

Estrategias de optimización con algoritmos genéticos complejos en
la resolución de problemas de asignación de canales en redes
celulares

Ing. Héctor Efrain Rodelo Torres

Mayo 2013

Índice general

1	Objetivo General	8
1.1	Objetivos Específicos	8
2	Marco Teórico	9
2.1	Problema de asignación de canales (CAP) en redes celulares	9
2.2	Algoritmos Genéticos (GA) aplicados a CAP en redes celulares	11
2.2.1	Individuos	11
2.2.2	Población	12
2.2.3	Función Objetivo	12
2.2.4	Operador de Selección	13
2.2.5	Operador de Cruce	14
2.2.6	Operador de Mutación	15
2.2.7	Operador de Re-inserción	16
2.2.8	Criterios de Finalización	17
2.3	GA con Operadores Sofisticados	17
2.3.1	Operador Ciclo de Vida	17
2.3.2	Operador Beeder	17
2.3.3	Estrategia Memética	18
2.3.4	Estrategia CHC	18
2.3.5	Estrategia de Relación Genética	18
2.3.6	Estrategia de Evolución Gramatical	19
2.3.7	Estrategia Lógica Difusa	19
2.3.8	Estrategia en Paralelo	19
2.3.8.1	Estrategia en Paralelo tipo Grano	19
2.3.8.2	Estrategia Masivamente Paralelo	20
2.3.8.3	Estrategia en Paralelo tipo Islas	20
2.3.8.4	Estrategia en Paralelo tipo Grillas	21
3	Desarrollo	22
3.1	GA Simple	22
3.2	GA Complejo Paralelo	24
3.2.1	GA Complejo Paralelo tipo Estrella	25
3.2.2	GA Complejo Paralelo tipo Anillo	25
3.2.3	GA Complejo Paralelo tipo Malla	26
4	Evaluación y Análisis de Resultados	28
4.1	GA Simple	28
4.2	GA Complejo Paralelo	29
4.3	Resultados Comparativos	31

5 Conclusiones	37
6 Anexos	43
6.1 Diagramas de bloques de los GA implementados	43
6.1.1 Diagrama de bloques GA-Simple	43
6.1.2 Diagrama de bloques GA-Complejo tipo Paralelo	44
6.1.3 Diagrama de bloques creación de la Población	45
6.1.4 Diagrama de bloques Función Aptitud	46
6.1.5 Diagrama de bloques Operador de Cruce	47
6.1.6 Diagrama de bloques Operador de Mutación	48
6.2 Representación Gráfica de la Solución de CAP	49
6.2.1 Solución Gráfica para P1	49
6.2.2 Solución Gráfica para P2	51
6.2.3 Solución Gráfica para P3	53
6.2.4 Solución Gráfica para P4	55
6.2.5 Solución Gráfica para P5	57
6.2.6 Solución Gráfica para P6	59
6.2.7 Solución Gráfica para P7	61
6.2.8 Solución Gráfica para P8	63
6.2.9 Solución Gráfica para P9	65
6.2.10 Solución Gráfica para P10	67
6.2.11 Solución Gráfica para P11	69

Índice de figuras

2.1	Diagrama de Bolques GA-Simple	11
2.2	Representación de un Individuo	12
2.3	Operadores de Cruce	15
2.4	Operadores de Mutación	16
2.5	Operador Ciclo de Vida	17
2.6	Estrategia de Evolución Gramatical	19
2.7	Estrategia en Paralelo tipo Islas	20
3.1	Representación genes dominantes	22
3.2	Representación Gráfica de Solución	24
3.3	Topología Estrella	25
3.4	Topología Anillo	26
3.5	Topología Malla	26
4.1	Diagrama de frecuencias y probabilidad acumulada del GA-Simple.	29
4.2	Diagrama de frecuencias y probabilidad acumulada de GA-Complejo Paralelo Anillo	31
4.3	Diagrama de frecuencia y probabilidad acumulada de GA-Complejo Paralelo Estrella	31
4.4	Diagrama de frecuencia y probabilidad acumulada de GA-Complejo Paralelo Malla	31
4.5	Análisis convergencia P1	32
4.6	Análisis convergencia P2	32
4.7	Análisis convergencia P3	33
4.8	Análisis convergencia P4	33
4.9	Análisis convergencia P5	34
4.10	Análisis convergencia P6	34
4.11	Análisis convergencia P7	35
4.12	Análisis convergencia P8	35
4.13	Análisis convergencia P9	36
4.14	Análisis convergencia P10	36
4.15	Análisis convergencia P11	36
6.1	Diagrama de Bloques GA-Simple	43
6.2	Diagrama de Bloques GA-Complejo Paralelo	44
6.3	Diagrama Creación de la Población	45
6.4	Diagrama Función Aptitud	46
6.5	Diagrama Operador Cruce	47
6.6	Diagrama Operador Mutación	48
6.7	Representación Gráfica Solución CAP para P1 GA-Simple	49
6.8	Representación Gráfica Solución CAP para P1 GA-Complejo Paralelo Estrella	49
6.9	Representación Gráfica Solución CAP para P1 GA-Complejo Paralelo Anillo	50
6.10	Representación Gráfica Solución CAP para P1 GA-Complejo Paralelo Malla	50

6.11	Representación Gráfica Solución CAP para P2 GA-Simple	51
6.12	Representación Gráfica Solución CAP para P2 GA-Complejo Paralelo Estrella	51
6.13	Representación Gráfica Solución CAP para P2 GA-Complejo Paralelo Anillo	52
6.14	Representación Gráfica Solución CAP para P2 GA-Complejo Paralelo Malla	52
6.15	Representación Gráfica Solución CAP para P3 GA-Simple	53
6.16	Representación Gráfica Solución CAP para P3 GA-Complejo Paralelo Estrella	53
6.17	Representación Gráfica Solución CAP para P3 GA-Complejo Paralelo Anillo	54
6.18	Representación Gráfica Solución CAP para P3 GA-Complejo Paralelo Malla	54
6.19	Representación Gráfica Solución CAP para P4 GA-Simple	55
6.20	Representación Gráfica Solución CAP para P4 GA-Complejo Paralelo Estrella	55
6.21	Representación Gráfica Solución CAP para P4 GA-Complejo Paralelo Anillo	56
6.22	Representación Gráfica Solución CAP para P4 GA-Complejo Paralelo Malla	56
6.23	Representación Gráfica Solución CAP para P5 GA-Simple	57
6.24	Representación Gráfica Solución CAP para P5 GA-Complejo Paralelo Estrella	57
6.25	Representación Gráfica Solución CAP para P5 GA-Complejo Paralelo Anillo	58
6.26	Representación Gráfica Solución CAP para P5 GA-Complejo Paralelo Malla	58
6.27	Representación Gráfica Solución CAP para P6 GA-Simple	59
6.28	Representación Gráfica Solución CAP para P6 GA-Complejo Paralelo Estrella	59
6.29	Representación Gráfica Solución CAP para P6 GA-Complejo Paralelo Anillo	60
6.30	Representación Gráfica Solución CAP para P6 GA-Complejo Paralelo Malla	60
6.31	Representación Gráfica Solución CAP para P7 GA-Simple	61
6.32	Representación Gráfica Solución CAP para P7 GA-Complejo Paralelo Estrella	61
6.33	Representación Gráfica Solución CAP para P7 GA-Complejo Paralelo Anillo	62
6.34	Representación Gráfica Solución CAP para P7 GA-Complejo Paralelo Malla	62
6.35	Representación Gráfica Solución CAP para P8 GA-Simple	63
6.36	Representación Gráfica Solución CAP para P8 GA-Complejo Paralelo Estrella	63
6.37	Representación Gráfica Solución CAP para P8 GA-Complejo Paralelo Anillo	64
6.38	Representación Gráfica Solución CAP para P8 GA-Complejo Paralelo Malla	64
6.39	Representación Gráfica Solución CAP para P9 GA-Simple	65
6.40	Representación Gráfica Solución CAP para P9 GA-Complejo Paralelo Estrella	65
6.41	Representación Gráfica Solución CAP para P9 GA-Complejo Paralelo Anillo	66
6.42	Representación Gráfica Solución CAP para P9 GA-Complejo Paralelo Malla	66
6.43	Representación Gráfica Solución CAP para P10 GA-Simple	67
6.44	Representación Gráfica Solución CAP para P10 GA-Complejo Paralelo Estrella	67
6.45	Representación Gráfica Solución CAP para P10 GA-Complejo Paralelo Anillo	68
6.46	Representación Gráfica Solución CAP para P10 GA-Complejo Paralelo Malla	68
6.47	Representación Gráfica Solución CAP para P11 GA-Simple	69
6.48	Representación Gráfica Solución CAP para P11 GA-Complejo Paralelo Estrella	69
6.49	Representación Gráfica Solución CAP para P11 GA-Complejo Paralelo Anillo	70
6.50	Representación Gráfica Solución CAP para P11 GA-Complejo Paralelo Malla	70

Índice de tablas

4.1	Resumen implementación Algoritmo Genético Simple	28
4.2	Resumen comparativo de implementar los Algoritmos Genéticos	30

Introducción

La masificación en el uso de sistemas de comunicaciones móviles debido al número de abonados en redes celulares ha crecido dramáticamente en los últimos años. Esto ha contribuido a que en las plataformas de radio comunicaciones tradicionales “broadcast”, fuese necesario una configuración con gran potencia de transmisión con torres elevadas, que garantizara una línea de vista con todos sus abonados con el fin de lograr su cobertura. Estos amplios radios de cobertura exigieron reservar el uso de radio canales en grandes área para una cantidad relativamente pequeña de usuarios. Lo que trajo como consecuencia la ineficacia en el uso del espacio espectral.

La tecnología celular abordó el problema de cobertura utilizando un gran número de transmisores de baja potencia en pequeñas áreas de cobertura. Las grandes áreas, antes cubiertas con un solo transmisor, fueron divididas en pequeñas zonas denominadas celdas. Debido a la baja potencia de los transmisores es posible reutilizar la misma frecuencia (canal) en otras celdas. Sin embargo, la re-utilización de los canales se convierte en emisor de interferencia electromagnética entre las celdas cercanas afectando la calidad del enlace.

Debido a que en el servicio de comunicaciones móviles, el número de canales es establecido por el Estado, se hace necesario su uso eficiente puesto que es un recurso limitado. Esto significa que el proceso de asignación de canales a las llamadas realizadas en cada celda, se debe ejecutar de tal manera que se minimicen las interferencias electromagnéticas asegurando la calidad del servicio.

Teniendo en cuenta los anteriores aspectos se plantea un esquema en donde las interferencias se puedan modelar en función de la cantidad de restricciones electromagnéticas al asignar un canal a una determinada celda. El planteamiento de optimización nace al encontrar un esquema óptimo de asignación de canales con calidad en el enlace y mínima interferencia electromagnética. Satisfaciendo así la demanda de tráfico impuesta por la red. Según [1] este problema de optimización es de tipo NP-Completo, es decir, un problema de decisión que puede ser resuelto por una máquina no determinística en un tiempo polinómico [2].

En 1975 John H. Holland [3] propone que el proceso de selección natural sea usado para la solución de problemas complejos, basado en un método que imita la evolución biológica de las especies propuesta por Charles Darwin, desarrollando lo que hoy se conoce como algoritmos genéticos. Los algoritmos genéticos son un conjunto de operaciones de búsqueda basados en los mecanismos de selección natural que combinan la supervivencia de las secuencias mejores adaptadas con cambios aleatorios de información. Es decir, una familia de procedimientos de búsqueda adaptativos. La adaptación de los algoritmos genéticos a los problemas de optimización ha demostrado ser una buena alternativa con resultados satisfactorios. A lo largo de los años se han propuestos varios métodos de resolución, algunos de los cuales están basados en técnicas de búsqueda locales

Los desarrollos e investigaciones en operadores genéticos incluye el cálculo de la probabilidad de los operadores mediante un análisis entrópico Shannon [4], la utilización de un método basado en coincidencia parcial de cruce (PMX) [5] y sobre-cruzamiento de 3 puntos [6] como operador de cruce, mutación uniforme basada en reglas heurísticas [1], modelado de un GA como un proceso de decisión tipo Markov [7], selección aleatoria de individuos con probabilidad sesgada según su aptitud [8], además de la combinación de diferentes técnicas [9, 10], produciendo un GA híbrido.

Este trabajo de investigación explora cuatro diferentes formas de adaptación y configuración de algoritmos genéticos en la solución del problema de asignación de canales en redes celulares. Se implementa un

modelo de algoritmo genético simple y tres topologías de conexión entre seis islas, del algoritmo genético paralelo, para luego confrontar los resultados de las simulaciones con los “Benchmarck” Philadelphia P1 hasta P11.

Capítulo 1

Objetivo General

Implementar y evaluar una estrategia de optimización fundamentada en algoritmos genéticos paralelos tipo “islas” para la resolución de CAP en redes celulares, utilizando un tipo de reproducción asexual simultánea con dos individuos.

1.1 Objetivos Específicos

- Implementar y comprobar un algoritmo genético simple para solucionar el problema de asignación de canales.
- Implementar un tipo de reproducción asexual simultáneo con dos individuos.
- Implementar un algoritmo genético paralelo tipo “islas”, utilizando las topologías de red, tipo estrella y anillo.
- Evaluar la efectividad del algoritmo desarrollado con problemas “Benchmark” internacionales de problemas CAP.

Capítulo 2

Marco Teórico

2.1 Problema de asignación de canales (CAP) en redes celulares

La anatomía del problema de asignación de canales (CAP) en redes celulares está compuesta por un escenario de n celdas hexagonales estratégicamente ubicadas con el fin de brindar una mayor cobertura en determinadas zonas geográficas, limitando el uso de canales a las restricciones electromagnéticas. Si existe un uso simultáneo del mismo canal en celdas vecinas se produce una interferencia de tipo co-canal (CC), en cambio si el uso simultáneo del canal se produce dentro de la misma celda la interferencia es de tipo canal-adyacente (CA), también se debe considerar una distancia mínima de separación entre canales para evitar la interferencia de tipo co-sitio (CS). La compatibilidad en las restricciones electromagnéticas es representada por una matriz simétrica $C = [C_{ij}]_{n \times n}$ donde C_{ij} representa la distancia mínima de separación de canales entre la celda i y la celda j para evitar interferencias electromagnéticas. La demanda de servicio en las celdas es representada por un vector fila $D = [d_i]_{1 \times n}$ donde d_i representa la demanda de canales en la i -ésima celda. El resultado del problema CAP se muestra como una matriz $F = [f_{ik}]_{n \times k}$, donde k es el número total de canales asignados en la red.

La óptima asignación de canales, respetando las restricciones electromagnéticas, permite mejorar la capacidad del sistema de comunicación incrementando el número de abonados simultáneos. CAP es un problema de optimización combinatoria de tipo NP-Completo el cual no se puede resolver eficientemente con una máquina determinística en un tiempo polinómico.

Según [11] el problema de mínima asignación de frecuencias (MS-FAP) supone un conjunto F de frecuencias (canales) que contiene los canales de mayor z_{max} y menor z_{min} asignación. Así mismo, la diferencia entre los canales máximos y mínimos usados y la duración de la asignación del canal, determinan el costo y cuales canales deben reducirse al mínimo $min z_{max} - z_{min}$. Por cada celda $v \in V$ un subconjunto $F(v) \subseteq F$ de canales disponibles, se especifica un subconjunto $m(v)$ de canales asignados a v . Dos celdas v y w pueden tener mutua interferencia en al menos un par de canales de transmisión, $\{v, w\} \in E$. Para cada par de canales $f \in F(v)$ y $g \in F(w)$ se penaliza la asignación conjunta en función del nivel de interferencia. Esta sanción se denota por la matriz $p_{vw}(f, g)$, que depende de v, w y la distancia entre canales $|f - g|$. La matriz de penalización p_{vw} es utilizada en combinación con un valor máximo de p_{max} , con el fin de establecer un umbral y prohibir la asignación a canales que supere dicho valor. El modelado matemático del problema se presenta en las siguientes ecuaciones.

$$x_{vf} = \begin{cases} 1 & \text{si canal } f \in F(v) \text{ es asignado } v \in V \\ 0 & \text{cualquier otro caso} \end{cases} \quad (2.1)$$

$$y_f = \begin{cases} 1 & \text{si canal } f \in F(v) \text{ es usado} \\ 0 & \text{cualquier otro caso} \end{cases} \quad (2.2)$$

$$\sum_{f \in F(v)} x_{vf} = m(v) \quad \forall v \in V \quad (2.3)$$

$$x_{vf} + x_{wg} \leq 1 \quad \forall \{v, w\} \in E, f \in F(v), g \in F(w) : p_{vw}(f, g) > p_{max} \quad (2.4)$$

$$z_{max} \geq fy_f \quad \forall f \in F \quad (2.5)$$

$$z_{min} \leq fy_f + f_{max}(1 - y_f) \quad \forall f \in F \quad (2.6)$$

$$x_{vf} \leq y_f \quad \forall v \in V, f \in F(v) \quad (2.7)$$

$$z_{min}, z_{max} \in \mathbb{Z}_+ \quad (2.8)$$

donde $f_{max} = \max_{f \in D} f$ es el máximo canal disponible. Las restricciones de (2.5) y (2.6) garantizan que estas variables tengan el valor correcto. Así mismo las ecuaciones (2.3) y (2.4) están encargadas de satisfacer las restricciones electromagnéticas. En (2.6), (2.7) y (2.8) se verifica la correcta asignación de un canal según la matriz F .

Debido a la dificultad que representa la optimización de CAP, la mayoría de los trabajos de investigación se han basado en métodos heurísticos. Como en [12, 13] donde desarrolla un método denominado Algoritmo Voraz, el cual construye una asignación interactiva de canales en una celda de acuerdo a una regla basada en las características locales de optimización de la función objetivo. Es decir, este método organiza los vértices de las celdas de acuerdo a los parámetros de primer grado, de último grado y de forma aleatoria. El parámetro de primer grado organiza los vértices por un grado mayor, mientras que el parámetro de último grado organiza los vértices por un grado menor. Otro ejemplo es el método de Búsqueda Local [14, 15], en donde se inicia con una solución previamente determinada, posteriormente se hacen iteraciones con pequeños cambios bidireccionales en la solución, en busca de mejorar las soluciones previamente establecidas. Según [14, 15] la Búsqueda Local es uno de los métodos de heurísticos más básicos desarrollados para mejorar la solución en problemas de combinatoria. Por otro lado, la Búsqueda Tabú [16] permite aumentar las posibilidades de mejora de soluciones, puesto que las iteraciones con pequeños cambios unidireccionales en la solución evita caer en ciclos viciosos. Es decir, las soluciones seleccionadas en la última iteración k son declaradas "soluciones tabú" y no podrán ser seleccionadas nuevamente. De hecho, la comprobación del estado tabú de una solución puede requerir tiempo excesivo de cálculo.

El Simulado Recocido es otro método de búsqueda heurística con una probabilidad de búsqueda que depende del tamaño de la población y su disminución general en el tiempo. Es decir, la mejor solución se acepta como una nueva solución actual, si y solo si, es mejor que la anterior o con una probabilidad que depende de su valor. Dicha probabilidad varía de acuerdo a la diferencia entre el valor actual y el nuevo. Además, la probabilidad de aceptación es controlada por un parámetro, la temperatura. Este parámetro disminuye a medida que aumenta el número de iteraciones (enfriamiento) es decir, a bajas temperaturas la probabilidad de aceptación es inferior. Cuando la temperatura es muy baja y las soluciones no presentan mejoras se termina el algoritmo [11]. Otra técnica heurística son las Redes Neuronales Artificiales (RNA), las cuales generan nuevas soluciones mediante la emulación de la conducta de una red de neuronas, donde cada neurona representa una posible solución [11, 17]. En el contexto de las Redes Neuronales, las soluciones se generan por una red de neuronas, cuyos estados representan los valores de las variables que intervienen en el modelo. Con el fin de minimizar una función objetivo, las neuronas cambian su estado dinámicamente en función de los estados de las neuronas vecinas. Los ingredientes básicos de un algoritmo de RNA son la definición de neurona, el mapeo entre los estados neuronales y las soluciones.

La Optimización por Colonia de Hormigas [18], es una clase de algoritmo heurístico basado en comportamiento social de las hormigas, el cual tiene un número fijo de agentes, donde cada hormiga puede ser

interpretada como un algoritmo secuencial codicioso iterativo que genera una solución mediante la mejora de soluciones parciales. Un movimiento es controlado por parámetros basados en la estructura del problema y el nivel de rastro de feromonas, que tiene en cuenta cuántas veces un paso ha sido exitoso. El rastro de feromonas se actualiza cuando todas las hormigas han completado su función de construir la solución final. Finalmente, el método de los Algoritmos Genéticos (GA) se inspiran en la teoría de la evolución de las especies propuesta por Charles Darwin. La metodología de estos algoritmos se basa en realizar un conjunto de operaciones con el fin de emular los pasos necesarios para crear nuevas descendencias, convirtiéndose en un método heurístico interesante en el desarrollo de soluciones del problema de asignación de canales en redes celulares. Este artículo se centrará en el estudio de GA como método de solución de problemas de CAP en redes celulares.

2.2 Algoritmos Genéticos (GA) aplicados a CAP en redes celulares

La estructura de los GA están compuestos por los individuos, una población, la función objetivo y varios operadores genéticos. Los individuos representan las posibles soluciones al problema modelado. La población está compuesta por individuos debidamente caracterizados, así mismo la función objetivo son un conjunto de reglas que determinan la calidad de cada individuo y los operadores genéticos (selección, cruce, mutación y re-inserción) modelan el proceso reproductivo de los organismos. Por último, los criterios de finalización determinan el fin del proceso antes descrito. En conjunto todos los componentes de GA modelan el proceso biológico evolutivo, como se muestra en la figura 2.1.

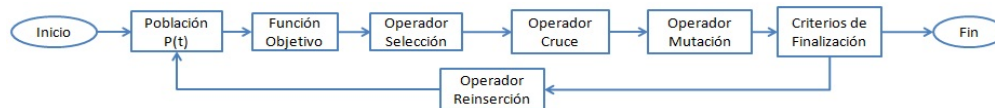


Figura 2.1: Diagrama de Bolques GA-Simple

El diagrama en bloques de un GA-Simple inicia con la generación de la población inicial $P(t := 0)$, luego el operador de selección, previa evaluación de la función objetivo, toma dos individuos de ciertas características para enviarlos al operador de cruce, quien determina cuales genes y como deben cruzarse los individuos para crear una nueva descendencia. El operador de mutación es el encargado de cambiar ciertos genes de cada individuo con el fin de producir diversidad genética en la población $P(t)$. Una vez obtenido los candidatos a la nueva generación, el operador de re-inserción determina los descendientes que serán re-insertados en $P(t)$ para conformar $P(t + 1)$. Por cada iteración del ciclo evolutivo, se hace un análisis del proceso ejecutado y según los criterios de finalización, culmina el proceso del GA-Simple. La implementación de GA ha mostrado ser una alternativa en la resolución de problemas de optimización tipo NP-Completo [19]. A continuación se detalla los componentes del diagrama de bloques en el contexto de solución del problema CAP.

2.2.1 Individuos

Los individuos están compuestos por genotipos y fenotipos [20, 21]. Los genotipos representan las características genéticas de los individuos, mientras que los fenotipos representan las características ya codificadas. El individuo está caracterizado por sus cromosomas, los cuales son una estructura de un grupo de genes. Así mismo, los genes están conformados por alelos y locus que simbolizan las características propias de cada individuo. Los alelos corresponden al valor del objeto del gen y el locus a la posición del alelo en una cadena dentro de un gen. En la figura 2.2 se muestra una posible representación de un individuo, donde en el Gen 1 locus 3 existe el alelo 9.

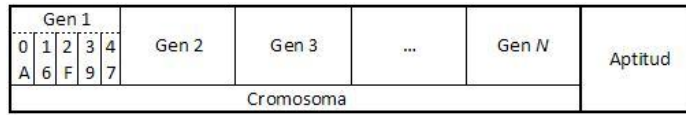


Figura 2.2: Representación de un Individuo

Los individuos pueden tener diferentes tipos de representaciones de acuerdo al planteamiento del problema a modelar. En CAP predomina la utilización de dos tipos de representaciones, detalladas a continuación. En la representación Binaria los genes son representados por "0" ó "1", según sea el autor la codificación binaria representa canales libres, ocupados o posibles incompatibilidades electromagnéticas entre canales. En [4, 6] se utiliza la codificación binaria para la representación genética de individuos con resultados favorables en la convergencia del algoritmo. Según los autores consultados en CAP, la ventaja fundamental de la representación binaria radica en el poco consumo de recursos computacionales al momento de procesar la información. En la representación Entera cada gen es simbolizado por un objeto $a \mid a \in \mathbb{Z}$, que puede representar canales libres, ocupados o posibles incompatibilidades electromagnéticas. En [9, 22, 23] se utiliza la codificación entera con miras a la simplificación de la validación de las restricciones electromagnéticas. Los resultados condujeron a la simplificación del préstamo de canales en celdas adyacentes y la transferencia del uso del canal a otra celda sin perder la conexión con el abonado, proceso denominado "Hand-Off". Por otro lado, en [20] se describen tres tipos de representaciones que hasta donde saben los autores no se encontró aplicación al problema de CAP. En la representación Real cada gen es caracterizado por un número $b \mid b \in \mathbb{R}$. En la representación Hexadecimal, los genotipos son simbolizados por elementos pertenecientes al conjunto de números hexadecimales. Por último, se encuentra la representación Octal, donde cada gen se simboliza con $0 \leq h \leq 7 \mid h \in \mathbb{Z}_+$.

2.2.2 Población

La población $P(t)$ es un conjunto de individuos, los cuales son afectados por los operadores genéticos con el fin de crear una nueva generación. Según [20] existen dos aspectos importantes en $P(t)$. El primer aspecto es la creación de la población inicial $P(0)$. Ésta generalmente es creada aleatoriamente, garantizando un conjunto heterogéneo de individuos. Ahora, si el objetivo es crear un conjunto homogéneo de individuos, estos deberán ser creados con un mecanismo no aleatorio. Sin embargo, en algunas poblaciones homogéneas se puede convertir en una desventaja debido a que el resultado podría estar anidado en los óptimos locales de la solución final [24]. Por lo general en CAP se hace una inicialización aleatoria de la población.

El segundo aspecto es el tamaño del conjunto $P(t)$. Una población pequeña corre el riesgo de no cubrir adecuadamente el espacio de búsqueda del conjunto del problema modelado, mientras que operar con poblaciones extremadamente grandes puede acarrear problemas relacionados con el excesivo costo computacional en la ejecución del algoritmo [24]. Por lo tanto, es recomendable hacer un análisis de la relación costo/beneficio al momento de decidir el tamaño del conjunto $P(t)$.

Según [5] en CAP es válido aplicar la ley de rendimiento decreciente, por ello la mayoría de los autores utilizan poblaciones iniciales no mayores a 150 individuos, garantizando una convergencia razonable del algoritmo. Un claro ejemplo de la aplicación de la ley se describe en [25], donde hacen una comparación entre dos tipos de GA aplicados al mismo problema. El primer algoritmo basado en la asignación dinámica de canales con $P(0) = 50$ individuos, encuentra una solución óptima a partir de la 100-ésima generación. El segundo algoritmo basado en el préstamo de canales entre celdas adyacentes con $P(0) = 10$ individuos, encuentra una solución óptima a partir de la 30-ésima generación.

2.2.3 Función Objetivo

La función objetivo califica y cuantifica las aptitudes y restricciones de cada individuo ó generación en particular. Los métodos más utilizados para la resolución de CAP descritos en [20] varían de acuerdo al modelado del problema. Entre las funciones más destacadas se encuentra la función Absolutista utilizada por

la mayoría de los autores en CAP, donde el proceso de evaluación de los individuos que desaprovechan las restricciones no serán considerados como tales, y el algoritmo continúa con los procesos evolutivos hasta obtener los individuos válidos. Es decir los individuos no válidos serán descartados. La función absolutista tiene la particularidad de operar exclusivamente con los individuos mejores adaptados ayudando a la convergencia del algoritmo. Sin embargo al descartar individuos menos aptos, se puede desaprovechar información genética valiosa que contribuya a la óptima solución. Diferentes estrategias han sido evaluadas, tales como en [23] que utilizan la función Reconstructiva, donde cada vez que se encuentra un individuo no apto, se reconstruye hasta convertirlo en un individuo válido. El costo computacional de la función reconstructiva es alto y de lenta convergencia, por ende los algoritmos que utilizan la función reconstructiva por lo general manejan poblaciones de hasta 50 individuos, tales como [4, 6, 8].

Otro ejemplo es [6] donde utilizan la Penalización de la función objetivo para cuantificar y calificar las aptitudes de cada individuo. Este método consiste en dividir la función objetivo del individuo por una cantidad (la penalización) que guarda relación con las restricciones que el individuo viola. Dicha cantidad puede simplemente tener en cuenta el número de restricciones violadas ó bien el denominado costo esperado de reconstrucción, es decir el coste asociado a la conversión de dicho individuo en otro que no viole ninguna restricción. Los resultados reportados por los autores al ejecutar la penalización de la función objetivo muestran una distribución más eficiente de los canales asignados, con la particularidad de permitir el préstamo de canales en celdas adyacentes, tal y como lo hacen en [6, 26].

2.2.4 Operador de Selección

Según la teoría de la evolución propuesta por Darwin, el proceso de selección en la naturaleza está basado en la supervivencia del individuo más fuerte y mejor adaptado al entorno. El operador de selección es el encargado de seleccionar aquellos individuos, que según sus características se consideren mejor adaptados, con el fin de diseminar sus genes en pro de su evolución. Existen diversas técnicas de selección, una de las primeras técnicas propuestas por [3] fue la selección aleatoria, donde los padres son seleccionados al azar de la población de individuos. En [27] se define el proceso de selección de acuerdo a ciertos métodos y parámetros previamente establecidos según el problema a modelar. El principio de selección del método de la Ruleta se basa en una búsqueda lineal proporcional al ancho de la ranura de la rueda, que pondera en proporción el valor de aptitud de cada individuo. Es decir, los individuos mejores adaptados tendrán mayor probabilidad de ser seleccionados. El valor esperado de un individuo es su aptitud dividida sobre la aptitud promedio de la población, donde a cada individuo se le asigna un trozo en la rueda de la ruleta, el tamaño de la rebanada es proporcional a la aptitud del individuo. La rueda se hace girar N veces, donde N es el número total de individuos de la población. En cada giro, en virtud del marcador de la rueda, el individuo es seleccionado para continuar con el proceso evolutivo. El método de la rueda de la ruleta es el más utilizado en el desarrollo de GA en CAP, ya que hace un barrido de toda la población permitiendo tomar los individuos con mayores probabilidades y aptitudes de ser seleccionados [5, 8, 25].

Como otra estrategia, en [28] utilizan la probabilidad de selección del método Ranking, el cual se basa en la clasificación de individuos en orden de importancia de la aptitud, es decir según su rango y no de su aptitud absoluta. Los individuos son organizados según el rango de forma descendente, la lista es encabezada por los individuos de mejor rango hasta llegar al individuo con peor rango. Este método selecciona los 2 mejores individuos del ranking. La particularidad del método de selección ranking está en listar los individuos, haciendo posible un posterior análisis en el bloqueo de llamadas. En [1] se utiliza la estrategia de selección del método Torneo, la cual permite hacer un barrido por toda la población y tomar lo mejor de ciertos subconjuntos, tal y como sucede en ambientes naturales reales. El torneo inicia con la creación aleatoria de dos subconjuntos contenidos en la población general, los individuos pertenecientes al subconjunto compiten entre sí para determinar quién tiene mejores aptitudes. El individuo ganador del primer subconjunto será cruzado con el individuo ganador del segundo subconjunto. Con este método se asegura la exploración total de la población. En [4] utilizan el método Elitista, donde se selecciona al individuo si el valor de aptitud de este en previas poblaciones $P(t-1)$ es mejor a los individuos en la población actual $P(t)$, para preservar al

individuo con mejor aptitud e incluirlo en futuras poblaciones $P(t + 1)$. La gran ventaja del método elitista radica en preservar sólo lo mejor de la población para poder utilizarlo en futuras generaciones, pero esa ventaja puede traer problemas genéticos tales como endogamia y malformaciones en los individuos, dando como resultado un desmejoramiento en la solución óptima. Si el elitismo se combina con los anteriores métodos de selección produce la ruleta de selección elitista, selección de torneo elitista y la selección de ranking elitista, convirtiéndolos en operadores híbridos.

2.2.5 Operador de Cruce

Una de las características distintiva de los GA es el uso del operador de cruce, que modela el apareamiento entre individuos. La función principal de este operador es la combinación de información genética proveniente de diferentes individuos previamente seleccionados de la población. El resultado del cruce genera una nueva descendencia con rasgos característicos de cada padre.

Holland en [3] explica el procedimiento básico del operador de cruce como la selección de una o varias ubicaciones de cadena en cada individuo. Estos puntos de cruce delimita el segmento de intercambio. Luego los segmentos se intercambian para combinar el material genético de cada padre, produciendo un par de hijos pertenecientes a la nueva generación. En CAP son muchos los métodos que se utilizan para modelar el apareamiento de individuos, quizás el más popular es el método de cruce Puntal. En [29] lo ejecutan teniendo en cuenta la posiciones elegidas previamente al azar, las sacadas son intercambiadas con las posiciones de cadena seleccionadas del segundo padre para crear al primer hijo. Ahora, para crear al segundo hijo, se toma las posiciones restantes de ambos padres para formarlo. Este tipo de cruce se utiliza cuando se quiere preservar rasgos específicos de los individuos. En [30] utilizan el cruce puntual con resultados favorables en la convergencia del algoritmo y reducción en las restricciones. En la figura 2.3a se muestra un cruce puntual.

En [3, 6, 8, 9, 25], se utiliza el método de Cruce de n puntos, donde se selecciona $n \mid n \in \mathbb{Z}_+$ posiciones de cadena en cada individuo progenitor para ser intercambiados y combinados entre los individuos y así generar la nueva descendencia. La mayoría de los autores de CAP coinciden en afirmar que el cruce de n puntos tiene una incidencia favorable en la preservación de cromosomas de individuos. En la figura 2.3b se hace una representación de un cruce con $n = 3$ puntos. La descripción realizada en [24, 31] del método de cruce Uniforme comienza con la creación de un vector binario denominado vector máscara, de igual longitud al tamaño de la cadena de los padres. Los genes de cada padre son copiados a cada hijo en función del vector máscara respetando su locus. Para la creación del primer hijo, se coloca el vector máscara debajo de la cadena del primer padre y mientras exista el valor de "1" lógico se transmiten los genes del primer padre, los espacios libres restantes son llenados con los genes del segundo padre, respetando su locus. Un procedimiento similar es utilizado para la generación del segundo hijo, se coloca el vector máscara debajo de la cadena del segundo padre y mientras exista el valor de "1" lógico se transmiten los genes del segundo padre, los espacios libres restantes son llenados con los genes del primer padre, respetando su locus. Otro tipo de cruce según los autores [24, 32], es el uniforme, donde los genes son transmitidos de acuerdo a una cadena generada mediante un proceso aleatorio. Aunque también cabe decir que en CAP la herencia aleatoria de genes puede crear incompatibilidades electromagnéticas entre canales adyacentes. En [22] ejecutan un cruce uniforme, tal como lo muestra la figura 2.3d. Los hijos están conformados por los genes de los padres según la posición del "1" lógico en el vector máscara. Si el gen que se hereda del padre ya se encuentra en la cadena del hijo, entonces se toma el gen de la siguiente posición y así hasta completar la nueva cadena del hijo. Otro caso es el método de cruce de Mapeo Parcial (PMX), donde según [31] es de vital importancia el locus de los genotipos. Al igual que en el cruce Uniforme, se debe generar un vector máscara. Para la creación del primer hijo, se coloca el vector máscara debajo de la cadena del primer padre y mientras exista el valor de "1" lógico se transmiten los genes del primer padre, pero si existe el valor de "0" lógico se transmiten los genes del segundo padre en estricto orden según locus. Un procedimiento similar es utilizado para la generación del segundo hijo, se coloca el vector máscara debajo de la cadena del segundo padre y mientras exista el valor de "1" lógico se transmite los genes del segundo padre, pero si existe el valor de "0" lógico se transmiten los genes del primer padre en estricto orden según locus. Este método de cruce es válido

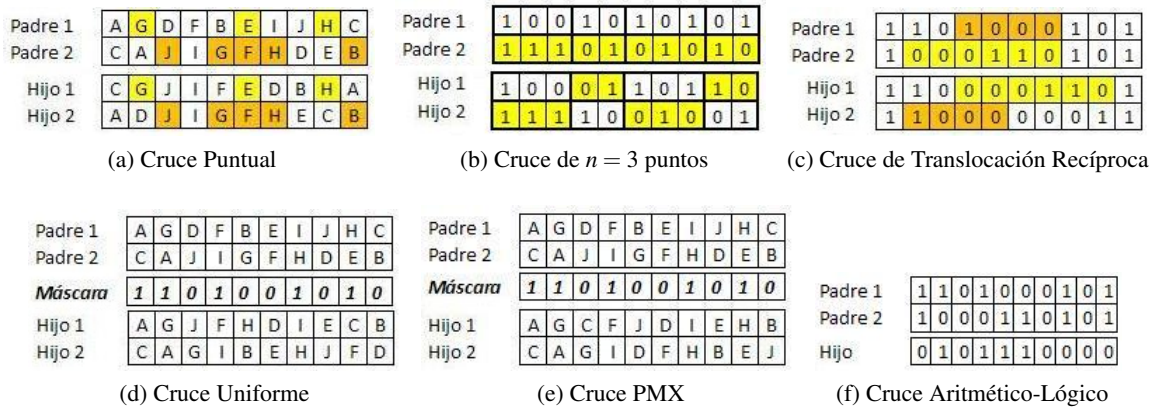


Figura 2.3: Operadores de Cruce

para cierta representación de individuos. En CAP, los autores [5] registraron una notable mejora en la calidad de la descendencia con la aplicación del cruce PMX, esto debido a que los hijos tienden a heredar lo mejor de cada padre. La figura 2.3e muestra un ejemplo del proceso [1]. Otro método de cruce utilizado en CAP es el Aritmético-Lógico, el cual ejecuta una operación aritmético-lógica entre padres. En [33] utilizaron una operación lógica tipo XOR entre padres, obteniendo resultados favorables en la aptitud de los individuos generados. En [15] utilizaron una operación de adición entre los genes de los padres, con resultados adversos en la aptitud de los nuevos individuos, atribuido al hecho de que el resultado de la suma puede tener una longitud mayor a la de los padres, variando el resultado final del cruce. En la figura 2.3f se muestra el cruce entre dos padres mediante una operación XOR. Quizás unos de los métodos menos utilizados en la literatura de CAP es el cruce de Traslocación Recíproca. En [10] utilizan el método de Traslocación Recíproca como operador de cruce. Este método inicia con la selección al azar de un segmento de cadena diferente por cada padre. Los segmentos pueden ser de tamaños diferentes. Para generar el primer hijo, se copia los genes del primer padre en estricto orden hasta llegar al punto del segmento seleccionado, en ese punto se copia el segmento del segundo padre. En los locus restantes se copia la información genética del primer padre. El proceso de la generación del segundo hijo inicia con la copia de los genes del segundo padre en estricto orden hasta llegar al punto del segmento seleccionado, en ese punto se copia el segmento del primer padre. Los locus restantes son llenados con la información genética del segundo padre. Según los autores de CAP la principal ventaja del método radica en que al seleccionar segmentos de genes se garantiza la herencia de características propias de cada individuo, acelerando considerablemente la convergencia del algoritmo. En la figura 2.3c se muestra un ejemplo del cruce de Traslocación Recíproca.

2.2.6 Operador de Mutación

Parte fundamental de la evolución es la adaptación de las especies a los cambios naturales de su entorno. En GA el operador de mutación es el encargado de hacer cambios con el fin de introducir diversidad en los individuos, produciendo nuevas soluciones e incrementando la variabilidad dentro de la población. En CAP se maneja una baja probabilidad de mutación debido a que los cambios producto de la mutación pueden generar nuevas interferencias electromagnéticas.

En [25] describen la mutación puntual como el cambio cuando cada gen tomado al azar muta independientemente del resto de genes. Este tipo de mutación es quizás la más utilizada en CAP, pues se puede controlar el gen a mutar, ayudando significativamente en el mejoramiento de la aptitud del individuo mutado. Por otra parte, la mutación inversiva del valor del alelo descrita en [26], cambia el valor del gen seleccionado por su correspondiente recíproco. Según los autores en CAP la mutación inversiva puede generar interferencia entre canales adyacentes. En la figura 2.4c se muestra un ejemplo de la mutación inversiva. Aun lo anterior, diversos esquemas de mutación existen. Por ejemplo, el método de mutación inversa [34], selec-

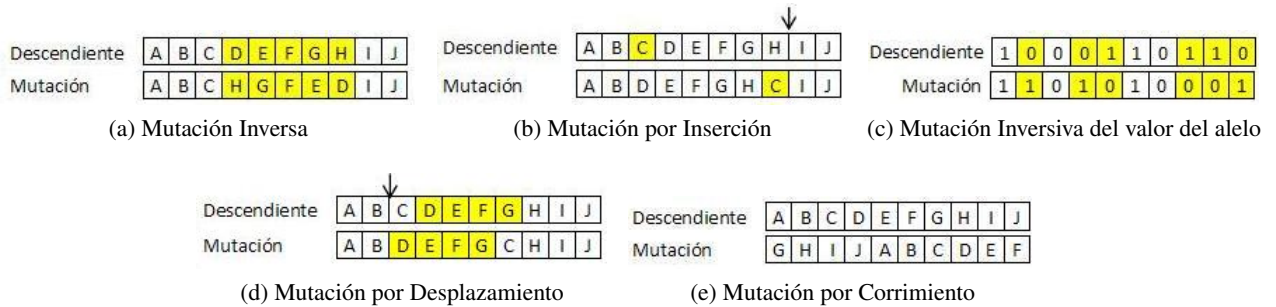


Figura 2.4: Operadores de Mutación

ción a dos posiciones al azar de un cromosoma invirtiendo la sub-cadena entre las posiciones, tal y como se muestra en la figura 2.4a. Los resultados de la aplicación del método de mutación inversiva tienen una gran variedad entre los autores de CAP, mientras [35, 36] afirman que el resultado tiene una incidencia enorme en la calidad de la aptitud de los individuos, [37, 38] afirman que tiene una baja incidencia sobre los individuos mutados. En [23] utilizan ejecutan una mutación por inserción, donde se selecciona un gen para insertarlo en una posición aleatoria, como lo muestra la figura 2.4b. La mutación por inserción es quizás uno de los métodos más populares en CAP, debido a la ventaja de poder reorganizar la cadena de un individuo con el fin de mejorar la aptitud del individuo mutado. En [32] se define la mutación por desplazamiento cuando se selecciona al azar una sub-cadena de genes para luego insertarlos en una posición aleatoria, ver figura 2.4d. En aplicaciones de CAP, el tamaño de la sub-cadena afecta la calidad del individuo, por lo tanto la mutación por inserción puede ser vista como un caso especial de la mutación por desplazamiento. Por último y quizás el menos utilizado, está el método de mutación por corrimiento [39]. Aquí los objetos avanzan posiciones dentro de la cadena en un sentido previamente definido. Según los autores, la gran desventaja del método radica en que la mutación modifica toda la cadena de genes del individuo, afectando negativamente la calidad de la aptitud del individuo mutado. En la figura 2.4e se ilustra por medio de un corrimiento de 4 posiciones hacia la derecha.

2.2.7 Operador de Re-inserción

En la evolución humana, los hijos tenderán a reemplazar a sus padres y así continuar con su legado. Esta condición no es ajena en los GA's, el operador de re-inserción tiene la tarea de tomar aquellos individuos con ciertas características y re-insertarlos en la población con el fin de continuar con el legado genético dejado por sus padres. Es decir, el operador de re-inserción se encarga de conformar la $P(t+1)$ población. Existen muchas estrategias para introducir la nueva descendencia en la población, pero quizás las 3 estrategias más utilizadas en CAP basan su proceso de inserción en una condición evolutiva.

La primera estrategia y quizás una de las más utilizadas por los autores en CAP es el reemplazo de la población de dos en dos [20]. El proceso se inicia con la selección de una pareja de padres, los cuales generan un par de descendientes. Los descendientes ingresan en la población para reemplazar a dos individuos con el peor valor de aptitud. Esta estrategia permite interactuar con todos los individuos de la población. En [27] describen la segunda estrategia como un reemplazo de toda la población. La estrategia inicia con la selección de $\frac{u}{2}$ parejas de padre, donde u es igual al tamaño de la población. Luego se crean u hijos, los cuales reemplazarán a toda la población. Según [40] dicha estrategia tiene un impacto favorable en la convergencia de los algoritmos genéticos en CAP, debido a la simplicidad del proceso de inserción. Por último, la tercera estrategia es una combinación de las anteriores. Según [41] los mejores individuos (máximo el 3% de la población) son preservados para la población $P(t+1)$, los demás individuos son creados con una de las dos estrategias anteriores. Esta estrategia es utilizada en CAP cuando se está trabajando con algoritmos diseñados bajo el concepto de elitismo. Los autores en CAP muestran el desempeño y convergencia del elitismo, pero también comentan el posible sesgo en las soluciones.

2.2.8 Criterios de Finalización

Al ser los algoritmos genéticos una familia de procedimientos de búsqueda adaptativos, se hace necesario definir criterios de finalización para evitar soluciones anidadas en los óptimos locales y converger a la óptima solución del problema modelado. En [27, 42, 43] describen las condiciones operacionales de los criterios de finalización de un GA, entre las que se destacan el máximo de generaciones, tiempo transcurrido, cambio generacional, intervalo generacional e intervalo de tiempo. Para el caso del cambio generacional, este finalizará el proceso genético si no hay un cambio en la aptitud media de la población durante un número de $s \mid s \in \mathbb{Z}_+$ generaciones. El intervalo generacional concluirá el algoritmo si no hay cambios significativos entre generaciones para una secuencia de $q \mid q \in \mathbb{Z}_+$ generaciones sucesivas. A diferencia del intervalo de tiempo finalizará el algoritmo si no hay cambios significativos entre generaciones durante un intervalo de tiempo.

Según la mayoría de los autores de GA en CAP la condición de intervalo generacional es la más utilizada, debido a lo simple de su implementación y efectiva en su proceso. Sin embargo el escoger una condición para los criterios de finalización no es suficiente, [20, 41] dicen que la combinación de uno o más criterios de finalización es la clave para estar seguros que la solución obtenida no está anidada en los posibles óptimos locales de la función.

2.3 GA con Operadores Sofisticados

Desde que Holland en 1975 desarrolló el GA, se han hecho diferentes tipos de investigaciones con el fin de crear un modelo que imite de manera más apropiada la evolución de las especies. Según las investigaciones hechas alrededor de GA, se sugiere agrupar las diferentes técnicas desarrolladas en dos grupos: Algoritmos Genéticos Simples y Algoritmos Genéticos Complejos. El primer grupo estaría compuesto por los estudios e investigaciones realizadas en torno a la estructura del algoritmo genético simple y todas sus posibles variantes. Para el segundo grupo, Algoritmos Genéticos Complejos (GA-Complejos), se han introducido operadores sofisticados y novedosas estrategias de búsqueda al esquema del algoritmo. La mayoría de los GA-Complejos no han sido aplicados en CAP y se presentan aquí con el objetivo de servir de referencia para futuras aplicaciones de este tipo de optimización. A continuación se expondrán los operadores y estrategias que integran los diferentes tipos de GA-Complejos.

2.3.1 Operador Ciclo de Vida

Documentado por [24], el algoritmo tiene la capacidad de modelar el ciclo de vida humana. Este operador cumple la función de asignarle un parámetro de vida a cada individuo, el cual está asociado a una cierta cantidad de generaciones ó tiempo de ejecución del algoritmo. Una vez cumplido con el parámetro, el operador de vida/muerte se encarga de eliminar (dar muerte) al individuo. Es decir, el periodo de vida de un individuo dependerá exclusivamente del operador de vida/muerte. La asignación del parámetro de vida/muerte se hace mediante un proceso aleatorio en el momento que el individuo es creado. En la figura 2.5 se muestra el esquema del funcionamiento general de un GA-Complejo basado en el operador Ciclo de vida.



Figura 2.5: Operador Ciclo de Vida

2.3.2 Operador Beeder

Un concepto similar al Operador Ciclo de vida se explica en [29, 42], donde se ejecutó un GA-Complejo llamado Beeder. Este algoritmo está basado en el modelado del ciclo de vida en un criadero. A diferencia

del Operador Ciclo de vida, en este algoritmo predomina el valor de aptitud de un individuo. Es decir, el parámetro de vida/muerte está directamente relacionado con el valor de aptitud. A mayor aptitud, mayor será la longevidad del individuo. En este caso un individuo extremadamente apto podrá perdurar en la población casi que de manera indefinida.

2.3.3 Estrategia Memética

En [44] se explica la estrategia de GA-Complejo denominada Memética. En la estrategia Memética los individuos reciben el nombre de “agentes”, debido a que ejecutan un comportamiento activo dentro del proceso evolutivo. Estos agentes son capaces de interactuar con otros agentes adyacentes. La interacción se centra en el intercambio de información propia. Es decir, el agente entrega y recibe información relacionada con su aptitud y la de sus vecinos. En definitiva, se puede construir un mapa general de toda la población con la información recolectada por los agentes en su entorno.

En la estrategia Memética se utilizan los operadores tradicionales de GA-Simple, con la variante de que el operador de selección también puede seleccionar al agente menos adaptado para que sea eliminado por el operador de re-inserción. Dicha eliminación se hace con el fin de generar los espacios necesarios para el ingreso de los nuevos agentes, ya que el tamaño de la población es constante. Los agentes mejores adaptados son los encargados de transmitir la información recolectada de su entorno. El operador de cruce intercambiará información genéticas entre agentes, donde los hijos resultantes heredarán la información del entorno proveniente de sus padres. El operador mutación cumple el rol de inyectar nueva información genética en la población de manera continua para no degradar al algoritmo en una simple búsqueda aleatoria. Una estrategia Memética puede verse como una colección de agentes que realizan una exploración autónoma del espacio de búsqueda. Este algoritmo no introduce un operador específico como tal, simplemente cambia el concepto y operatividad de los operadores convencionales.

2.3.4 Estrategia CHC

En [20] explican una estrategia encargada de evitar el incesto entre individuos, denominada Estrategia CHC. A diferencia del proceso del GA-Simple, en la estrategia CHC existe un operador de cruce heterogéneo y un operador de mutación cataclísmica. El operador de cruce heterogéneo solo intercambia la mitad de la información genética entre los padres que crearán a los hijos. Estos hijos tendrán una distancia equidistante entre sí y sus padres, convirtiéndolos en individuos con algún tipo de relación genética entre sí. Si el algoritmo llega a un punto en que todos los individuos están genéticamente relacionados entre sí. Entonces, entra en operación el operador de mutación cataclísmica. Este operador realiza un mutación de casi toda la información genética contenida en el individuo, con el fin de generar diversidad genética en la población. La estrategia CHC posee la particularidad de que todos los individuos tienen la misma probabilidad de ser seleccionados. Pero, si dos individuos seleccionados están genéticamente relacionados, se descarta aleatoriamente uno de ellos y se selecciona otro individuo. Este algoritmo basa su funcionamiento en una estrategia de mantenimiento de la diversidad genética, mediante la prohibición del incesto.

2.3.5 Estrategia de Relación Genética

El concepto de la relación genética explicado en [29], tiene el fin de identificar la relación existente entre individuos que tengan algún ancestro en común. Este GA-Complejo denominado Estrategia de Relación Genética, modifica la estructura básica de un individuo introduciendo una variable denominada LineOpt.

LineOpt es la línea ascendente de cada individuo, donde se almacena la información del valor de aptitud de los padres y sus respectivas ascendencias. Entonces, una vez que el operador de selección escoge los posibles padres, el operador de cruce analiza el LineOpt de ambos individuos. Si dos individuos tienen LineOpt con condiciones de aptitud idénticas, entonces se escoge aleatoriamente uno de los individuos y el otro se descarta. Pero si los individuos tienen condiciones semejantes de LineOpt, entonces, se escoge al ancestro más cercano a la generación en curso y se descarta el otro individuo. El proceso general de operación

de la Estrategia de Relación Genética es igual al GA-Simple, con la particularidad de que la Estrategia de Relación Genética basa su procedimiento en evitar problemas de endogamia entre individuos, mediante la prohibición del apareamiento entre individuos genéticamente relacionados.

2.3.6 Estrategia de Evolución Gramatical

En [45] se explica la Estrategia de Evolución Gramatical, como un GA-Complejo con una población de individuos con cadenas de longitud variable. El enfoque de esta estrategia radica en la optimización de la información genética mediante la duplicación del material genético importante y eliminación de la información genética irrelevante, según una regla gramatical. La regla gramatical es un conjunto de parámetros que determinan que información genética es importante, según el problema a modelar. Es decir, cuántos y cuáles genes de un individuo se deben eliminar o reemplazar por nueva información genética.

Este algoritmo inicia como un GA-Simple, pero introduce dos nuevos operadores. El operador de duplicidad, selecciona y copia los mejores genes de cada individuo según la regla gramatical. Las copias de los genes se colocan a continuación del último gen del individuo. El operador de poda elimina todos los genes que no se utilizan, previa correspondencia entre el gen y la regla gramatical. En figura 2.6 se muestra un diagrama de bloques de la estrategia de evolución gramatical.



Figura 2.6: Estrategia de Evolución Gramatical

2.3.7 Estrategia Lógica Difusa

En [31, 46] explican un GA-Complejo basado en lógica difusa, denominado Estrategia Lógica Difusa. Este algoritmo tiene los operadores tradicionales de GA-Simple, con la particularidad de que en esta estrategia todo su funcionamiento se basa en operadores difusos [47]. Es decir, la representación basada en operadores difusos permite establecer unas condiciones en donde la probabilidad de adaptación está ligada a cierta incertidumbre de su valor. Es decir, para la función objetivo sobreviven los individuos que tienden a ser mejor adaptados. Los individuos tienen valores ponderados similares de ser seleccionados, el operador de cruce es flexible en el intercambio de información genética, el operador de mutación es escalable, los criterios de finalización y el operador de re-inserción operan bajo la lógica difusa. En definitiva, individuos con condiciones similares de aptitud pueden tener la misma capacidad de sobrevivir.

2.3.8 Estrategia en Paralelo

Un algoritmo es paralelo si puede ejecutar varios procesos simultáneos en un mismo intervalo de tiempo. Para el desarrollo de algoritmos en paralelo eficientes hay que poder crear, destruir, especificar e interactuar entre los procesos. En [48] explican un GA-Complejo con los atributos antes mencionados, denominado GA-Paralelo. En esencia los GA-Paralelo operan como un GA-Simple con ciertos atributos especiales. Los GA-Paralelo están clasificados según el tipo de relación y/o comunicación entre los diferentes procesos ejecutados simultáneamente.

2.3.8.1 Estrategia en Paralelo tipo Grano

Los GA-Paralelo tipo Grano comienzan su ejecución tal y como lo haría un GA-Simple, con la particularidad de que los diferentes procesos simultáneos comparten la población de cada algoritmo según el tipo de grano [49].

- GA-Paralelo de Grano fino: Utiliza una única población centralizada controlada por el procesador maestro, el mismo que se encarga de la selección y asignación de cada uno de los esclavos, quienes reciben uno a uno varios individuos a los cuales evalúan, cruzan y mutan para finalmente devolver los resultados al maestro.
- GA-Paralelo de Grano medio: Cada individuo forma parte de varias sub-poblaciones, solapadas entre sí, que determinan la adyacencia entre individuos de acuerdo a una topología de conexión. La alta conectividad entre vecinos incrementa la difusión de los individuos mejor adaptados.
- GA-Paralelo de Grano grueso: Este modelo maneja los individuos en poblaciones independientes. Cada proceso determina cómo y cuando se realiza el intercambio de individuos. El intercambio de individuos entre poblaciones está regido según los requerimientos del problema modelado.

2.3.8.2 Estrategia Masivamente Paralelo

En [20, 50] se desarrolla una estrategia en paralelo denominada Masivamente paralelo. Esta estrategia basa su funcionamiento, en poblaciones estructuradas con solapamiento entre si a fin de compartir material genético entre poblaciones. Esta estrategia permite una transferencia gradual de información genética sin la introducción súbita de cromosomas. En esencia, este algoritmo desarrolla su proceso tal y como lo hace un GA-Simple la diferencia se encuentra en el tratamiento de la población.

2.3.8.3 Estrategia en Paralelo tipo Islas

La estrategia en paralelo tipo Islas se basa en la estructura espacial distribuida de la población. La población se divide en subpoblaciones disjuntas donde en cada una de ellas se ejecuta en forma paralela un GA-Simple con intercambios regulares de individuos entre estas sub-poblaciones según una cantidad de generaciones preestablecida y algunos parámetros determinados [50, 51]. El operador de migración es el encargado de enviar y recibir los individuos de cada isla, ver figura 2.7.

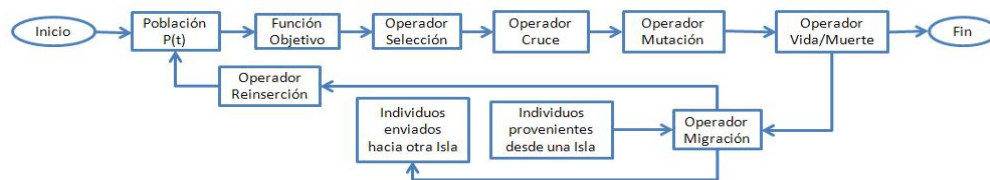


Figura 2.7: Estrategia en Paralelo tipo Islas

El enfoque de islas de esta estrategia permite tener varias opciones de configuración en la comunicación de las islas. Las 3 configuraciones más comunes son:

- Comunicación en estrella: Existe una sub-población que es seleccionada como maestra (aquella que tiene mejor media en el valor de la función objetivo), las demás sub-poblaciones son consideradas esclavas. Las sub-poblaciones esclavas migran sus mejores individuos a la sub-población maestra, la cual ejecuta un proceso igual al de un GA-Simple. El proceso del GA ejecutado en la sub-población maestra es el que determina cuales y hacia donde migran los individuos de las sub-poblaciones esclavas.
- Comunicación en red: No existe una jerarquía entre las sub-poblaciones, por lo tanto los individuos con mejor valor de aptitud migran desde y hacia cada una de las isla. Este tipo de comunicación realiza un intercambio con todas las islas adyacentes.
- Comunicación en anillo: Cada sub-población envía sus mejores individuos a una isla vecina, efectuándose la migración en un único sentido.

2.3.8.4 Estrategia en Paralelo tipo Grillas

El modelo de Grillas (modelo celular) basa su trabajo en individuos distribuidos sobre una malla bidimensional, un individuo por cada celda. La interacción genética está restringida a los vecinos “ceranos” de cada individuo [52]. La diferencia con respecto al GA-Simple radica en el proceso del operador de selección. Con una probabilidad uniforme, este operador selecciona cierta cantidad de individuos en poblaciones adyacentes para un posterior cruce con el individuo destacado de la población actual. La estrategia en Paralelo tipo Grillas no necesita una implementación sobre toda la población, si no más bien puede ser implementado sobre sub-poblaciones organizadas espacialmente.

Capítulo 3

Desarrollo

Los algoritmos genéticos implementados en este trabajo de investigación están basados en la optimización de la asignación de canales en redes celulares. La programación se desarrolló sobre la librería GAOT de Matlab. Esta librería ofrece un paquete predefinido de los operadores más comunes. Sin embargo para cumplir con el objetivo principal fue necesario implementar un operador de cruce con prioridad a los genes dominantes que permitiera mejorar el valor aptitud promedio. Esta investigación implementó un GA-Simple y tres topologías de conexión para un GA-Complejo en paralelo, basado en una estrategia de optimización según las ecuaciones (2.1), (2.2), (2.3), (2.4) y fundamentado en un modelo tipo “islas”, considerando genes dominantes. Los algoritmos evaluados ejecutan los mismos operadores genéticos salvo la creación de la población inicial, que varía de acuerdo a la topología de conexión del GA-Complejo Paralelo a ejecutar y el operador de migración que envía y recibe a los individuos de acuerdo a la topología de conexión entre las islas.

A continuación se detallan los algoritmos genéticos implementados con sus respectivos operadores bioinspirados en los postulados de la teoría “Evolución biológica por selección natural” expuesta por Charles Darwin en 1859.

3.1 GA Simple

El GA-Simple implementado (anexo 6.1) permite la optimización de la asignación de canales en redes celulares. La codificación del individuo se hace con un arreglo de objetos $a \mid a \in \mathbb{Z}$, considerando una lista prioritaria de genes dominantes. En la figura 3.1, muestra una lista prioritaria de 3 genes dominantes.

$$Cel_i = [1 \ 2 \ 3 \ 4 \ 4 \ 4]$$

Figura 3.1: Representación genes dominantes

El vector Cel_i representa la asignación de celdas base creadas a partir del vector demanda D . Por ejemplo, el vector D tiene una demanda de 6 canales distribuidos en 4 celdas. Entonces, se creará un vector Cel_i que contenga una repetición ordenada las celdas según la cantidad de canales requeridos.

$$D = [1 \ 1 \ 1 \ 3] \quad (3.1)$$

$$Cel_i = [1 \ 2 \ 3 \ 4 \ 4 \ 4] \quad (3.2)$$

Para la creación de la población (anexo 6.3) se realiza una permutación genética del vector Cel_i dando origen a la matriz de celdas $pop_{n \times m}$ donde n es el tamaño de la población y m la sumatoria de los canales demandados. La creación de la matriz de canales $PobCanales_{n \times m}$ se inicia con la mínima posible asignación del canal asociado según el ordenamiento de la matriz pop . La función aptitud (anexo 6.4) evalúa los canales

asignados mediante el modelado de las ecuaciones (2.1), (2.2), (2.3), (2.4), (2.5), (2.6), (2.7), (2.8) para crear la matriz de puntuación llamada $scores_n$, donde se registra el valor aptitud de los individuos. Cada fila ordenada de las matrices $PobCanales$, pop y $scores$ conforman una posible solución al problema.

Si aplicamos el ejemplo expuesto en [53], tenemos al vector demanda D y matriz de compatibilidad C . Entonces se genera el vector Cel_i de celdas índice ordenado.

$$D = [1 \ 1 \ 1 \ 3] \quad (3.3)$$

$$C = \begin{bmatrix} 5 & 4 & 0 & 0 \\ 4 & 5 & 0 & 1 \\ 0 & 0 & 5 & 2 \\ 0 & 1 & 2 & 5 \end{bmatrix} \quad (3.4)$$

$$Cel_i = [1 \ 2 \ 3 \ 4 \ 4 \ 4] \quad (3.5)$$

Para la creación de la población, se permuta genéticamente el vector Cel_i generando la matriz pop con un número de filas igual al tamaño de la población, luego se crean las matrices $PobCanales$ y $scores$. El operador de aptitud evalúa la información genética de las matrices $PobCanales$ y pop para determinar el valor aptitud de cada individuo.

$$pop_{inicial} = \begin{bmatrix} 4 & 2 & 3 & 4 & 1 & 4 \\ 3 & 4 & 4 & 4 & 2 & 1 \\ 4 & 2 & 1 & 4 & 4 & 3 \\ 3 & 4 & 2 & 4 & 4 & 1 \\ 4 & 3 & 1 & 2 & 4 & 4 \end{bmatrix} = \begin{bmatrix} pop_{inicial}\{1\} \\ pop_{inicial}\{2\} \\ pop_{inicial}\{3\} \\ pop_{inicial}\{4\} \\ pop_{inicial}\{5\} \end{bmatrix} \quad (3.6)$$

$$PobCanales_{inicial} = \begin{bmatrix} 1 & 3 & 4 & 6 & 10 & 11 \\ 1 & 3 & 8 & 13 & 1 & 5 \\ 1 & 2 & 6 & 6 & 11 & 3 \\ 1 & 3 & 1 & 8 & 13 & 5 \\ 1 & 3 & 1 & 5 & 6 & 11 \end{bmatrix} = \begin{bmatrix} PobCanales_{inicial}\{1\} \\ PobCanales_{inicial}\{2\} \\ PobCanales_{inicial}\{3\} \\ PobCanales_{inicial}\{4\} \\ PobCanales_{inicial}\{5\} \end{bmatrix} \quad (3.7)$$

$$scores_{inicial} = [11 \ 13 \ 11 \ 13 \ 11] \quad (3.8)$$

El operador de selección tipo ruleta, escoge los padres para cruzar. El proceso del operador de cruce, está inspirado en la reproducción asexual (anexo 6.5), según una fracción de cruce definida en su ejecución. El cruce inicia con un desplazamiento lineal en donde los puntos a cruzar son seleccionados aleatoriamente teniendo en cuenta la prioridad ordenada de los genes dominantes según la cadena de cada padre. La función $flipplr(x)$ voltea simultáneamente en ambos padres, el vector en los puntos seleccionados de izquierda a derecha. Es decir, devuelve el valor de x con toda la información genética de los hijos. Esta condición asegura la no destrucción del material genético del individuo durante la creación de la descendencia.

Continuando con el ejemplo presentado en [53], el operador de selección tipo ruleta toma de forma aleatoria dos individuos $pop_{inicial}\{1\}$ y $pop_{inicial}\{3\}$ para generar el cruce. Teniendo en cuenta una lista prioritaria de 2 genes dominantes, el operador de cruce genera $xoverKids\{1\}$ y $xoverKids\{2\}$.

$$xoverKids = \begin{bmatrix} 4 & 1 & 2 & 4 & 4 & 3 \\ 4 & 2 & 1 & 3 & 4 & 4 \end{bmatrix} = \begin{bmatrix} xoverKids\{1\} \\ xoverKids\{2\} \end{bmatrix} \quad (3.9)$$

El operador de mutación puntual (anexo 6.6), intercambia el alelo entre dos valores de locus seleccionados aleatoriamente sin tener en cuenta la lista prioritaria de genes dominantes. El operador de mutación genera el vector $MutChildren$, que almacena la información genética mutada. La rata de mutación es inversamente proporcional a la fracción de cruce. Es decir, a mayor cantidad de individuos cruzados, son menos los individuos a mutar

$$MutChildrens = [4 \ 1 \ 3 \ 2 \ 4 \ 4] \quad (3.10)$$

La función de aptitud analiza la descendencia para determinar su valor aptitud. Siguen con el ejemplo de [53], si la descendencia tiene mejor valor aptitud que su ascendencia entonces los nuevos individuos reemplazaran al grupo de individuos peores adaptados. Conformando la siguiente generación en *pop*, *PobCanales* y *scores*.

$$pop = \begin{bmatrix} 4 & 2 & 3 & 4 & 1 & 4 \\ 4 & 1 & 3 & 2 & 4 & 4 \\ 4 & 2 & 1 & 4 & 4 & 3 \\ 4 & 2 & 1 & 3 & 4 & 4 \\ 4 & 3 & 1 & 2 & 4 & 4 \end{bmatrix} = \begin{bmatrix} pop\{1\} \\ pop\{2\} \\ pop\{3\} \\ pop\{4\} \\ pop\{5\} \end{bmatrix} \quad (3.11)$$

$$PobCanales = \begin{bmatrix} 1 & 3 & 4 & 6 & 10 & 11 \\ 1 & 7 & 3 & 5 & 6 & 1 \\ 1 & 2 & 6 & 6 & 11 & 3 \\ 1 & 2 & 6 & 3 & 11 & 6 \\ 1 & 3 & 1 & 5 & 6 & 11 \end{bmatrix} = \begin{bmatrix} PobCanales\{1\} \\ PobCanales\{2\} \\ PobCanales\{3\} \\ PobCanales\{4\} \\ PobCanales\{5\} \end{bmatrix} \quad (3.12)$$

$$scores = [11 \ 11 \ 11 \ 11 \ 11] \quad (3.13)$$

Luego de la ejecución del AG-Simple, el individuo mejor adaptado es el resultado de la combinación del vector *Cel* celda con el vector *F* canales. La representación gráfica del individuo se muestra en la figura 3.2. El criterio de finalización está configurado para finalice el proceso del GA al ejecutar 100 generaciones.

$$Cel = [4 \ 2 \ 3 \ 4 \ 1 \ 4] \quad (3.14)$$

$$F = [1 \ 3 \ 4 \ 6 \ 10 \ 11] \quad (3.15)$$

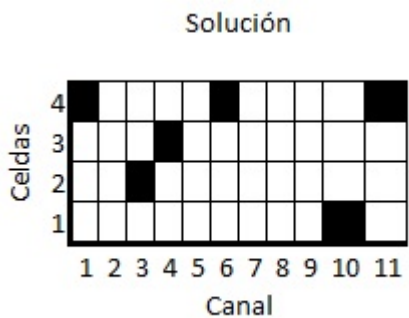


Figura 3.2: Representación Gráfica de Solución

3.2 GA Complejo Paralelo

El GA-Complejo Paralelo aplicado con seis islas (anexo 6.2) utiliza todos los operadores implementados en GA-Simple. Sin embargo, para poder generar un intercambio de información genética es necesario crear un mecanismo de comunicación entre las islas. El operador de migración introduce el conjunto migrante de individuos ordenados provenientes de las islas según la topología de conexión. El conjunto de individuos nativos menos apto es reemplazado por los inmigrantes según el porcentaje de migración configurado. Por último, se tiene un marcador que indica el fin de las migraciones según la isla actual.

A continuación se detallan las topologías de conexión implementadas en GA-Complejo Paralelo tipo islas

3.2.1 GA Complejo Paralelo tipo Estrella

La configuración tipo estrella (figura 3.3), realiza el proceso de migración basado en el segundo planteamiento de la teoría “Evolución de las especies por Selección Natural”, “Los animales generalmente habitan la misma región que sus antepasados”. La isla-1 es la isla principal de la topología del problema CAP. Las islas 2, 4, 5, y 6 modelan un ecosistema que circunda la isla principal. La isla 3 está en iguales condiciones a las demás islas, pero con el operador de selección diferente. Esto genera un tipo de individuos capaces de adaptarse a su entorno hasta convertirse en individuos exitosos. Dentro de la población de la isla principal los individuos inmigrantes se organizan según el orden de llegada, en las últimas posiciones de la población en curso.

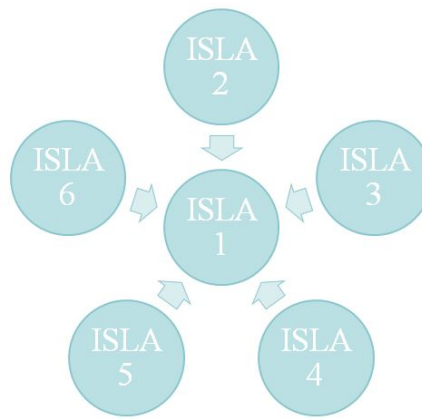


Figura 3.3: Topología Estrella

Para la configuración del GA-Complejo paralelo tipo “Estrella” las islas 2, 3, 4, 5 y 6, tienen un tamaño de población de 13 individuos por cada isla, envían un conjunto de 4 individuos mejores adaptados hacia la isla-1, con un tamaño de población de 70 individuos, cada 10 generaciones hasta completar un total de 100 generaciones ejecutadas. La fracción del operador de cruce de las islas 2, 3, 4, 5, y 6 es de 0,5 teniendo en cuenta 2 genes dominantes. En cambio la isla-1 tiene configurada una fracción de cruce de 0,2 teniendo en cuenta 15 genes dominantes. Las islas 1 y 3 ejecutan un operador de selección tipo torneo, a diferencia de las islas 2, 4, 5, y 6 que ejecutan un operador de selección uniforme.

3.2.2 GA Complejo Paralelo tipo Anillo

Para el caso de la topología tipo anillo (figura 3.4), el proceso de migración está basado en el quinto planteamiento de la teoría “Evolución de las especies por Selección Natural”, “Las etapas iniciales del desarrollo de las especies son similares y solo se diferencia en las etapas finales”. La isla-1 es la isla principal en la topología tipo anillo. Su configuración le aporta diversidad a la población. Adicionalmente por ser la isla de inicio y finalización del anillo, se ejecuta en 3 oportunidades. Las islas 2, 3, 4, 5 y 6 modelan las diferentes etapas a las que puede estar sometido un individuo, simulando todo el proceso de “selección natural” en donde sólo los mejores adaptados sobreviven.

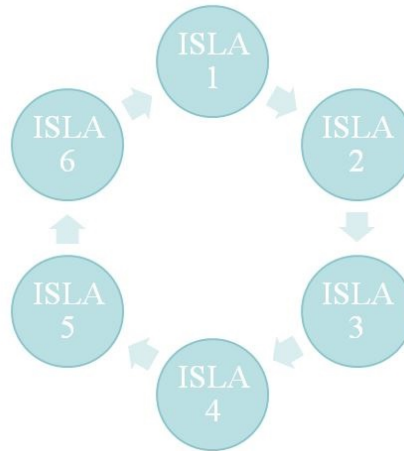


Figura 3.4: Topología Anillo

La configuración del GA-Complejo paralelo tipo “Anillo” en su primera iteración ejecuta una permutación en la población inicial de 70 individuos creados aleatoriamente, luego según la isla y topología de conexión, se introducen los individuos inmigrantes, escogidos un con operador de selección tipo torneo. La fracción del operador de cruce varía de acuerdo a la isla en ejecución. Para $Isla - 1 \Rightarrow 0,2$; $Isla - 2 \Rightarrow 0,3$; $Isla - 3 \Rightarrow 0,5$; $Isla - 4 \Rightarrow 0,7$; $Isla - 5 \Rightarrow 0,8$; $Isla - 6 \Rightarrow 1,0$, teniendo en cuenta 12 genes dominantes en todas las islas de la topología tipo anillo. Las islas 2, 3, 4, 5, 6 migran un conjunto de individuos cada 5 generaciones. En cambio la isla-1 en su primera y segunda iteración realiza 16 generaciones. En la iteración final la isla-1 ejecuta 18 generaciones, para así completar un total de 100 generaciones ejecutadas en dos giros del anillo.

3.2.3 GA Complejo Paralelo tipo Malla

En la implementación del GA-Complejo paralelo tipo “Malla” (figura 3.5) está inspirado en el primer planteamiento de la teoría “Evolución de las especies por Selección Natural”, “El proceso de la evolución es gradual, lento y continuo, sin saltos discontinuos o cambios súbitos”. La isla-1 tiene 0,2 de fracción de cruce, con 2 genes dominantes, operador de selección tipo uniforme y con una migración en la generación 15. Las islas 2, 3, 4, 5 tienen una fracción de cruce de 0,5 con 15 genes dominantes y un operador de selección tipo torneo y migran en la generación 15 hacia la siguiente isla. La isla-6 tiene una fracción de cruce de 0,7 con 15 genes dominantes y 25 generaciones para así completar un total del 100 generaciones ejecutadas.

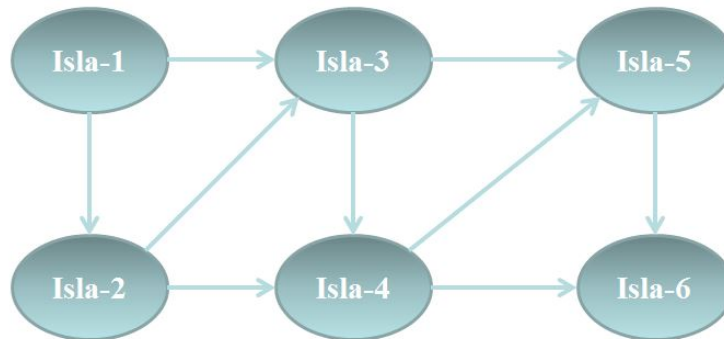


Figura 3.5: Topología Malla

El proceso de migración inicia luego de que la isla-1, con una población inicial creada aleatoriamente, llega a la generación 15. Entonces, envía un grupo ordenado de 14 individuos a la isla-2. La isla-2 inicializa

con una población nativa aleatoria, más un conjunto ordenado de 14 inmigrantes proveniente de la isla-1, los cuales son ubicados en las últimas posiciones de la población en curso. Luego de 15 generaciones la isla-2 envía 28 individuos emigrantes, repartidos en 14 individuos para la isla-3 y 14 individuos para la isla-4. La isla-3 inicializa con la población nativa aleatoria, más un conjunto ordenado de 14 inmigrantes provenientes de la isla-1 y 14 individuos ordenados provenientes de la isla-2, los cuales son ubicados en las últimas posiciones de la población en curso, según el orden de recepción. Luego de 15 generaciones la isla-3 envía 2 grupos ordenados de 14 individuos emigrantes, hacia las islas 4 y 5, respectivamente. La isla-4 inicializa con la población nativa aleatoria, más un conjunto ordenado de 14 inmigrantes provenientes de la isla-2 y un grupo ordenado de 14 individuos proveniente de la isla-3. Luego de 15 generaciones la isla-4 envía un grupo ordenado de 14 individuos emigrantes hacia la isla 5 y un conjunto ordenado de 14 individuos emigrantes hacia la isla-6. La isla-5 inicializa con la población nativa aleatoria, más un conjunto ordenado de 14 inmigrantes provenientes de la isla-3 y un grupo ordenado de 14 individuos proveniente de la isla-4. Luego de 15 generaciones la isla-5 envía un conjunto ordenado de 14 individuos emigrantes, hacia las islas-6. La isla-6 inicializa con la población nativa aleatoria, más un conjunto ordenado de 14 inmigrantes provenientes de la isla-4 y un grupo ordenado de 14 individuos provenientes de la isla-5. Luego de 25 generaciones la isla-6 selecciona al individuo mejor adaptado. La población inicial nativa de todas las islas es creada aleatoriamente. Los individuos con menos probabilidades de selección de la población nativa son reemplazados por los individuos inmigrantes.

Capítulo 4

Evaluación y Análisis de Resultados

En esta sección se evalúa la efectividad de los algoritmos implementados en un modelamiento con base en la plataforma “Philadelphia Benchmarks” [54–56] aplicada al problema CAP.

4.1 GA Simple

Los resultados de la implementación y comprobación del algoritmo genético simple para solucionar el problema de asignación de canales, se presentan en la tabla 4.1. Debido a la convergencia en la primera generación, los resultados obtenidos de los “Philadelphia Benchmarks” P12-P19 estos no brindan un aporte significativo a la investigación. Por lo tanto, se determinó no tenerlos en cuenta en la comparación frente al GA-Complejo paralelo tipo islas. Para el GA-Simple se implementó una distribución de cromosomas conformados por 6 genes dominantes.

Philadelphia Ref.		F-DR	GA-Simple		
Benchmark	Fval	Fval	Fval	Generación	Tiempo (hrs)
P1	239	259	249	99	0,84
P2	257	292	280	20	0,99
P3	426	448	435	80	1,53
P4	855	895	873	98	11,98
P5	1.713	1.801	1.771	96	94,40
P6	179	240	208	43	0,58
P7	252	269	260	86	0,89
P8	426	476	436	33	1,19
P9	524	593	550	46	1,54
P10	426	467	436	51	1,31
P11	426	472	437	99	1,28
P12	532	533	533	1	4,64
P13	308	310	309	1	0,67
P14	532	533	533	1	0,97
P15	308	315	309	1	0,89
P16	532	533	533	1	2,25
P17	220	222	221	1	1,13
P18	380	381	381	1	0,83
P19	528	533	529	1	2,13

Tabla 4.1: Resumen implementación Algoritmo Genético Simple

La columna “Fval” de “Ref. Philadelphia” muestra la cota inferior $\min\{\max\{F_{ik}\}\}$ para cada “Benchmark”. “F-DR” es un algoritmo genético implementado en [57, 58] con el cual se realizó la comparación de resultados. La columna “GA-Simple” muestra los resultados obtenidos con la ejecución del algoritmo implementado, así mismo la columna “Generación” muestra en cual generación encontró al individuo mejor adaptado. En general los resultados obtenidos por GA-Simple evidencia una mejora de hasta 10 canales en el valor aptitud del individuo mejor adaptado. El histograma presentado en la gráfica 4.1, muestra la repetibilidad y reproducibilidad en la convergencia del algoritmo durante la ejecución de P1.

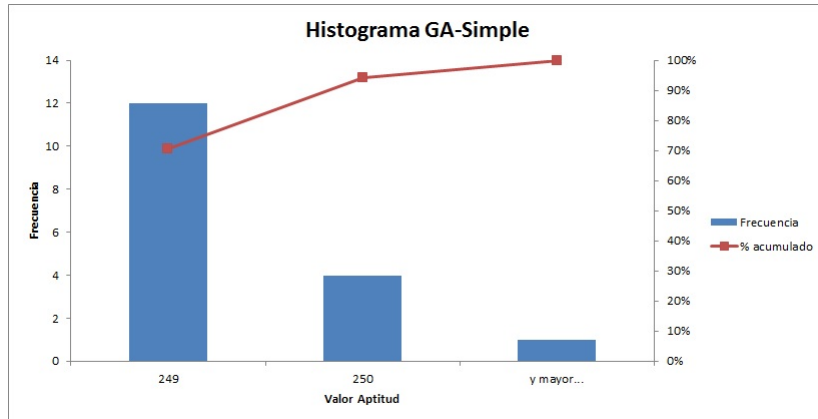


Figura 4.1: Diagrama de frecuencias y probabilidad acumulada del GA-Simple.

4.2 GA Complejo Paralelo

Los resultados obtenidos al implementar y evaluar la estrategia de optimización fundamentada en algoritmos genéticos paralelos tipo “islas” para la resolución de CAP en redes celulares, ofrecen una mejora en la pendiente de convergencia del algoritmo, con individuos mejores adaptados, que reflejan una óptima asignación de canales. En la tabla 4.2, se presenta un resumen comparativo detallado de los resultados de las diferentes implementaciones y simulaciones del algoritmo. La eficiencia del algoritmo se midió teniendo en cuenta el coste computacional reflejado en los tiempos de ejecución. Los casos de P4 y P5 se caracterizaron por ser escenarios con un gran coste computacional con tiempos de convergencia superiores a 24 horas con respecto a lo registrado por los demás “Benchmarks”.

En general se destaca la eficiencia del algoritmo con topología tipo estrella en el que encontró más numero de veces al individuo mejor adaptado con mejores valores de aptitud.

Para el caso de la topología tipo malla, esta encuentra al individuo mejor adaptado en menor cantidad de generaciones. Es decir, mejora la tasa de convergencia en relación a las demás topologías.

La topología tipo anillo demostró ser robusta debido a los valores reportados para el caso de P4, donde el vector demanda y la matriz de compatibilidad requieren un gran costo computacional para encontrar una solución. Los histogramas presentados en las gráficas 4.2 - 4.3 y 4.4 evidencian el desempeño de la convergencia del algoritmo para optimizar la solución del problema CAP. La probabilidad acumulada marca una tendencia creciente. Los resultados de la topologia malla muestran la repetibilidad y reproducibilidad en el valor aptitud del individuo mejor adaptado.

Philadelphia Ref. Benchmark	F-DR		AG-Simple		AGP-Estrella		AGP-Anillo		AGP-Malla	
	Fval	Fval	Fval	Tiempo (hr)	Fval	Tiempo (hr)	Fval	Tiempo (hr)	Fval	Tiempo (hr)
P1	239	259	249	0,84	248	0,75	247	0,76	247	0,73
P2	257	292	280	0,99	270	1,26	273	1,32	275	1,24
P3	426	448	435	1,53	429	1,65	431	1,71	432	1,64
P4	855	895	873	11,98	869	13,19	865	16,87	869	13,02
P5	1.713	1.801	1.771	94,40	1.757	96,45	1.749	97,35	1.767	96,21
P6	179	240	208	0,58	206	0,61	206	0,62	206	0,66
P7	252	269	260	0,89	259	0,96	260	1,19	259	1,18
P8	426	476	436	1,19	431	1,62	430	1,29	430	1,53
P9	524	593	550	1,54	539	2,20	545	2,25	542	2,15
P10	426	467	436	1,31	431	1,38	434	1,74	434	1,79
P11	426	472	437	1,28	435	1,70	432	1,34	436	1,68

Tabla 4.2: Resumen comparativo de implementar los Algoritmos Genéticos

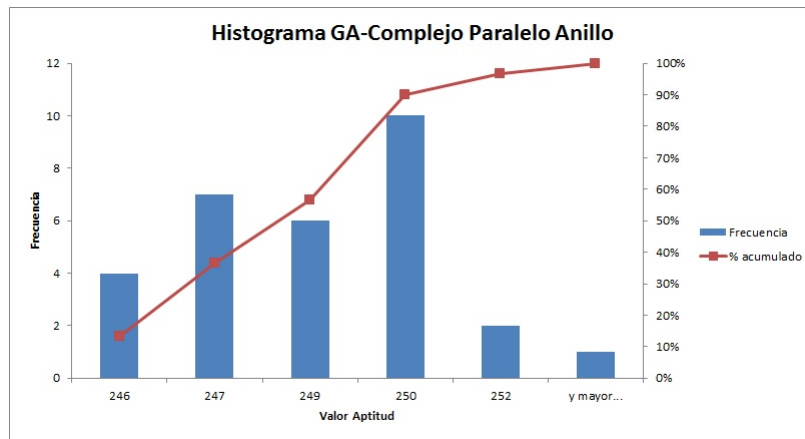


Figura 4.2: Diagrama de frecuencias y probabilidad acumulada de GA-Complejo Paralelo Anillo

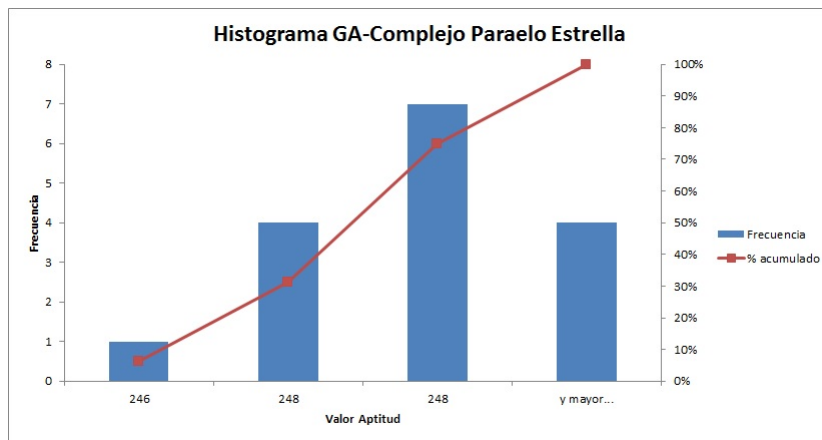


Figura 4.3: Diagrama de frecuencia y probabilidad acumulada de GA-Complejo Paralelo Estrella

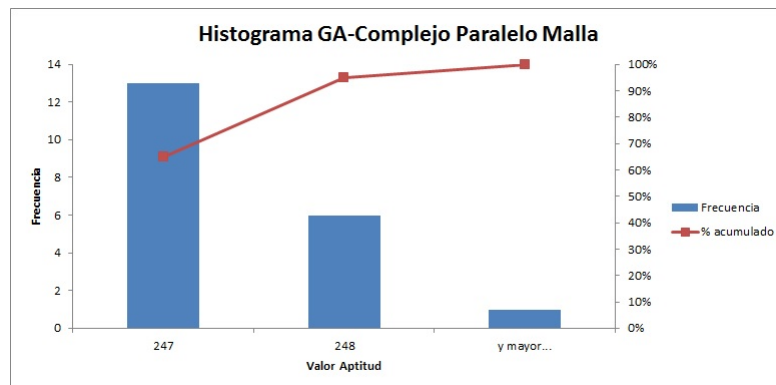


Figura 4.4: Diagrama de frecuencia y probabilidad acumulada de GA-Complejo Paralelo Malla

4.3 Resultados Comparativos

El modelado del “Benchmarck” P1, arrojó que las topologías anillo y malla encontraron en las generaciones 55 y 58 al individuo mejor adaptado, con igual valor de aptitud. En la figura 4.5, se hace una comparación de las curvas de convergencia de los algoritmos modelados. La topología anillo muestra una mejor tasa

de convergencia y bajo coste computacional, reflejados en un uso eficiente en la asignación de canales en redes celulares. Sin embargo, para el modelamiento del “Benchmark” P2 (figura 4.6) y P3 (figura 4.7), la topología tipo estrella obtuvo los mejores resultados en la asignación de canales, con una ágil tasa de convergencia a bajo coste computacional.

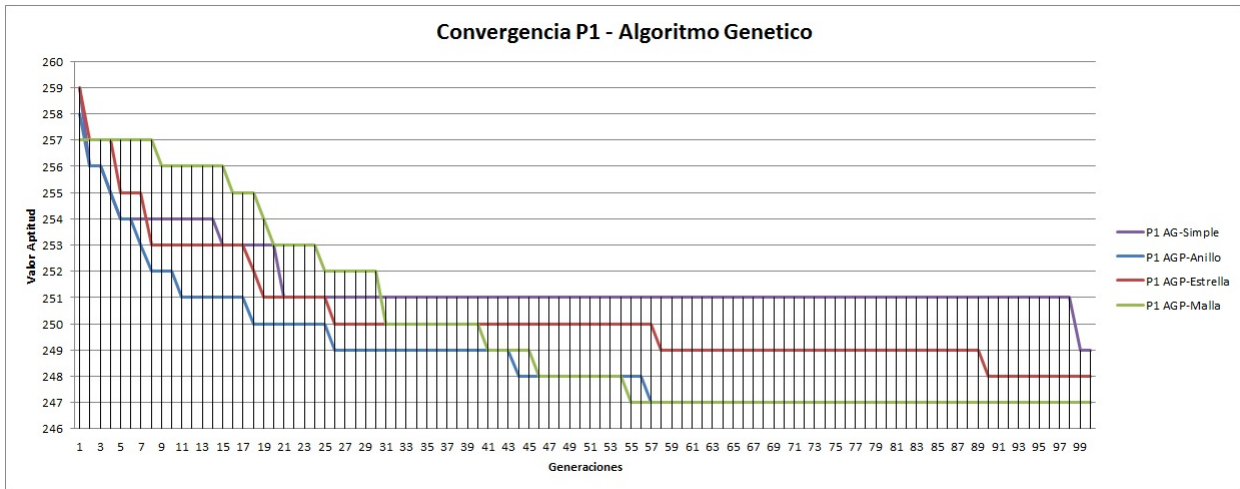


Figura 4.5: Análisis convergencia P1

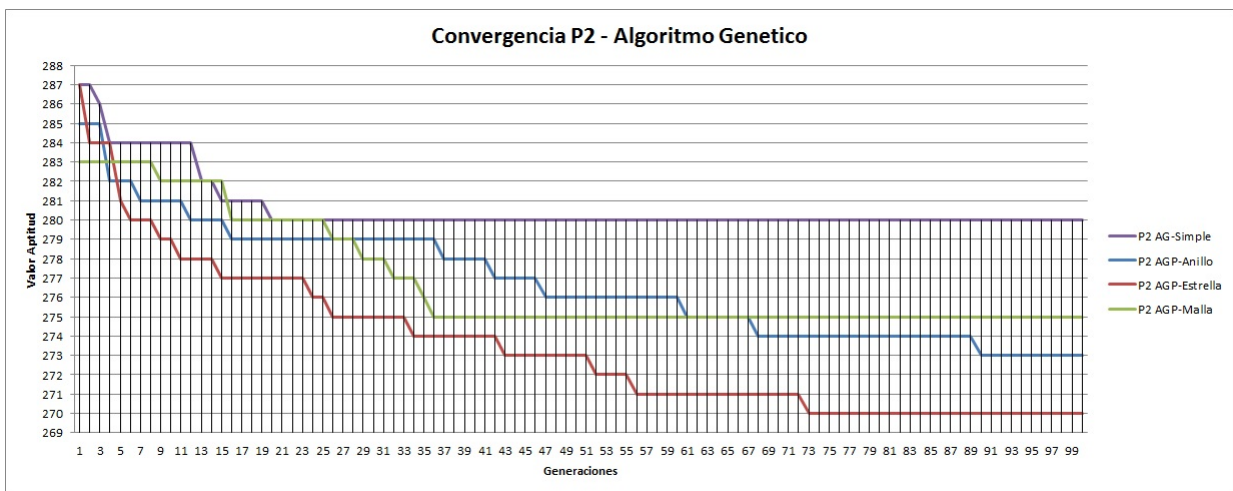


Figura 4.6: Análisis convergencia P2

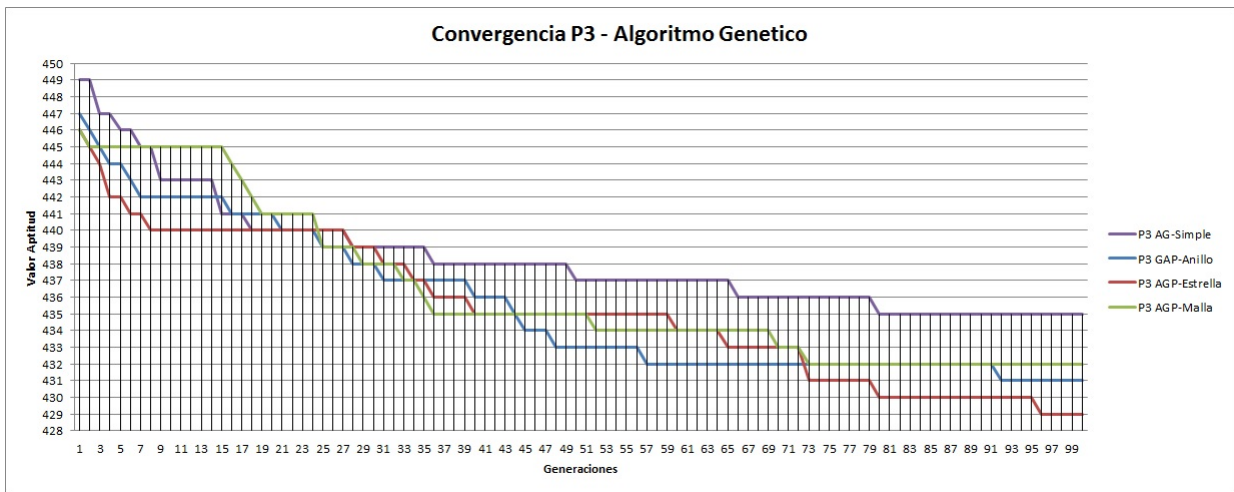


Figura 4.7: Análisis convergencia P3

Debido a los requerimientos del vector demanda y matriz de compatibilidad los “Benchmark” P4 y P5, se convierten en un modelamiento de alta exigencia computacional. La topología anillo, ofrece una clara convergencia y mejor valor aptitud del individuo mejor adaptado (figura 4.8) en P4. Esto se debe a que en la topología tipo anillo los individuos emigrantes migran cada 5 generaciones hacia una isla. Es decir, los individuos emigrantes se convierten en una población nómada, que distribuyen sus genes con el paso por las islas. Del mismo modo, para el caso de P5 (figura 4.9), la topología tipo anillo dio como resultado una pendiente con tendencia constante en la curva de convergencia de la población y una mejora del valor aptitud del individuo mejor adaptado de 9 canales.

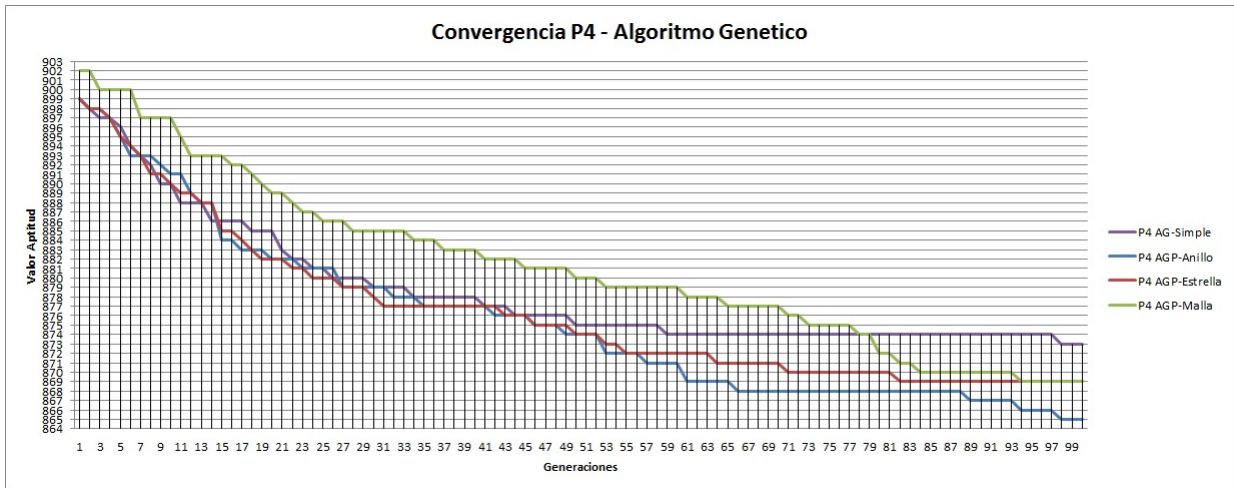


Figura 4.8: Análisis convergencia P4

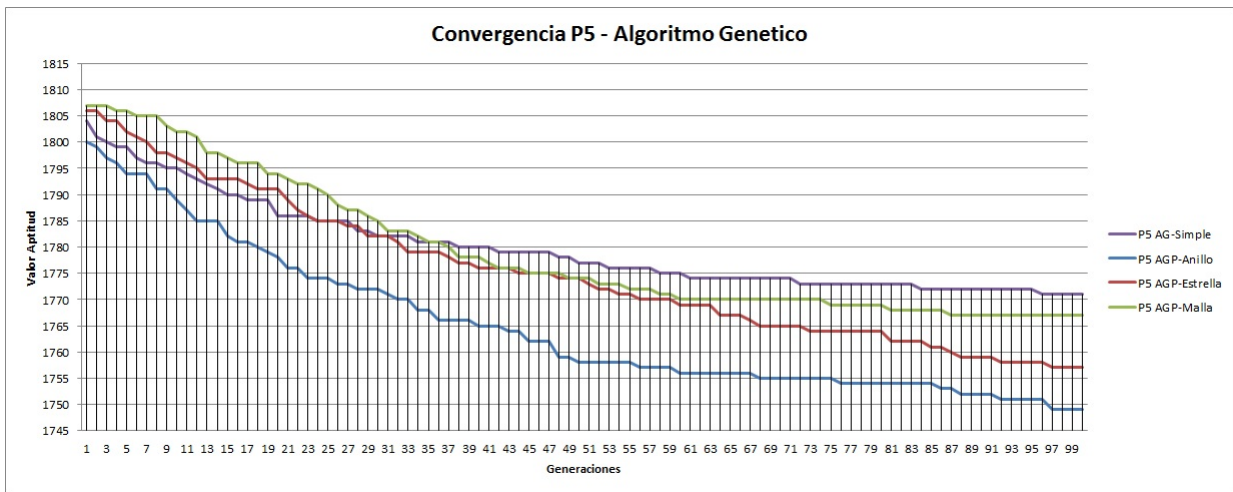


Figura 4.9: Análisis convergencia P5

En el caso del “Benchmark” P6, las topologías tipo malla, anillo y estrella tuvieron una convergencia similar hasta la generación 55, 59 y 63 puntos en los cuales recibieron un conjunto ordenado de individuos inmigrantes mejores adaptados. A partir de la recepción del mencionado conjunto las tres topologías encuentran al individuo mejor adaptado (figura 4.10). El valor aptitud del individuo mejor adaptado encontrado por las tres topologías es el mismo. Sin embargo su información genética es diferente. Para poder determinar cual de los tres individuos es realmente el mejor adaptado es necesario contemplar nuevas variables al problema de asignación de canales planteado.

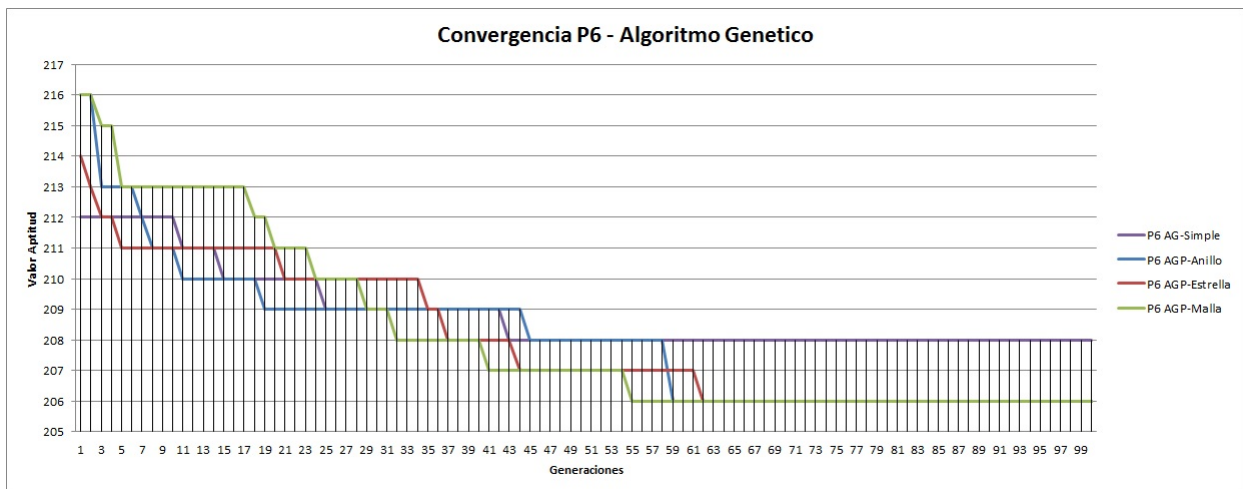


Figura 4.10: Análisis convergencia P6

Con la implementación de la topología tipo estrella en el “Benchmark” P7, se obtuvo como resultado una convergencia rápida y dinámica de la población hasta encontrar al individuo mejor adaptado en la generación 31. En cambio la topología tipo malla, encontró al individuo mejor adaptado en la generación 76 (figura 4.11). Es decir, la topología estrella resultó estar mejor preparada para enfrentar un problema de características iguales a P7, ofreciendo resultados satisfactorios con bajo coste computacional. A diferencia de los resultados obtenidos en el modelamiento del “Bechmark” P8, en donde la topología tipo malla demostró una rápida convergencia poblacional hasta encontrar al individuo mejor adaptado en la generación 64 (figura 4.12).

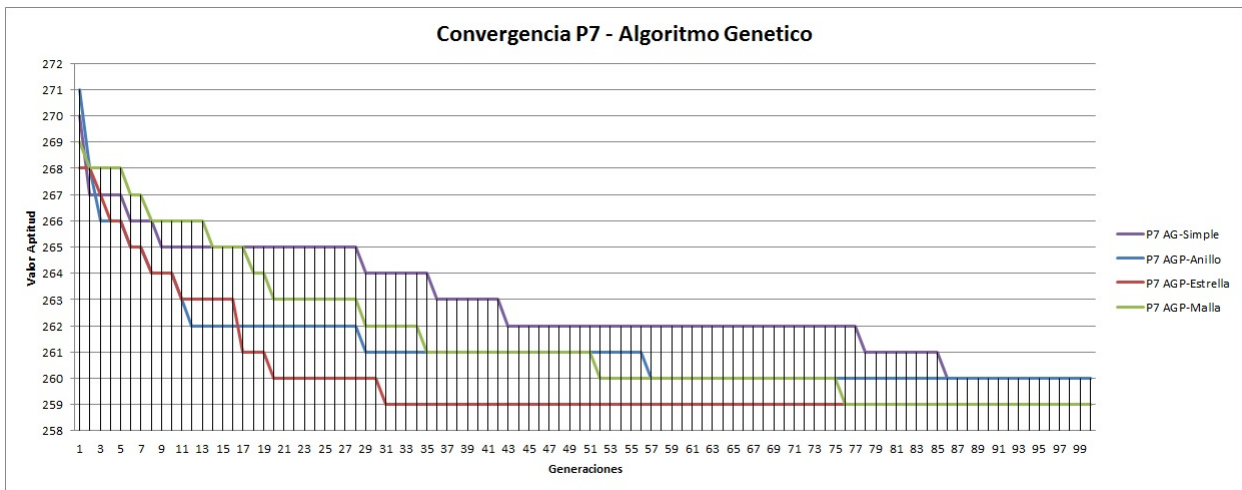


Figura 4.11: Análisis convergencia P7

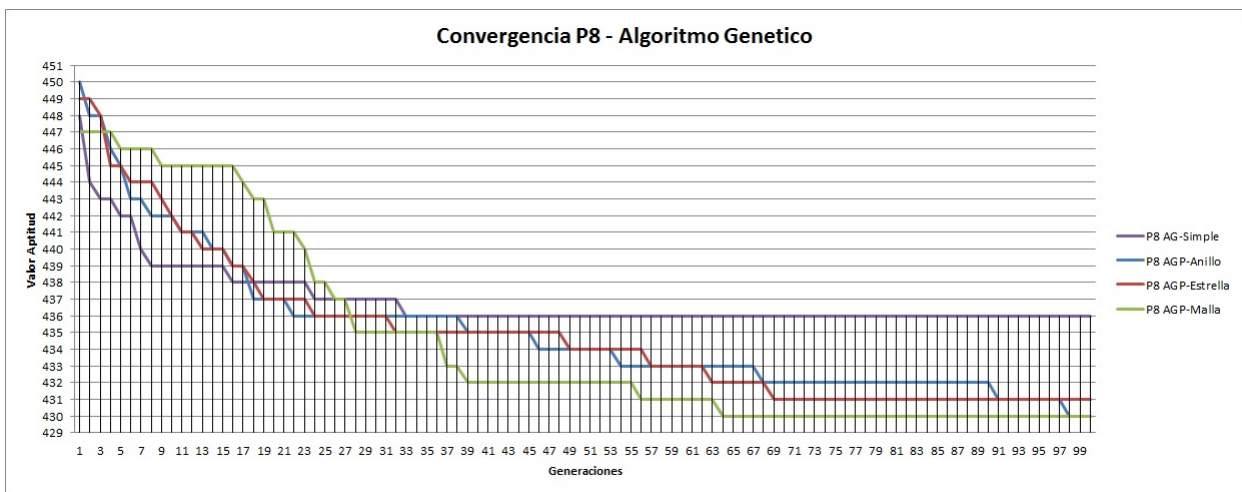


Figura 4.12: Análisis convergencia P8

Para los “Benchmark” P9 y P10 (figuras 4.13-4.14), la topología de conexión tipo estrella presentó una convergencia con pendiente constante hasta encontrar al individuo mejor adaptado. Además los valores de aptitud del mejor individuo adaptado presentan una mejora de 3 canales en la asignación de frecuencias en redes celulares. Es decir, se está maximizando la utilización del espectro. Por último, esta la evaluación del “Benchmark” P11 (figura 4.15), en donde la topología tipo anillo encontró al individuo mejor adaptado en la generación 92 presentando una lenta convergencia poblacional. Sin embargo, en todas las simulaciones y comparaciones realizadas entre el GA-Simple, F-DR y GA-Complejo paralelo los resultados de las topologías en paralelo siempre encontraron un individuo mejor adaptado que en los demás algoritmos genéticos analizados.

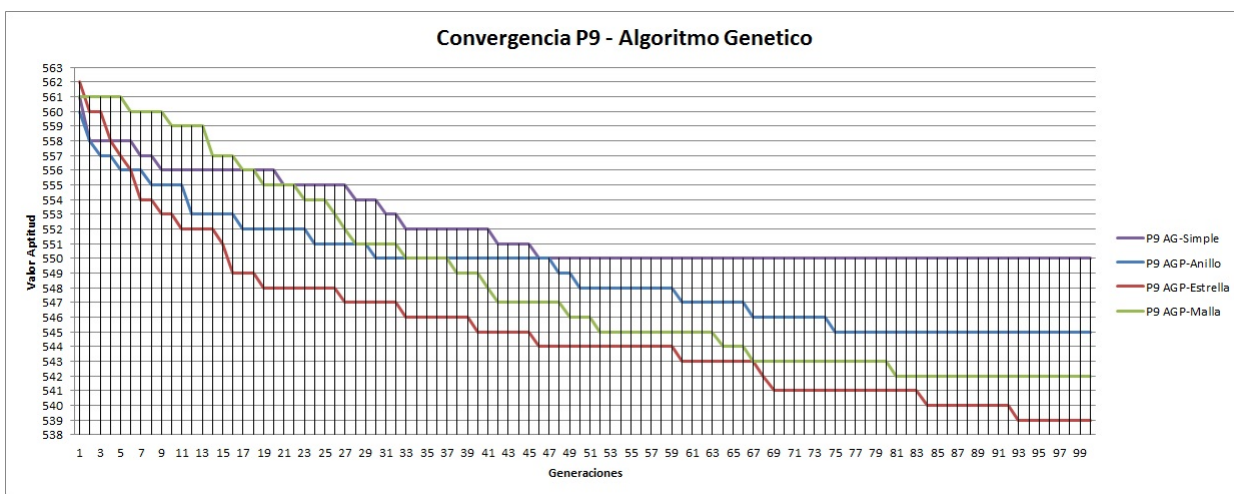


Figura 4.13: Análisis convergencia P9

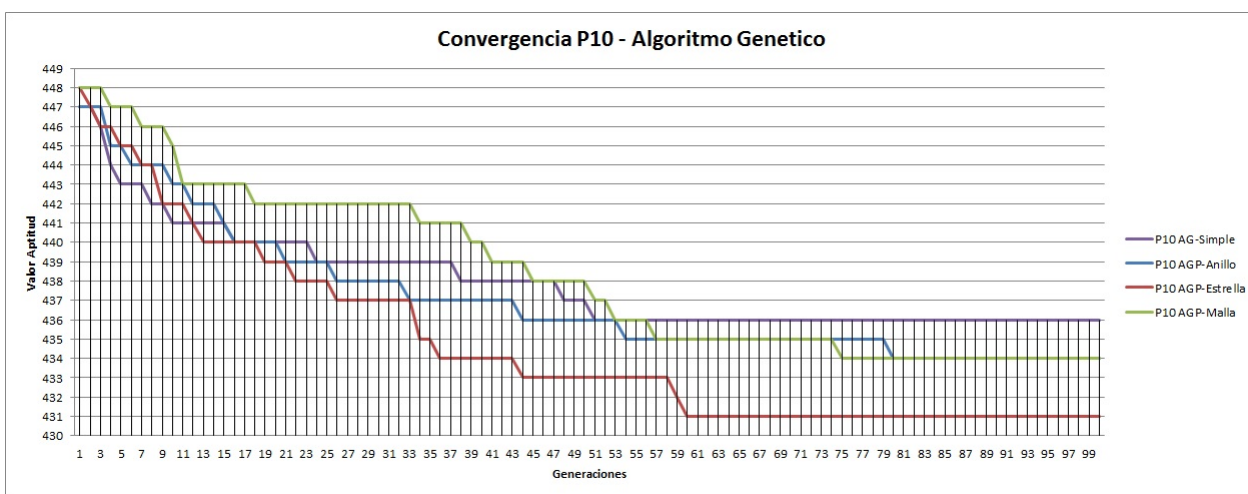


Figura 4.14: Análisis convergencia P10

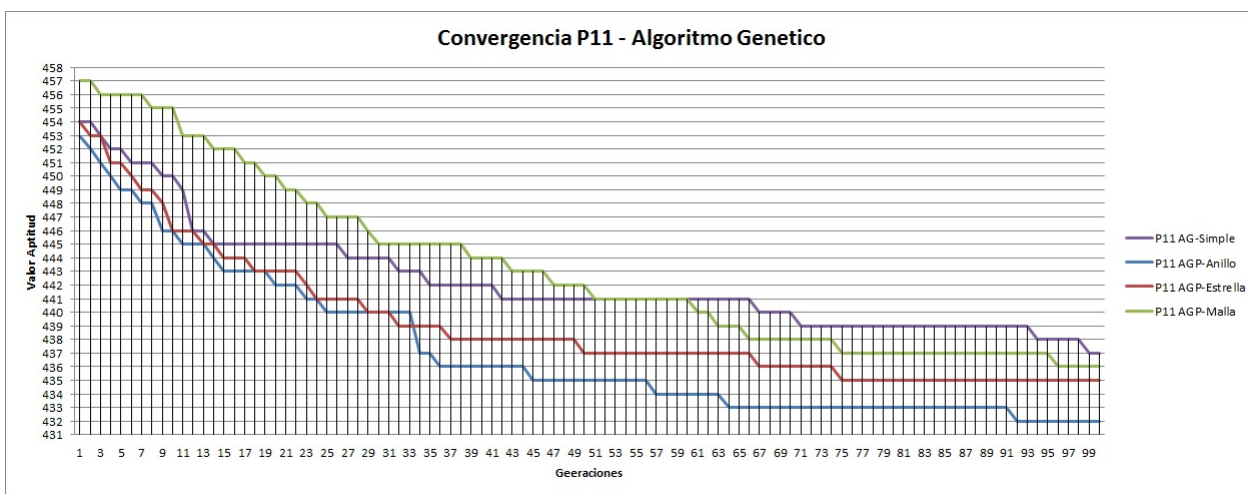


Figura 4.15: Análisis convergencia P11

Capítulo 5

Conclusiones

Los resultados de esta investigación muestran el cumplimiento del objetivo general inicialmente propuesto. Se implementó y evaluó una estrategia de optimización fundamentada en algoritmos genéticos paralelos tipo “islas” para la resolución de CAP en redes celulares, utilizando un tipo de reproducción asexual simultánea con dos individuos. Este trabajo de investigación se desarrolló con la utilización de la librería GOAT de Matlab por cuanto ofrece una amplia gama de opciones de configuración y personalización de los componentes de un algoritmo genético. Prueba de ello es que permitió manipular su código a tal punto que se implementó un operador de cruce bio-inspirado en la reproducción asexual de las especies.

Los resultados obtenidos con la implementación de GA-Simple, demostraron una variabilidad de individuos creados a partir de la permutación en la población en cada generación ejecutada. El GA-Simple consiguió individuos mejores adaptados hasta en 9 canales, en comparación con los resultados obtenidos del algoritmo F-DR.

Al implementar GA-Simple en los “Benchmark” Philadelphia P12 hasta P19, dio como resultado una convergencia del individuo mejor adaptado en la primera generación. Dado que los resultados obtenidos no son un aporte significativo a esta investigación, se determinó realizar la evaluación desde P1 hasta P11.

Los “Benchmark” P4 y P5 de Philadelphia se caracterizaron por ser una configuración con exigencias computacionales en tiempo de convergencia. La evaluación de P4 y P5 dio como resultado una convergencia con pendiente constante y mejoras en el valor de aptitud de 5 y 10 canales respectivamente. El tiempo de convergencia superó las 14 y 98 horas continuas de simulación, durante este tiempo los equipos de cómputo no presentaron fallas por falta de capacidad de procesamiento. Para el caso de la topología tipo anillo, su estructura cíclica le permitió establecer un esquema robusto de optimización basado en tiempos cortos de convergencia. Esta topología, obtuvo una pendiente suave y constante en el modelamiento de P4 y P5 hasta obtener el mejor valor aptitud de las topologías implementadas.

Los GA-Complejos Paralelos dieron como resultado valores de aptitud de individuos mejores adaptados con mejoras de hasta 10 canales de asignación. Los resultados obtenidos por la topología tipo estrella, evidenciaron como la centralización de los mejores individuos permitió encontrar mejores soluciones al problema de CAP.

En la configuración de la topología malla, los resultados de aptitud de los mejores individuos estuvieron dentro de la media. Esto es debido a que un individuo mejor adaptado de la isla-1 no alcanzaría a migrar hasta la isla-6. Este esquema de migración modela las etapas a las que puede estar sometida las especies para propiciar su evolución.

Como trabajos a futuro se recomienda continuar con trabajos de investigación en la evaluación sistemática de los GA emulando procesos biológicos con un operador de migración y genes dominantes. Este operador hace que los GA tengan una mayor concordancia con la adaptación a los cambios en la evolución de las especies.

Agradecimientos a la Pontificia Universidad Javeriana y al Departamento de Electrónica de dicha universidad, por facilitar los recursos necesarios de Software, bases de datos y personal para realizar a cabo la presente investigación. Este trabajo está dedicado a mis padres, hermanos, mi compañera inseparable y a

nuestro hermoso bebé. Espero poder retribuirles el apoyo, el cariño y la comprensión que me han brindado durante todos estos años de grandes sacrificios.

Bibliografía

- [1] F. Jaimes-Romero, D. Munoz-Rodriguez, and S. Tekinay, “Channel assignment in cellular systems using genetic algorithms,” in *Vehicular Technology Conference, 1996. 'Mobile Technology for the Human Race'*, IEEE 46th, vol. 2, pp. 741 –745 vol.2, 28 1996.
- [2] C. Augusto, “Teoria de la complejidad computacional y teoria de la computabilidad,” *Investigacion sistemas informaticos*, vol. 1, pp. 102 –105, feb 2004.
- [3] J. H. Holland, *Adaptation in natural systems an introductory analysis with applications to biology, control and artificial intelligence*. Cambridge: MIT Press, sixth edition ed., 2001.
- [4] L. S. Jose-Revuelta, “A new adaptive genetic algorithm for fixed channel assignment,” *Information Sciences*, vol. 177, pp. 2655 – 2678, sept. 2007.
- [5] W. Lai and G. Coghill, “Channel assignment through evolutionary optimization,” *Vehicular Technology, IEEE Transactions on*, vol. 45, pp. 91 –96, feb 1996.
- [6] M. Lima, A. Araujo, and A. Cesar, “Adaptive genetic algorithms for dynamic channel assignment in mobile cellular communication systems,” *Vehicular Technology, IEEE Transactions on*, vol. 56, pp. 2685 –2696, sept. 2007.
- [7] L. Idoumghar and R. Schott, “A new hybrid ga-mdp algorithm for the frequency assignment problem,” in *Tools with Artificial Intelligence, 2006. ICTAI '06. 18th IEEE International Conference on*, pp. 18 –25, nov. 2006.
- [8] S. Pinagapany and A. Kulkarni, “Solving channel allocation problem in cellular radio networks using genetic algorithm,” in *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, pp. 239 –244, 6-10 2008.
- [9] C.-F. Tsai, F.-C. Lin, and C.-W. Tsai, “An efficient ga-based approach for fixed channel assignment in cellular radio networks,” in *Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on*, vol. 1, pp. 384 – 389 vol.1, 16-20 2003.
- [10] A. Acan, H. Altincay, Y. Tekol, and A. Unveren, “A genetic algorithm with multiple crossover operators for optimal frequency assignment problem,” in *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, vol. 1, pp. 256 – 263 Vol.1, 8-12 2003.
- [11] K. I. Aardal, S. P. V. Hoesel, A. M. Koster, C. Mannino, and A. Sassano, “Models and Solution Techniques for Frequency Assignment Problems,” 2001.
- [12] J. A. Zoellner and C. L. Beall, “A breakthrough in spectrum conserving frequency assignment technology,” *IEEE Transaction on Electromagnetic Compatibility*, vol. 19, no. 1, pp. 313–319, 1977.
- [13] A. Thavarajah and W. Lam, “A heuristic algorithm for channel assignment in cellular mobile systems,” *VTC '98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution*, no. 4, pp. 1690–1694, 1998.

- [14] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: Algorithms and complexity*. Prentice-Hall, 1982.
- [15] M.-H. Jin, H.-K. Wu, J.-T. Horng, and C.-H. Tsai, “An evolutionary approach to fixed channel assignment problems with limited bandwidth constraint,” in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 7, pp. 2100–2104 vol.7, 2001.
- [16] D. Gozuepek, G. Genc, and C. Ersoy, “Channel assignment problem in cellular networks: A reactive tabu search approach,” in *24th International Symposium on Computer and Information Sciences*, pp. 298–303, IEEE, Sep 2009.
- [17] H.-s. Lee, D.-w. Lee, and J. Lee, “Multi-stage neural networks for channel assignment in cellular radio networks,” *LNCS 3174*, no. 1, pp. 287–292, 2004.
- [18] B. Baran and M. Almiron, “Colonia de hormigas en un ambiente paralelo asincrono,” *Universidad Nacional de Asuncion, Centro Nacional de Computacion, San Lorenzo, Paraguay*, vol. 4, pp. 1–11, 2001.
- [19] I. Chambers, *The Practical Handbook of Genetic Algorithms Applications*. Lodon: Chapman Hall/CRC, second edit ed., 2000.
- [20] S. Sivanandam and S. Deepa, *Introduction to Genetic Algorithms*. Berlin: Springer, first edit ed., 2008.
- [21] C. P. L.E. Ruiz, “Un algoritmo genetico fijo para solucionar el problema de asignacion de canales,” *Segundo Congreso Internacional de la Region Andina*, vol. 48, pp. 27–34, Marzo 2004.
- [22] X. Fu, Y. Pan, A. Bourgeois, and P. Fan, “A three-stage heuristic combined genetic algorithm strategy to the channel-assignment problem,” in *Parallel and Distributed Processing Symposium, 2003. Proceedings. International*, p. 8 pp., 22–26 2003.
- [23] S. Patra, K. Roy, S. Banerjee, and D. Vidyarthi, “Improved genetic algorithm for channel allocation with channel borrowing in mobile computing,” *Mobile Computing, IEEE Transactions on*, vol. 5, pp. 884–892, July 2006.
- [24] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer-Verlag, third edit ed., 1996.
- [25] H. G. Sandalidis, P. Stavroulakis, and J. Rodriguez-Tellez, “Genetic inspired channel assignment schemes for cellular systems,” *Mathematics and Computers in Simulation*, vol. 51, pp. 273 – 286, sept. 2000.
- [26] G. Chakraborty and B. Chakraborty, “A genetic algorithm approach to solve channel assignment problem in cellular radio networks,” in *Soft Computing Methods in Industrial Applications, 1999. SM-Cia/99. Proceedings of the 1999 IEEE Midnight-Sun Workshop on*, pp. 34–39, 1999.
- [27] W. Banzhaf, R. E. Keller, F. D. Francone, and P. Nordin, *Genetic Programming An Introduction*. San Francisco: Morgan Kaufmann Publishers, first edit ed., 1998.
- [28] L. Wang, S. Arunkumaar, and W. Gu, “Genetic algorithms for optimal channel assignment in mobile communications,” in *Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference on*, vol. 3, pp. 1221 – 1225 vol.3, 18–22 2002.
- [29] J. S. A.E. Eiben, *Introduction to Evolutionary Computing*. Oxford: Springer, second edit ed., 1996.

- [30] L. Wang, S. Li, S. C. Lay, W. H. Yu, and C. Wan, "Genetic algorithms for optimal channel assignments in mobile communications," *Scholl of Electrical and Electronic Engineering Nanyang Technological University*, pp. 1–33, 2001.
- [31] M. Sakawa, *Genetic Algorithms and Fuzzy Multiobjective Optimization*. Massachusetts: Kluwer Academic Publishers, second edit ed., 2002.
- [32] J. R. Koza, *Genetic Programming On the Programming of Computers by Means of Natural Selection*. Massachusetts: The MIT Press, sixth edit ed., 1998.
- [33] T. Back, *Evolutionary Algorithms in Theory and Practice*. Oxford: Oxford University Press, second edit ed., 1996.
- [34] V. Estivill-Castro, "Adaptive genetic operators," in *Proceedings Intelligent Information Systems. IIS'97*, (Australia), pp. 194–198, IEEE Comput. Soc, 1997.
- [35] P. Bentley, T. Gordon, J. Kim, and S. Kumar, "New trends in evolutionary computation," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, (London), pp. 162–169, IEEE Comput. Soc, 2001.
- [36] S. Matsui, I. Watanabe, and K.-i. Tokoro, "An efficient hybrid genetic algorithm for a fixed channel assignment problem with limited bandwidth," in *GECCO 2003*, (Berlin), pp. 2240–2251, Springer-Verlag, 2003.
- [37] S. Alireza, G. Shirazi, and H. Amindavar, "A hybrid method for channel assignment problems in cellular radionetworks," in *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, vol. 00, pp. 1260–1265, IEEE, 2006.
- [38] A. Ghassempour Shirazi and H. Amindavar, "Channel assignment for cellular radio using extended dynamic programming," *AEU - International Journal of Electronics and Communications*, vol. 59, pp. 401–409, Nov 2005.
- [39] W. Youping, L. Liang, and C. Lin, "An advanced genetic algorithm for traveling salesman problem," in *2009 Third International Conference on Genetic and Evolutionary Computing*, (Pekin), pp. 101–104, IEEE, Oct 2009.
- [40] M. Melanie, *An Introduction to Genetic Algorithms*. Massachusetts: A Bradford Book the MIT Press, first edit ed., 1999.
- [41] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms*. Lodon: springer, first edit ed., 2010.
- [42] J. C. Bean, Y. Chen, C. A. C. Coello, K. Deb, A. E. Eiben, E. Ghashghai, T. Goyal, EmmaHart, F. Hillier, M. Hillier, R. Hinterding, J. A. Joines, M. G. Kay, A. Li, A. Z.-Z. Lin, Z. Michalewicz, B. A. Norman, R. L. Rardin, P. Ross, T. Runarsson, R. Sarker, M. Schmidt, A. E. Smith, B. Wah, I. Wegener, C. C. W. III, K. P. Wong, and X. Yao, *Evolutionary Optimization*. New York: Kluwer Academic Publishers, first edit ed., 2003.
- [43] I. Hitoshi, P. Topon Kumar, and H. Yoshihiko, *Applied Genetic Programming and Machine Learning*. New York: CRC International Series on Computational Intelligence, first edit ed., 2010.
- [44] C. Cotta, "Una vision general de los algoritmos memeticos," *Lenguas y Ciencias de la Computacion, ETSI Informatica*, vol. 1, pp. 1–21, 2007.
- [45] J. Koza, *Genetic Programming Theory and Practice II*. Boston: Springer, first edit ed., 2005.

- [46] M. Boulif and K. Atif, "A new fuzzy genetic algorithm for the dynamic bi-objective cell formation-problem considering passive and active strategies," *International Journal of Approximate Reasoning*, vol. 47, pp. 141–165, feb 2008.
- [47] Y. Maeda, M. Ishita, and Q. Li, "Fuzzy adaptive search method for parallel genetic algorithm with island combination process," *International Journal of Approximate Reasoning*, vol. 41, no. 1, pp. 59–73, 2006. Aggregation Operators and Decision Modeling.
- [48] E. Alba Torres, *Analisis y Diseno de Algoritmos Geneticos Paralelos Distribuidos*. PhD thesis, Universidad de Malaga, 1999.
- [49] S. Nesmachnow, "Evolucion en el diseno y clasificacion de algoritmos geneticos paralelos," 2002.
- [50] S. E. Eklund, "A massively parallel architecture for distributed genetic algorithms," *Parallel Computing*, vol. 30, no. 5-6, pp. 647–676, 2004. Parallel and nature-inspired computational paradigms and applications.
- [51] E. Cantu-Paz and D. E. Goldberg, "Efficient parallel genetic algorithms: theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 4, pp. 221–238, 2000.
- [52] R. Poveda Ch, J. Gomez P., and E. Leon G, "Grislas : Un algoritmo genetico paralelo que combina los modelos de grillas e islas para encontrar soluciones optimas cercanas al problema del agente viajero," *Avances en Sistemas e Informatica*, vol. 5, no. 3, pp. 13–20, 2008.
- [53] F. Nobuo and T. Yoshiyasu, "A neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Transactions on vehicular technology*, vol. 41, pp. 430–437, 1992.
- [54] L. Anderson, "A simulation study of some dynamic channel assignment algorithms in a high capacity mobile telecommunications system," *Vehicular Technology, IEEE Transactions on*, vol. 22, pp. 210 – 217, nov. 1973.
- [55] C. Valenzuela, S. Hurley, and D. Smith, "A permutation based genetic algorithm for minimum span frequency assignment," in *Parallel Problem Solving from Nature PPSN V* (A. Eiben, T. Back, M. Schoenauer, and H.-P. Schwefel, eds.), vol. 1498 of *Lecture Notes in Computer Science*, pp. 907–916, Springer Berlin / Heidelberg, 1998. 10.1007/BFb0056932.
- [56] A. Avenali, C. Mannino, and A. Sassano, "Minimizing the span of d-walks to compute optimum frequency assignments," *Mathematical Programming*, vol. 91, pp. 357–374, 2002. 10.1007/s101070100247.
- [57] S. L. M. Jose-Revuelta, "An heuristic search technique for fixed frequency assignment in non-homogeneous demand systems," *Signal Processing*, vol. 88, pp. 1461–1476, Jun 2008.
- [58] K. L. Yeung and T.-s. P. Yum, "Fixed Channel Assignment Optimization for Cellular Mobile Networks," *IEICE TRANS. COMMUN.*, vol. E83-B, no. 8, pp. 1783–1791, 2000.

Capítulo 6

Anexos

6.1 Diagramas de bloques de los GA implementados

6.1.1 Diagrama de bloques GA-Simple

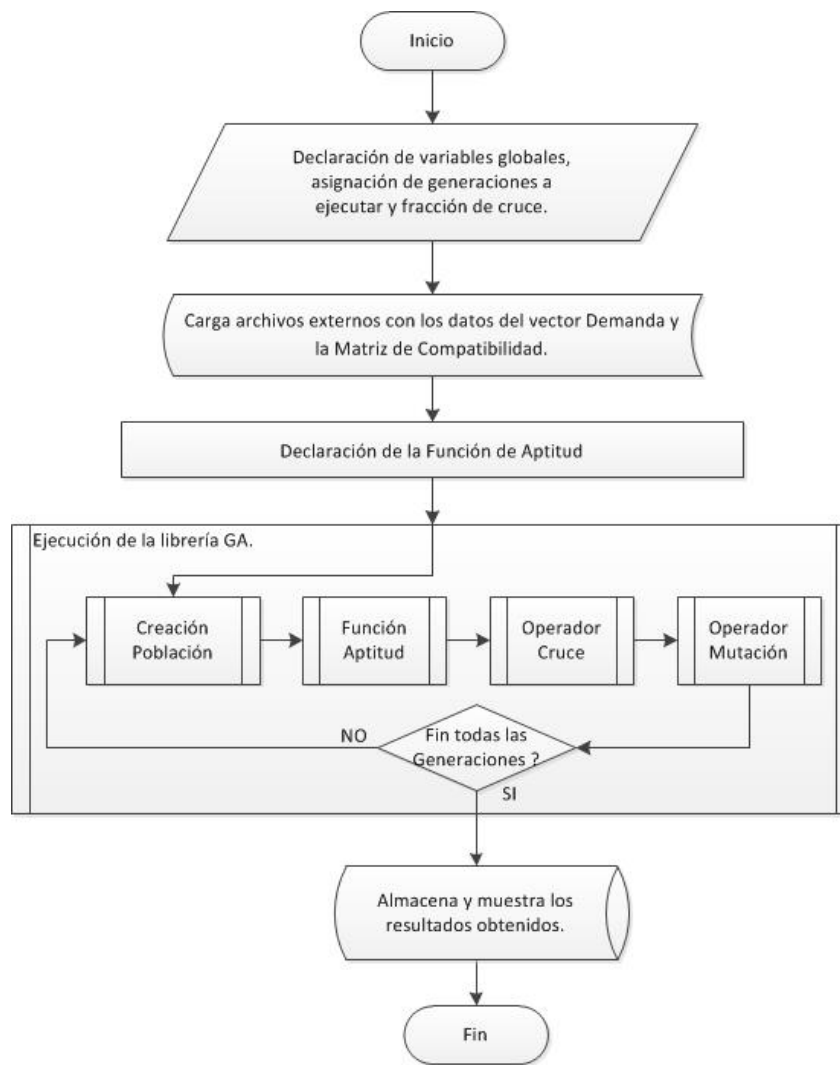


Figura 6.1: Diagrama de Bloques GA-Simple

6.1.2 Diagrama de bloques GA-Complejo tipo Paralelo

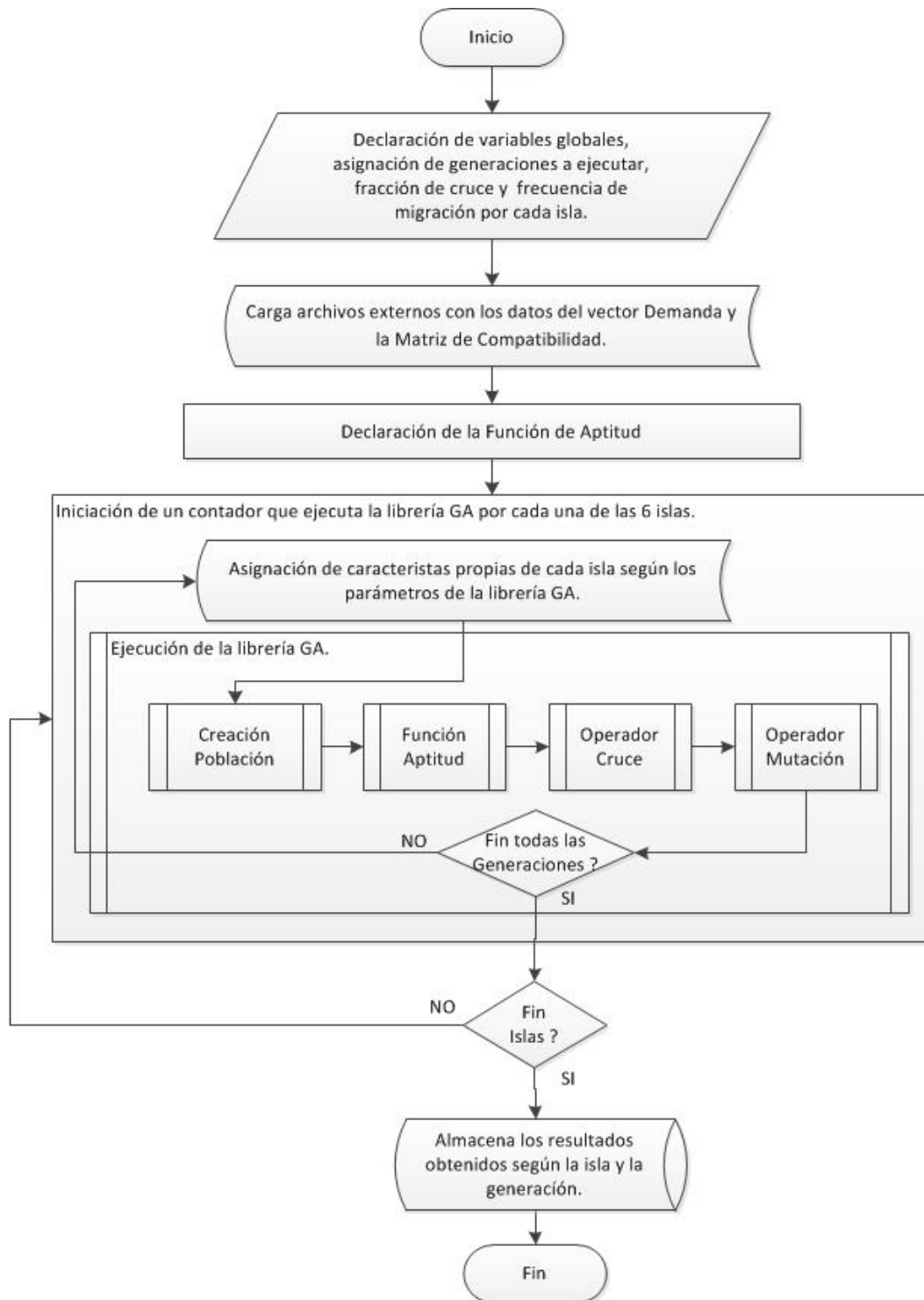


Figura 6.2: Diagrama de Bloques GA-Complejo Paralelo

6.1.3 Diagrama de bloques creación de la Población

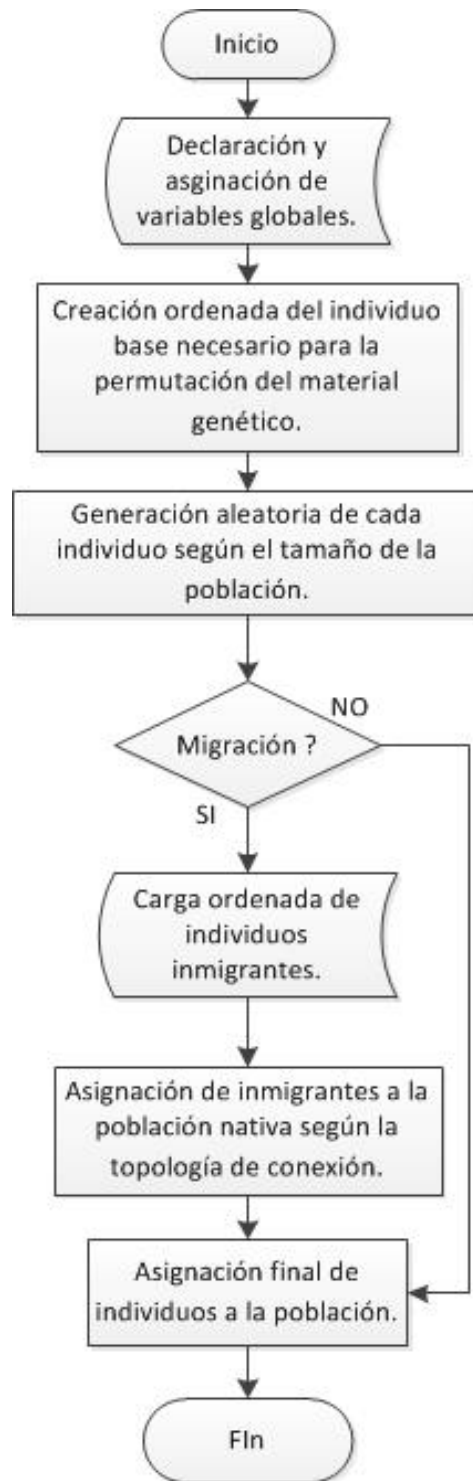


Figura 6.3: Diagrama Creación de la Población

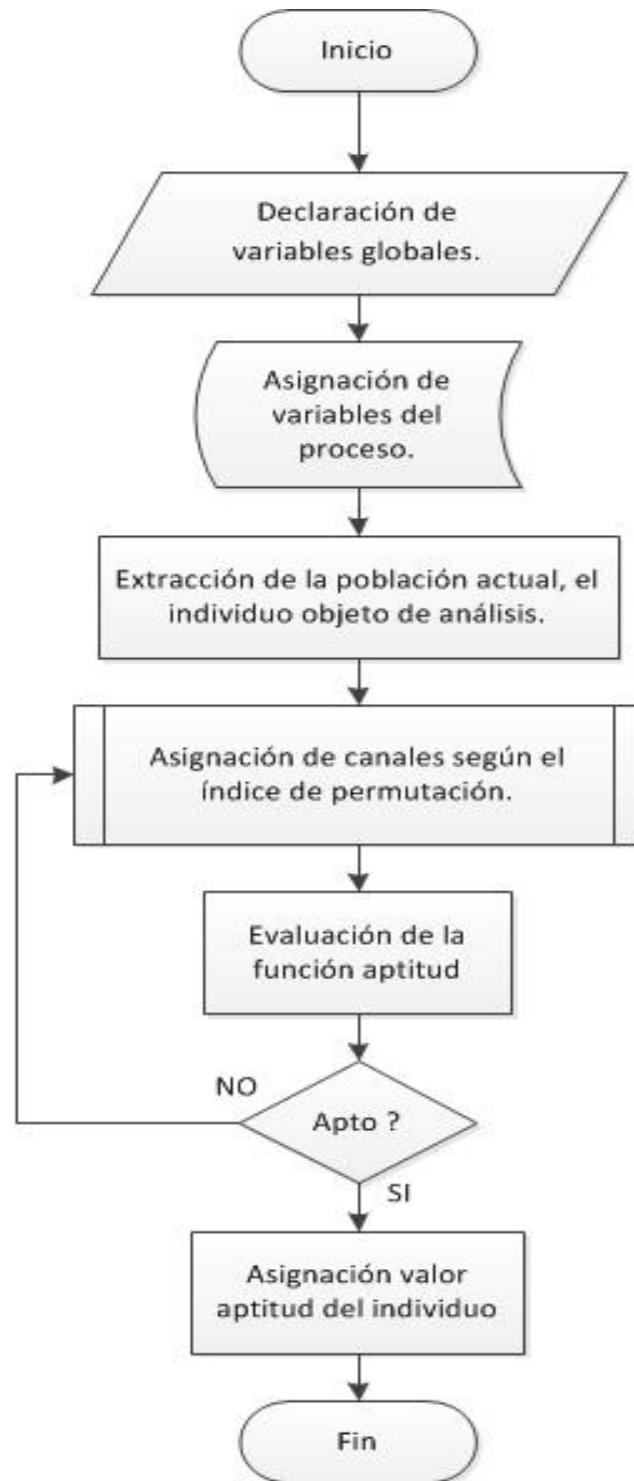
6.1.4 Diagrama de bloques Función Aptitud

Figura 6.4: Diagrama Función Aptitud

6.1.5 Diagrama de bloques Operador de Cruce

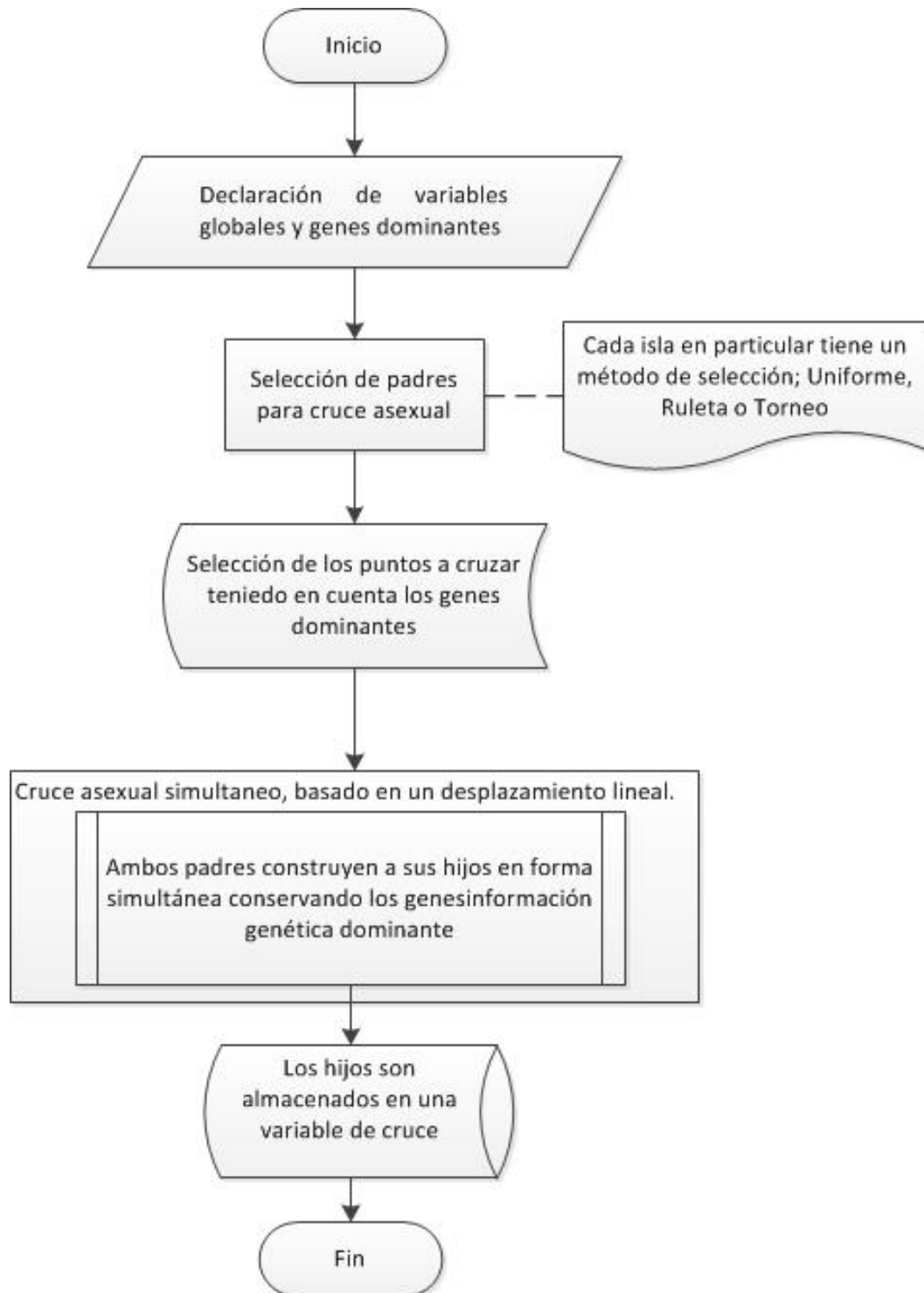


Figura 6.5: Diagrama Operador Cruce

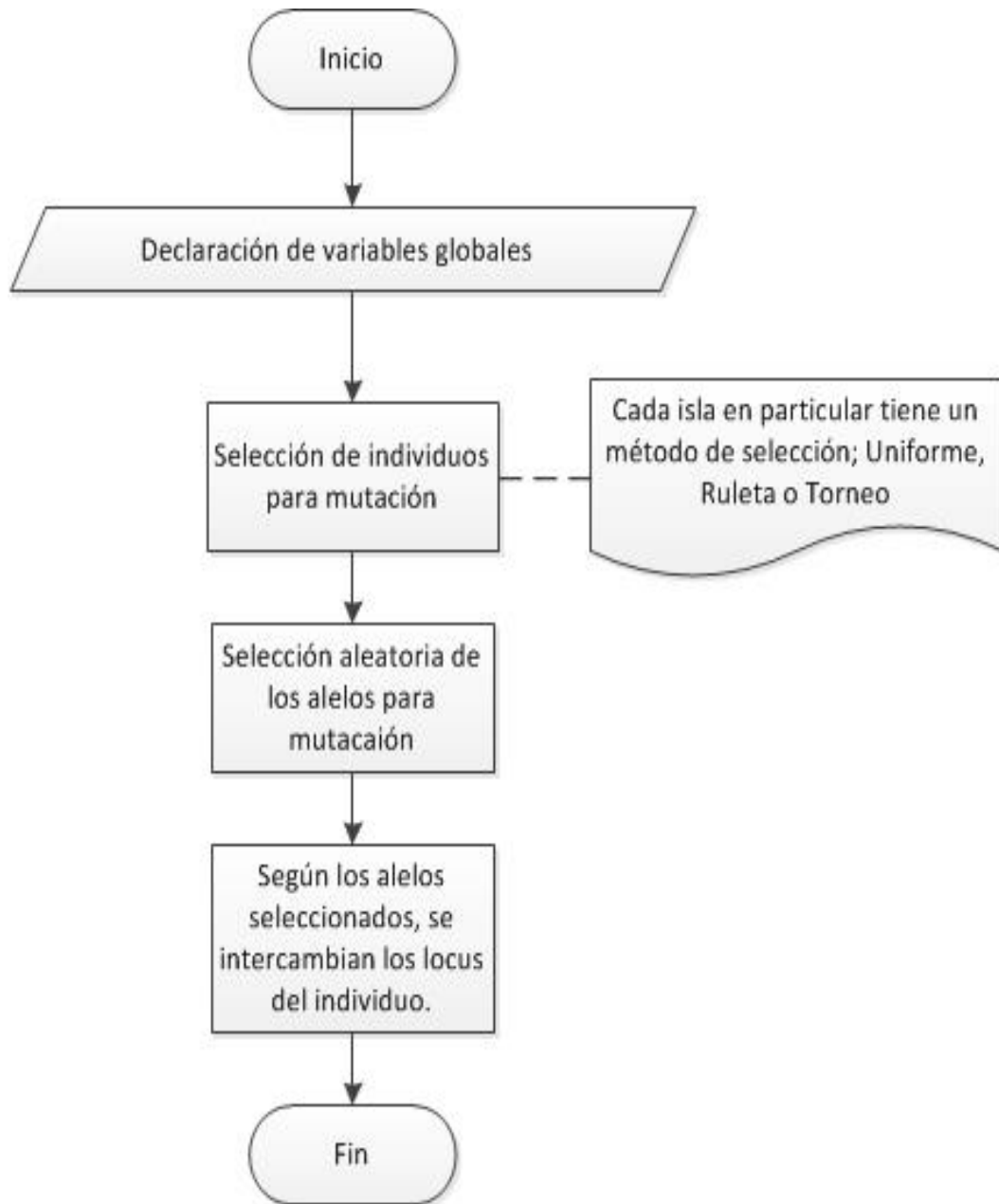
6.1.6 Diagrama de bloques Operador de Mutación

Figura 6.6: Diagrama Operador Mutación

6.2 Representación Gráfica de la Solución de CAP

6.2.1 Solución Gráfica para P1

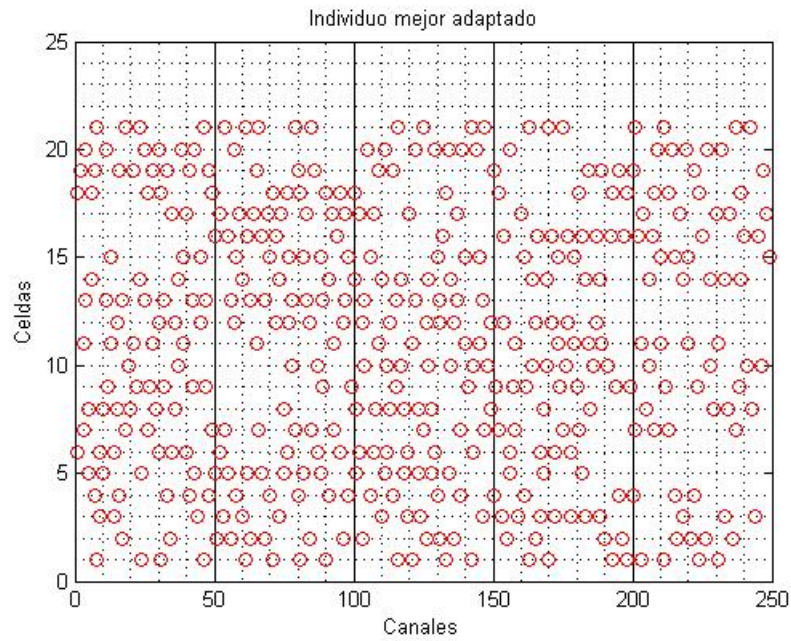


Figura 6.7: Representación Gráfica Solución CAP para P1 GA-Simple

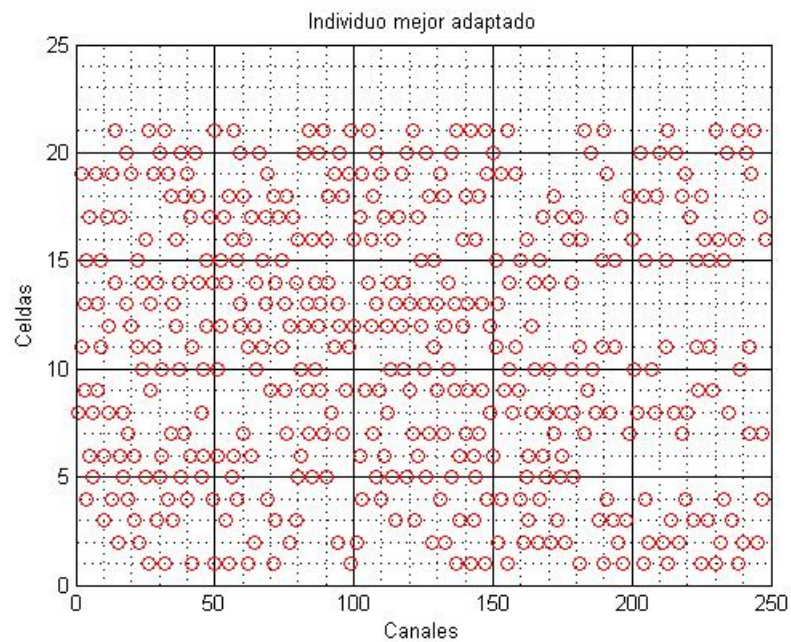


Figura 6.8: Representación Gráfica Solución CAP para P1 GA-Complejo Paralelo Estrella

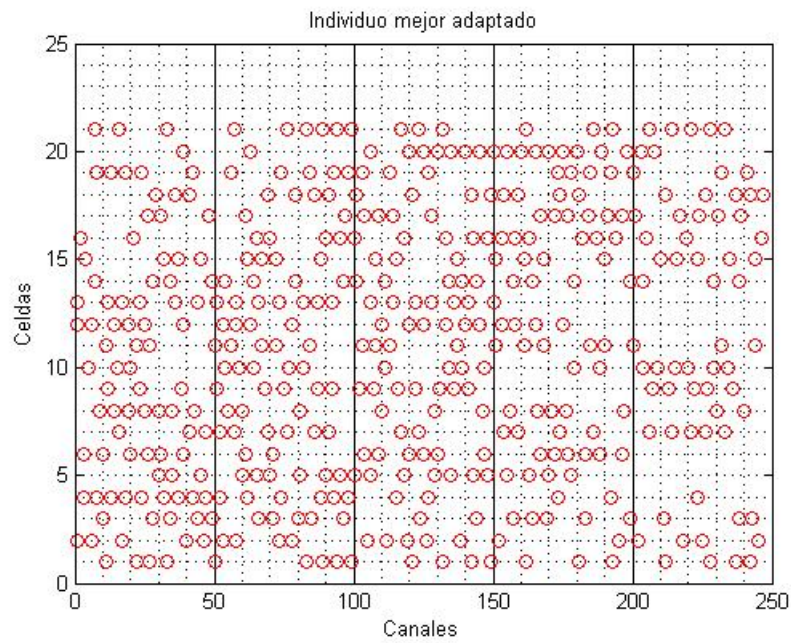


Figura 6.9: Representación Gráfica Solución CAP para P1 GA-Complejo Paralelo Anillo

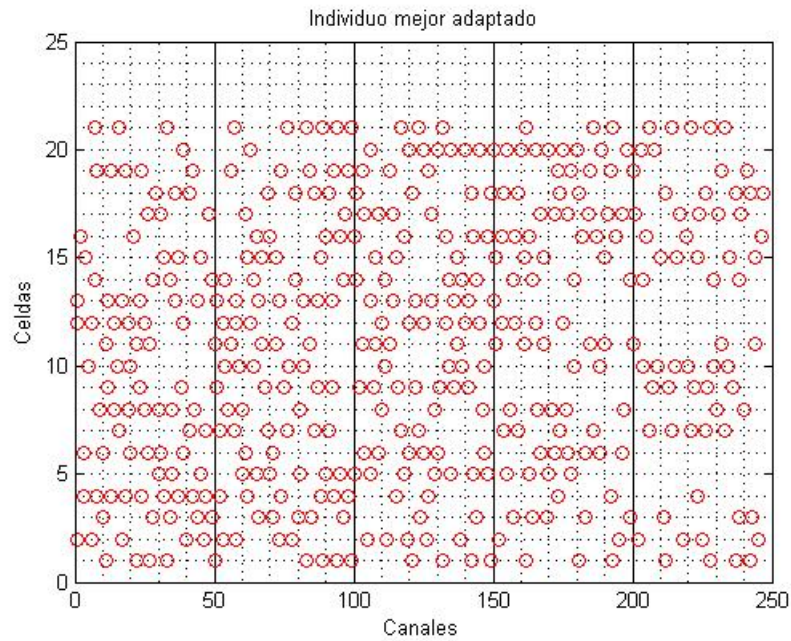


Figura 6.10: Representación Gráfica Solución CAP para P1 GA-Complejo Paralelo Malla

6.2.2 Solución Gráfica para P2

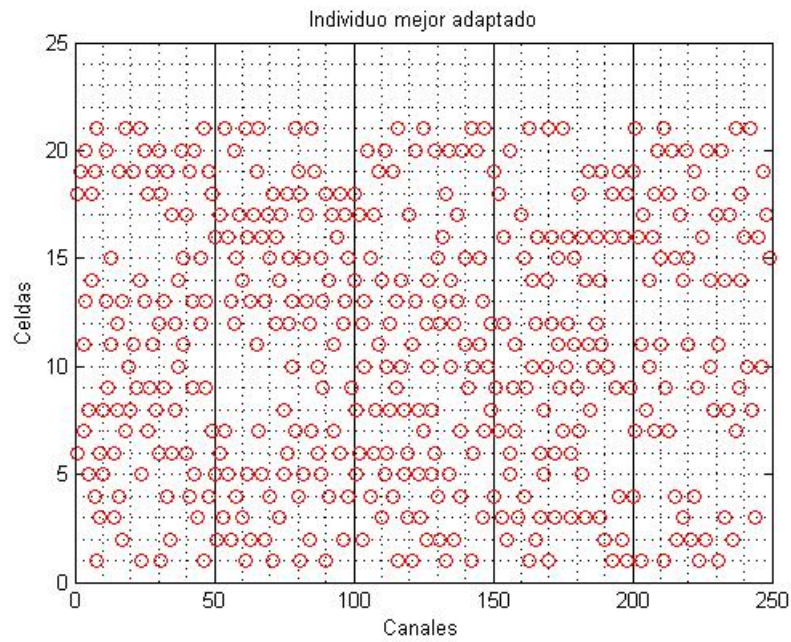


Figura 6.11: Representación Gráfica Solución CAP para P2 GA-Simple

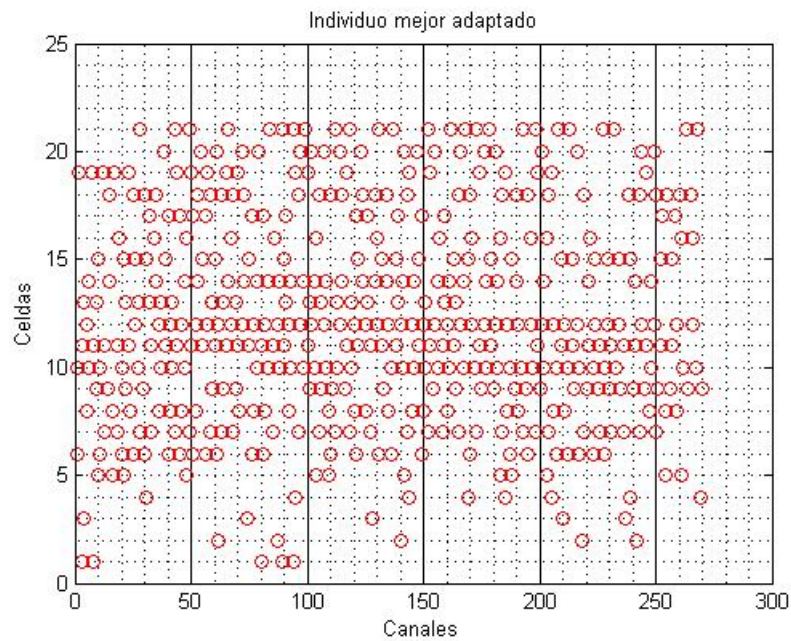


Figura 6.12: Representación Gráfica Solución CAP para P2 GA-Complejo Paralelo Estrella

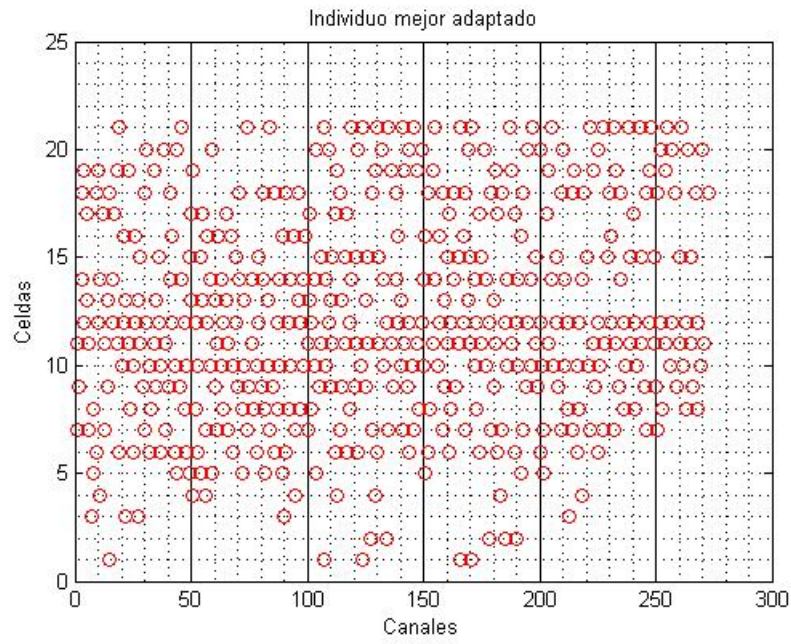


Figura 6.13: Representación Gráfica Solución CAP para P2 GA-Complejo Paralelo Anillo

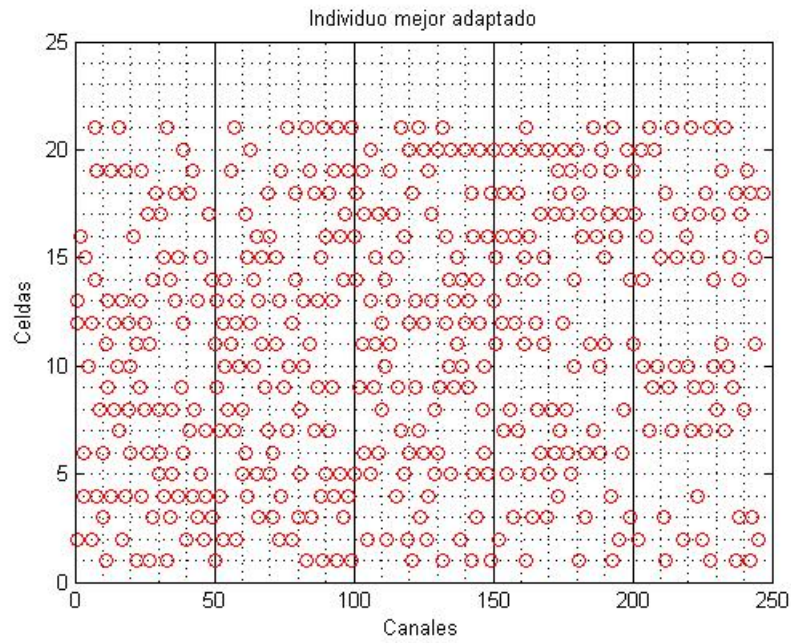


Figura 6.14: Representación Gráfica Solución CAP para P2 GA-Complejo Paralelo Malla

6.2.3 Solución Gráfica para P3

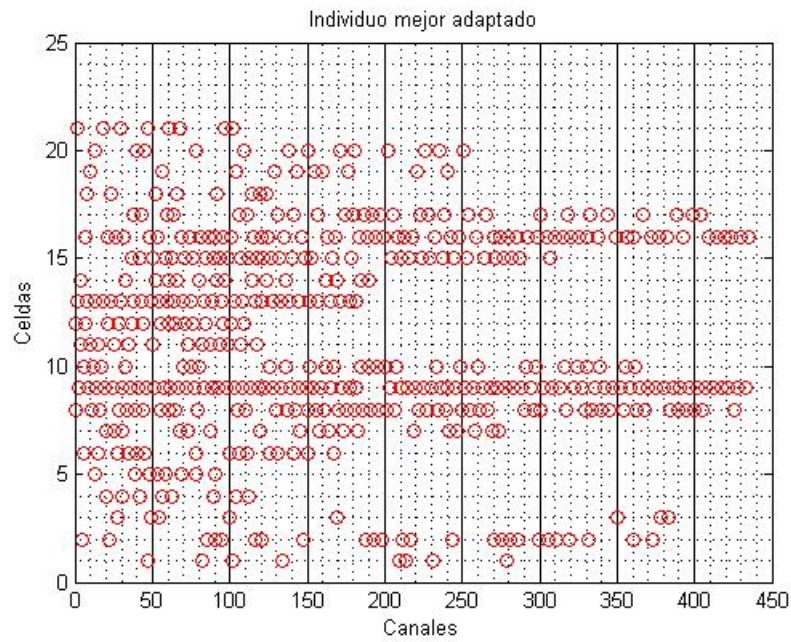


Figura 6.15: Representación Gráfica Solución CAP para P3 GA-Simple

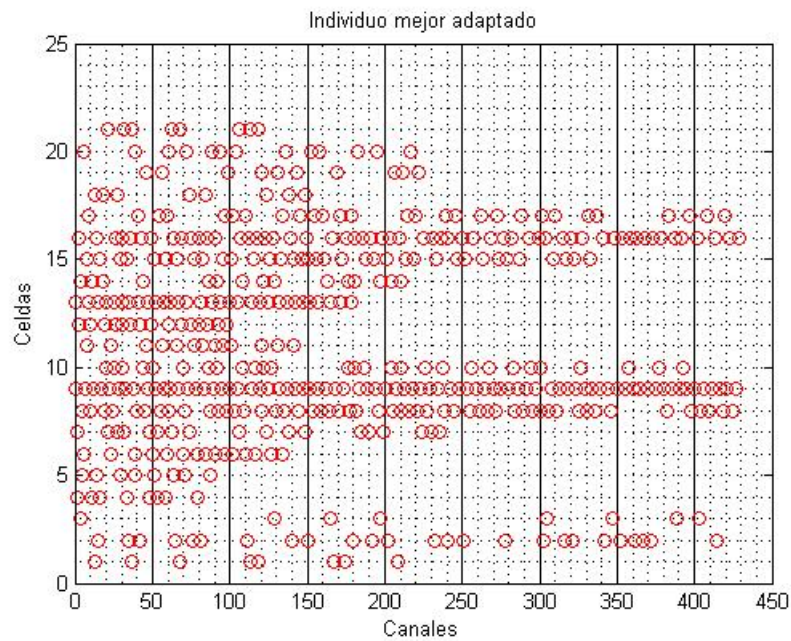


Figura 6.16: Representación Gráfica Solución CAP para P3 GA-Complejo Paralelo Estrella

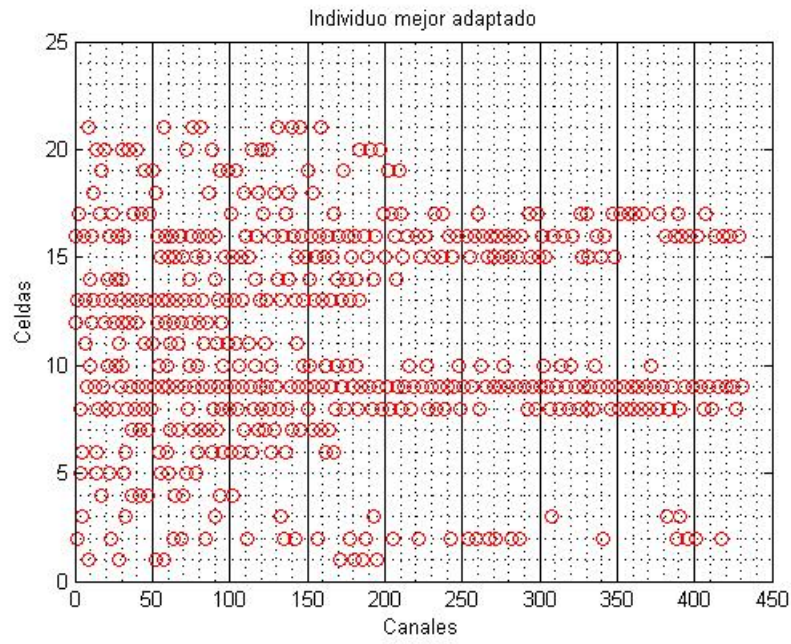


Figura 6.17: Representación Gráfica Solución CAP para P3 GA-Complejo Paralelo Anillo

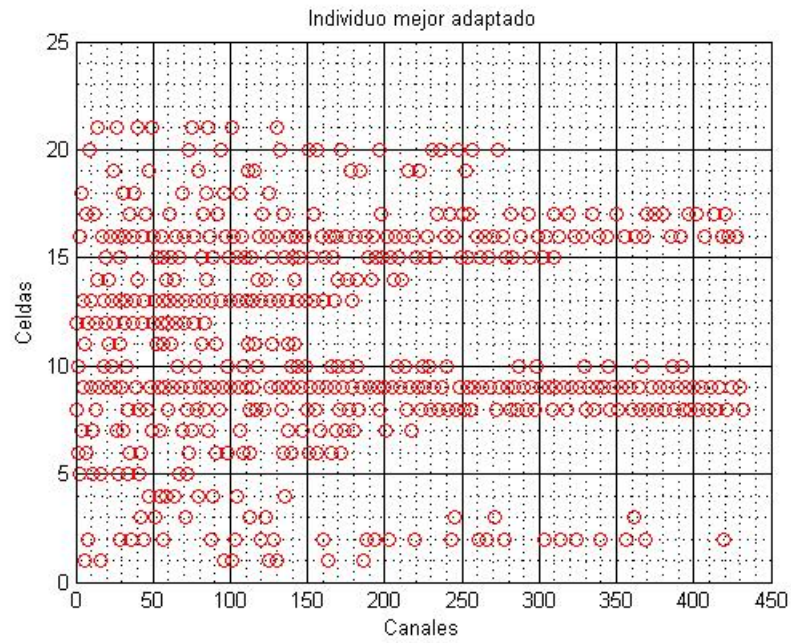


Figura 6.18: Representación Gráfica Solución CAP para P3 GA-Complejo Paralelo Malla

6.2.4 Solución Gráfica para P4

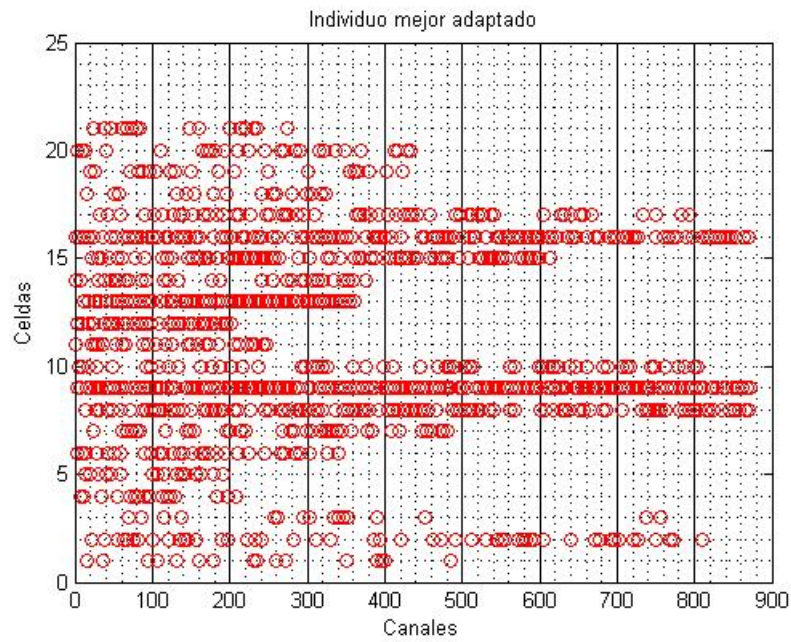


Figura 6.19: Representación Gráfica Solución CAP para P4 GA-Simple

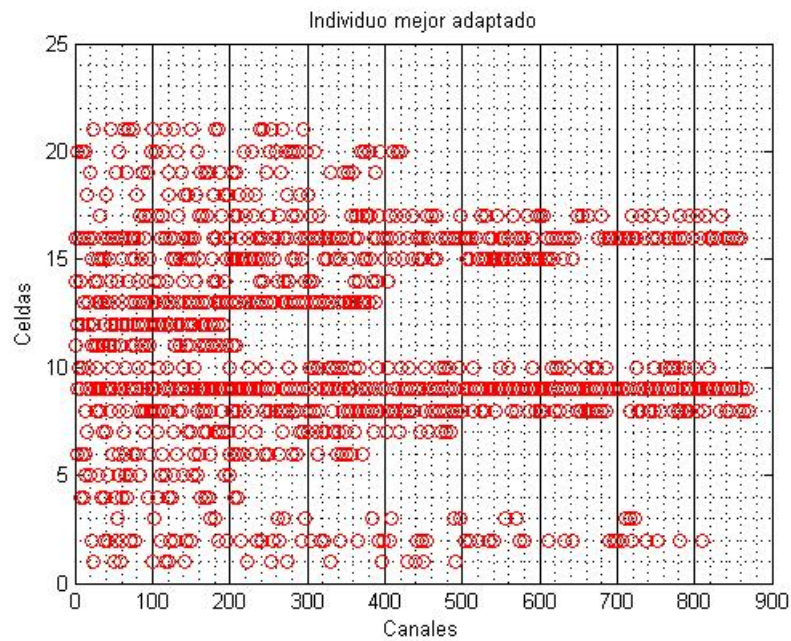


Figura 6.20: Representación Gráfica Solución CAP para P4 GA-Complejo Paralelo Estrella

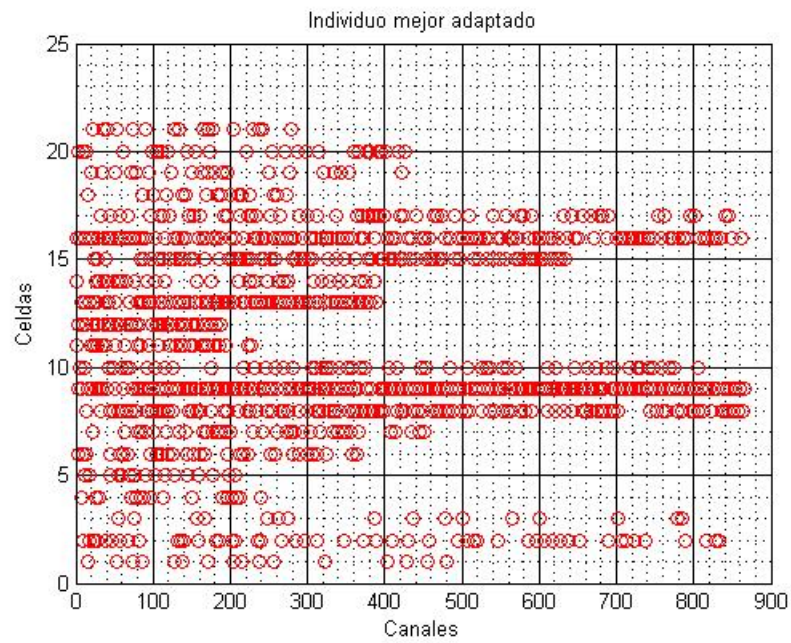


Figura 6.21: Representación Gráfica Solución CAP para P4 GA-Complejo Paralelo Anillo

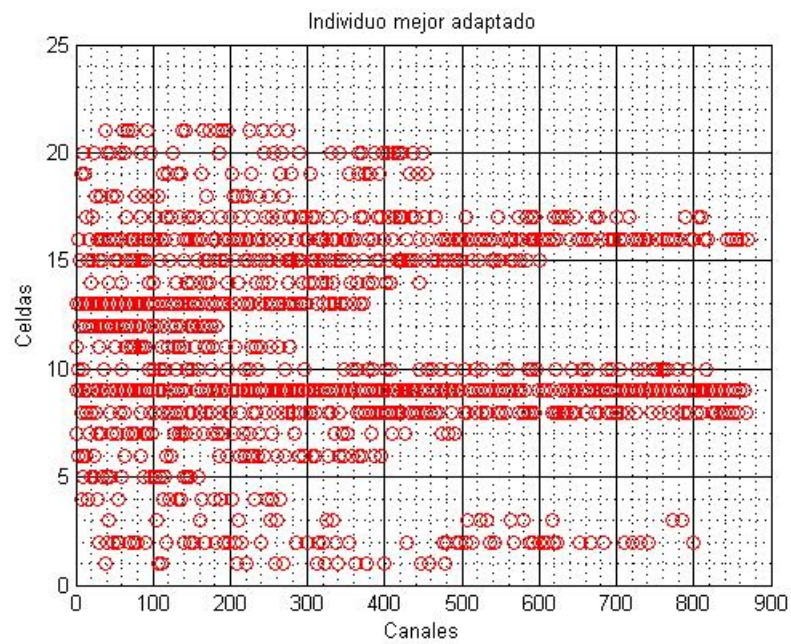


Figura 6.22: Representación Gráfica Solución CAP para P4 GA-Complejo Paralelo Malla

6.2.5 Solución Gráfica para P5

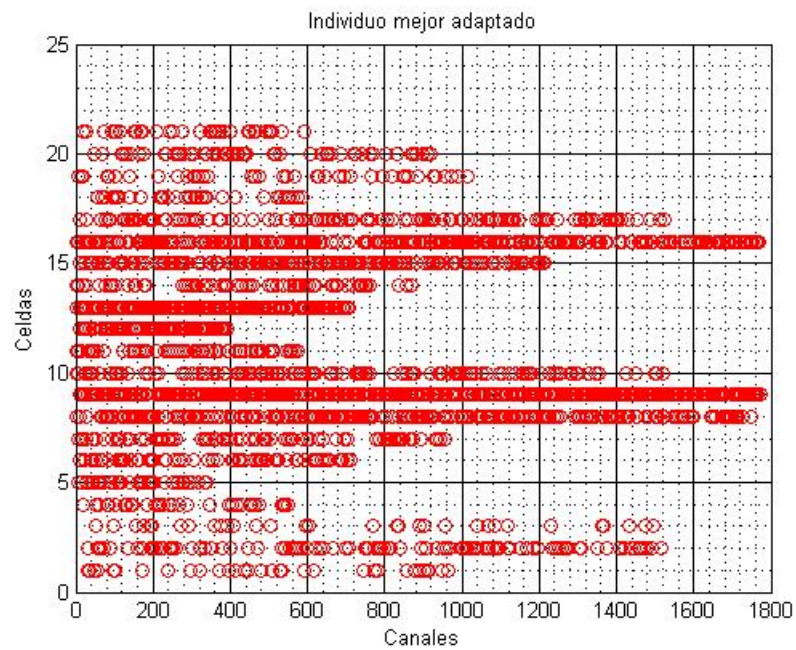


Figura 6.23: Representación Gráfica Solución CAP para P5 GA-Simple

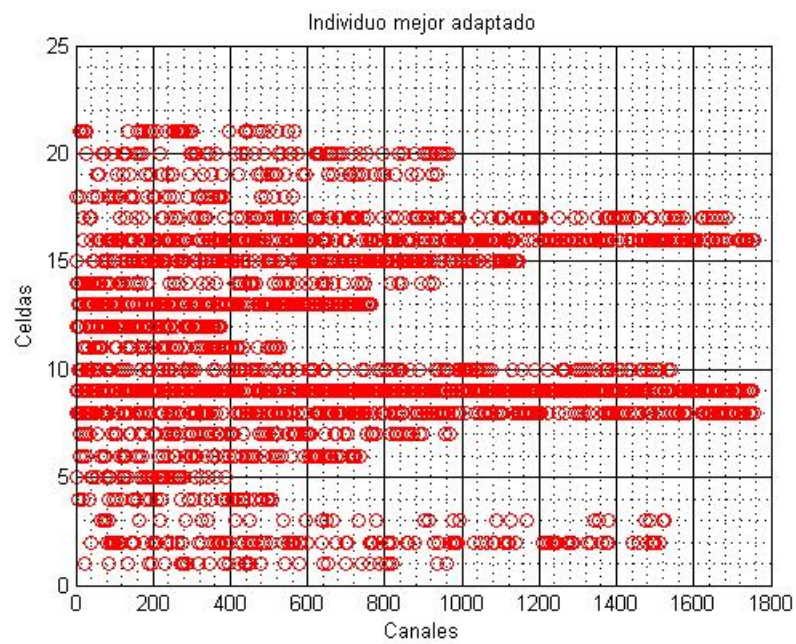


Figura 6.24: Representación Gráfica Solución CAP para P5 GA-Complejo Paralelo Estrella

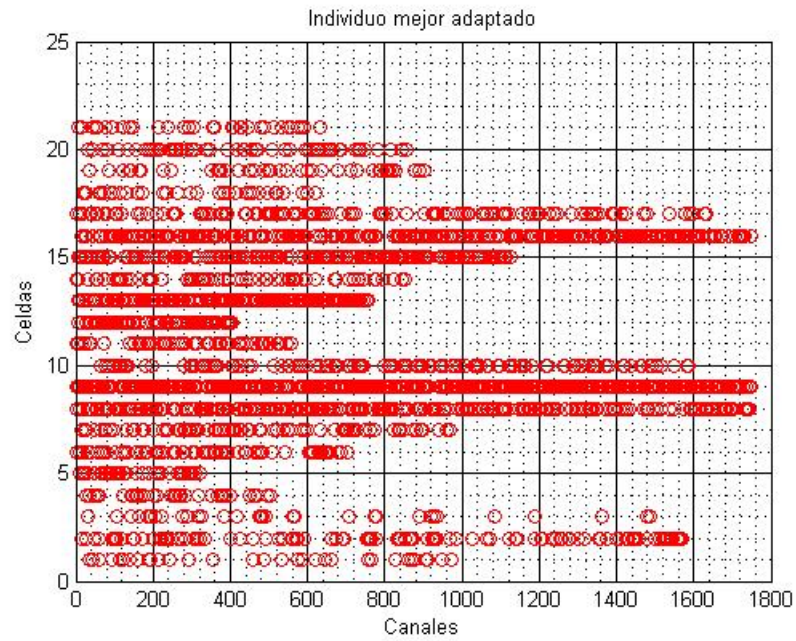


Figura 6.25: Representación Gráfica Solución CAP para P5 GA-Complejo Paralelo Anillo

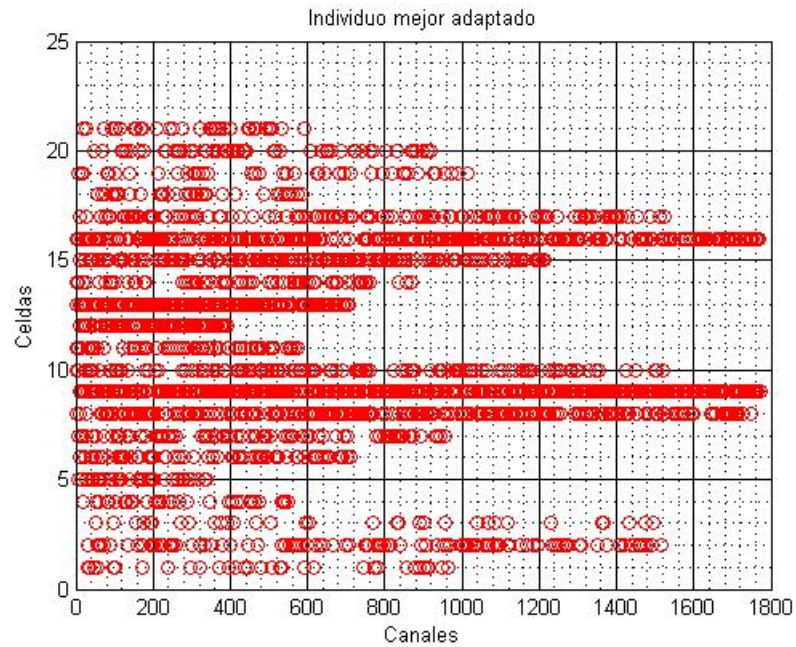


Figura 6.26: Representación Gráfica Solución CAP para P5 GA-Complejo Paralelo Malla

6.2.6 Solución Gráfica para P6

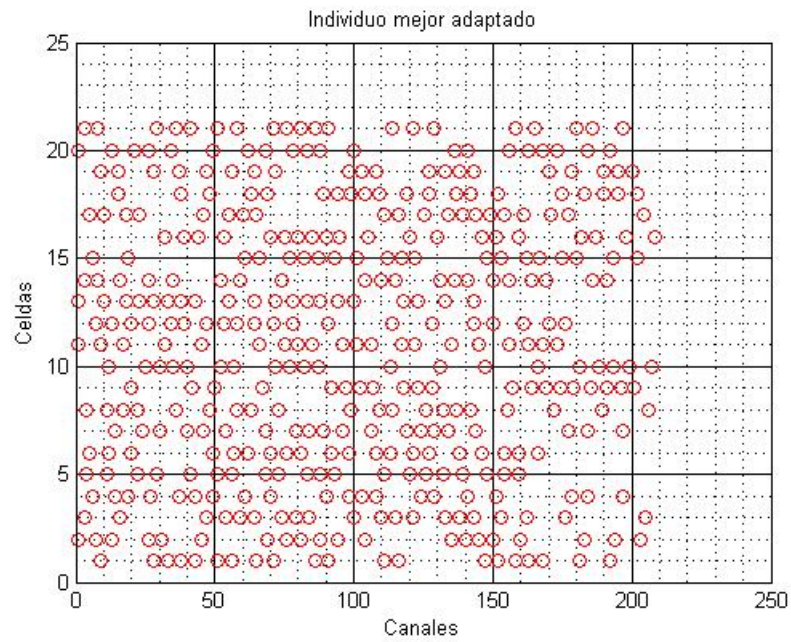


Figura 6.27: Representación Gráfica Solución CAP para P6 GA-Simple

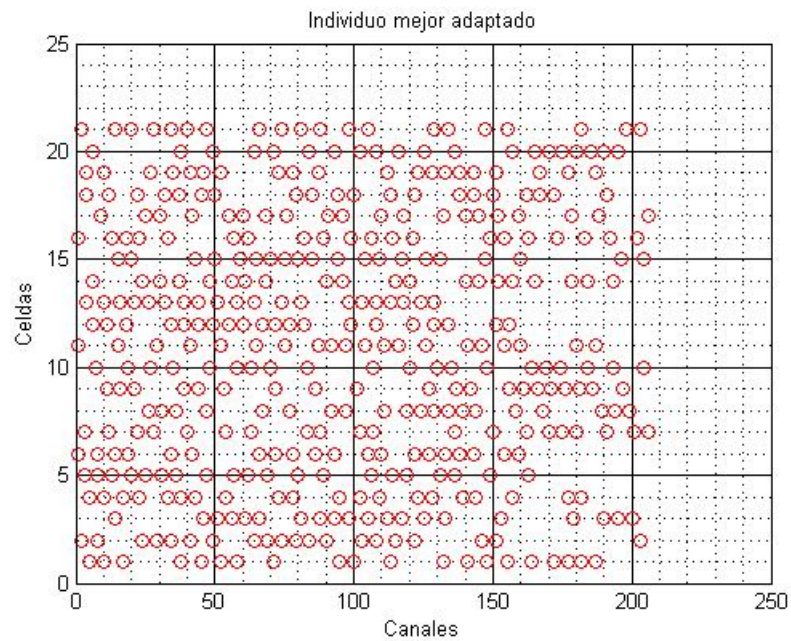


Figura 6.28: Representación Gráfica Solución CAP para P6 GA-Complejo Paralelo Estrella

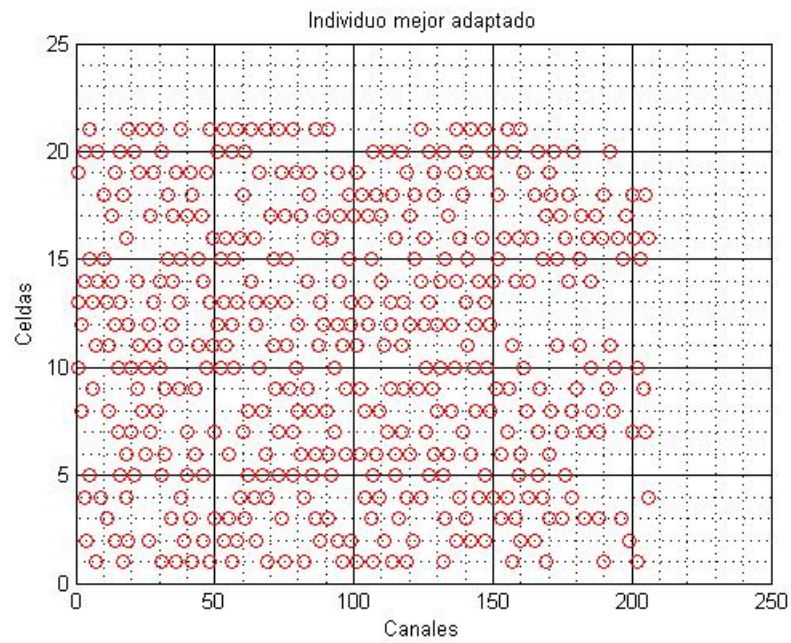


Figura 6.29: Representación Gráfica Solución CAP para P6 GA-Complejo Paralelo Anillo

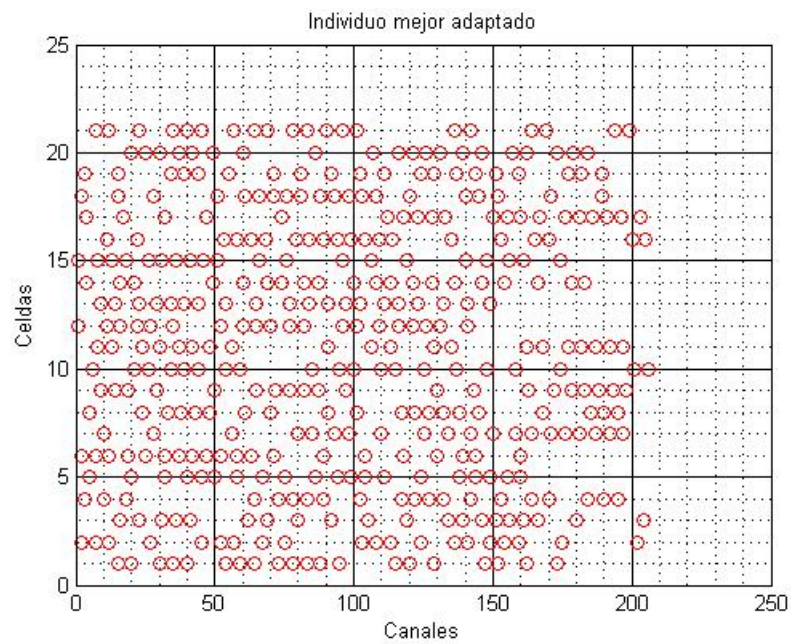


Figura 6.30: Representación Gráfica Solución CAP para P6 GA-Complejo Paralelo Malla

6.2.7 Solución Gráfica para P7

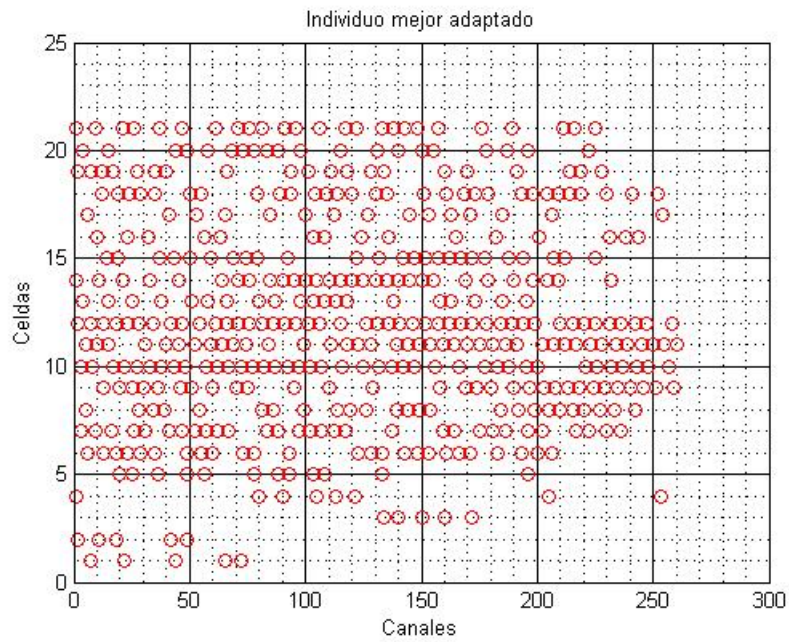


Figura 6.31: Representación Gráfica Solución CAP para P7 GA-Simple

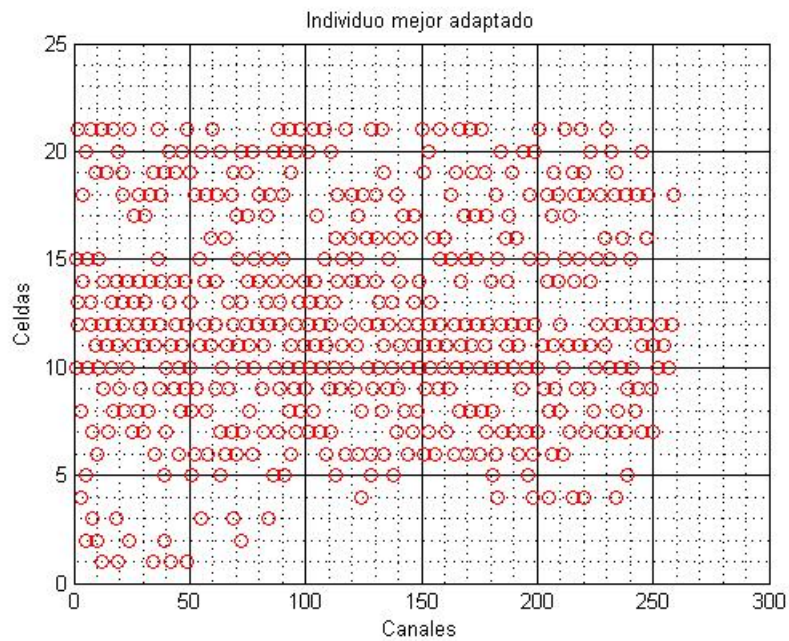


Figura 6.32: Representación Gráfica Solución CAP para P7 GA-Complejo Paralelo Estrella

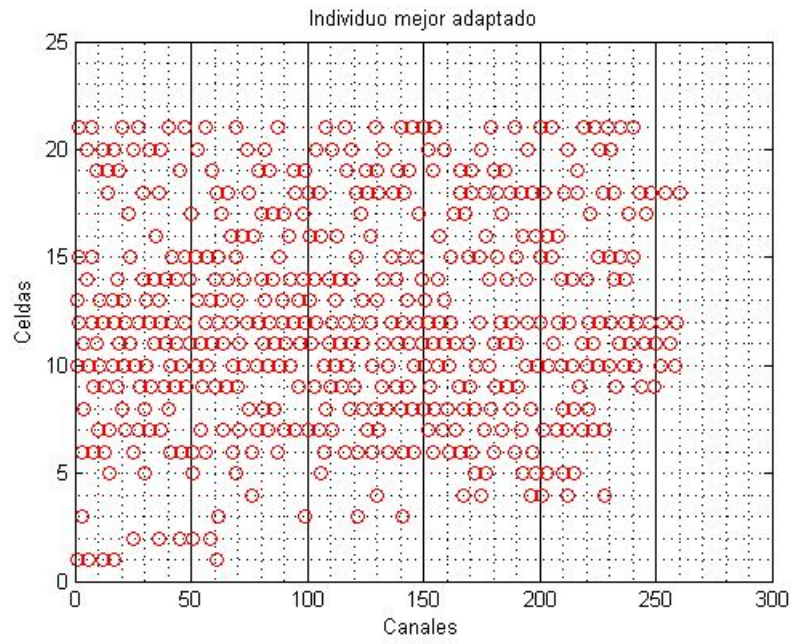


Figura 6.33: Representación Gráfica Solución CAP para P7 GA-Complejo Paralelo Anillo

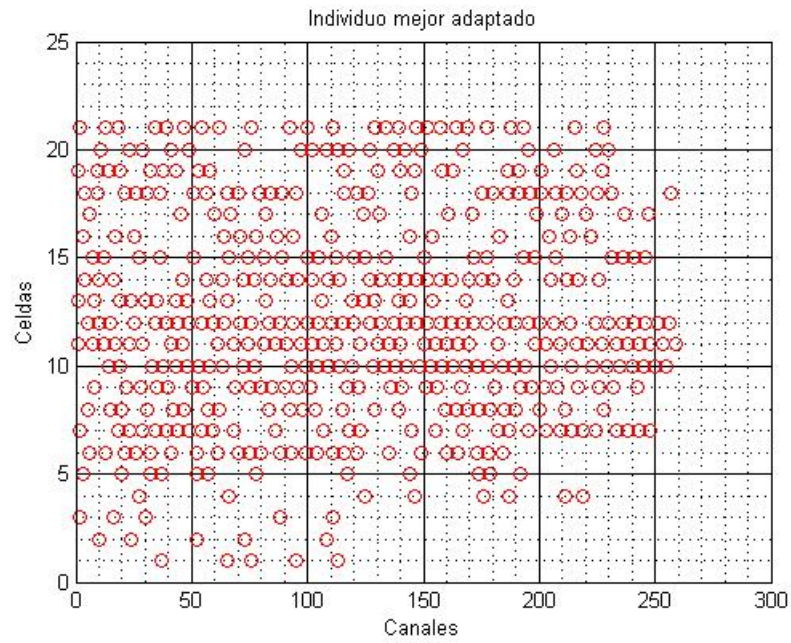


Figura 6.34: Representación Gráfica Solución CAP para P7 GA-Complejo Paralelo Malla

6.2.8 Solución Gráfica para P8

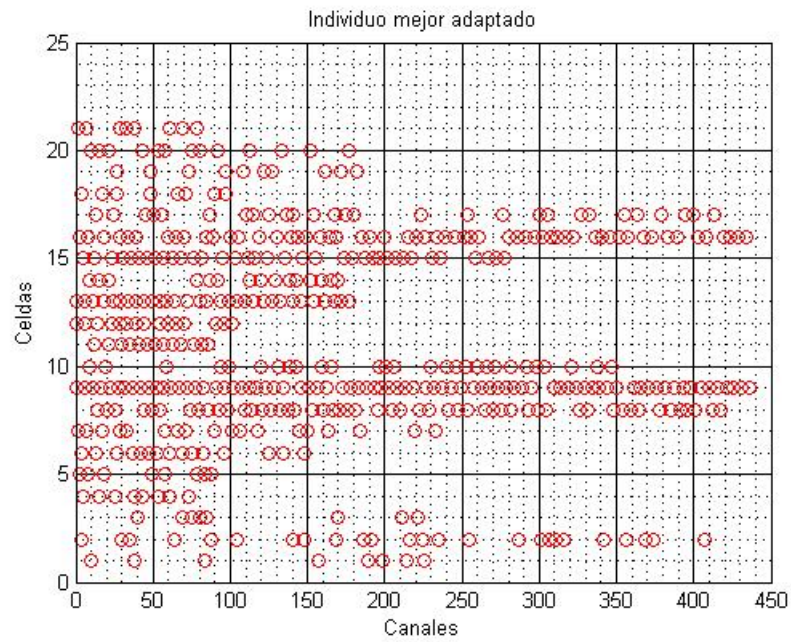


Figura 6.35: Representación Gráfica Solución CAP para P8 GA-Simple

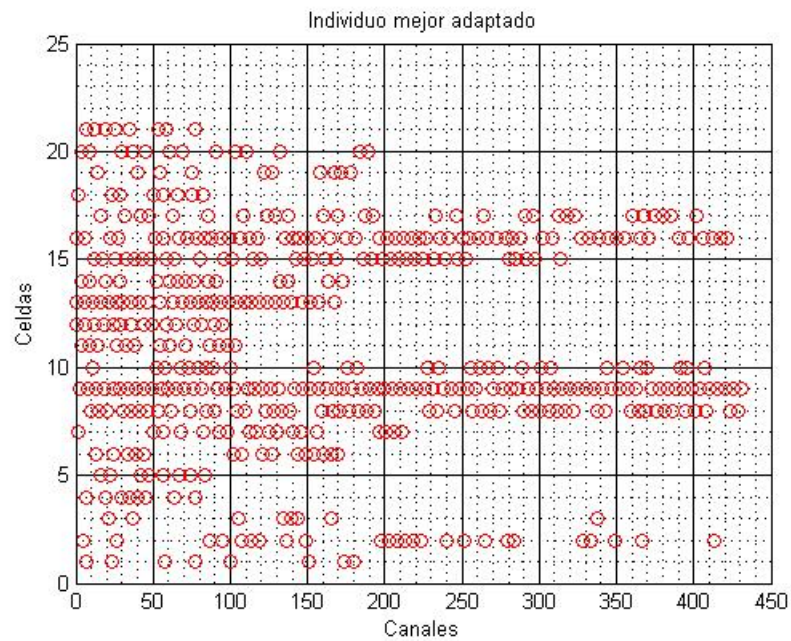


Figura 6.36: Representación Gráfica Solución CAP para P8 GA-Complejo Paralelo Estrella

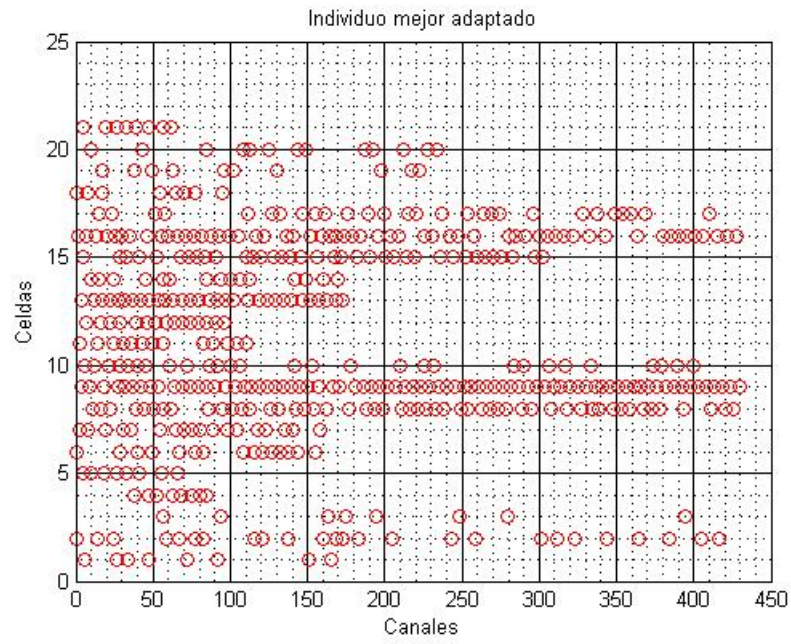


Figura 6.37: Representación Gráfica Solución CAP para P8 GA-Complejo Paralelo Anillo

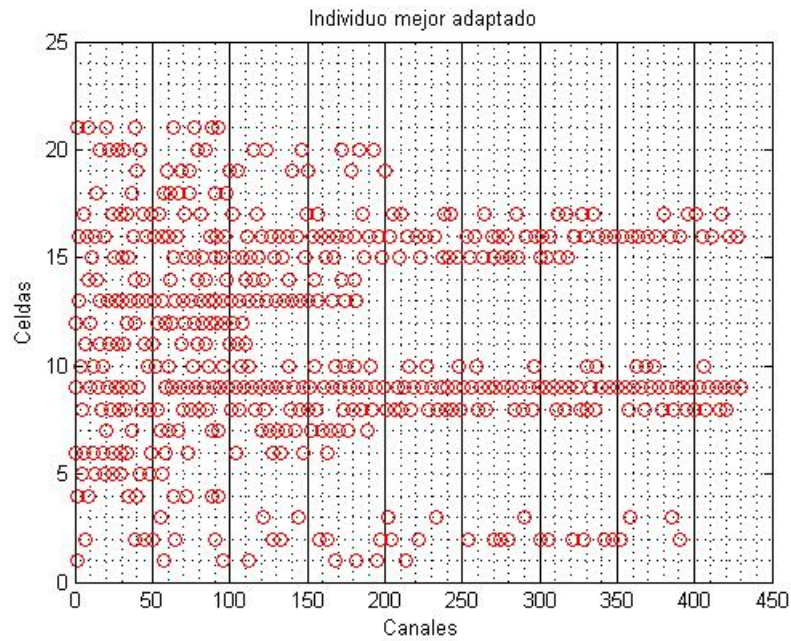


Figura 6.38: Representación Gráfica Solución CAP para P8 GA-Complejo Paralelo Malla

6.2.9 Solución Gráfica para P9

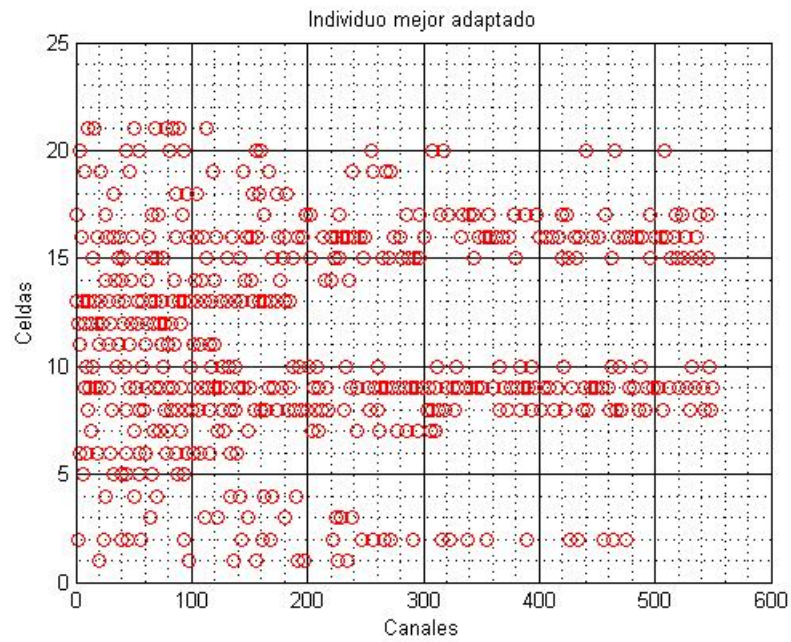


Figura 6.39: Representación Gráfica Solución CAP para P9 GA-Simple

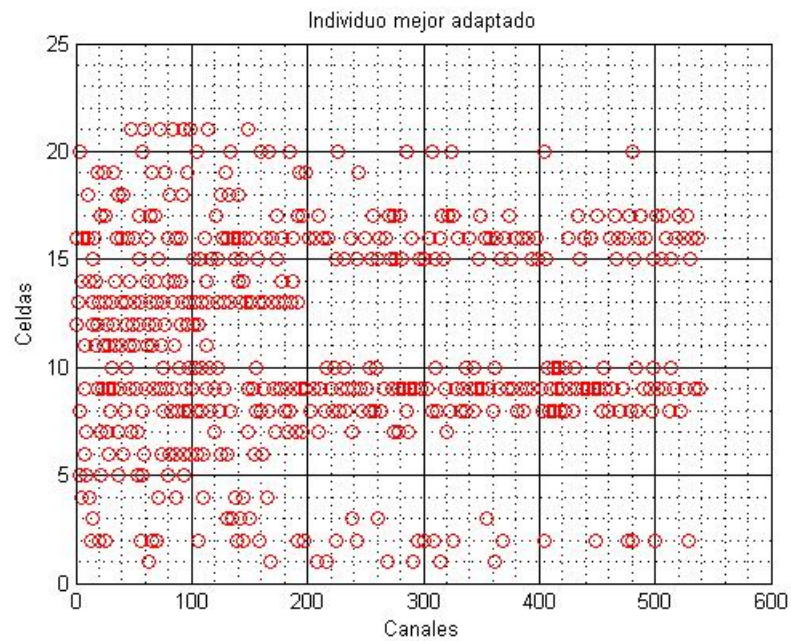


Figura 6.40: Representación Gráfica Solución CAP para P9 GA-Complejo Paralelo Estrella

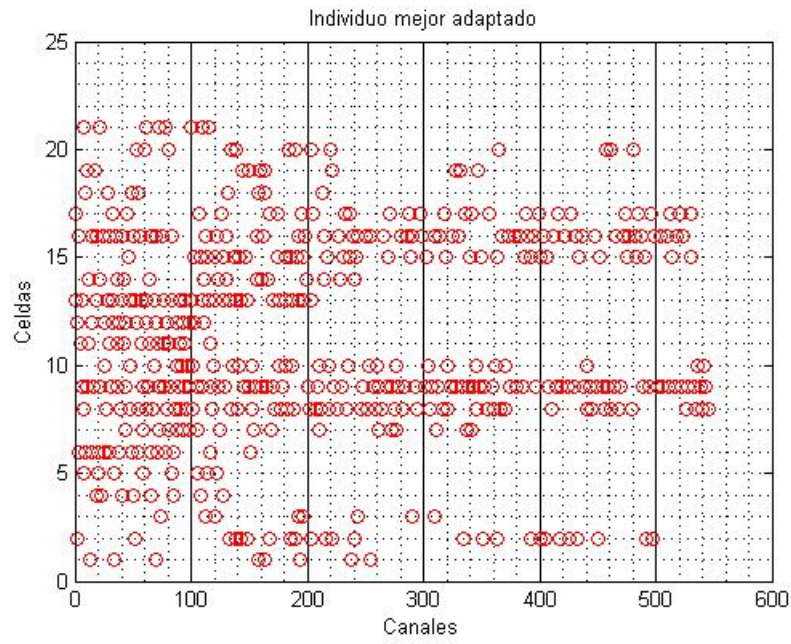


Figura 6.41: Representación Gráfica Solución CAP para P9 GA-Complejo Paralelo Anillo

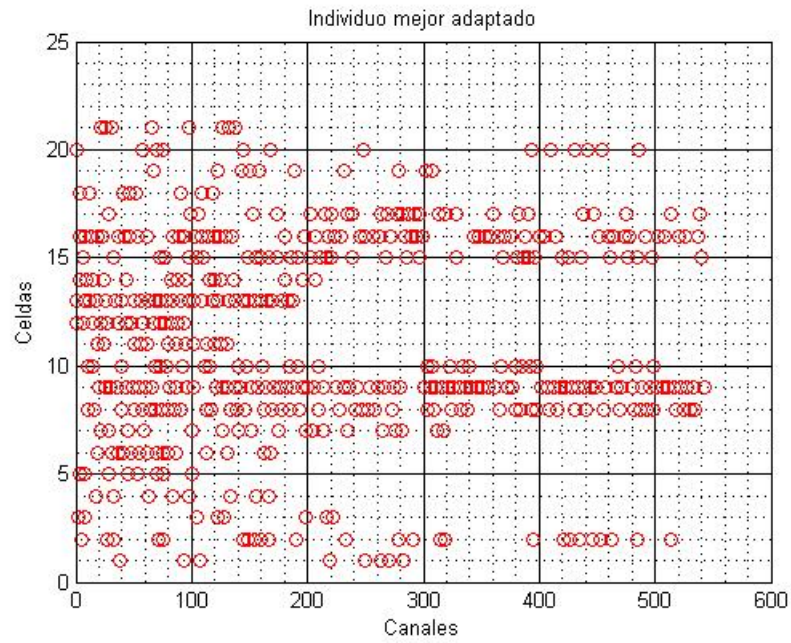


Figura 6.42: Representación Gráfica Solución CAP para P9 GA-Complejo Paralelo Malla

6.2.10 Solución Gráfica para P10

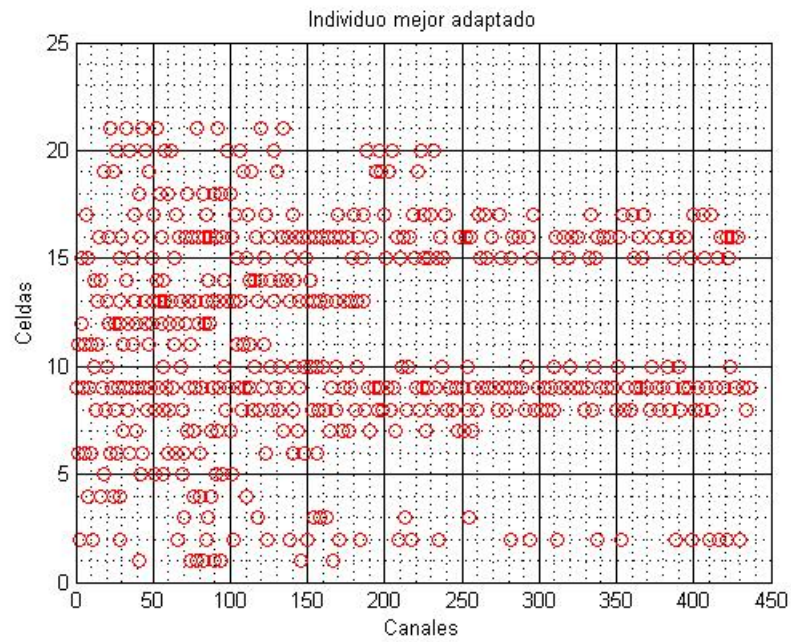


Figura 6.43: Representación Gráfica Solución CAP para P10 GA-Simple

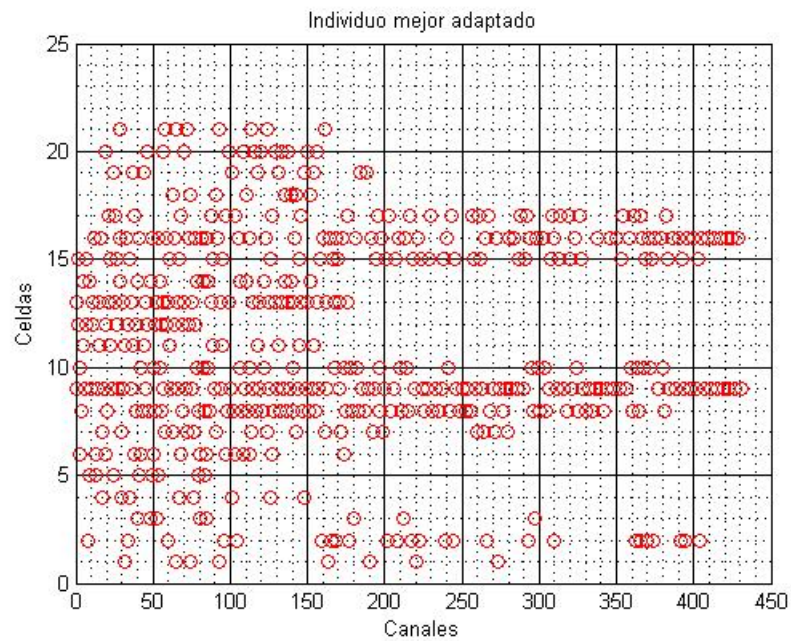


Figura 6.44: Representación Gráfica Solución CAP para P10 GA-Complejo Paralelo Estrella

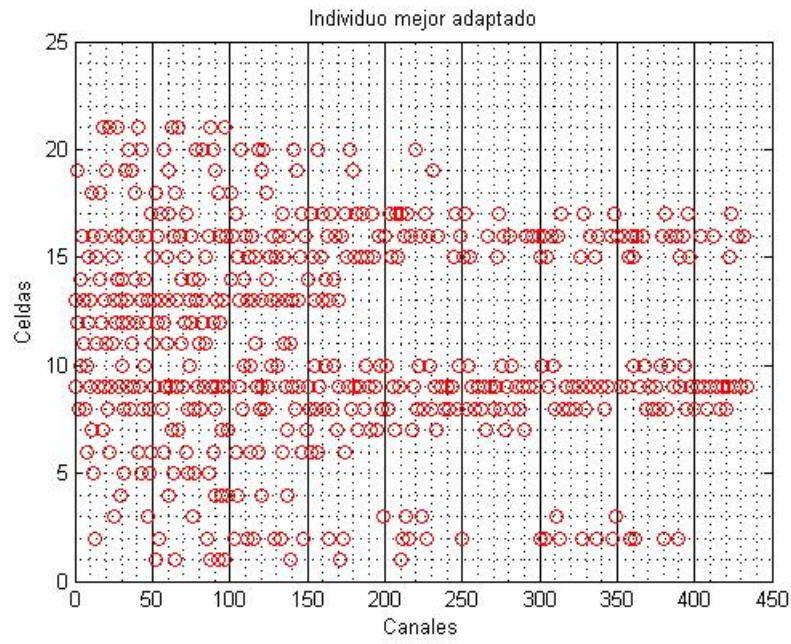


Figura 6.45: Representación Gráfica Solución CAP para P10 GA-Complejo Paralelo Anillo

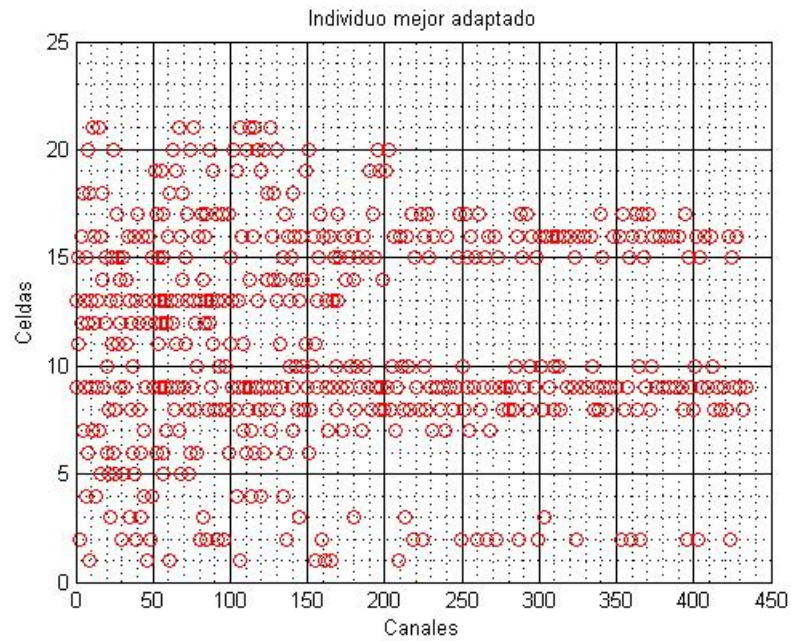


Figura 6.46: Representación Gráfica Solución CAP para P10 GA-Complejo Paralelo Malla

6.2.11 Solución Gráfica para P11

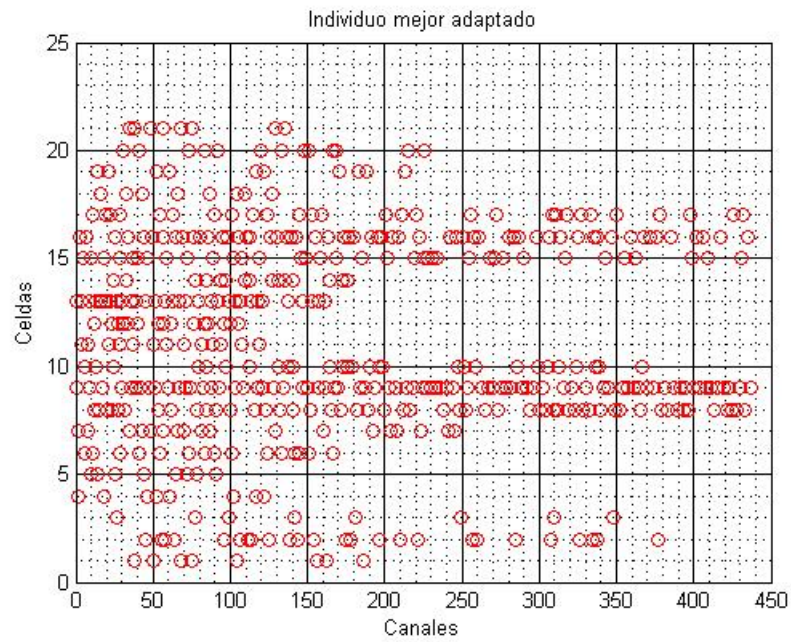


Figura 6.47: Representación Gráfica Solución CAP para P11 GA-Simple

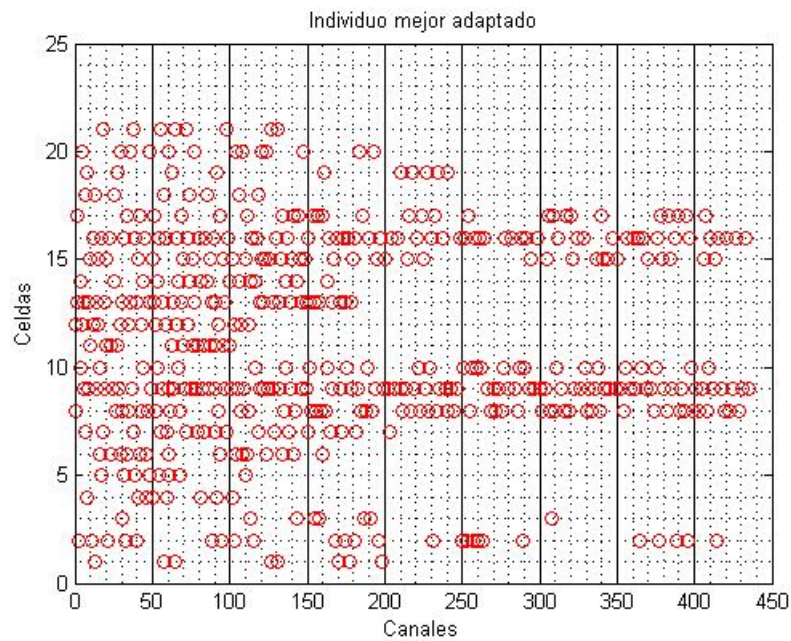


Figura 6.48: Representación Gráfica Solución CAP para P11 GA-Complejo Paralelo Estrella

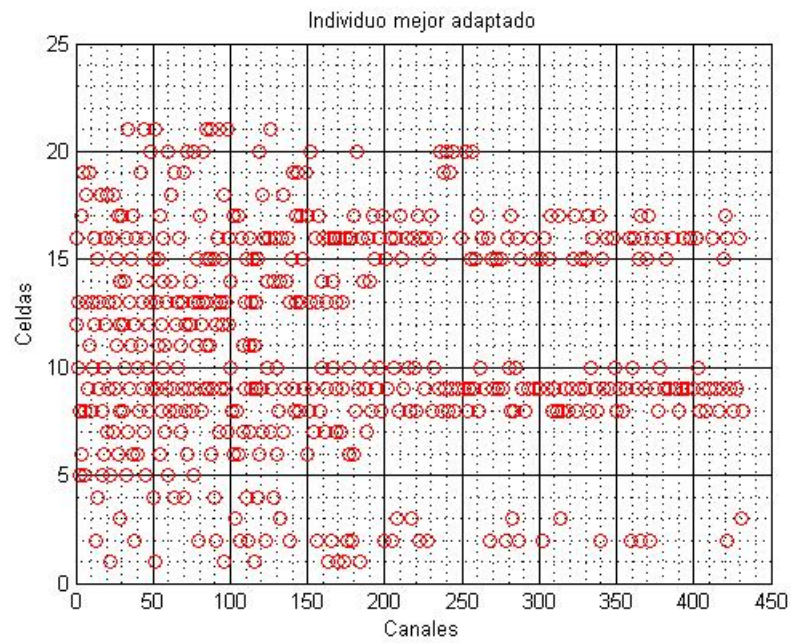


Figura 6.49: Representación Gráfica Solución CAP para P11 GA-Complejo Paralelo Anillo

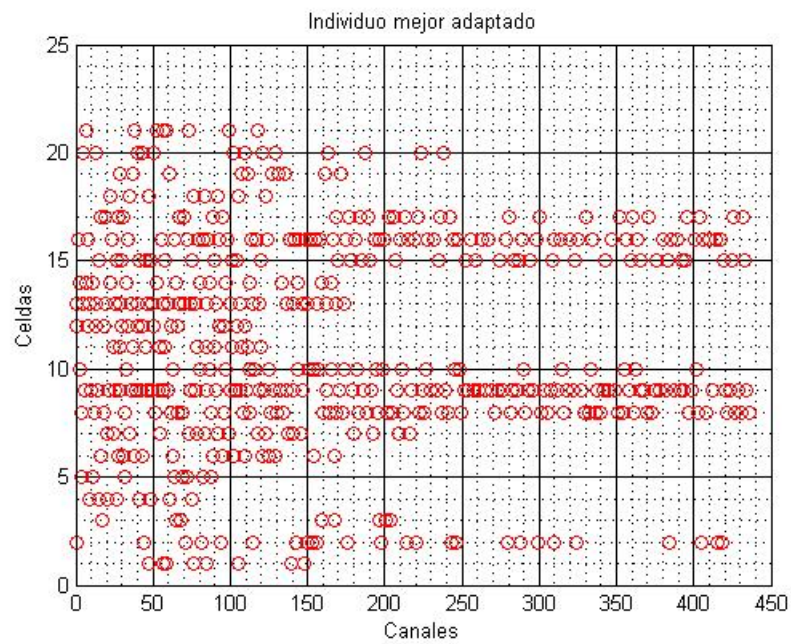


Figura 6.50: Representación Gráfica Solución CAP para P11 GA-Complejo Paralelo Malla