo por padecer durante un año y medio la convivencia conmigo. También debo decir gracias a (en orden de aparición): Mariano, Polaco, Javi, Marcelo, Negro, Peteco, Renzo, el Oso, Gastón, Pipa, Florencia, Pau, Pecho, Sebacar y Álvaro, pues siempre me alentaron a continuar en el camino del estudio y se bancaron mil faltas a reuniones importantes (esas son reuniones importantes!).

Párrafo aparte merece la familia. No basta con decirles gracias a ellos una ni mil veces, pues creo que la única manera de reconocer realmente todo lo que han hecho por mí es comprando cal, cemento y hacerles un monumento a cada uno: Mamá Mónica, Papá Norberto (gracias por volver), Mariana y Abuela Ángela gracias, los amo.

AI.2 - Agradecimientos personales de Lucas Rojas

Deseo enfatizar en el agradecimiento de quienes más me han apoyado durante todo este importante período de mi vida personal y profesional. Confieso que durante la mayor parte del tiempo dedicado al desarrollo de este trabajo, he disfrutado del proceso y reconozco que, quizás, quienes se vieron más perjudicados en esta etapa fueron mis afectos. La dedicación requerida por este desafío me llevó a no estar presente en muchos momentos importantes. Agradezco profundamente el apoyo y comprensión aquellas personas que conforman mi círculo más íntimo: mi novia (Maite), mis padres (Agustín y Alicia), mis hermanos (Luis y Cecilia), mi sobrina (Candela), y mis entrañables amigos (que no nombro para no dejar accidentalmente afuera a nadie).

También quiero agradecer a Eduardo Gargiulo y a los *G&G Boys* (Julito y Mauro), quienes colaboraron abiertamente con la causa.

Finalmente llega el turno de los *Dreamsys Boys* (Cristian y Gerardo), con quienes además de realizar este trabajo tuve que convivir durante casi 15 meses.

Una vez más a todos ¡Muchas gracias!

REFERENCIAS BIBLIOGRÁFICAS

"Que otros se jacten de las páginas que han escrito; a mi me enorgullecen las que he leído."

JORGE LUIS BORGES
Un lector, 1969

- [Aardal et al., 2001] – Aardal, K., Van Hoesel, S., Koster, A., Mannino, C., Sassano, A. Models and Solution Techniques for Frequency Assignment Problems. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2001.
- [Alba y Cotta, 2003] – Alba, E., Cotta, C. Tutorial on evolution computation. Disponible en <http://www.lcc.uma.es/~ccottap/semEC/ec.html>. Consultado en Marzo 2006.
- [Alba et al., 2001] – Alba, E., Almeida, F., Blesa, M., Cabeza, J., Cotta, C., Diaz, M., Dorta, I., Gabarro, J., Leon, C., Luna, J., Moreno, L., Pablos, C., Petit, J., Rojas, A., Xhafa, F. MALLBA: A library of skeletons for combinatorial optimization. Proceedings of the Euro-Par, Paderborn (GE), LNCS 2400, pp.927-932, 2002.
- [Alonso, 1997] – Alonso, J. M. Programación de aplicaciones paralelas con MPI (Message Passing Interface). Facultad de Informática. Universidad del País Vasco/Euskal Herriko Unibertsitatea. Disponible en www.urjc.es/cat/hidra/manuales/MPI/tutorial_MPI.pdf. Consultado en Marzo 2006.
- [COST259, 1996] – Eisenblätter, A., Koster, A. COST 259. Wireless Flexible Personalized Communications. Disponible en <http://fap.zib.de/problems/COST259/>. Consultado en Marzo 2006.
- [Darwin, 1859] – Darwin, C. El origen de las especies. Editorial Longseller. 2004.
- [DC5_IM, 1998] – Börndorfer, R., Eisenblätter, A., Grottschel, M., Martin, A. DSATUR with Costs (5%) followed by Iterated 1-Opt - R. Frequency Assignment in Cellular Phone Networks. Annals of Operations Research, 76:73-93, 1998.
- [DTS, 2001] – Smith, D.H., Moon, J., Montemanni, R. Dynamic Tabu Search. U. Glamorgan, Reino Unido. An Improved Tabu Search Algorithm for the Fixed-Spectrum Frequency. Assignment Problem. IEEE Transactions on vehicular technology, vol. 52, No. 3, May 2003.
- [Eisenblätter, 1998] – Eisenblätter, A. COST 259. File Formats for Standard Scenarios for Frequency Planning. Disponible en http://fap.zib.de/problems/COST259/documentation/file_formats-1_0/index.html. Consultado en Marzo 2006.
- [FAP Web, 2006] – Eisenblätter, A., Koster, A. A website about Frequency Assignment Problems. Scientific Computing. Optimization. Konrad-Zuse-Zentrum für Informationstechnik Berlin. Disponible en <http://fap.zib.de/index.php>. Consultado en Marzo 2006.
- [Fogel et al., 1966] – Fogel, L.J., Owens, A.J. and Walsh, M.J. Artificial Intelligence Through Simulated Evolution.
- [Goldberg, 1989] – Goldberg, D. Genetics Algorithms in search, optimization and machine learning. Addison-Wesley. 1998.

- [Holland, 1975] – Holland, J. *Adaptation in natural and artificial systems*. University of Michigan Press. 1975.
- [INTI, 2004] – Mollo, J. El control del espectro radioeléctrico. Disponible en <http://www.inti.gov.ar/sabercomo/inti-03-04/inti12.php>. Consultado en Marzo 2006.
- [K-THIN, 2002] – Mannino, C., Oriolo, G., Ricci, F. Solving Stability Problems on a Superclass of Interval Graphs. Disponible en http://www.optimization-online.org/DB_FILE/2002/04/465.pdf. 2002. Consultado en Marzo 2006.
- [Koster, 1998] – Koster, A. *Frequency Assignment, Models and Algorithms*. Ph.D. Thesis, Universiteit Maastricht, Maastricht, The Netherlands. 1999.
- [Koza, 1990] – Koza, J. *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*, Stanford University Computer Science Department technical report STAN-CS-90-1314, 1990.
- [Koza, 2003] – Koza, J. PowerPoint (PPT) file of overview lecture (Nro. 1 - September 24, 2003). Disponible en <http://www.genetic-programming.com/c2003lecture1modified.ppt>. Consultado en Marzo 2006.
- [Mallba Library, 2004] – Mallba Architecture. Disponible en <http://neo.lcc.uma.es/mallba/easy-mallba/html/mallba.html>. Consultado en Marzo 2006.
- [Merelo, 1997] – Merelo Guervós, J. J. *Informática Evolutiva*. Disponible en <http://geneura.ugr.es/~jmerelo/ie/index.html>. Consultado en Marzo 2006.
- [Michalewicz, 1999] – Michalewicz, Z. *Genetic algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1999.
- [MPI, 2005] – Message-Passing Interface. Disponible en <http://www-unix.mcs.anl.gov/mpi/mpich/>. Consultado en Marzo 2006.
- [Mitchell, 1999] – Mitchell, M. *An Introduction to Genetic Algorithms*. MIT Press, 1999.
- [Moore, 2001] – Moore, M. Teaching Students to use Genetic Algorithms to solve optimization problems. *Journal of Computing in Small Colleges*, vol. 16, number 3, March 2001. With presentation to the 12th Annual South Central Conference of the Consortium for Computing in Small Colleges, April, 2001.
- [Nesmachnow, 2004] – Nesmachnow, S. *Algoritmos Genéticos Paralelos y su aplicación al diseño de redes de comunicaciones confiables*. Universidad de la República. Montevideo, Uruguay. Disponible en <http://www.fing.edu.uy/inco/pedeciba/bibliote/tesis/tesis-nesmachnow.pdf>. Consultado en Marzo 2006.
- [Nolan, 2004] – Nolan, J. *Frequency Assignment and Spectrum Trading for Radio Services*. Tesis Doctoral, University of Oxford, 2004. Disponible en <http://www.smithinst.ac.uk/Projects/PG2000/CASE-RA/reading/Thesis/Thesis.pdf>. Consultado en Marzo 2006.
- [Nowostawski y Poli, 1999] – Nowostawski, M., Poli, R. Parallel Genetic Algorithm Taxonomy. In *Proceedings of the Third International Conference on Knowledge-based Intelligent Information Engineering Systems KES'99*, pages 88-92. IEEE Computer Society, Aug. 1999.

- [Obitko, 1998] – Obitko, M. Introduction to genetic algorithms with Java applets. Disponible en <http://cs.felk.cvut.cz/~xobitko/ga/>. Consultado en Marzo 2006.
- [SA, 1999] – Beckmann, D., Killat, U. Simulated Annealing. Frequency Planning with respect to Interference Minimization in Cellular Radio Networks. 8th COST 259 Meeting, Vienna, Austria, 1999.
- [SA, 2000] – Jordán de Urries, L., Diaz Guerra, M. A., Berberana, I. Simulated Annealing. Frequency Planning using Simulated Annealing. Tech. report, COST 259, Bergen Kongress Senter, Bergen (N), on 27-28 Apr. 2000.
- [TA, 2000] – Hellebrandt, M., Heller, H. Threshold Accepting. A new heuristic method for frequency assignment. Technical Report TD(00) 003, COST 259, Valencia, Spain, Jan. 2000.
- [TA, 2000b] – Hellebrandt, M., Heller, H. Threshold Accepting. A new heuristic method for frequency assignment. Technical Report TD(00) 003, COST 259, Valencia, Spain, Jan. 2000.
- [Tanenbaum, 2003] – Tanenbaum, A. S. Redes de computadoras. Cuarta edición. Pearson, 2003.
- [Thede, 2004] – Thede, S. M. An introduction to genetic algorithms. Journal of Computing Sciences in Colleges archive. Volume 20, Issue 1 (October 2004). pp 115 – 123. 2004 .
- [Whitley, 1994] – Whitley, D. A Genetic Algorithm Tutorial, Statistics and Computing, Volume 4, pp. 65-85. Disponible en <http://www.cs.colostate.edu/~genitor/MiscPubs/tutorial.ps.gz>. Consultado en Marzo 2006.
- [Wiley, 2001] – Wiley, J. Wireless flexible personalised communications. Disponible en <http://www.wiley.co.uk/wileychi/commstech/correia.html>. Consultado en Marzo 2006.