

ALGORITMO DE ASIGNACIÓN DE FRECUENCIAS POR ENJAMBRE DE PARTÍCULAS EN UNA ARQUITECTURA CLUSTER

Viviana Paola Leyva Cifuentes

Director

Ing. Carlos Iván Páez Rueda Ph.D.

Pontificia Universidad Javeriana
Facultad de Ingeniería
Departamento de Electrónica



24 de Noviembre de 2017

Resumen

Este trabajo de grado solucionó algunos problema de asignación fija de canales en frecuencia para redes simples y complejas, usando *Particle Swarm Optimization* en una arquitectura computacional convencional y en una distribuida en paralelo de alto rendimiento (HPC) de la Pontificia Universidad Javeriana, buscando mejorar el desempeño de la red en términos de interferencia. Se diseñaron otros problemas de asignación de canales para Wi-Fi y se implementó una estrategia de inicialización que ayuda a la convergencia del algoritmo.

Dedicado a mi familia, gracias por su apoyo durante toda mi carrera.
A ti amor mio, por ayudarme a levantar cuando pensé que no lo lograría.
A Páez por su infinita paciencia, pocos directores como usted.

Índice general

1. Introducción	8
2. Marco Teórico	10
2.1. Problemas de asignación de frecuencias y métodos heurísticos	10
2.2. Optimización por enjambre de partículas	11
2.2.1. Concepto y planteamiento del algoritmo	11
2.2.2. Variantes de PSO	13
2.3. Arquitecturas computacionales	14
2.3.1. Clusters	14
2.3.2. ZINE Javeriana	15
2.3.3. MPI (<i>Message Passing Interface</i>)	16
3. Objetivos	17
3.1. Objetivo específico 1	17
3.2. Objetivo específico 2	17
3.3. Objetivo específico 3	18
3.4. Objetivo específico 4	18
4. Descripción de los problemas de asignación	19
4.1. Problemas de asignación de frecuencias	19
4.1.1. Restricciones	19
4.2. Problemas de referencia	20
4.2.1. Problema EX1: 4 celdas, 11 canales	20
4.2.2. Problema de Philadelphia P3: 21 celdas, 533 canales	20
4.3. Problema de asignación de frecuencias en la Facultad de Ingeniería de la Pontifica Universidad Javeriana	22
4.3.1. Medidas de potencia y distancia	22
4.3.2. Diseño de las matrices de compatibilidad electromagnética	29
5. Diseño	33
5.1. Descripción general	33
5.1.1. Diseño de variables y constantes del programa	33
5.1.2. Diseño de evaluación de la función objetivo y actualización de óptimos	35
5.1.3. Diseño de actualización de posición y velocidad de las partículas	40
5.2. Modificación del algoritmo	41
5.3. Descripción del algoritmo en MPI	46

6. Pruebas y evaluación	48
6.1. EX1: 4 celdas,11 canales	48
6.1.1. Inicialización aleatoria	48
6.1.2. Generación aleatoria	50
6.1.3. Inicialización por intervalos y celdas <i>HotSpot</i>	51
6.2. PHI3: 21 celdas, 533 canales	51
6.2.1. Inicialización aleatoria	53
6.2.2. Generación aleatoria	56
6.2.3. Inicialización por intervalos y celdas <i>HotSpot</i>	56
6.3. PUJ1: 11 celdas, 11 canales	62
6.3.1. Inicialización aleatoria	62
6.3.2. Generación aleatoria	63
6.4. PUJ2: 11 celdas, 11 canales	64
6.4.1. Inicialización aleatoria	64
6.4.2. Generación aleatoria	67
6.5. PUJ3: 11 celdas, 11 canales	68
6.5.1. Inicialización aleatoria	68
6.5.2. Generación aleatoria	71
6.5.3. Inicialización por intervalos y celdas <i>HotSpot</i>	72
6.6. Comparación de la asignación original y las asignaciones del trabajo de grado	75
7. Conclusiones	78
8. Anexos	79
8.1. Anexo 1	79
8.2. Anexo 2	79
8.3. Anexo 3	79
8.4. Anexo 4	79

Índice de figuras

2.1. Conceptos de PSO. Tomado de [19, 20]	12
2.2. Diagrama de flujo de PSO en paralelo [20]	13
2.3. Componentes de un cluster. Tomado de [6]	14
2.4. Características de <i>HPC</i> y <i>HTC</i> . Tomado de [6]	15
2.5. Software disponible en Zine	15
4.1. Concepto de asignación de frecuencias	19
4.2. Ejemplo de asignación según C y d.	20
4.3. Aplicación móvil, Wifi Analyzer	22
4.4. Ubicación espacial de los AP de la facultad de ingeniería de la PUJ	23
4.5. Niveles de potencia y distancia de AP 1	24
4.6. Niveles de potencia y distancia de AP 2	25
4.7. Niveles de potencia y distancia de AP 3	25
4.8. Niveles de potencia y distancia de AP 4	26
4.9. Niveles de potencia y distancia de AP 5	26
4.10. Niveles de potencia y distancia de AP 6	27
4.11. Niveles de potencia y distancia de AP 7	27
4.12. Niveles de potencia y distancia de AP 8	28
4.13. Niveles de potencia y distancia de AP 9	28
4.14. Niveles de potencia y distancia de AP 10	29
4.15. Niveles de potencia y distancia de AP 11	29
4.16. Área de cobertura a -45dBm	30
4.17. Área de cobertura a -60dBm	31
4.18. Área de cobertura a -70 dBm	32
5.2. Cambio de interpretación de las matriz x.	33
5.1. Diagrama de flujo de la solución de <i>FAP</i> con <i>PSO</i>	34
5.3. Diagrama de flujo de la inicialización aleatoria de las partículas y sus velocidades.	35
5.4. Diagrama de flujo de la evaluación del fitness y actualización de óptimos.	36
5.5. Diagrama de flujo de la función objetivo para canal adyacente y co-canal.	37
5.6. Diagrama de flujo de la función objetivo para co-sitio.	38
5.7. Diagrama de flujo de la actualización del óptimo local.	39
5.8. Diagrama de flujo de actualización del óptimo global.	39
5.9. Diagrama de flujo de la actualización de la posición y velocidad de las partículas	40
5.10. Diagrama de flujo del cálculo del cambio de posición y velocidad	41
5.11. Diagrama de flujo de la evaluación de los límites de los cambios de posiciones y velocidades	41
5.12. Diagrama de flujo de la inicialización de las partículas con la técnica de HotSpot e intervalos.	42
5.13. Diagrama de flujo del algoritmo para la ubicación de la celdas HotSpot.	43

5.14. Diagrama de flujo de la inicialización de la primera partícula.	44
5.15. Diagrama de flujo de la inicialización de la primera celda prioritaria.	45
5.16. Diagrama de flujo del proceso de copia de la asignación a las demás partículas.	45
5.17. Diagrama de flujo de la generación de posiciones aleatorias con intervalos.	46
5.18. Diagrama de flujo de la versión paralela del algoritmo PSO.	47
6.1. Curvas de convergencia para EX1, inicialización aleatoria.	49
6.2. Curvas de las 3 mejores convergencias para EX1, inicialización aleatoria.	49
6.3. Malla de asignación para EX1, inicialización aleatoria con 60 partículas.	50
6.4. Función objetivo de las asignaciones aleatorias para EX1.	50
6.5. Malla de asignación para EX1, inicialización con intervalos y <i>Hotspot</i> con 60 partículas.	51
6.6. Comparación de los tiempos de ejecución para PHI3.	52
6.7. Curvas de convergencia para PHI3, inicialización aleatoria con 20 partículas.	53
6.8. Curvas de convergencia para PHI3, inicialización aleatoria con 40 partículas.	53
6.9. Curvas de convergencia para PHI3, inicialización aleatoria con 60 partículas.	54
6.10. Curvas de las mejores convergencias para PHI3, inicialización aleatoria.	54
6.11. Malla de asignación para PHI3, inicialización aleatoria con 60 partículas.	55
6.12. Función objetivo de las asignaciones aleatorias para PHI3.	56
6.13. Curvas de convergencia para PHI3, inicialización con intervalos y Hotspot, prioridad 1.	56
6.14. Curvas de convergencia para PHI3, inicialización con intervalos y Hotspot, prioridad 4.	57
6.15. Curvas de convergencia para PHI3, inicialización con intervalos y Hotspot, prioridad 8.	57
6.16. Curvas de las mejores convergencias para PHI3, inicialización con intervalos y HotSpot.	58
6.17. Malla de asignación para PHI3, inicialización con intervalos y HotSpot con prioridad 1.	59
6.18. Malla de asignación para PHI3, inicialización con intervalos y HotSpot con prioridad 4.	60
6.19. Malla de asignación para PHI3, inicialización con intervalos y HotSpot con prioridad 8.	61
6.20. Malla de asignación para PUJ1, inicialización aleatoria con 20 partículas, entorno de prueba convencional.	62
6.21. Malla de asignación para PUJ1, inicialización aleatoria con 40 partículas, entorno de prueba convencional.	63
6.22. Malla de asignación para PUJ1, inicialización aleatoria con 60 partículas, entorno de prueba convencional.	63
6.23. Función objetivo de las asignaciones aleatorias para PUJ1.	64
6.24. Curvas de convergencia para PUJ2, inicialización aleatoria, entorno de prueba convencional.	65
6.25. Malla de asignación para PUJ2, inicialización aleatoria con 20 partículas, entorno de prueba convencional.	66
6.26. Malla de asignación para PUJ2, inicialización aleatoria con 40 partículas, entorno de prueba convencional.	66
6.27. Malla de asignación para PUJ2, inicialización aleatoria con 60 partículas, entorno de prueba convencional.	67
6.28. Función objetivo de las asignaciones aleatorias para PUJ1.	67
6.29. Curvas de convergencia para PUJ3, inicialización aleatoria con 20 partículas.	68
6.30. Curvas de convergencia para PUJ3, inicialización aleatoria con 40 partículas.	69
6.31. Curvas de convergencia para PUJ3, inicialización aleatoria con 60 partículas.	69
6.32. Malla de asignación para PUJ3, inicialización aleatoria con 20 partículas.	70

6.33. Malla de asignación para PUJ3, inicialización aleatoria con 40 partículas.	70
6.34. Malla de asignación para PUJ3, inicialización aleatoria con 60 partículas.	71
6.35. Función objetivo de las asignaciones aleatorias para PUJ1.	71
6.36. Cambio de método para la ubicación de la celda HotSpot.	72
6.37. Curvas de convergencia para PUJ3, inicialización con intervalos y celdas <i>HotSpot</i> , entorno de prueba convencional.	72
6.38. Curvas de las mejores convergencias para PUJ3, inicialización con intervalos y HotSpot.	73
6.39. Malla de asignación para PUJ3, inicialización con intervalos y celdas <i>HotSpot</i> , prioridad 2.	73
6.40. Malla de asignación para PUJ3, inicialización con intervalos y celdas <i>HotSpot</i> , prioridad 3.	74
6.41. Malla de asignación para PUJ3, inicialización con intervalos y celdas <i>HotSpot</i> , prioridad 4.	74
6.42. Asignación original de canales evaluados con PUJ1.	75
6.43. Asignación original de canales evaluados con PUJ2.	76
6.44. Asignación original de canales evaluados con PUJ3.	76
6.45. Comparación de los resultados de la evaluación de la función objetivo para el problema de asignación de la PUJ.	77

Índice de cuadros

2.1. Valores y rangos de las variables de PSO.	12
2.2. Resumen de características de los cluster de ZINE. Tomado de [17]	15
2.3. Funciones básicas de MPI.	16
2.4. Tipos de datos en MPI.	16
4.1. Direcciones físicas de los AP y número de identificación.	24
6.1. Tasas de convergencia del problema EX1.	48
6.2. Tasas de convergencia del problema PHI3.	52
6.3. Tiempos de ejecución de PHI3.	52
6.4. Tasas de convergencia del problema PUJ1.	62
6.5. Tasas de convergencia del problema PUJ2.	64
6.6. Tasas de convergencia del problema PUJ3.	68
6.7. Direcciones físicas de los AP y número de identificación.	75

Capítulo 1

Introducción

El crecimiento desmedido y la alta demanda de servicio Wi-Fi de los últimos años, ha dificultado la administración, planificación y control de las redes inalámbricas. Adicionalmente, tener un espectro en frecuencia limitado y alta congestión, ha generado la necesidad de re-usar canales para la comunicación. La calidad de una señal recibida puede caracterizarse a partir de tres conceptos: la potencia de recepción, el nivel de ruido (*background noise*) y la potencia de interferencia. La primera característica depende de la distancia entre transmisor y receptor, y la segunda no puede ser cambiada. Sin embargo, se pueden buscar estrategias que permitan minimizar la interferencia, cumpliendo las restricciones de compatibilidad electromagnética (EMC o *ElectroMagnetic Compatibility*) [13, 19].

La interferencia es una señal no deseada que afecta el rendimiento de un sistema de telecomunicaciones. Puede ser leve o muy grave, provocando desde la pérdida de algunos bits hasta el detrimento total de la señal, reduciendo la calidad de la información. Como se menciona en [13], *Gamst y Rave* definieron el modelo general de un problema de asignación de canales en una red celular heterogénea, donde se plantean que las EMC de una celda se describen como una matriz $n \times n$ llamada matriz de compatibilidad C , y el número de frecuencias requeridas para asignar en una celda n se representan con el vector de demanda $D[n]$. En resumen, el problema de asignación de canales en una red celular, consiste en encontrar una asignación libre de conflictos con la menor cantidad de frecuencias disponibles delimitadas por C y D .

El problema de asignación de canales (FAP o *Frequency assignment problems*) puede abordarse de diferentes maneras, puede ser desde la perspectiva del problema genérico (F-FAP) o desde el planteamiento de maximización de una función objetivo (MAX-FAP), donde las dos siguen la estructura de árbol de búsqueda. La diferencia radica en que la primera se basa en una relación de programación lineal, mientras la segunda toma la idea combinatorial, es decir cambian los modelos de la función objetivo. Una característica importante de F-FAP y MAX-FAP, es la facilidad para pasar de un método a otro, a diferencia de los problemas de asignación de frecuencia con mínima interferencia (MI-FAP o *minimum interference frequency assignment problem*) [3]. En la actualidad el problema de asignación de canales se ha trabajado en sus diferentes clasificaciones, sin embargo, aún existen problemas con los tiempos de convergencia y los embotellamientos debido a la falta de balanceo de carga computacional.

En telecomunicaciones el recurso más limitado es la frecuencia, por lo tanto, crear algoritmos que optimicen su uso y rendimiento es una necesidad. Se busca definir hasta qué punto puedo mejorar una característica sin afectar negativamente otras, como la calidad de servicio, capacidad, disponibilidad, retardo, confiabilidad, tasa de error de bit, entre otras. Dado que el problema de asignación en frecuencia es un problema *NP-hard*, no se puede asegurar una solución óptima. Para tener resultados válidos es necesario un tiempo de convergencia o procesamiento que crezca exponencialmente con el tamaño de las instancias. Al optimizar la asignación de canales de frecuencia en los *Access Point*, es posible disminuir el número de canales solapados, pues para redes Wi-Fi IEEE 802.11b solo existen

tres canales libres de interferencia *inter-channel* dentro de su rango de frecuencia a 2,4 GHz.

El presente trabajo de grado pretende aportar un proceso que disminuya la interferencia *inter-channel*, que se presenta cuando existen varios AP (*Access Points*) en un espacio con medio compartido, lo que aumenta el rendimiento en un sistema de comunicaciones, mejorando la calidad de servicio, el ancho de banda, la relación señal a ruido y la velocidad de transmisión. Al mejorar dichas características es posible hacer que la red funcione de mejor manera, lo que posibilita incrementar el número de usuarios y la tasa de transferencia, sin perder el desempeño deseado. Para resolver el problema de los tiempos de convergencia que surge en la implementación a gran escala de algoritmos complejos, este trabajo de grado pretende emplear una plataforma distribuida en paralelo de alto rendimiento computacional (HPC o *high performance computing*).

Este documento se desarrolla de la siguiente manera, comienza en el capítulo 2 mostrando lo que se ha desarrollado hasta ahora en el campo de los algoritmos bio-inspirados, trabajos similares para la solución de los problemas de asignación de frecuencias, una contextualización a la historia, definición y características del algoritmo de optimización por enjambre de partículas, PSO, y un acercamiento a las arquitecturas computacionales de alto rendimiento. Luego, en el capítulo 4 se hace una profundización del problema de asignación de frecuencias, la definición de las restricciones (*hard y soft*), y el modo de asignación en Wi-Fi. En la sección 5, se muestra el diseño de la solución planteada en este trabajo de grado, sus restricciones y métodos para una arquitectura de computación estándar y una distribuida. Finalmente, en el capítulo 6 se organizan los resultados de las pruebas realizadas y la evaluación de los mismos que se resumen en las conclusiones en el capítulo 7.

Capítulo 2

Marco Teórico

2.1. Problemas de asignación de frecuencias y métodos heurísticos

Sobre el desarrollo de algoritmos para asignación de canal existen numerosas investigaciones, por ejemplo, en [11] se muestra el planteamiento y resultados de dos algoritmos de asignación de canales, el algoritmo módulo 3 y el algoritmo RAA (*Reference Array Algorithm*) los cuales asignan los tres canales sin interferencia *inter-channel* del protocolo IEEE 802.11b. Con respecto a los algoritmos bio-inspirados en el campo de aplicación de asignación de recursos, en [14] se muestra el uso de los algoritmos de optimización de colonia de hormigas (ACO, *Ant colony optimization*), en [19] se hace una comparación en el desempeño de los algoritmos genéticos y los algoritmos PSO (*Particle Swarm Optimization*) en el manejo de asignación de canales en frecuencia. En [3] se muestran también los modelos para *FAP* a partir de la perspectiva de los algoritmos *greedy*, búsqueda local, búsquedas taboo, enfriamiento simulado, redes neuronales artificiales.

Los algoritmos de optimización por enjambres de partículas (PSO, *Particle Swarm Optimization*) surgieron en el sexto simposio internacional de micro-máquinas y ciencias humanas de 1995 [12], como un nuevo modelo de optimización propuesto por *Russel Eberhart* y *James Kennedy*. Otros autores han basado sus trabajos en este planteamiento, por ejemplo, en [20] se muestran técnicas de optimización bio-inspiradas de diseño para antenas de comunicaciones a partir de los algoritmos por enjambres de partículas. Benameur [5] muestra el proceso de concepción y desarrollo del modelo discreto de PSO (DPSO) para problemas de asignación de frecuencias.

Existen evidencias de aplicación de los algoritmos de optimización por enjambres de partículas en los tres modos de asignación de canales, por ejemplo en [21] se muestra el desarrollo y resultados de PSO en FCA (*Fixed channel assignment*, en español asignación de canal fija) el cual como su nombre indica no es adaptativo, [15] es un ejemplo en DCA (Dynamic channel assignment, en español asignación de canal dinámica) el cual puede ser centralizado o distribuido, con el inconveniente de requerir un alto tiempo de evaluación y un nivel elevado de esfuerzo computacional y en [19] tienen el problema de HCA (*Hybrid channel assignment*, en español asignación de canal híbrida) la cual reúne las ventajas de las dos primeras, pues tiene sub-grupos según FCA y otros según DCA.

En [9] se plantea el modelo para PSO bi-objetivo (BOOP), donde FAP deja de abordarse con una sola restricción y se tienen en cuenta el nivel de interferencia y el número de violaciones de las restricciones de la función fitness. En [8] se usa PSO en una versión cooperativa en paralelo, en una arquitectura *GPU*, donde se busca subdividir el problema de optimización en sub secciones más simples y diferentes entre ellas. En complemento a otros proyectos o artículos aquí referenciados, este trabajo de grado buscó solucionar el problema de asignación fija de canales en frecuencia para redes simples y complejas, usando PSO en la arquitectura distribuida en paralelo de alto rendimiento (*HPC*) de la Pontificia Universidad Javeriana.

2.2. Optimización por enjambre de partículas

Como se mencionó en 2.1, los algoritmos de optimización por enjambre de partículas (PSO, *Particle Swarm Optimization*) surgieron en el sexto simposio internacional de micro-máquinas y ciencias humanas de 1995 [12], como un nuevo modelo de optimización propuesto por *Russel Eberhart y James Kennedy*. En este primer planteamiento se presentó un algoritmo bio-inspirado, diseñado para solucionar problemas no lineales como, por ejemplo: el entrenamiento de redes neuronales, tareas de aprendizaje para robots, etc. Este algoritmo tiene la característica de ser computacionalmente económico, pues se maneja una alta velocidad, requiere poco uso de memoria y además emplea operaciones matemáticas primitivas.

PSO está incluido en la tecnología de vida artificial (*A-life*), esto se debe a que requiere un conocimiento mutuo entre cada partícula y con esta información el «enjambre» se aproxima hacia la respuesta más acertada según el conocimiento colectivo, por ello este algoritmo se considera evolutivo. El planteamiento inicial de PSO, se basa en el mecanismo de comunicación de los enjambres de abejas, los bancos de peces o los nichos de aves, donde cada partícula viaja sobre el campo de búsqueda teniendo una visión individual y sesgada del mismo, esta visión se complementa con la visión parcial de sus compañeras, lo que favorece a un conocimiento colectivo y completo del campo de búsqueda. De esta forma, las abejas pueden encontrar la región del campo de flores con más alimento, lo que en un problema general representa la mejor solución encontrada por las partículas, según unos criterios de evaluación.

2.2.1. Concepto y planteamiento del algoritmo

Cada partícula tiene una posición y una velocidad asociada, las cuales se van modificando de acuerdo a dos paradigmas planteados por *Eberhart y Kennedy*, *pBest* y *gBest*. La mejor posición localmente encontrada por cada partícula (*pBest*), se refiere a la región del campo de flores con más alimento, según una abeja en particular bajo su visión parcial. La mejor posición global del enjambre (*gBest*), la cual se selecciona entre las *pBest* de todas las partículas. Los criterios de evaluación permiten saber si una región del espacio de búsqueda es la mejor solución, donde hay «más alimento» para las abejas, la cual determina la *gBest*. En la figura 2.1a se muestra el comportamiento de las partículas en el enjambre.

El objetivo es que en cada iteración las partículas encuentren una posición más cerca a la comida, para ello cada vez que se encuentra una nueva *gBest* las partículas cambian su velocidad asociada, una de sus componentes es dirigida a la mejor posición global, otra componente hacia la mejor posición personal y otra componente inercial (referida a donde originalmente estaba dirigida la partícula antes de la actualización de *gBest*), este comportamiento se ilustra en la figura 2.1b.

De acuerdo a [18] existen 4 parámetros fundamentales que afectan directamente el desarrollo del algoritmo de optimización por enjambre de partículas: El peso inercial, el parámetro cognitivo, el parámetro social y el factor de restricción.

El primero de ellos permite que la partícula tenga una «memoria» respecto a su posición anterior, con esto se controla la convergencia de todo el enjambre. Los parámetros cognitivo y social, se encargan de relacionar la velocidad de la partícula con el óptimo local *pBest* y el óptimo global *gBest*. El factor de restricción definido por la literatura como 0.729, relaciona la velocidad calculada con la nueva posición de la partícula n -ésima.

En (2.1) se muestra como calcular la velocidad para una partícula i , en la iteración $n+1$, el primer término es el inercial donde relaciona el peso inercial ω con la velocidad de la partícula en la iteración n . El segundo término es el cognitivo, donde se multiplica el parámetro cognitivo, un número aleatorio y la diferencia entre el óptimo local *pBest* y la posición de la partícula en la iteración inmediatamente anterior. El tercero es el término social, donde de manera similar al cognitivo, se multiplica el parámetro social con un número aleatorio y la diferencia entre el *gBest* y

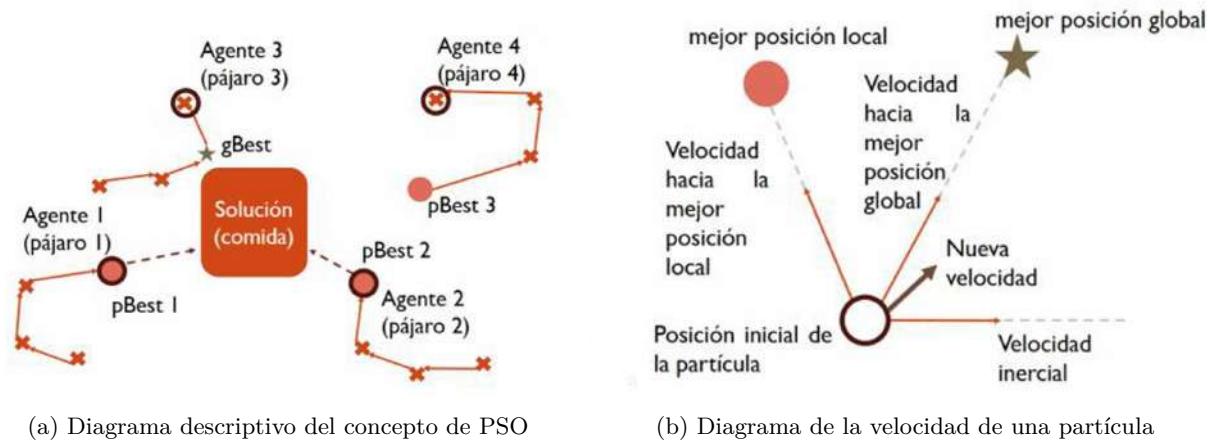


Figura 2.1: Conceptos de PSO. Tomado de [19,20]

la posición previa de la partícula.

$$V_i^{n+1} = \omega V_i^n + C_1 r_{i1}^n (P_i^n - X_i^n) + C_2 r_{i2}^n (P_g^n - X_i^n) \quad (2.1)$$

En (2.2) se define la nueva posición de una partícula i -ésima, sumando la posición previa con el factor de restricción multiplicado por la velocidad recientemente encontrada con (2.1).

$$X_i^{n+1} = X_i^n + \chi V_i^{n+1} \quad (2.2)$$

En la tabla 2.1 se resumen los valores o rangos de cada variable usada en (2.1) y (2.2).

El diagrama de flujo de la figura 2.2 muestra el algoritmo básico de la optimización por enjambre de partículas de forma paralela. Como se ve, es necesaria una comunicación permanente entre las partículas, y una evaluación constante de los criterios de convergencia los cuales son dependientes del problema que se este solucionando.

Variable	Símbolo	Valor o rango
Peso inercial	ω	[0,1, 1]
Constante cognitiva	C_1	2
Constante social	C_2	2
Aleatorio cognitivo	r_1	[0, 1]
Aleatorio social	r_2	[0, 1]
Factor de restricción	χ	0.729

Cuadro 2.1: Valores y rangos de las variables de PSO.

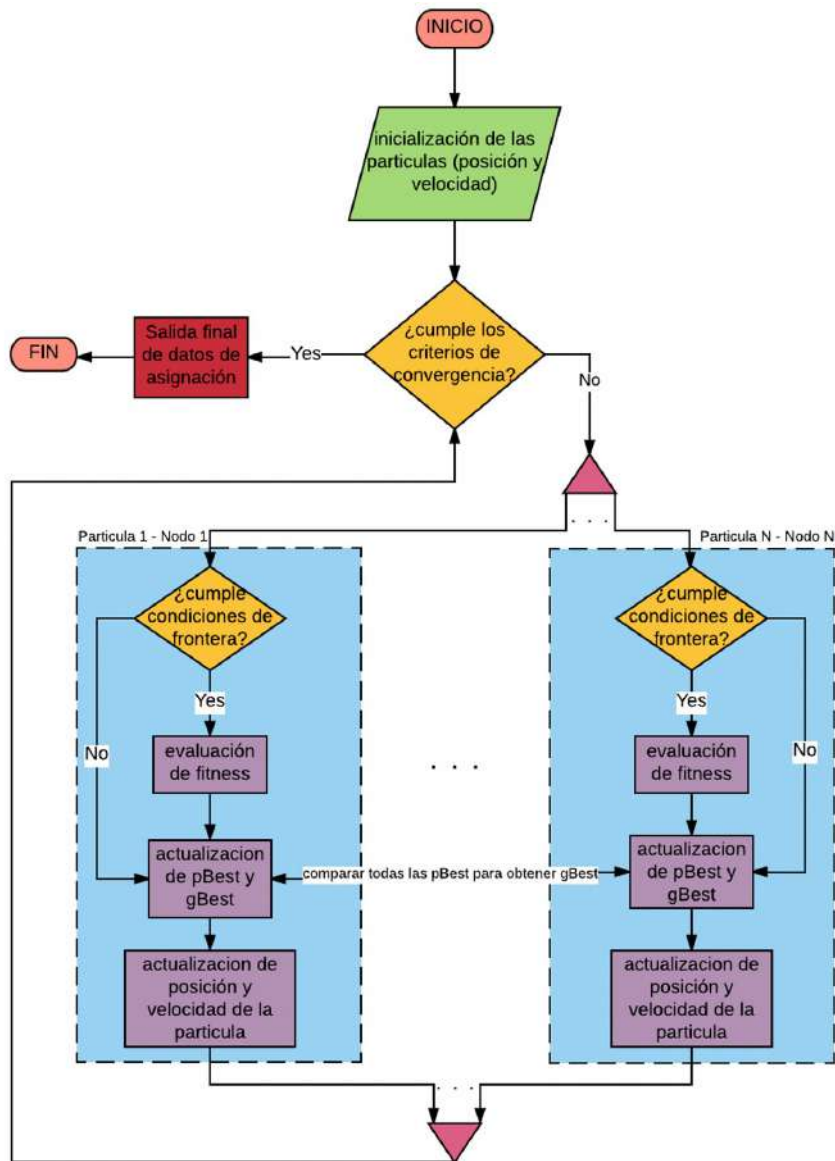


Figura 2.2: Diagrama de flujo de PSO en paralelo [20]

2.2.2. Variantes de PSO

Hay varias formas de implementar PSO, pueden clasificarse en tres categorías: La primera es cambiar la forma de distribución de las partículas sobre el espacio de búsqueda, la segunda forma es variando los parámetros de cambio de la velocidad y posición, y la tercera depende del tipo de datos usados para la implementación.

Dentro del primer método podemos encontrar la configuración convencional de PSO, donde todas las partículas están «volando» sobre todo el espacio, inicialmente de manera indiscriminada. La otra modalidad es la de sub-dividir el espacio de búsqueda en pequeñas regiones, y crear partículas aleatorias sobre dichas regiones, entonces existen: un $pBest$ referente a cada partícula, un $gBest$ de cada sub-región y un $gGlobal$ el cual es el óptimo encontrado en todo el espacio dentro de las $gBest$.

En el segundo método se realiza un cambio en el uso de los parámetros de las formulas de cambio de posición y velocidad, pueden hacerse en su modo convencional conocido en [18] como

PSO-Bo, en el cual se usan las variables de peso inercial y factor de restricción al tiempo. Otra variante es PSO-In, usando solo el peso inercial, y finalmente PSO-Co, que usa únicamente el factor de restricción [18].

La tercera modalidad de PSO se refleja en el tipo de variables o soluciones a implementar [13], puede usarse una representación binaria o real (decimal o entera), y se puede pasar de una a la otra forma fácilmente, según sea el caso.

En este trabajo de grado se usará PSO-Bo en su definición original de espacio de búsqueda, con variables enteras usando un truncamiento que se explicará en el capítulo 5.

2.3. Arquitecturas computacionales

2.3.1. Clusters

Un cluster es un conjunto de computadores conectados a través de una red, donde el usuario lo percibe como un solo equipo [6]. Los componentes de un cluster se muestran en la figura 2.3, consiste en un software y los nodos, los cuales pueden ser:

- Nodo principal
- Nodos trabajadores
- Nodos cliente

También están las conexiones de red, que pueden ser *Ethernet*, *FastEthernet*, *GigabitEthernet*, *Myrinet*, o *Infiniband*. El *Middleware* se encarga de optimizar el sistema, manejar las migraciones, el balanceo de carga y la tolerancia a los fallos entre otros. En otras palabras es lo que soporta la escalabilidad del sistema.

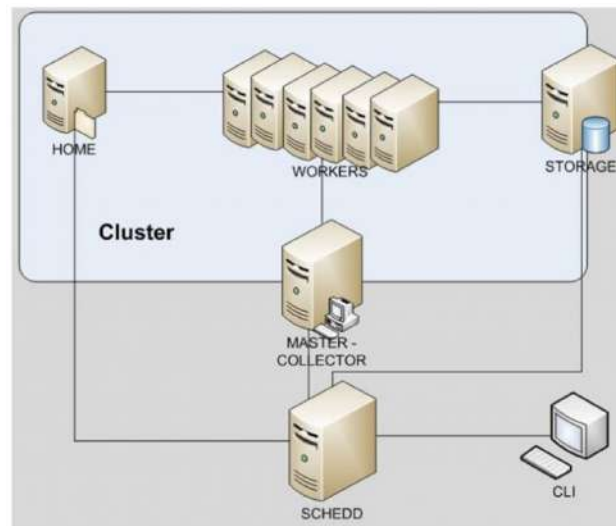


Figura 2.3: Componentes de un cluster. Tomado de [6]

Los clusters pueden manejarse de dos formas, como plataformas de alto rendimiento (HTC o *High Throughput Computing*), o de alto desempeño computacional (HPC o *High Performance Computing*). La figura 2.4, muestra algunas características y deferencias clave de cada uno.

HPC	HTC
Programas altamente paralelos	Usa muchos nodos
Requiere alta velocidad de red, memoria y disco	No requiere tantos recursos de red, memoria y disco
Sincronización entre nodos	Se puede dividir el trabajo en partes separadas e independientes
Cambiar código de programas, MPI, PVM o librerías especializadas	Entre más máquinas, más trabajos ejecutará.
Recursos dedicados, bloqueo a otros usuarios	Importa más la cantidad de resultados que la velocidad de procesamiento

Figura 2.4: Características de *HPC* y *HTC*. Tomado de [6] .

2.3.2. ZINE Javeriana

En este trabajo de grado se usaron los recursos del centro de alto rendimiento computacional de la Pontificia Universidad Javeriana, *ZINE*, el cual consiste en un sistema de procesamiento en paralelo o distribuido, que consta de un conjunto de computadores independientes, interconectados entre sí de tal manera que funciona como un solo recurso computacional. Del cluster se esperan principalmente 4 cosas: alto rendimiento, alta disponibilidad, equilibrio de carga y escalabilidad.

La plataforma *ZINE* de la Javeriana cuenta con 3 recursos disponibles, el primero llamado Zine01 es un cluster con 384 cores, sigue el cluster Ingsistemas con 368 cores y por último Condor un cluster de 54 cores. Cada uno cuenta con diferentes características que se detallan en la tabla ??.

Característica	Zine01	IngSistemas	Condor
Servidor	HP ProLiant SL390s G7	DELL 710 y HP SL250s G8	SUN MicroSystems
Procesador	Intel Xeon L5640	Intel Xeon X5460	AMD 2X – X86_64
Conectividad	Infiniband	Ethernet	Ethernet
Total Cores	384	368	54

Cuadro 2.2: Resumen de características de los cluster de *ZINE*. Tomado de [17]

Zine permite a sus usuarios usar diferentes paquetes de software dependiendo de sus necesidades que se lista en la figura 2.5 , cuenta con 3 compiladores: GCC, Phyton y Perl.

454_data_analysis-2.9	fastx_toolkit-0.0.13.2	hmmer-3.0	R-2.15.0
Artemis	fftw-3.3.2	lammps-4Dec12	R-2.15.2
Autodock-vina-1.1.2	fftw-3.3.3	libgtextutils-0.6.1	R-2.15.3
Autodock4.2.5.1	freesurfer-5.3.0	mothur-1.32.1	r-3.0.2
Biom-format-1.1.2	goatbrot	mpich-3.0.4	seismic_unix-43R3
blast_plus-2.2.28	gromacs_mpi-4.5.5	NAMD_2.9	trinityrnaseq-r2013_08_14
blast-2.2.26	gromacs-333	numpy-1.5.1	wrf
cmake-2.8.10.2	gromacs-405	openmpi-165	
coin-Cbc	gromacs-455	pycogent-1.5.3	
creatools	gromacs-461	qiime-1.7.0	

Figura 2.5: Software disponible en *Zine*

2.3.3. MPI (*Message Passing Interface*)

Es una interfaz estandarizada para la realización de aplicaciones paralelas basadas en paso de mensajes [4], no exige una determinada implementación del mismo y otorga una serie de funciones básicas que permiten la comunicación entre procesos necesaria en las aplicaciones a diseñar.

Para hacer uso de la interfaz de paso de mensajes en lenguaje C es necesario incluirla, usando el comando «`#include <mpi.h>`». La unidad básica en MPI son los **procesos** [10], los cuales tienen espacios de memoria independientes, cada uno cuenta con un número identificador comunmente llamado *rank*. La tabla 2.3, muestra algunas de las funciones básicas de MPI, y la tabla 2.4, define los tipos de datos propios de la interfaz.

Función	Uso
MPI_INIT	Inicialización
MPI_COMM_WORLD	Comunicador
MPI_COMM_RANK	Identificación de procesos
MPI_COMM_SIZE	Procesos en el comunicador
MPI_FINALIZE	Finalización
MPI_BCAST	Transferencia de datos múltiples nodos
MPI_REDUCE	Transferencia de datos múltiples nodos

Cuadro 2.3: Funciones básicas de MPI.

Tipos de dato
MPI_SHORT
MPI_INT
MPI_CHAR
MPI_LONG
MPI_UNSIGNED_CHAR
MPI_FLOAT
MPI_DOUBLE
MPI_LONG_DOUBLE
MPI_BYTE
MPI_PACKED

Cuadro 2.4: Tipos de datos en MPI.

Existen varias implementaciones de MPI y dependiendo del entorno en donde se dese ejecutar se debe escoger cual conviene en cada ocasión, en [1] se dan algunas características claves que ayudan a escoger la mejor opción entre MPICH y OpenMPI. En este trabajo de grado se decidió trabajar con OpenMPI por el hecho de que MPICH no soporta la conectividad *InfiniBand*, con la cual opera el cluster **ZINE**. En el anexo 8.4 se muestra el proceso de instalación y configuración de un cluster casero, creado para pruebas de este trabajo de grado.

Capítulo 3

Objetivos

Se implementó un algoritmo de optimización por enjambre de partículas (PSO, *Particle Swarm Optimization*), que permitió asignar canales en frecuencia reduciendo la interferencia en una red IEEE 802.11b.

Los objetivos específicos planteados al comienzo del trabajo de grado son:

1. Implementar un algoritmo de asignación de canales usando la estrategia de optimización por enjambre de partículas (PSO) en una arquitectura computacional estándar.
2. Formular e implementar el algoritmo de asignación de canales PSO en la arquitectura computacional de alto rendimiento (HPC: High Performance Computing) de la Pontificia Universidad Javeriana.
3. Comparar los resultados de una red usando la asignación de canales en frecuencia de PSO y una asignación aleatoria.
4. Validar los resultados del algoritmo de asignación de canal basado en PSO con los benchmark y en redes compuestas por cientos de Access Points (APs).

3.1. Objetivo específico 1

Se implementó el algoritmo PSO en una arquitectura de computación convencional en lenguaje C, logrando el desarrollo de una guía sobre la instalación en *Windows 10*, ejecución y pruebas sencillas de funcionamiento de MinGW, la cual se encuentra adjunta en el anexo 8.3. Se probó PSO con problemas de computación entera como los señalados en [18], obteniendo resultados de desempeño acertados y comparables con las pruebas de escritorio desarrolladas en paralelo, logrando así pasar al desarrollo de la función objetivo para asignación fija de canales en frecuencia, descrita en el capítulo 5.

3.2. Objetivo específico 2

Luego de cumplir el objetivo específico 1, se hicieron las modificaciones correspondientes que permitieron transformar el algoritmo en paralelo y ejecutarlo en la plataforma *ZINE* del centro de alto rendimiento computacional Javeriano. Adicionalmente, como el cluster es un entorno de producción no era posible ejecutar pruebas continuamente para verificación, por lo tanto se construyó un cluster «casero» y se diseñó una guía para una reproducción que está en el anexo 8.4.

3.3. Objetivo específico 3

Se realizaron varias comparaciones de los 5 problemas estudiados en este trabajo de grado:

- Respecto a su entorno de pruebas, pudiendo ser en un computador convencional o en la plataforma de computación distribuida
- Cambiando la forma de inicialización de las posiciones y velocidades de las partículas, una de manera aleatoria o con el método explicado en la sección 5.2, llamado técnica de celdas *HotSpot*, donde también se hace uso de unos intervalos de ventana variable.

En cada caso se podían cambiar características para la ejecución del programa, tales como el tamaño del enjambre (cantidad de partículas) o el nivel de prioridad en el caso de la inicialización con celdas *HotSpot*.

Se realizaron las pruebas de comparación de los problemas aquí estudiados, entre las asignaciones de canales completamente aleatorias y las respuestas obtenidas usando PSO en alguna de sus configuraciones, estos resultados se encuentran detallados en 6.

3.4. Objetivo específico 4

Se usaron 2 problemas ampliamente conocidos en la literatura. El primer problema se conoce como **EX1**, que consiste en una matriz de compatibilidad electromagnética de 4x4, teniendo 4 celdas en el sistema y 11 canales para asignación correspondientemente. Este problema al ser «simple», se usó para las pruebas de escritorio del algoritmo en sus primeras etapas de configuración, en el planteamiento de la función objetivo y en las pruebas del diseño de la estrategia de soporte para la convergencia con la técnica *HotSpot*.

El segundo problema de referencia, se denomina comúnmente como **PHI3**, es una de las combinaciones usadas en las formulaciones de Philadelphia. Cuenta con una matriz de compatibilidad de 21x21, con 21 celdas y 533 canales para asignación, tiene una dificultad considerablemente más alta que el primer problema y se evaluó para el desempeño de PSO en un entorno de pruebas convencional y en un entorno distribuido en paralelo.

Además de estos dos *benchmarks*, se diseñaron 3 matrices de compatibilidad electromagnética, de acuerdo a un estudio de los niveles de potencias de los *Access Points* 802.11, realizado en la Facultad de Ingeniería de la Pontificia Universidad Javeriana. Sin embargo, quedó pendiente realizar las pruebas con cientos de AP, pues aunque se solicitaron los datos, no fueron suministrados oportunamente.

Capítulo 4

Descripción de los problemas de asignación

4.1. Problemas de asignación de frecuencias

El objetivo del problema de asignación de canales es distribuir las frecuencias disponibles de acuerdo al requerimiento de cada una de las celdas del sistema. El problema surge por la baja disponibilidad de las frecuencias, como se mencionó en el capítulo 1 existe un incremento en su demanda, esto provoca la necesidad de compartir el medio, la idea es hacerlo de manera inteligente de tal forma que se reduzca la interferencia, figura 4.1.

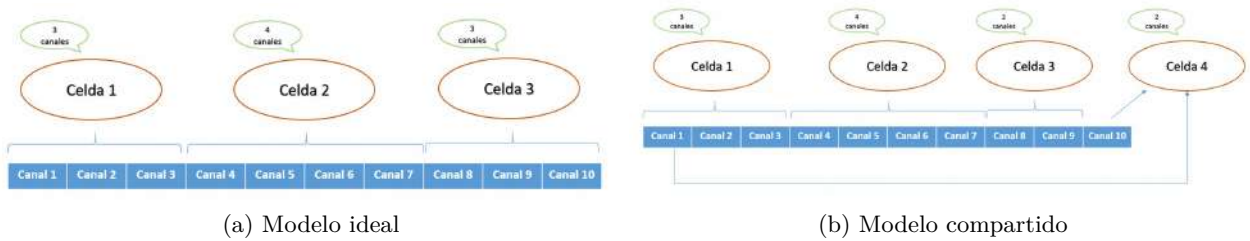


Figura 4.1: Concepto de asignación de frecuencias

4.1.1. Restricciones

Para entender el problema de asignación de frecuencias es necesario hablar de las restricciones de asignación, existen dos tipos: fuertes y suaves [7].

Las fuertes, son aquellas que obligatoriamente deben cumplirse en mayor o menor grado, llamadas las restricciones de compatibilidad electromagnética (*EMC*) y las condiciones de demanda. Hay tres tipos de interferencia: co-Canal, donde dos celdas adyacentes operan bajo la misma frecuencia. La interferencia de canal adyacente, en la cual dos celdas adyacentes tienen frecuencias adyacentes. Y la interferencia de co-Sitio, dos canales adyacentes en la misma celda. Las EMC, definen la distancia mínima de separación entre los canales respecto a las celdas del sistema. El vector demanda d describe la cantidad de canales solicitados por cada celda.

Un ejemplo de como funcionan y se relacionan la matriz C y el vector d se encuentra en la figura 4.2, donde la diagonal en la matriz, define las restricciones de co-Sitio (*CSC*, *co-site constraint*), y el resto de posiciones diferentes de 0 definen las restricciones de co-canal (*CCC*, *co-channel constraint*), y de canal adyacente (*ACC*, *Adjacent channel constraint*). Mientras que el vector de demanda muestra que para la celda 1,2 y 3 necesitan 1 canal, mientras que la tercera celda 3 canales.

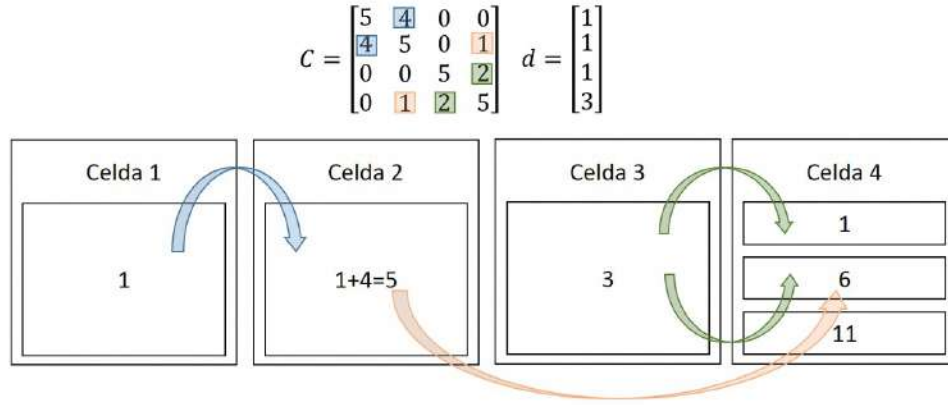


Figura 4.2: Ejemplo de asignación según C y d .

Las restricciones suaves son aquellas que pueden mejorar los resultados pero que no son totalmente necesarias, por ejemplo las condiciones de empaquetado donde el objetivo es asignar la mínima frecuencia posible que cumpla las EMC. También esta la condición de resonancia, que solo aplica cuando se tiene una geometría «simétrica» en la ubicación y distribución de las celdas, que permita repetir la asignación de canales cada cierto número de nodos bajo un esquema de re-uso. Otra restricción es la condición de mínimos movimientos, que se usa cuando se tiene un problema de asignación dinámica, con la cual cuando exista un cambio en la demanda se hace la re asignación buscando no cambiar demasiado lo previamente asignado.

4.2. Problemas de referencia

En esta sección se mostrarán las matrices de compatibilidad electromagnética y los vectores de demanda de cada uno de los problemas con los cuales se probó el algoritmo.

4.2.1. Problema EX1: 4 celdas, 11 canales

Se muestra la matriz de compatibilidad C del problema de 4 celdas en (4.1), y (4.2) es el vector demanda. [7]

$$C = \begin{bmatrix} 5 & 4 & 0 & 0 \\ 4 & 5 & 0 & 1 \\ 0 & 0 & 5 & 2 \\ 0 & 1 & 2 & 5 \end{bmatrix} \quad (4.1)$$

$$d = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 3 \end{bmatrix} \quad (4.2)$$

4.2.2. Problema de Philadelphia P3: 21 celdas, 533 canales

La matriz de compatibilidad electromagnética (4.3), y (4.4) es el vector demanda del problema de Philadelphia de 21 celdas y 533 canales. [16]

4.3. Problema de asignación de frecuencias en la Facultad de Ingeniería de la Pontifica Universidad Javeriana

4.3.1. Medidas de potencia y distancia

Para el estudio de la asignación de frecuencias para los *Access Points Wi-Fi* de la facultad de ingeniería se hicieron pruebas usando **WiFi Analyzer**, una aplicación móvil creada por *Kevin Yuan* disponible en la *Play Store de Android*, con un celular *Lenovo K6 (k33b36)*.



Figura 4.3: Aplicación móvil, Wifi Analyzer

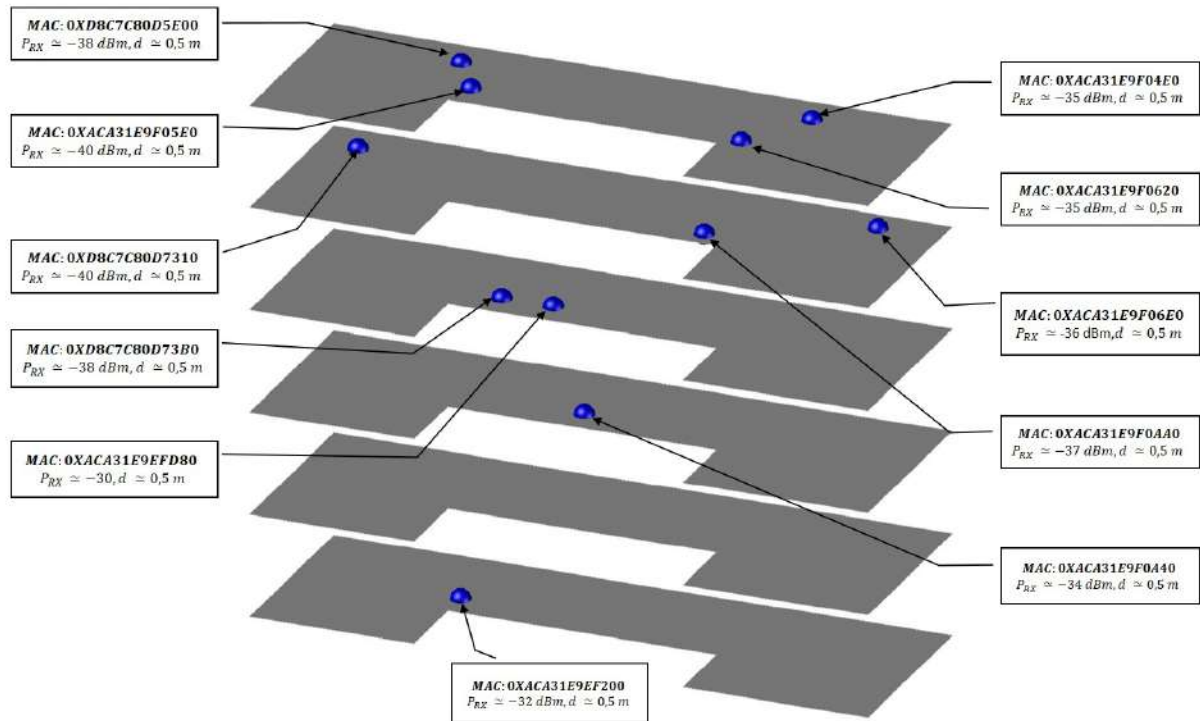
Se hicieron mediciones de la distancia tomando cada *AP* como punto $(0,0,0)$, en cada caída de -10 dB en cada piso y luego se estimó el punto más cercano verticalmente en los entre pisos.

Para identificar cada nodo se usa su dirección física, sin embargo, por facilidad se asignó un número a cada *AP* de acuerdo a la tabla, véase en la figura 4.4.

Las medidas y valores de potencia de cada *AP* se registran en las figuras 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14 y 4.15, donde se define con color azul la celda aproximada de potencia a -45 dBm, en color naranja la respectiva celda a -60 dBm y en color verde la celda a -70 dBm.



(a) Edificio José Gabriel Maldonado, Facultad de Ingeniería, tomado de [2]



(b) Modelo 3D simplificado, ubicación de los AP

Figura 4.4: Ubicación espacial de los AP de la facultad de ingeniería de la PUJ

ID	Dirección MAC
1	0XACA31E9EF200
2	0XACA31E9F0A40
3	0XD8C7C80D73B0
4	0XACA31E9EFD80
5	0XACA31E9F06E0
6	0XACA31E9F0AA0
7	0XD8C7C80D7310
8	0XACA31E9F04E0
9	0XACA31E9F0620
10	0XACA31E9F05E0
11	0XD8C7C80D5E00

Cuadro 4.1: Direcciones físicas de los AP y número de identificación.

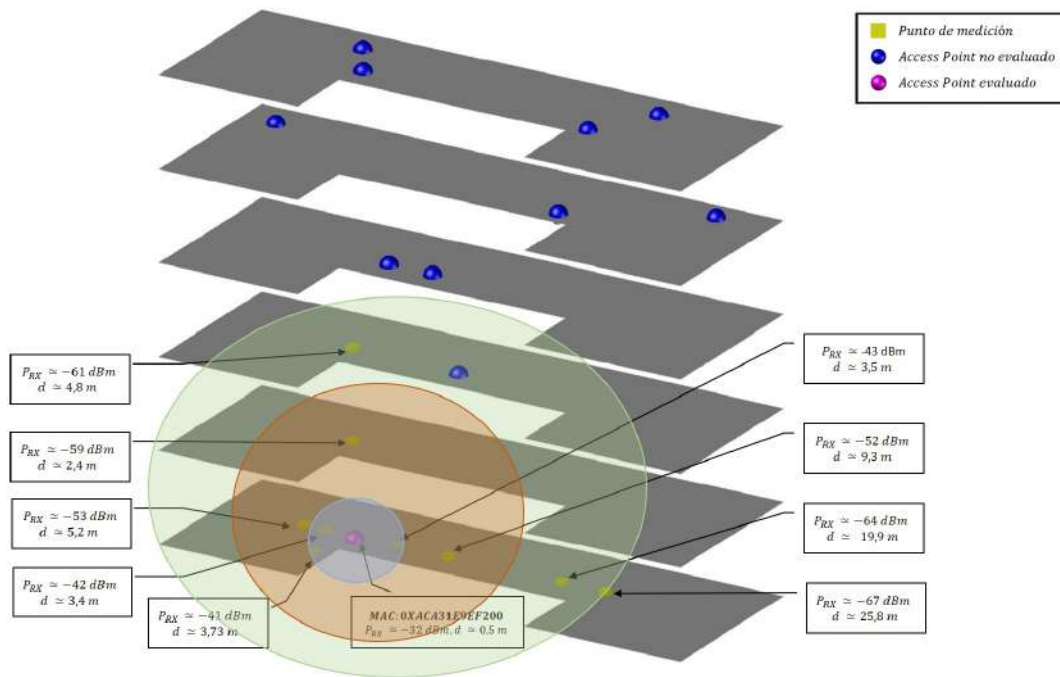


Figura 4.5: Niveles de potencia y distancia de AP 1

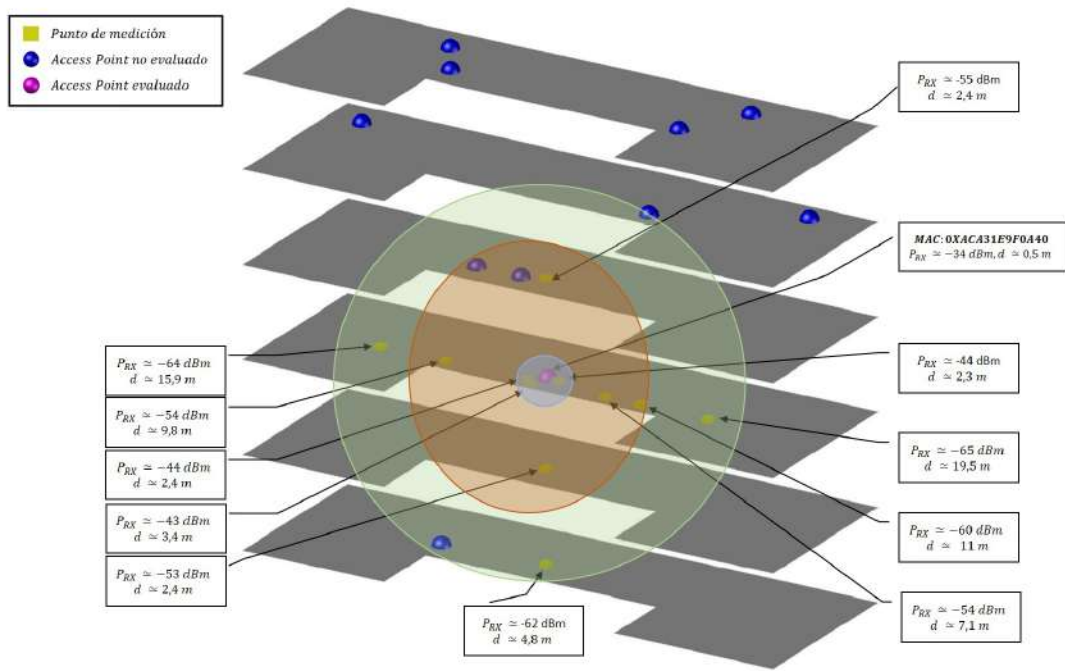


Figura 4.6: Niveles de potencia y distancia de AP 2

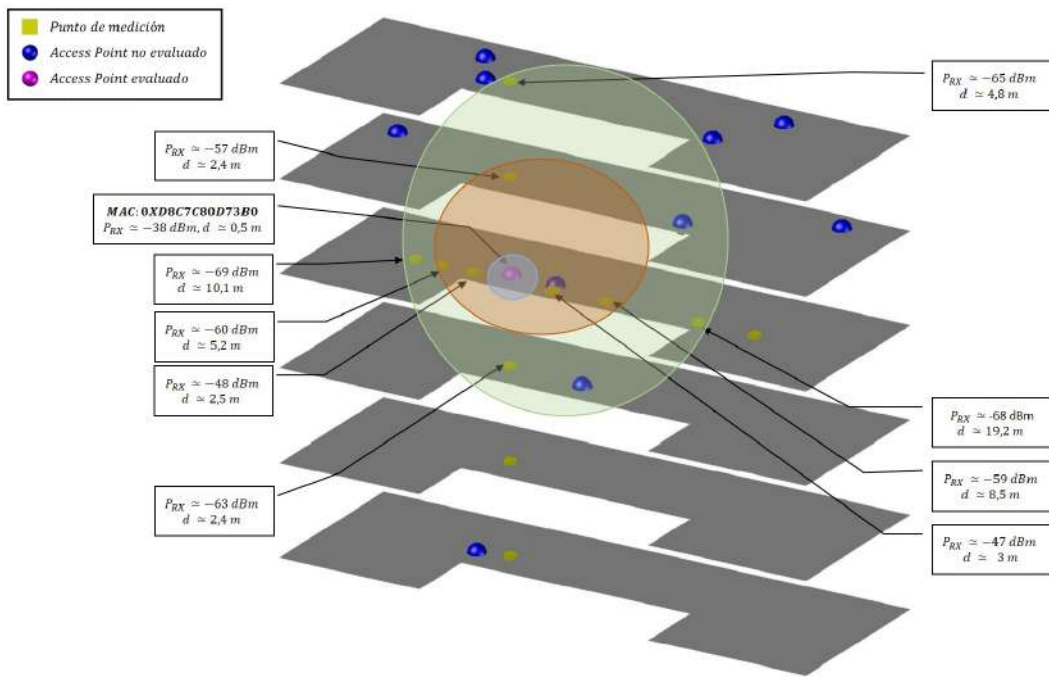


Figura 4.7: Niveles de potencia y distancia de AP 3

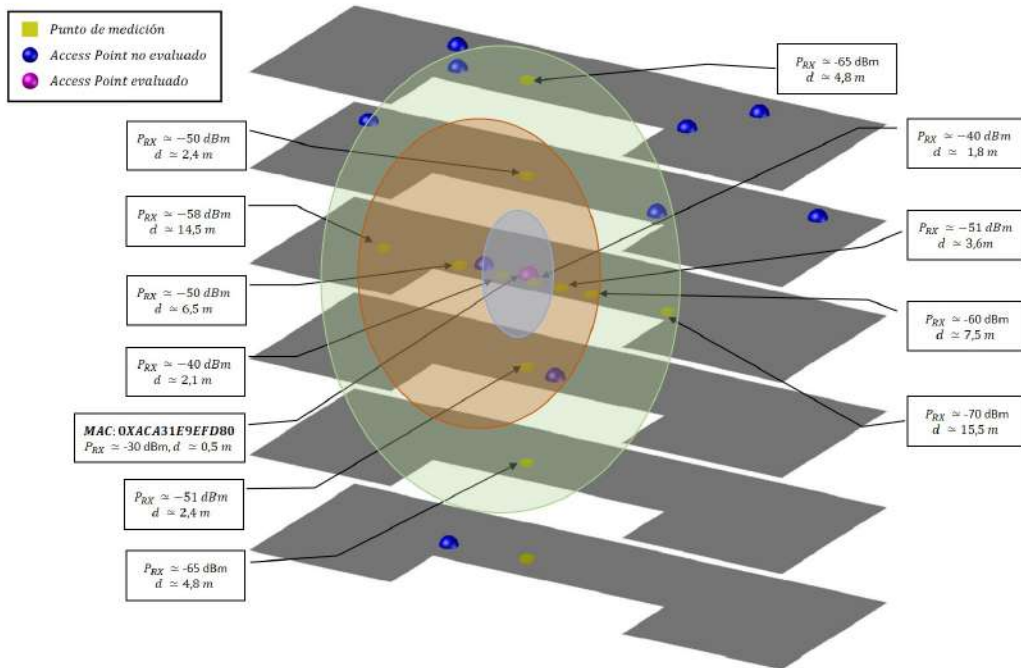


Figura 4.8: Niveles de potencia y distancia de AP 4

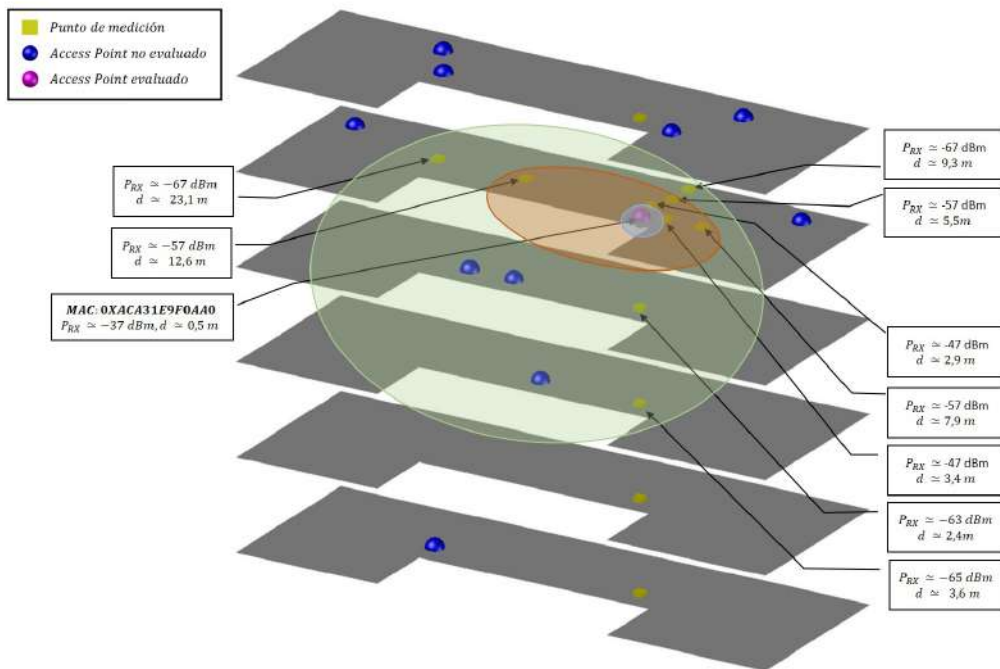


Figura 4.9: Niveles de potencia y distancia de AP 5

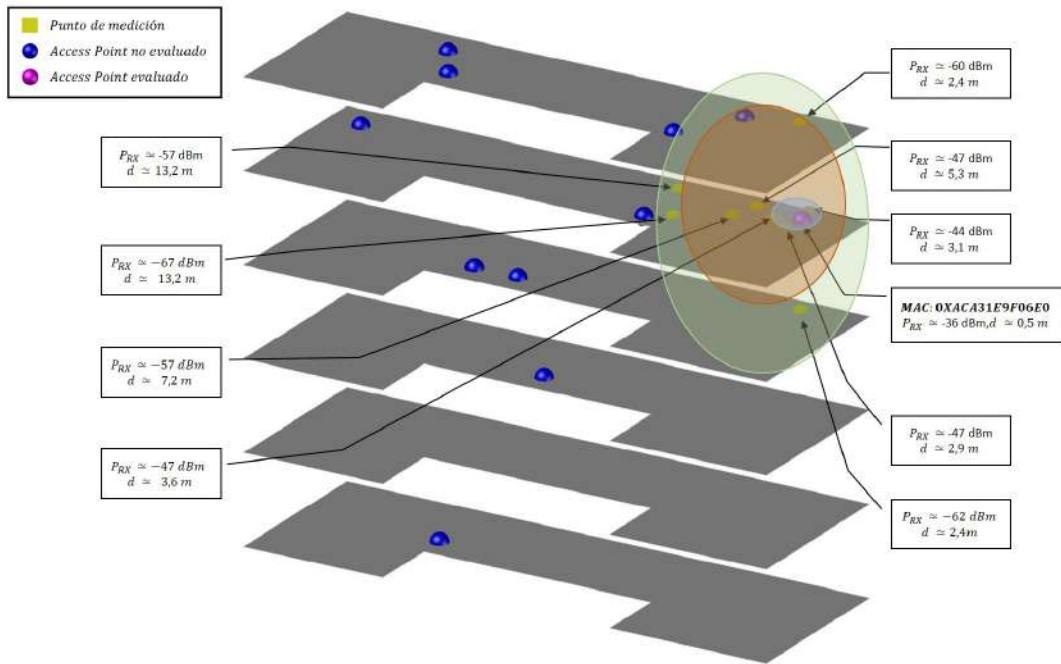


Figura 4.10: Niveles de potencia y distancia de AP 6

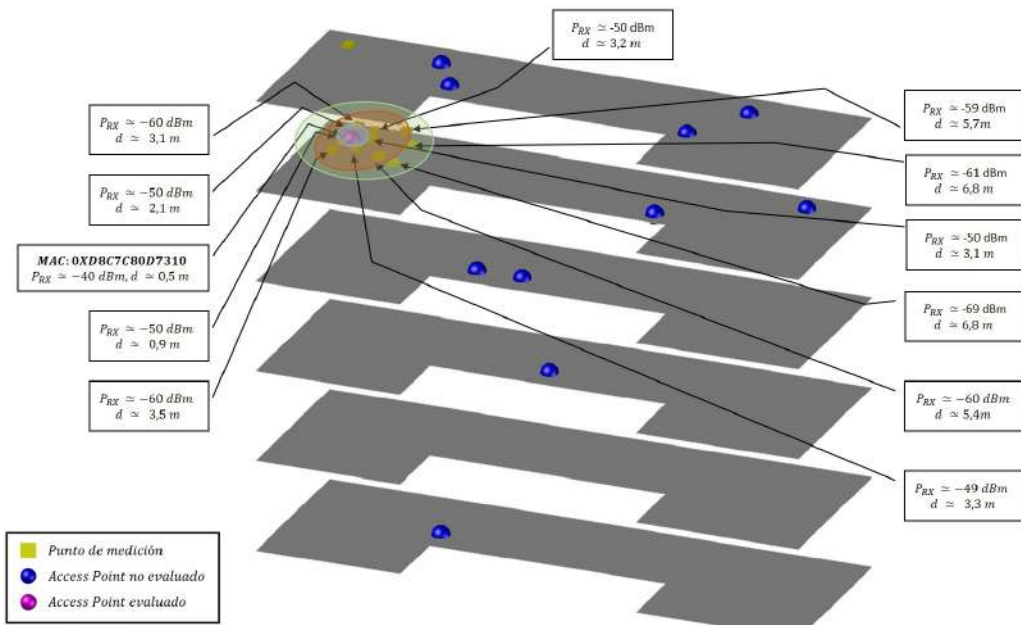


Figura 4.11: Niveles de potencia y distancia de AP 7

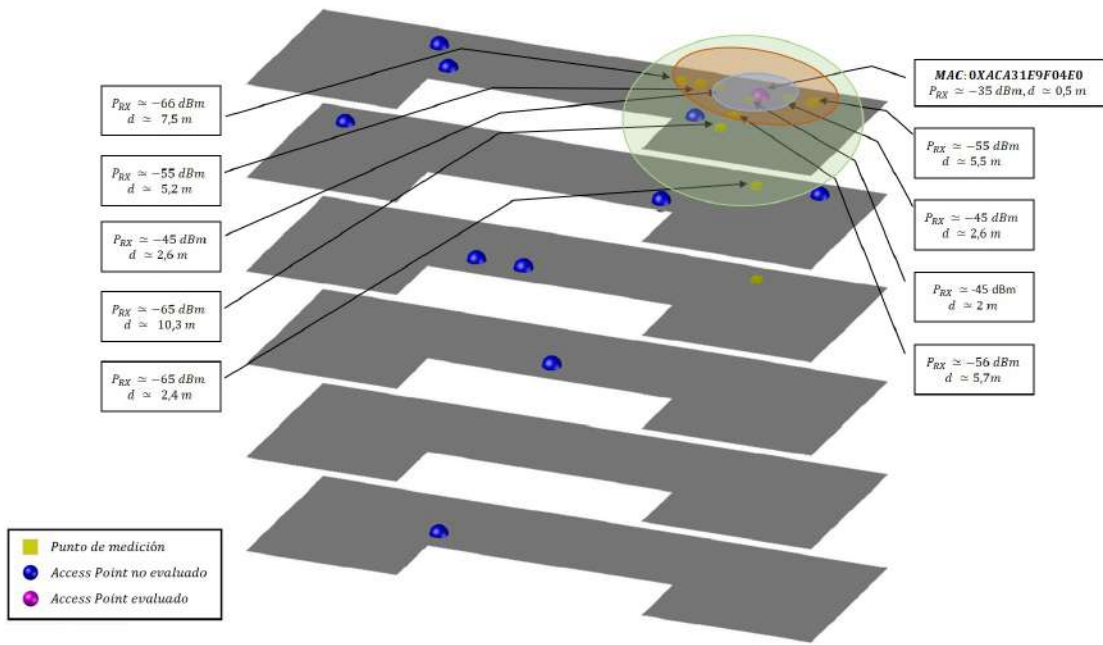


Figura 4.12: Niveles de potencia y distancia de AP 8

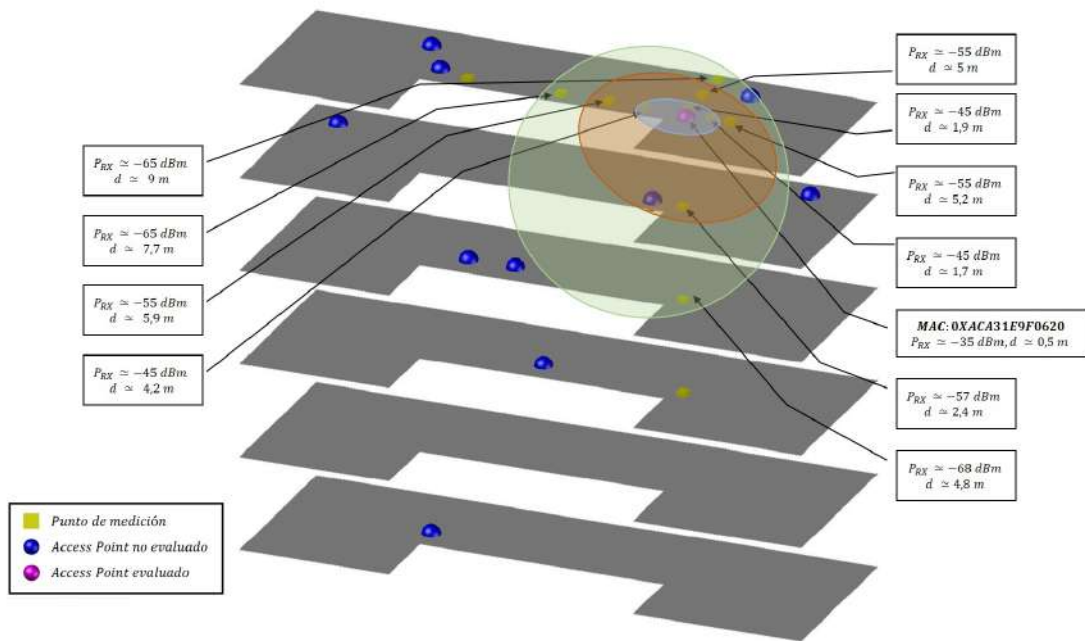


Figura 4.13: Niveles de potencia y distancia de AP 9

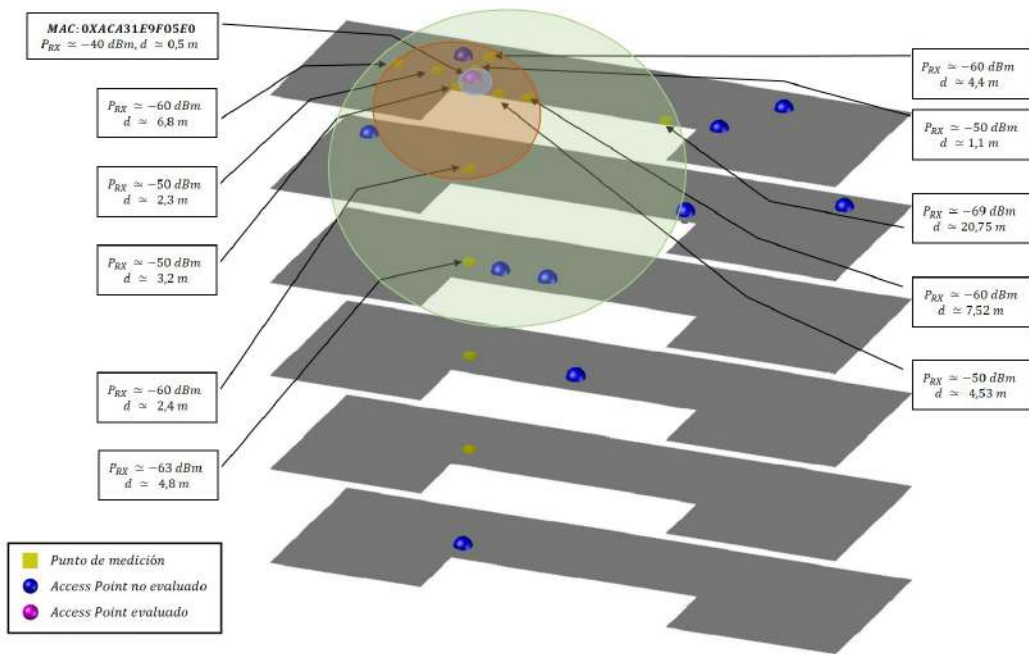


Figura 4.14: Niveles de potencia y distancia de AP 10

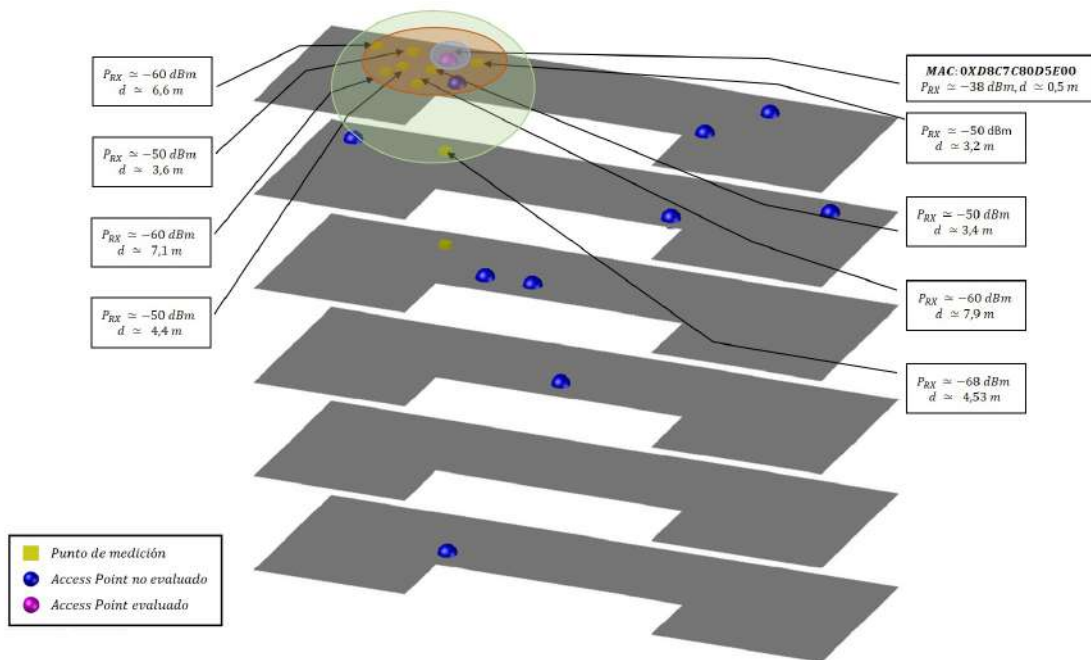


Figura 4.15: Niveles de potencia y distancia de AP 11

4.3.2. Diseño de las matrices de compatibilidad electromagnética

Con la información recolectada se crearon 3 matrices diferentes de acuerdo a un umbral, estimado primero en -45 dBm, luego en -60dBm y finalmente en -70 dBm. De manera gráfica se definieron las áreas de interferencia entre celdas y con esto se dio un peso a las restricciones de compatibilidad

electromagnética de 0 a 5, donde 0 es poca o nula interferencia y 5 es completamente interferido.

En la figura 4.16 se muestra las áreas de cobertura a -45 dBm y (4.5) es la matriz de compatibilidad $PUJ1$ correspondiente.

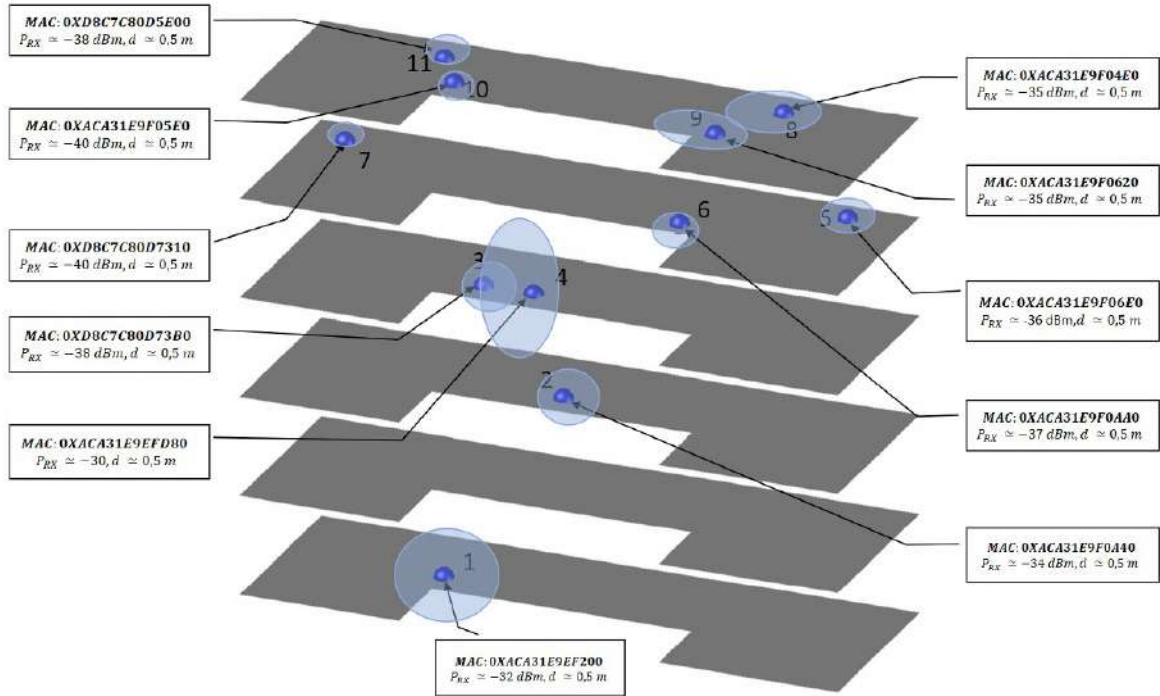


Figura 4.16: Área de cobertura a -45dBm

$$PUJ1 = \begin{bmatrix}
 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 5 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 5 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 5 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5
 \end{bmatrix} \tag{4.5}$$

La figura 4.17 muestra las áreas de cobertura a -60 dBm y (4.6) es la matriz de compatibilidad $PUJ2$ determinada.

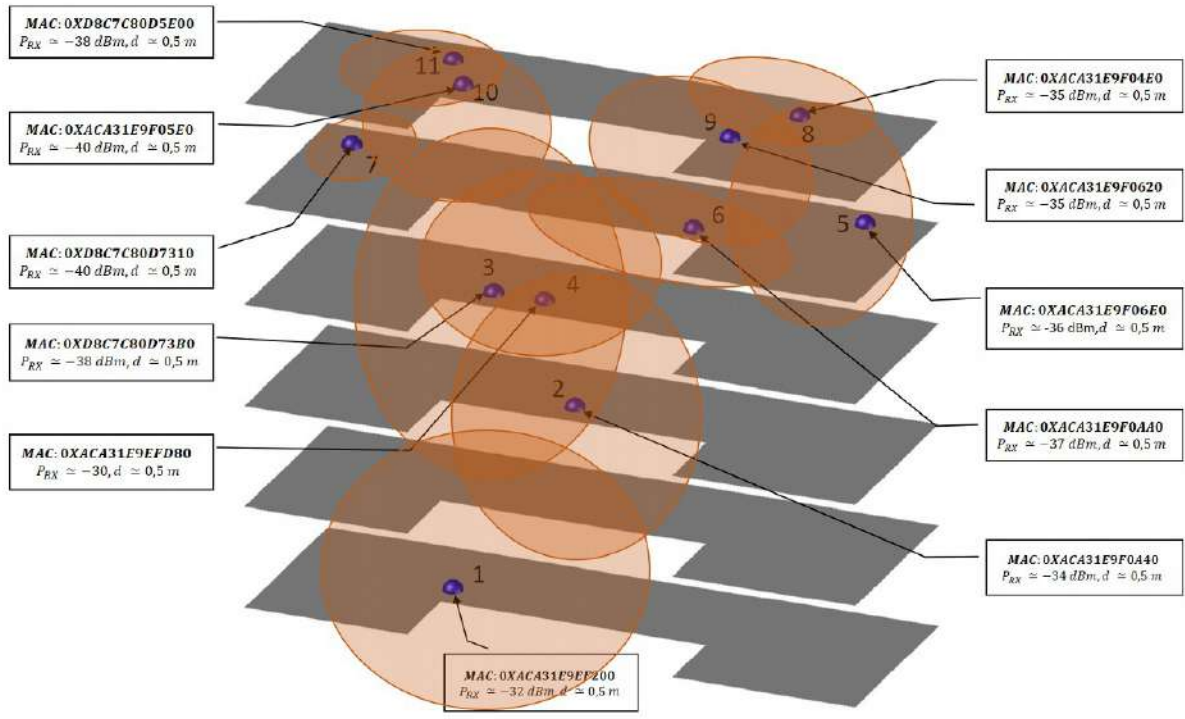


Figura 4.17: Área de cobertura a -60dBm

$$P U J 2 = \begin{bmatrix} 5 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 5 & 4 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 5 & 5 & 0 & 3 & 0 & 0 & 1 & 3 & 0 \\ 0 & 3 & 5 & 5 & 0 & 2 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 5 & 2 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 3 & 2 & 2 & 5 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 5 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 3 & 0 & 3 & 5 & 0 & 0 \\ 0 & 0 & 3 & 3 & 0 & 0 & 3 & 0 & 0 & 5 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 5 \end{bmatrix} \quad (4.6)$$

La figura 4.18 muestra las áreas de cobertura a -70 dBm y (4.7) corresponde a la matriz de compatibilidad $P U J 3$ hallada.

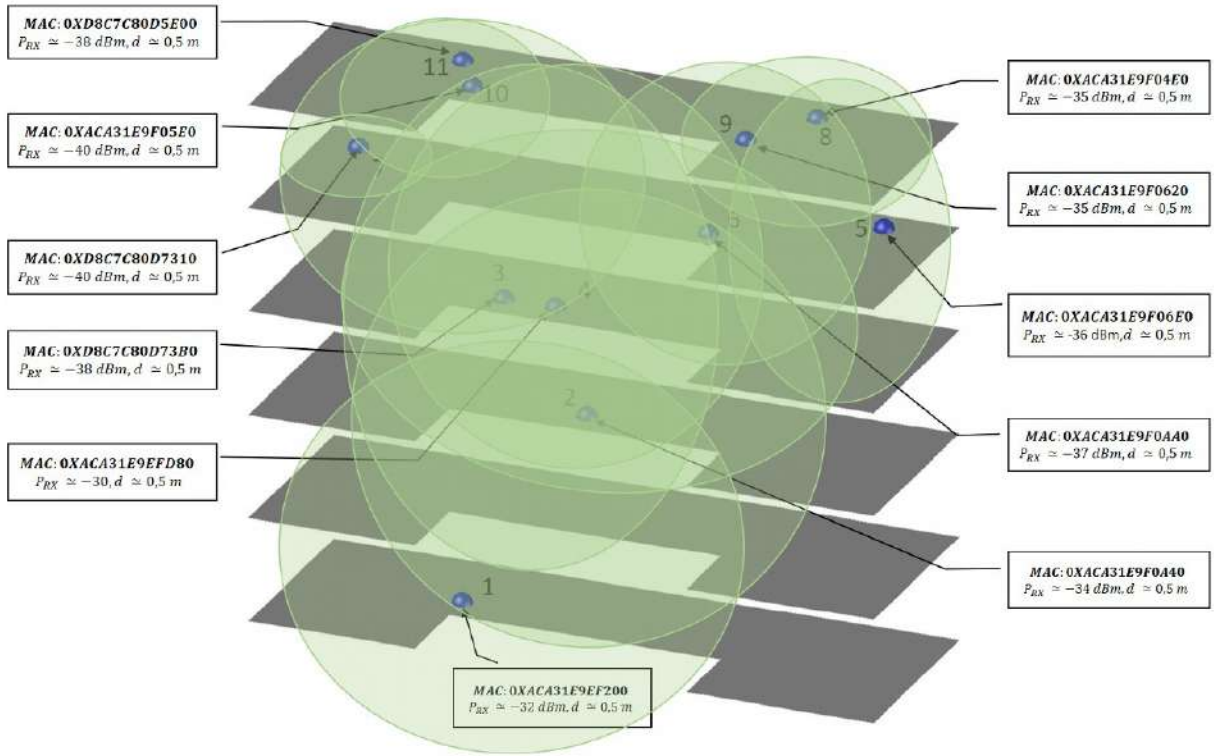


Figura 4.18: Área de cobertura a -70 dBm

$$P U J 3 = \begin{bmatrix} 5 & 4 & 2 & 3 & 0 & 2 & 0 & 0 & 0 & 1 & 0 \\ 4 & 5 & 4 & 4 & 2 & 4 & 0 & 0 & 3 & 3 & 0 \\ 2 & 4 & 5 & 5 & 2 & 4 & 2 & 2 & 4 & 4 & 3 \\ 3 & 4 & 5 & 5 & 1 & 5 & 4 & 2 & 4 & 5 & 3 \\ 0 & 2 & 2 & 1 & 5 & 3 & 0 & 4 & 4 & 1 & 0 \\ 2 & 4 & 4 & 5 & 3 & 5 & 1 & 2 & 4 & 4 & 1 \\ 0 & 0 & 2 & 4 & 0 & 1 & 5 & 0 & 0 & 5 & 4 \\ 0 & 0 & 2 & 2 & 4 & 2 & 0 & 5 & 4 & 2 & 0 \\ 0 & 3 & 4 & 4 & 4 & 4 & 0 & 4 & 5 & 4 & 0 \\ 1 & 3 & 4 & 5 & 1 & 4 & 5 & 2 & 4 & 5 & 5 \\ 0 & 0 & 3 & 3 & 0 & 1 & 4 & 0 & 0 & 5 & 5 \end{bmatrix} \quad (4.7)$$

Capítulo 5

Diseño

5.1. Descripción general

En este trabajo de grado se solucionaron problemas de programación enteros, siguiendo el procedimiento descrito en la figura 5.1. Las entradas del programa son las matrices de compatibilidad electromagnética C , y el vector de demanda d . El código desarrollado en C se encuentra en el Anexo 8.1.

5.1.1. Diseño de variables y constantes del programa

El siguiente paso es definir las constantes del programa, están los valores de la tabla 2.1, la cantidad de partículas y el número máximo de iteraciones depende de la dimensión y complejidad de cada problema.

Las variables del sistema son x , que son las posibles soluciones que se inicializan aleatoriamente sobre el espacio de búsqueda. Asociado a esta posición hay una velocidad v , igualmente aleatoria definida entre $(-V_{max}, V_{max})$. Por facilidad para la implementación x (y v correspondientemente) se implemento como una matriz, donde las filas corresponden a las partículas del sistema y las columnas son la demanda máxima d_{max} solicitada en d multiplicada por el número de celdas del sistema. En la figura 5.2 se detalla mejor este cambio.

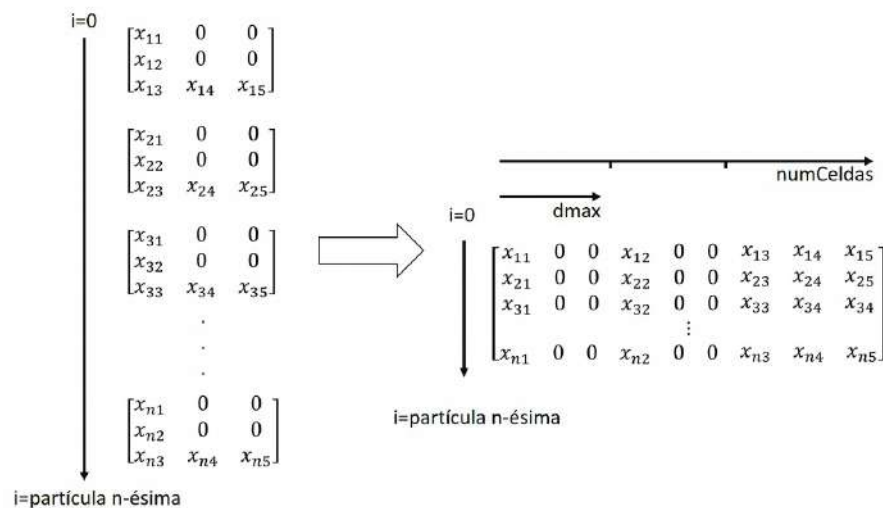


Figura 5.2: Cambio de interpretación de las matriz x .

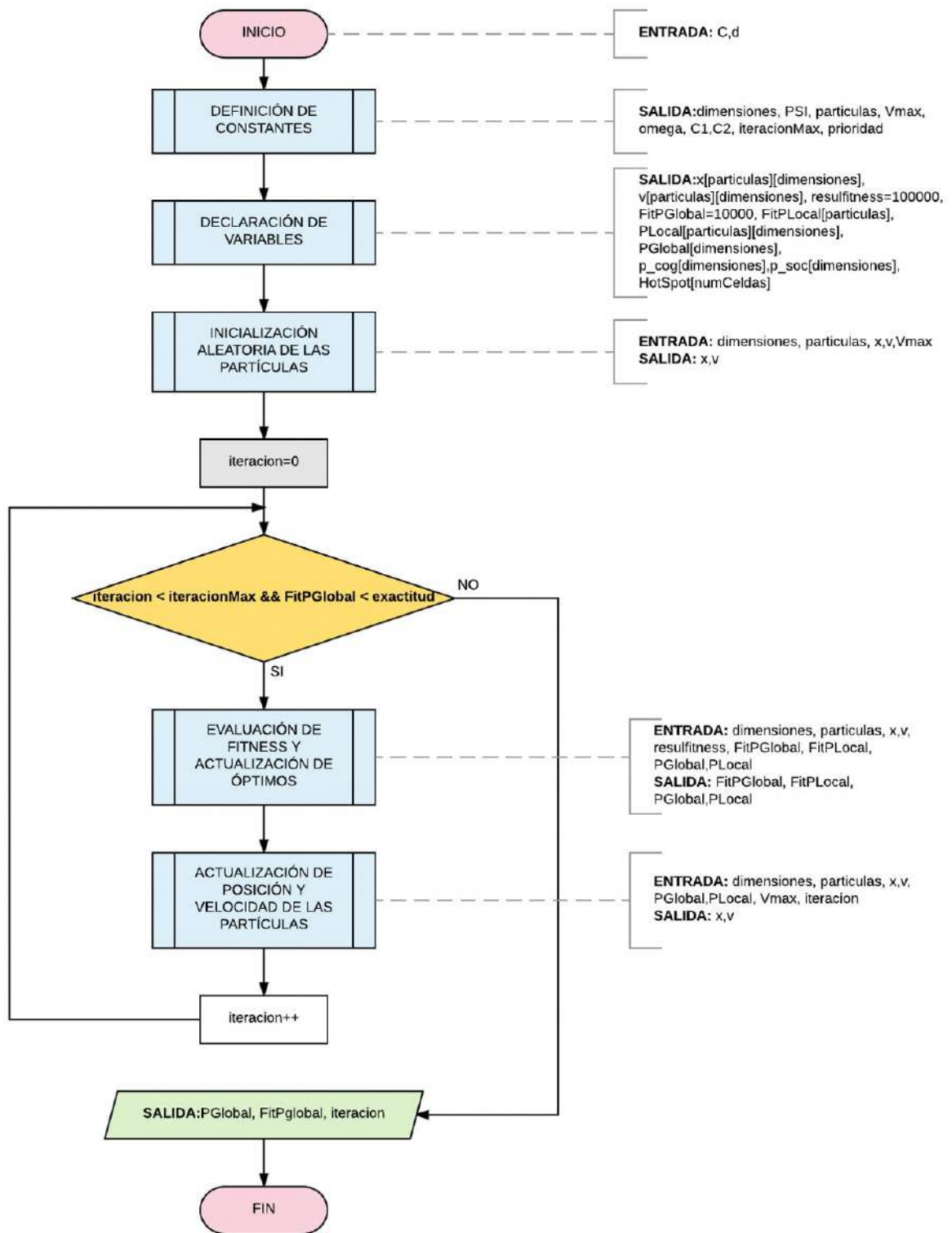


Figura 5.1: Diagrama de flujo de la solución de *FAP* con *PSO*.

PLocal corresponde a la solución con mejor respuesta a la función fitness encontrada de cada partícula, y PGlobal corresponde a la mejor solución encontrada por todo el enjambre en el histórico de iteraciones. Estos dos se van actualizando de acuerdo a PSO con (2.1) y (2.2), ayudados por FitPGlobal y FitPLocal, variables que guardan los resultado de la evaluación de la función fitness de PGlobal y cada uno de los PLocales.

Para la inicialización de las variables x y v , se siguió el diagrama de flujo de la figura 5.3, el cual funcionó con las pruebas iniciales del algoritmo, pero debió ser modificado como se explicará en la sección 5.2 para ayudar con la convergencia de los problemas más grandes.

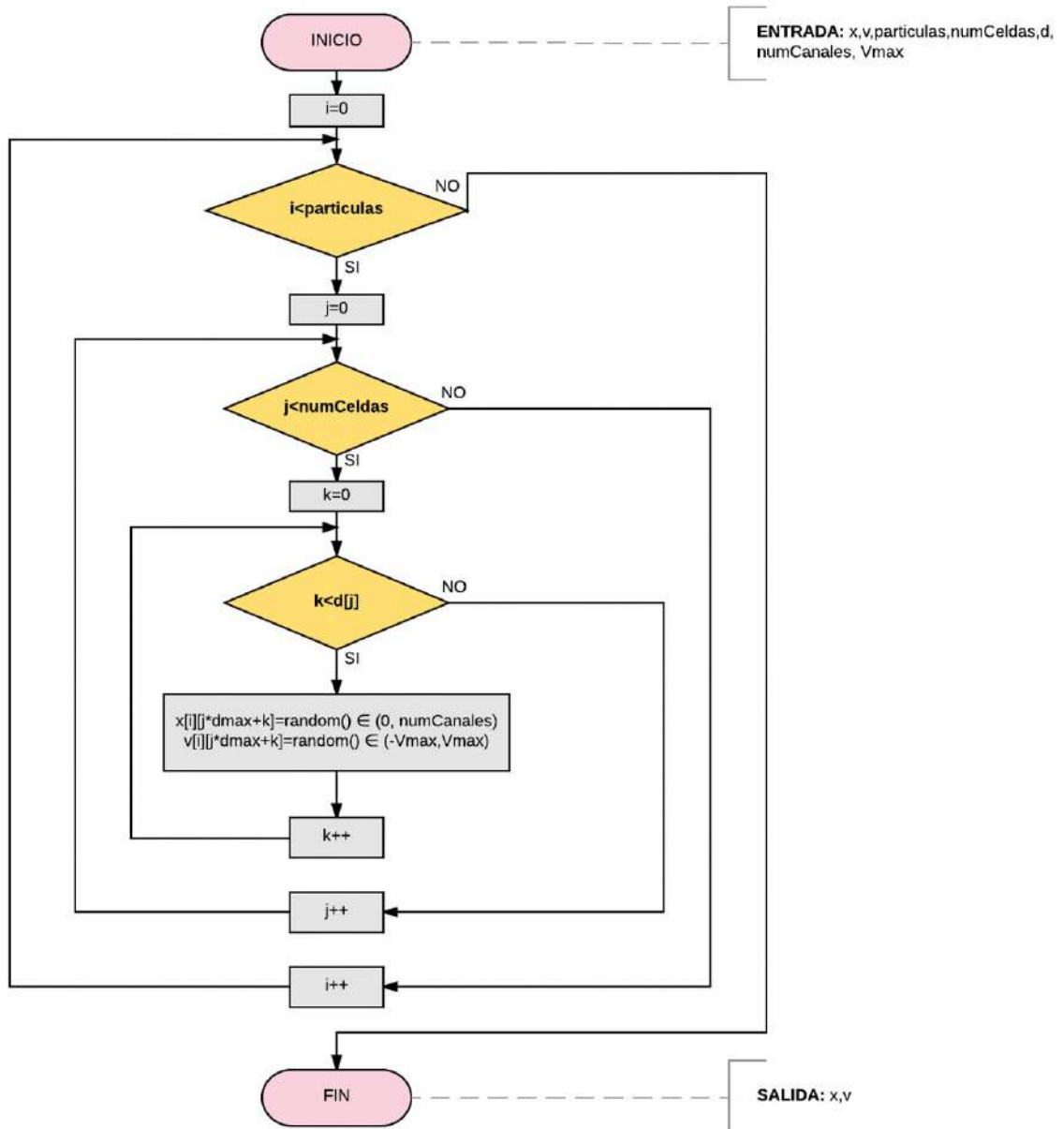


Figura 5.3: Diagrama de flujo de la inicialización aleatoria de las partículas y sus velocidades.

5.1.2. Diseño de evaluación de la función objetivo y actualización de óptimos

El proceso de la figura 5.4, se desarrollo en 3 bloques fundamentalmente:

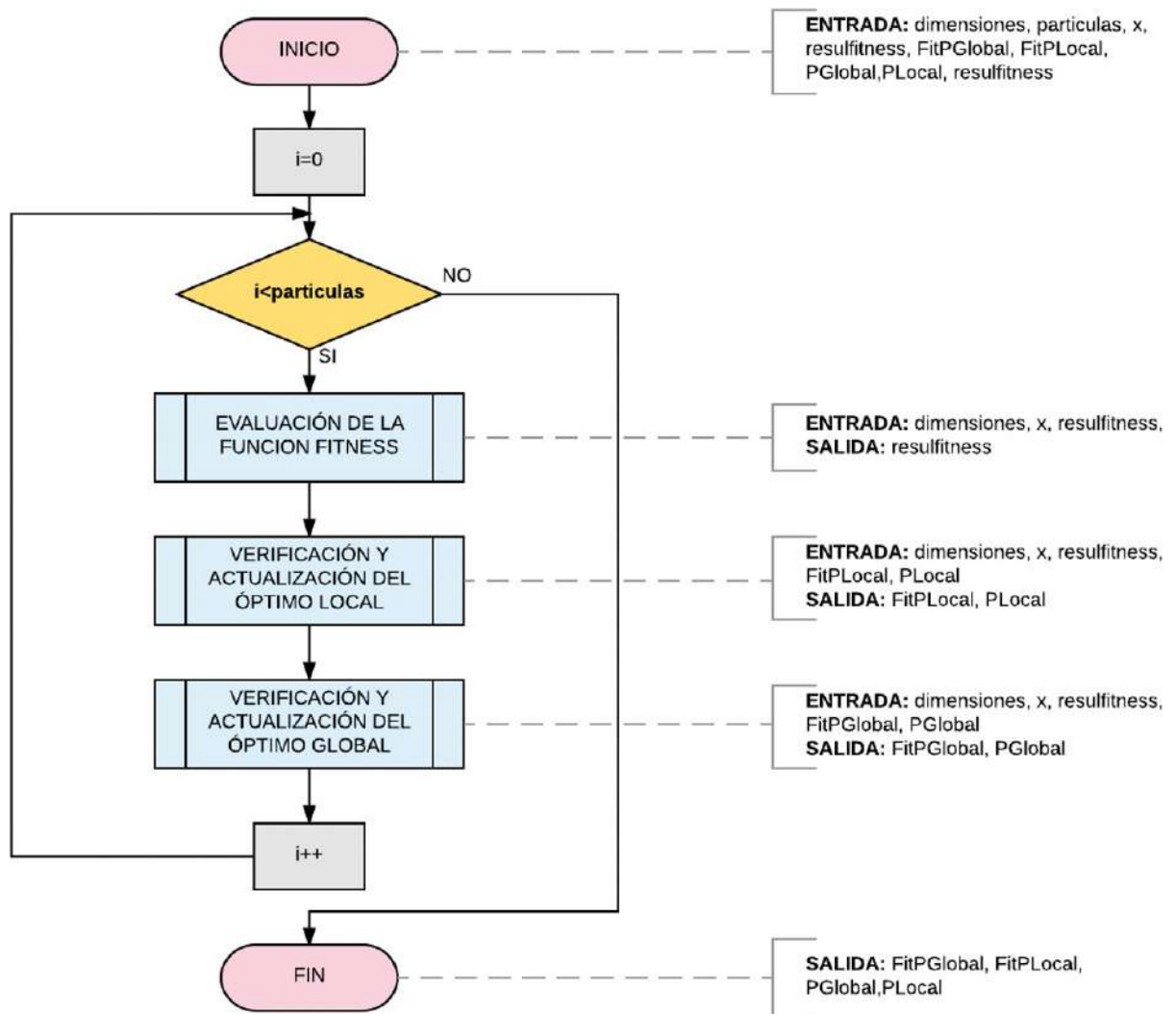


Figura 5.4: Diagrama de flujo de la evaluación del fitness y actualización de óptimos.

- La evaluación de la función objetivo para las restricciones de canal adyacente, co-Canal y co-Sitio, explicados en los diagramas 5.5 y 5.6 , teniendo en cuenta que por la implementación no se puede incumplir la demanda.

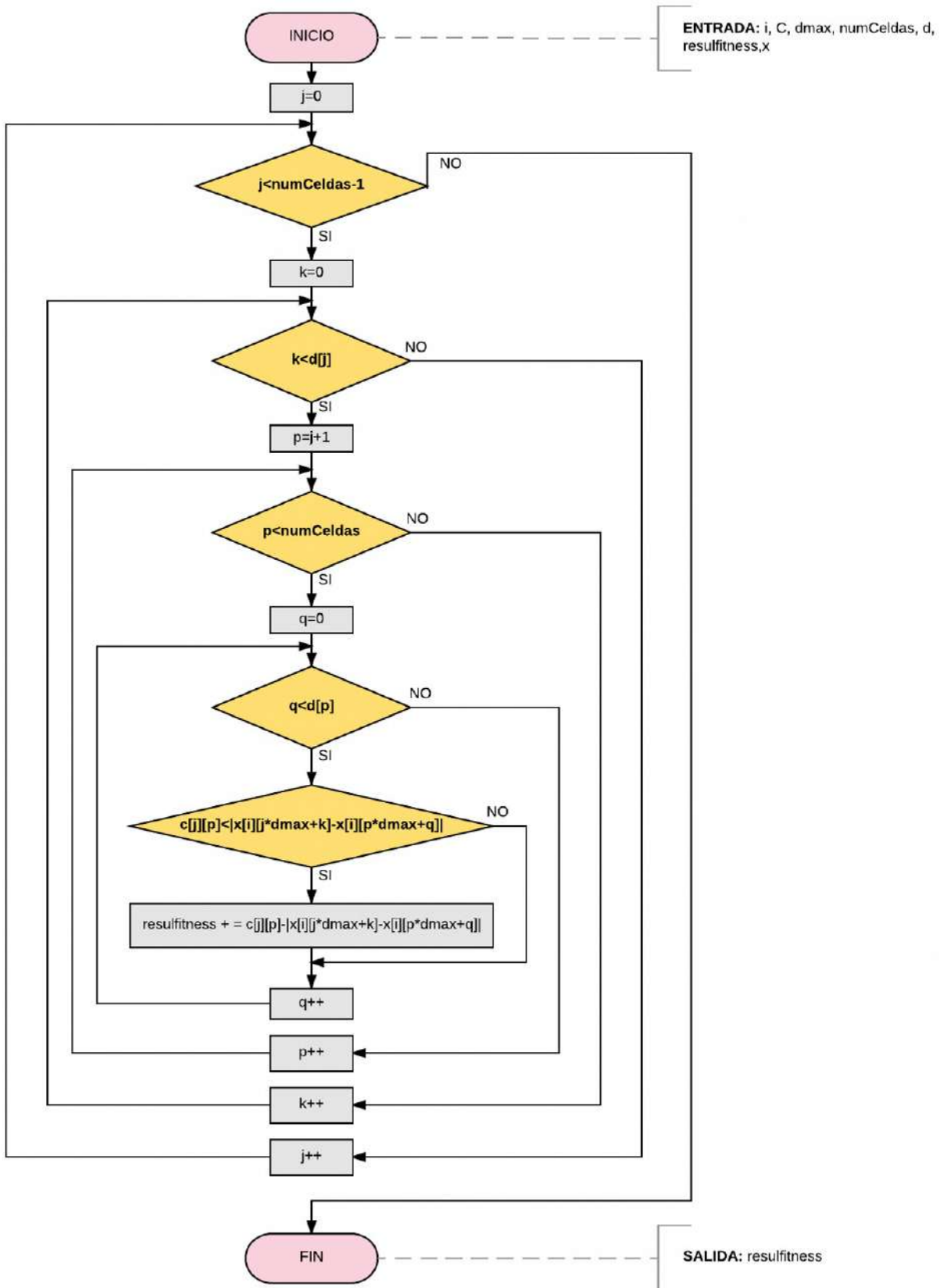


Figura 5.5: Diagrama de flujo de la función objetivo para canal adyacente y co-canal.

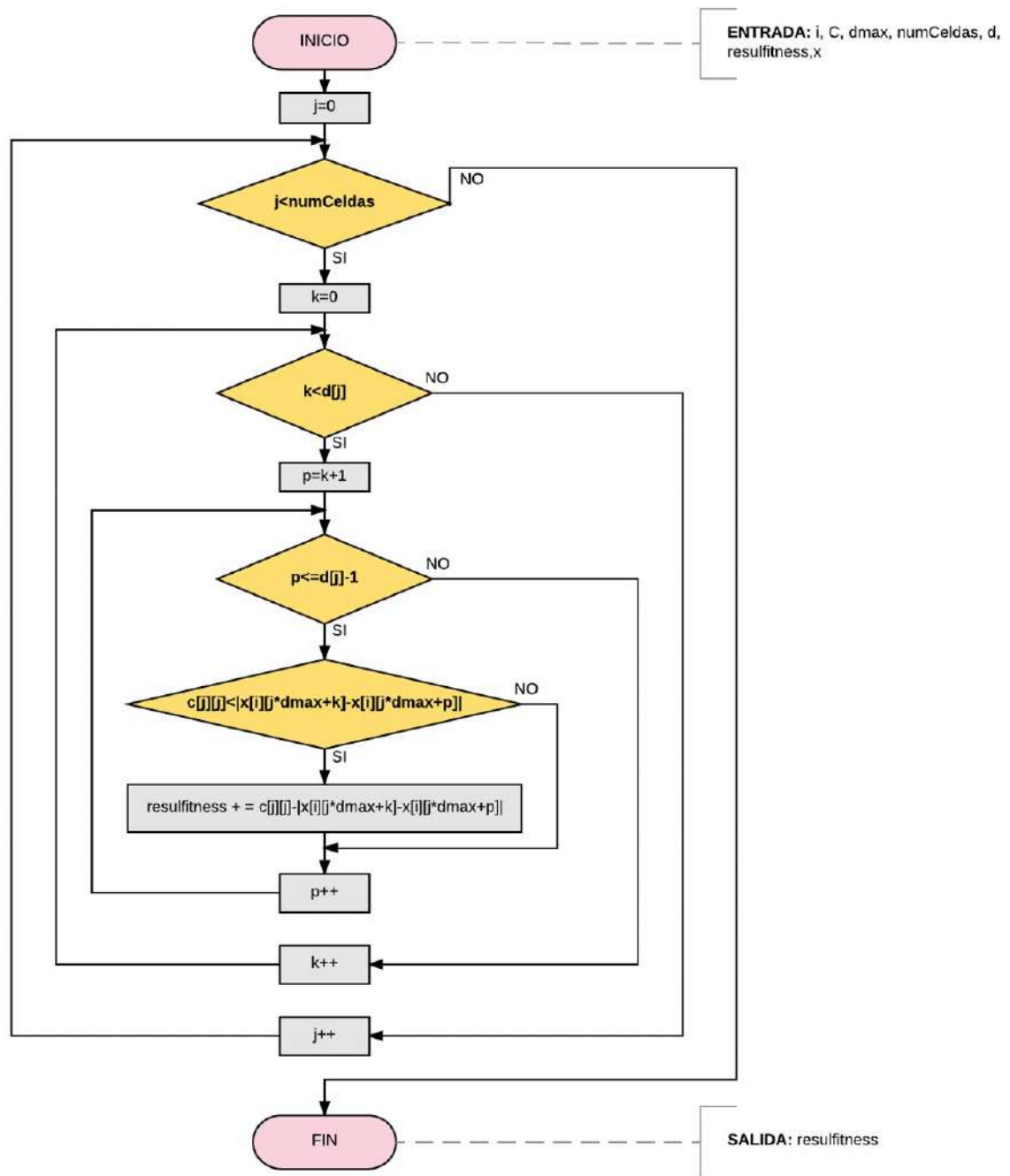


Figura 5.6: Diagrama de flujo de la función objetivo para co-sitio.

- La actualización del óptimo local, se encuentra detallado en la figura 5.7.

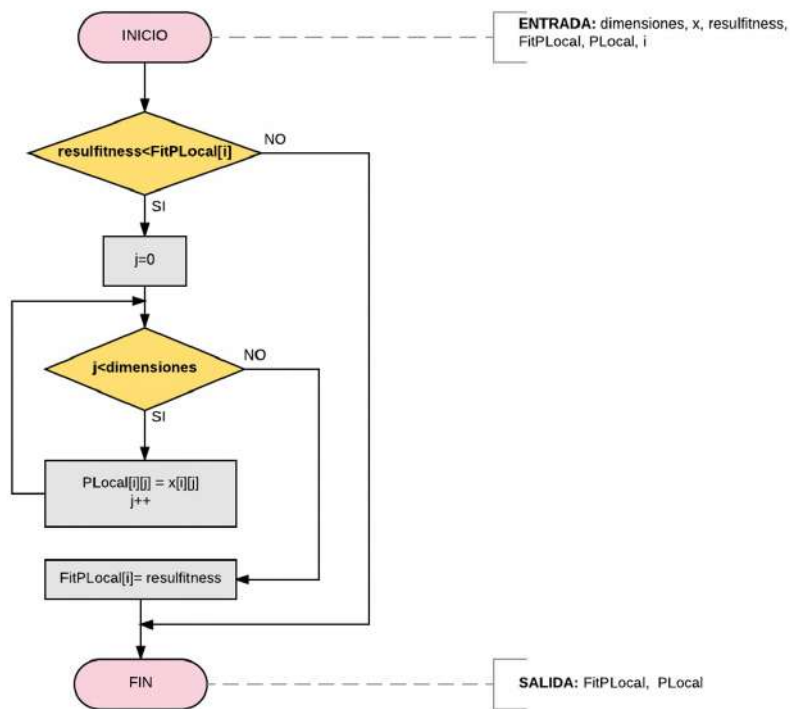


Figura 5.7: Diagrama de flujo de la actualización del óptimo local.

- La actualización del óptimo global, descrito en el diagrama de la figura 5.8.

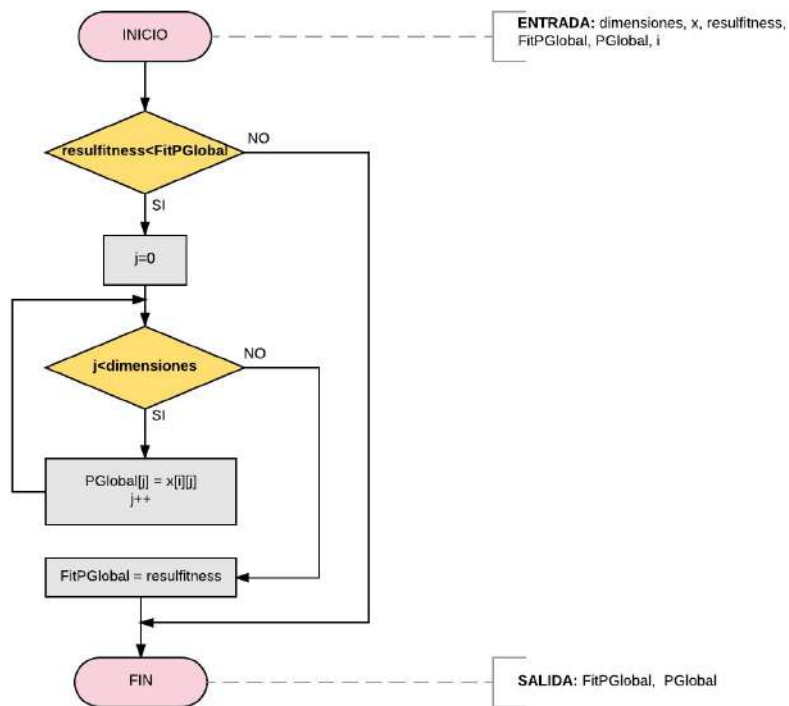


Figura 5.8: Diagrama de flujo de actualización del óptimo global.

5.1.3. Diseño de actualización de posición y velocidad de las partículas

Para hacer el cambio de la posición y velocidad de las partículas, es necesario calcular el valor de ω , que depende del número de la iteración pues tiene una caída lineal respecto al tiempo. De igual manera, en cada asignación se cambian los términos aleatorios del parámetro cognitivo y social de (2.1). El proceso se describe en la figura 5.9.

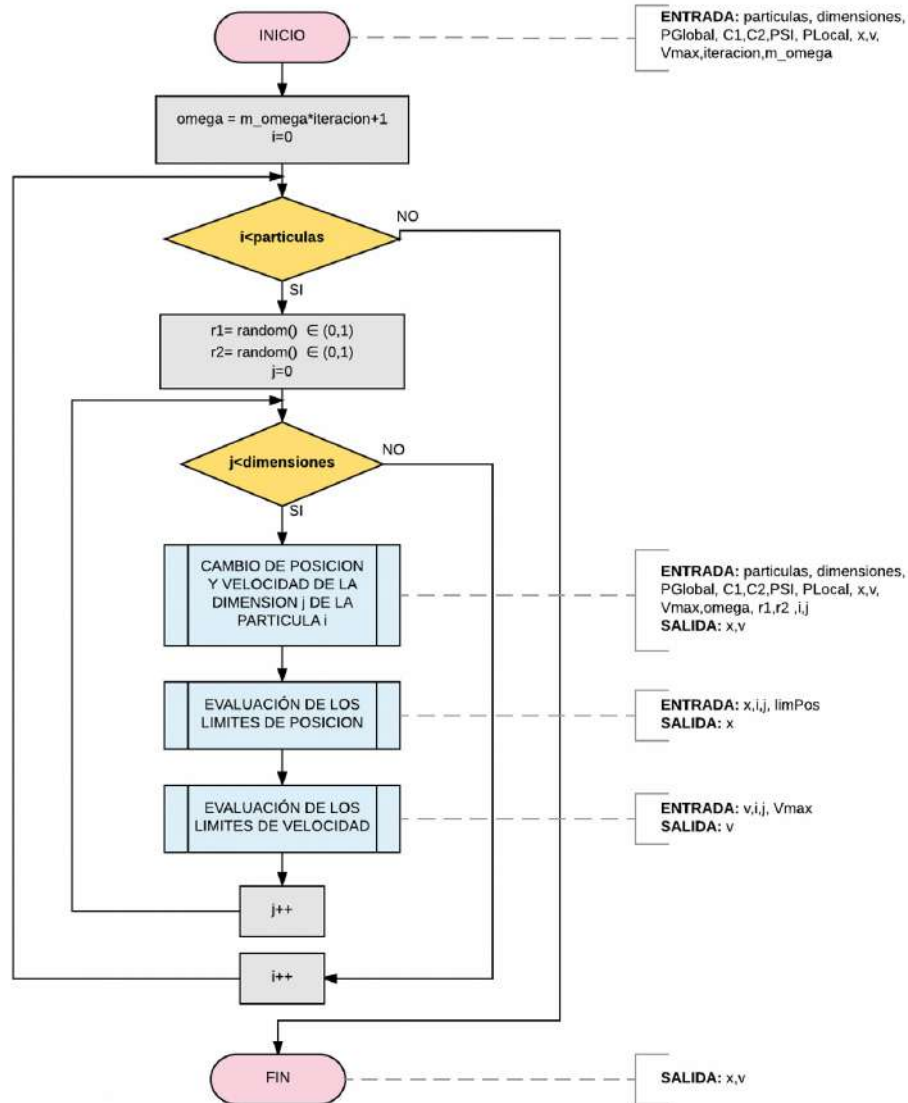


Figura 5.9: Diagrama de flujo de la actualización de la posición y velocidad de las partículas

El diagrama 5.10 muestra como se calcula el cambio de velocidad y posición por dimensión de cada partícula.

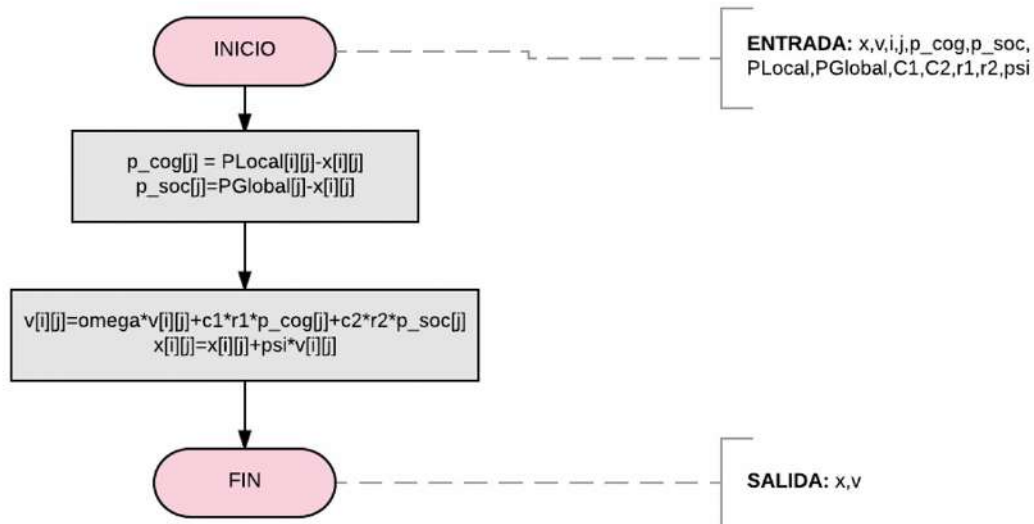
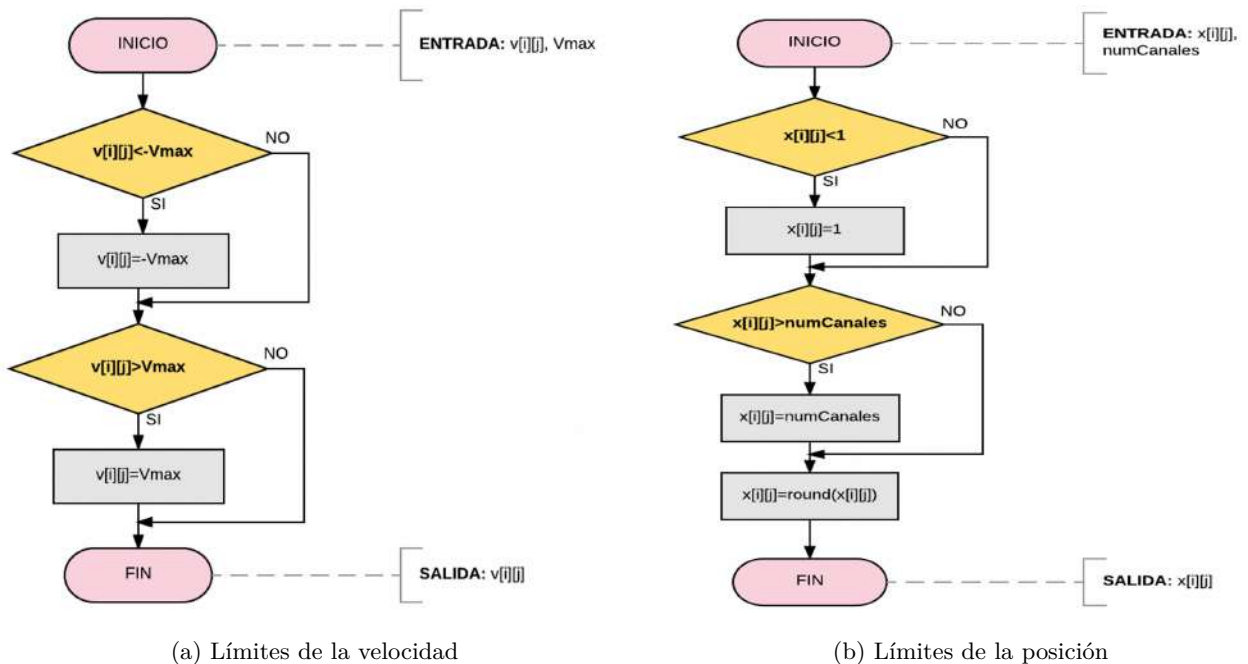


Figura 5.10: Diagrama de flujo del cálculo del cambio de posición y velocidad

La figura 5.11a describe la evaluación de los límites de la velocidad, que como se mencionó, está limitado por $(-Vmax, Vmax)$, de manera similar en la figura 5.11b, se muestra como se limitaron las posiciones de las partículas y se redondeó a su entero más cercano en cada iteración.



(a) Límites de la velocidad

(b) Límites de la posición

Figura 5.11: Diagrama de flujo de la evaluación de los límites de los cambios de posiciones y velocidades

5.2. Modificación del algoritmo

Dada la dificultad encontrada para la convergencia del algoritmo, se implementaron dos métodos adicionales a PSO en la inicialización de las partículas, que permitieron ubicar las partículas sobre

una región sub-óptima del espacio de búsqueda. La primera es la técnica *HotSpot*, con la cual se seleccionan n número de celdas prioritarias, a las cuales se les hace una asignación «manual» de canales, bajo ciertos criterios. Se tienen en cuenta las restricciones de compatibilidad electromagnética, la condición de demanda y especialmente la condición de empaquetado. Y el segundo método es los intervalos de límites ajustables, con el cual se asegura que las celdas que nos son prioritarias sean asignadas con canales de orden ascendente, dentro del rango de canales disponibles, en donde se procura hacer que los canales respeten la restricción co-site. El procedimiento se detalla en la figura 5.12.

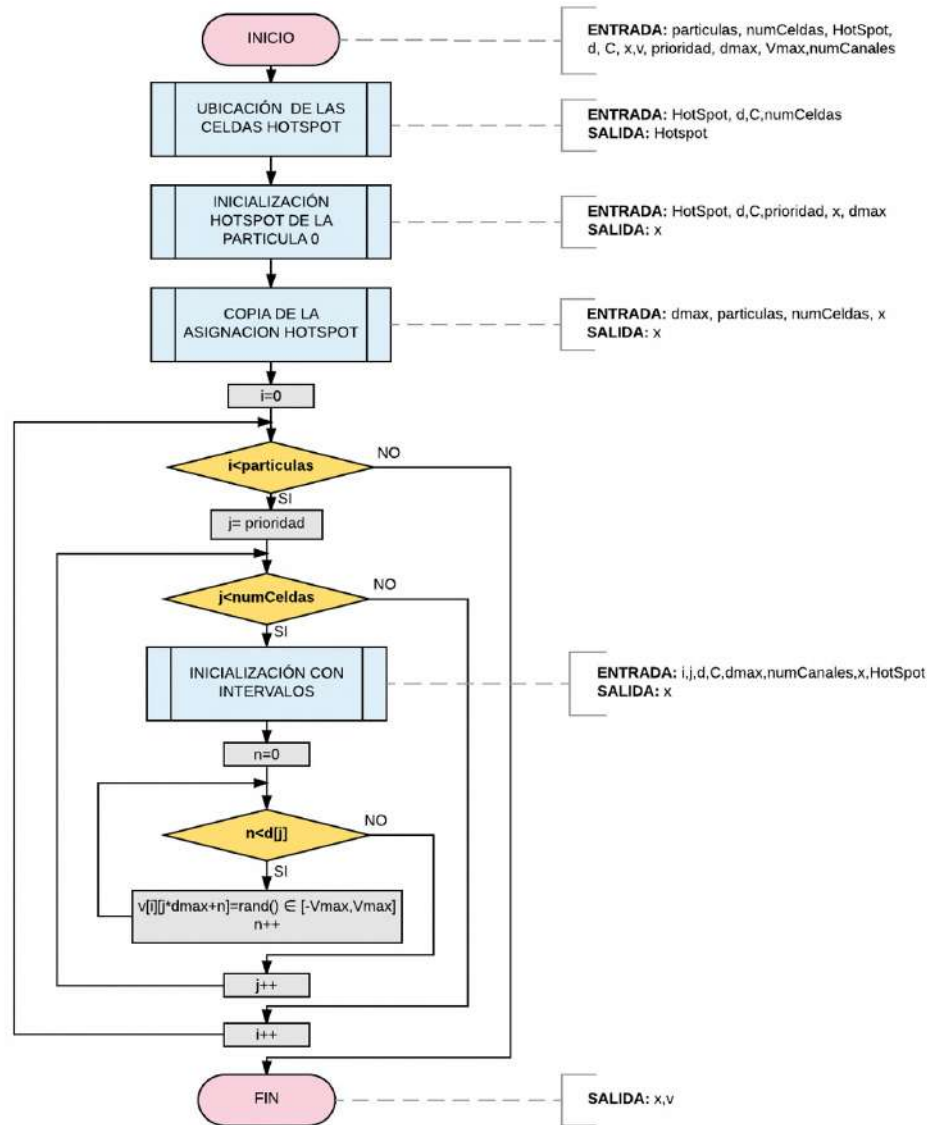


Figura 5.12: Diagrama de flujo de la inicialización de las partículas con la técnica de HotSpot e intervalos.

Primero se deben ubicar las celdas HotSpot de acuerdo a la prioridad definida como parámetro, se organizan los índices de las celdas de acuerdo al valor obtenido multiplicando los valores de la matriz de compatibilidad en su diagonal C_{jj} , con el valor correspondiente por celda de la demanda d_j , la ubicación de las celdas se describe en la figura 5.13.

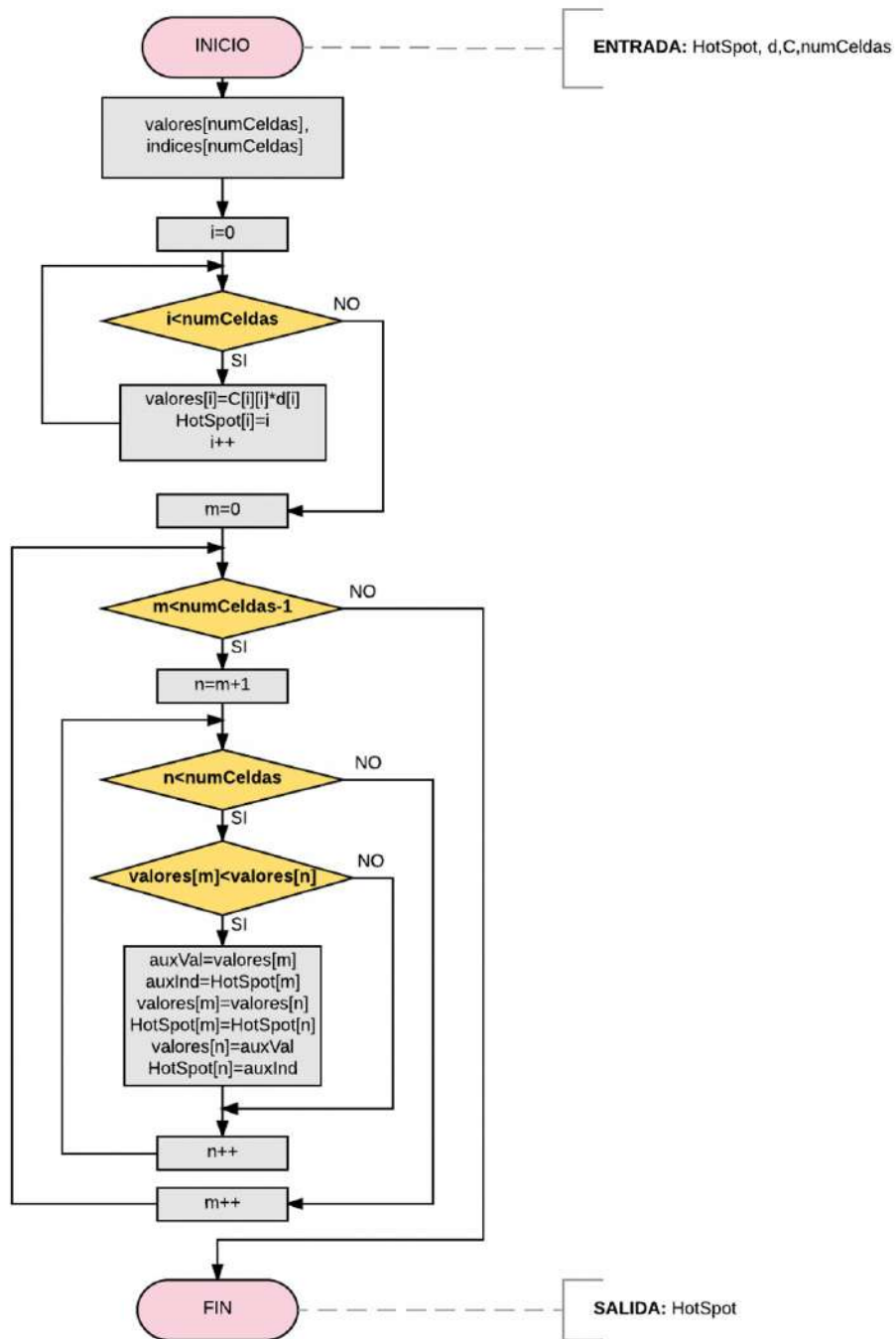


Figura 5.13: Diagrama de flujo del algoritmo para la ubicación de la celdas HotSpot.

Se debe tener en cuenta que la asignación se realiza para la partícula 0 y luego se copia a las demás, a todas las partículas se les ubica sobre el mismo sub-espacio de búsqueda. El proceso se describe en la figura 5.14.

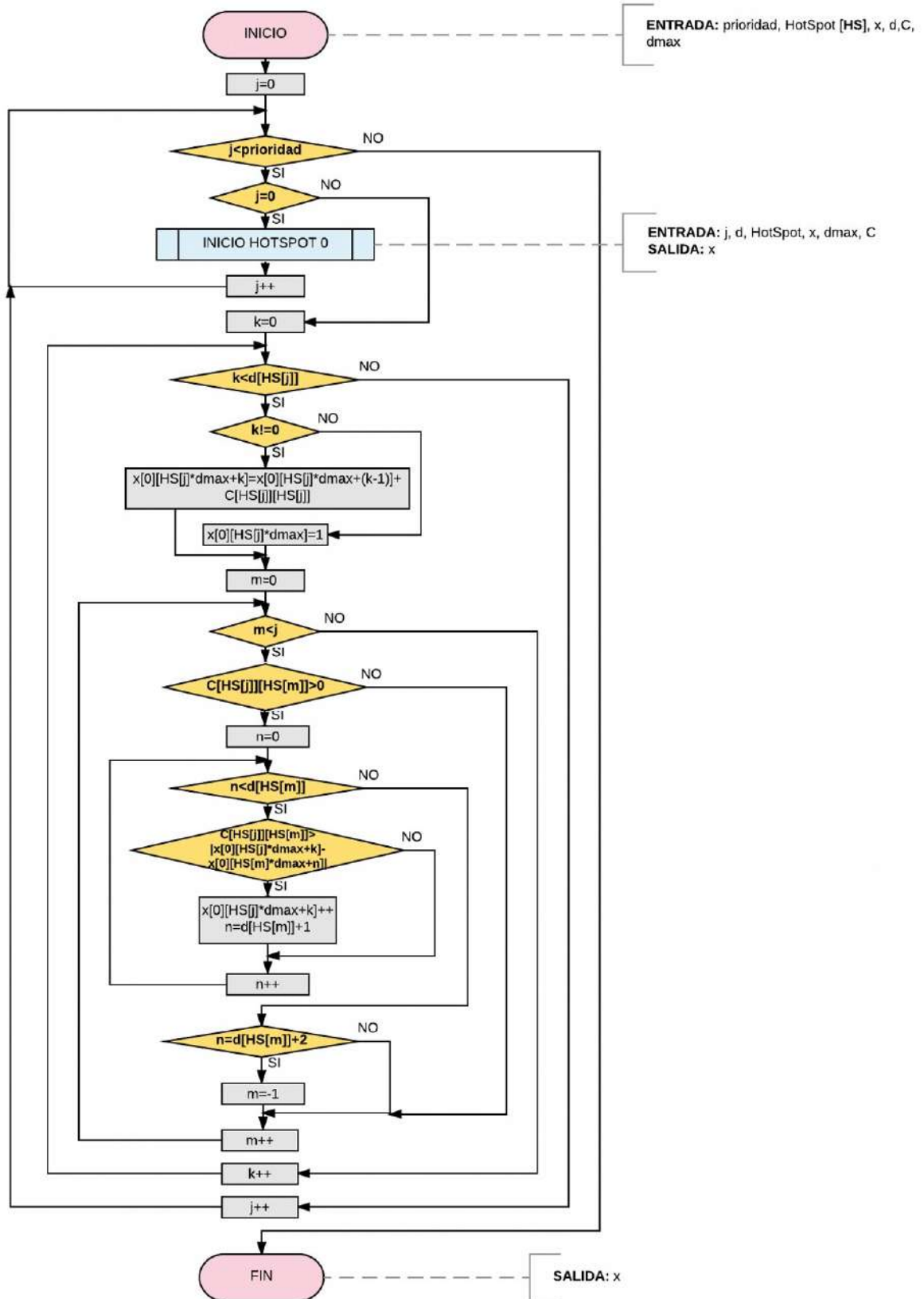


Figura 5.14: Diagrama de flujo de la inicialización de la primera partícula.

Además, en la figura 5.15 se muestra el sub-proceso de la inicialización de la primera celda prioritaria de la partícula 0, y en la figura 5.16 se muestra el proceso de copiado a las demás partículas.

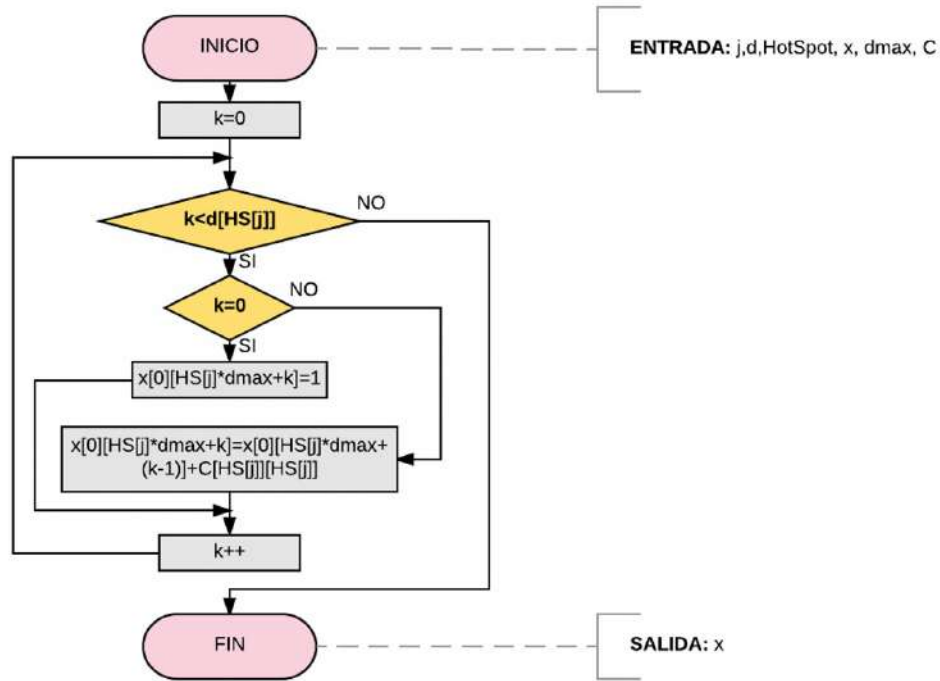


Figura 5.15: Diagrama de flujo de la inicialización de la primera celda prioritaria.

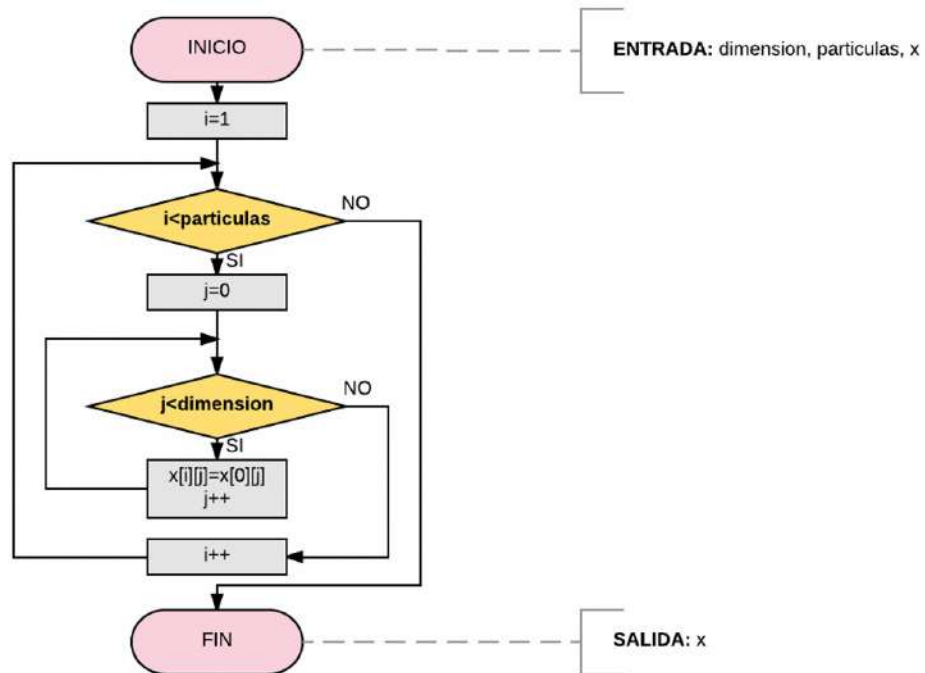


Figura 5.16: Diagrama de flujo del proceso de copia de la asignación a las demás partículas.

Luego se hace la inicialización semi-aleatoria de las demás celdas en cada partícula, como se muestra en la figura 5.12, los intervalos se definen de acuerdo a (5.1), donde a cambia según el algoritmo descrito en la figura 5.17.

$$x[i][HotSpot[j] * dmax + k] = \left[a, \left\lfloor \frac{numCanales - a + 1}{d[HotSpot[j]] - k} \right\rfloor + (a - 1) \right] \quad (5.1)$$

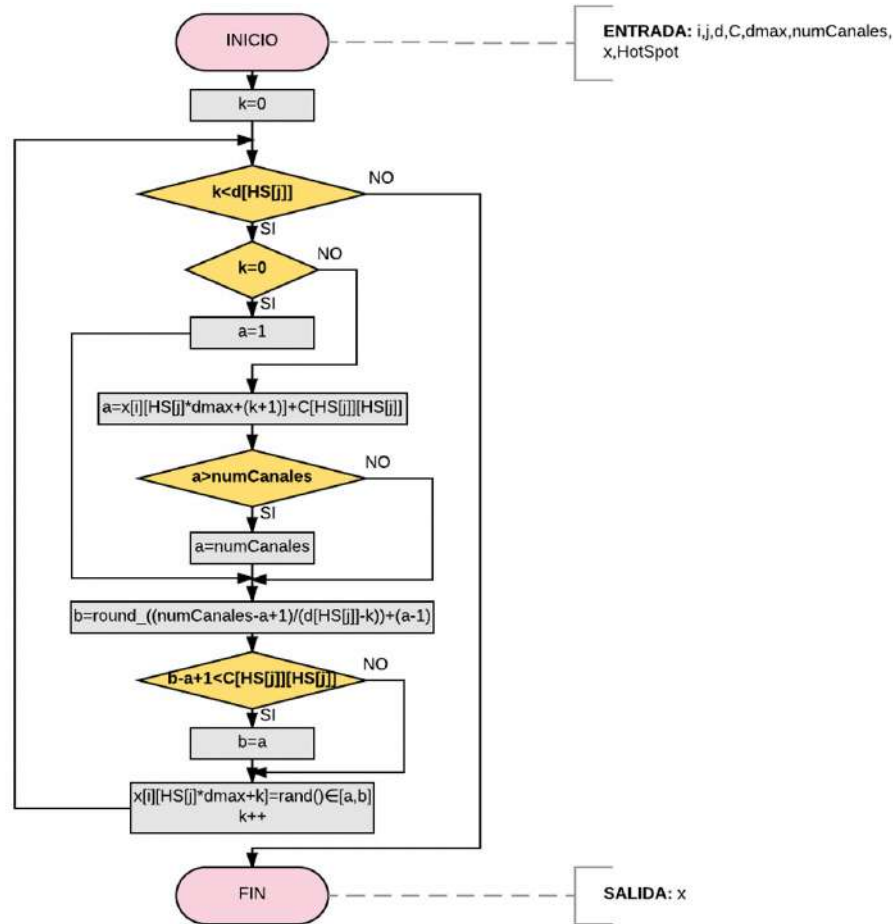


Figura 5.17: Diagrama de flujo de la generación de posiciones aleatorias con intervalos.

5.3. Descripción del algoritmo en MPI

Para implementarlo en una plataforma MPI es necesario modificar el algoritmo para que sean procesos paralelos, esto se hace haciendo uso de algunos de los métodos básicos de MPI descritos en la sección 2.3.3. Prácticamente lo que se hizo fue crear un enjambre diferente en cada procesador, de esta forma se hace una mayor cobertura inicial del espacio de búsqueda, y luego se definió el mejor resultado obtenido, de acuerdo a su curva de respuesta del fitness global. El proceso se detalla en el diagrama de la figura 5.18. el código desarrollado se encuentra en el anexo 8.2.

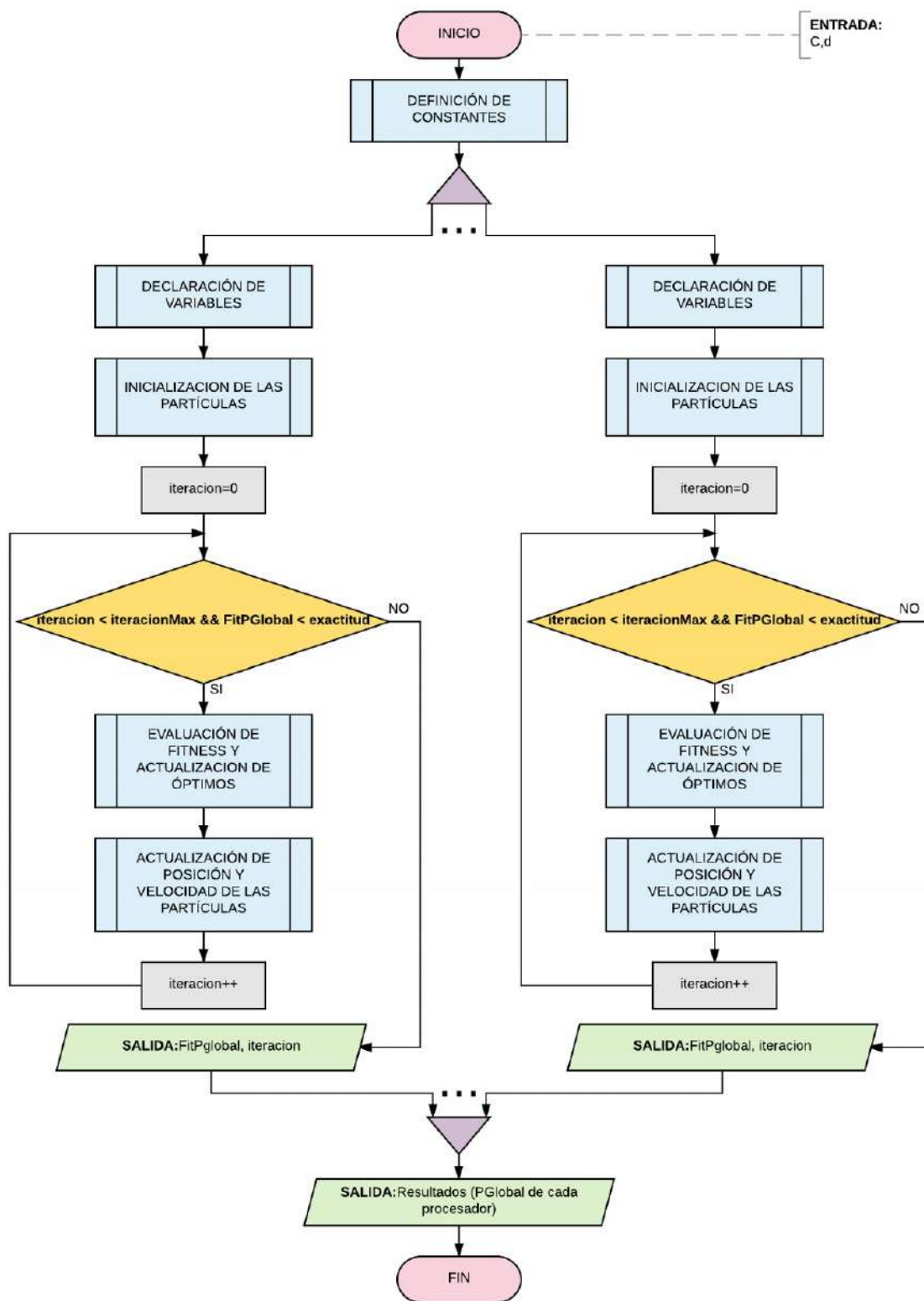


Figura 5.18: Diagrama de flujo de la versión paralela del algoritmo PSO.

Capítulo 6

Pruebas y evaluación

Se realizaron varias pruebas en el entorno de prueba de un computador convencional *Lenovo ThinkCentre* con un procesador Intel Core i7-4765T de 2 GHz y 8 GB de RAM. Cada prueba se realizó 30 veces con 800 iteraciones máximas antes de descartar un óptimo local. Se varió el tipo de inicialización de las partículas: aleatorio y con el método de intervalos y celdas *HotSpot*.

Con la técnica de inicialización aleatoria se hizo un análisis de la tasa de convergencia respecto a el tamaño del enjambre, variando en 20, 40 y 60 la cantidad de partículas. Y con la inicialización por intervalos y celdas *HotSpot* se hizo una variación en el nivel de prioridad otorgado en cada prueba en 1, 2 y 4, y se definieron sus curvas de convergencia en cada caso. Además, se agregó una comparación entre los mejores resultados obtenidos con 60 partículas con ambas técnicas de inicialización, sus asignaciones e interferencia, apoyado con unas mallas de asignación de canales.

6.1. EX1: 4 celdas,11 canales

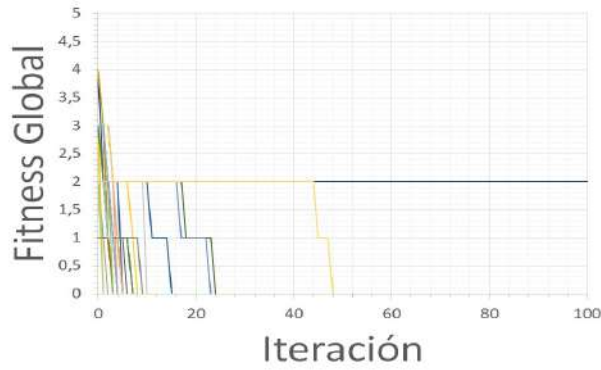
La tabla 6.1 muestra las tasas de convergencia del algoritmo bajo los diferentes parámetros definidos para el protocolo de pruebas, con el problema EX1 descrito en la sección 4.2.1. Para esta prueba se realizaron pruebas de escritorio, buscando posibles fallas de los algoritmos implementados, los cuales están descritos en el capítulo 5.

Entorno de prueba	Inicialización	Tamaño del enjambre	Prioridad	Tasa de convergencia
Convencional	Aleatorio	20	NA	28/30
Convencional	Aleatorio	40	NA	30/30
Convencional	Aleatorio	60	NA	30/30
Convencional	Generación Aleatoria	NA	NA	0/30
Convencional	intervalos & HS	60	1	30/30
Convencional	intervalos & HS	60	2	30/30
Convencional	intervalos & HS	60	4	30/30

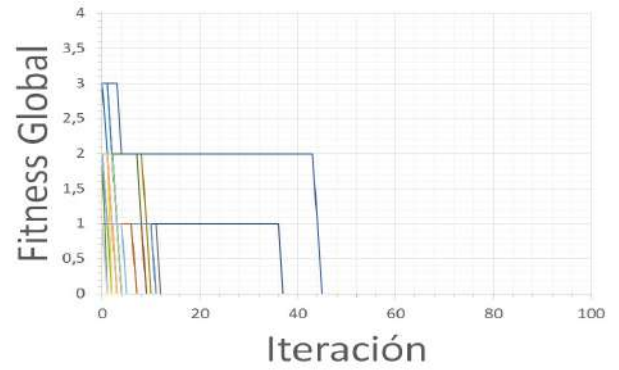
Cuadro 6.1: Tasas de convergencia del problema EX1.

6.1.1. Inicialización aleatoria

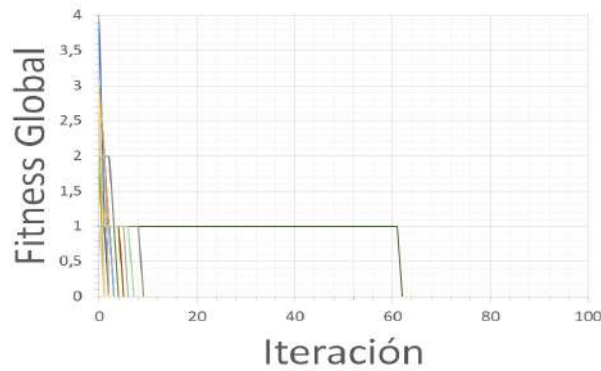
En la figura 6.1 se muestran los resultados de la prueba de la función objetivo para el problema EX1 con inicialización aleatoria con diferentes tamaños del enjambre. Se registran las primeras 100 iteraciones con el objetivo de ver claramente los cambios de la función objetivo, pero las pruebas se hicieron 800 iteraciones máximas al igual que las demás.



(a) 20 partículas.



(b) 40 partículas.



(c) 60 partículas.

Figura 6.1: Curvas de convergencia para EX1, inicialización aleatoria.

La figura 6.2 muestra los 3 mejores resultados obtenidos por la evaluación de la función fitness para cada tamaño de enjambre respectivamente, como el problema sencillo en algunos casos con la inicialización aleatoria se asignan canales que cumplen las características de la matriz (4.1).

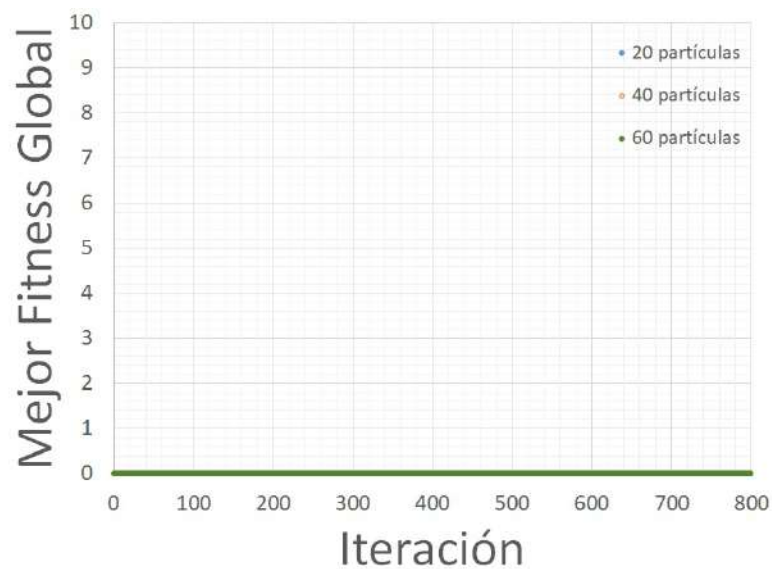


Figura 6.2: Curvas de las 3 mejores convergencias para EX1, inicialización aleatoria.

La figura 6.3 muestra la asignación de canales realizada por una de las mejores partículas del enjambre de 60, teniendo en cuenta el orden dado por (6.1).

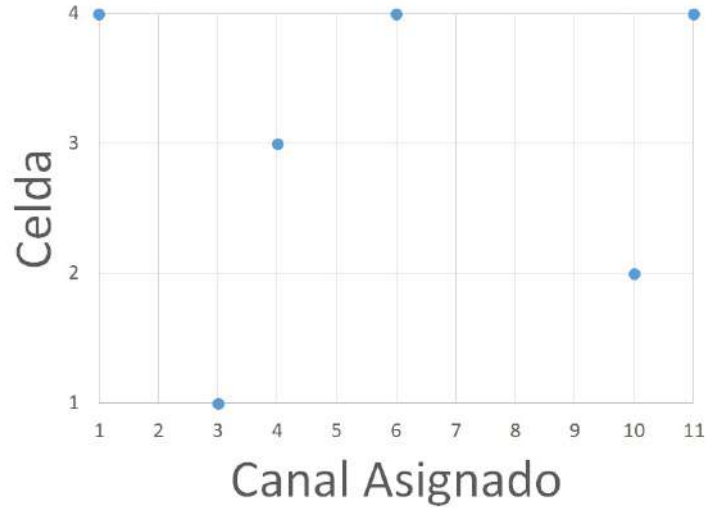


Figura 6.3: Malla de asignación para EX1, inicialización aleatoria con 60 partículas.

$$A_{EX1aleatorio} = \begin{bmatrix} 3 & 0 & 0 \\ 10 & 0 & 0 \\ 4 & 0 & 0 \\ 11 & 6 & 1 \end{bmatrix} \quad (6.1)$$

6.1.2. Generación aleatoria

Se realizaron 30 generaciones de asignaciones aleatorias en un computador convencional, pero ninguna tuvo convergencia, llegando a tener resultados de 1 en la evaluación de la función objetivo. El valor promedio de *Fitness* fue de 6,83. Los resultados se encuentran en la figura 6.4.

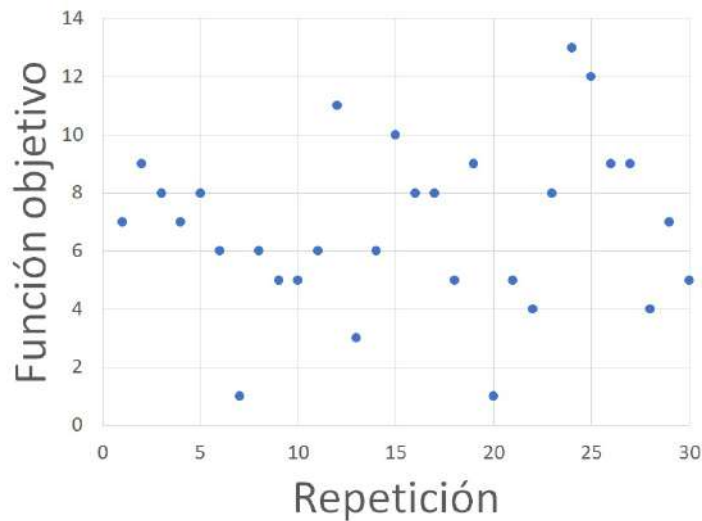


Figura 6.4: Función objetivo de las asignaciones aleatorias para EX1.

6.1.3. Inicialización por intervalos y celdas *HotSpot*

Independientemente del nivel de prioridad asignado en estas pruebas se obtuvo una convergencia en la primera iteración, es decir que con solo la asignación se obtenía una asignación que cumplía las restricciones dadas por la matriz (4.1). La figura 6.5 muestra una de las asignaciones realizadas por el enjambre de 60 partículas con prioridad 1, teniendo en cuenta el orden de (6.2).

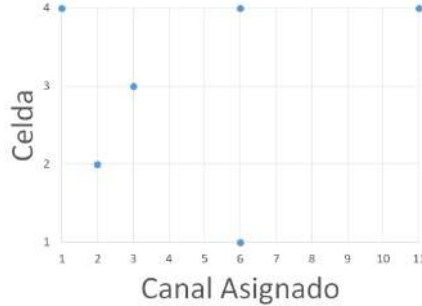


Figura 6.5: Malla de asignación para EX1, inicialización con intervalos y *Hotspot* con 60 partículas.

$$A_{EX1Int\&HS} = \begin{bmatrix} 6 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 0 \\ 1 & 6 & 11 \end{bmatrix} \quad (6.2)$$

Luego de verificar que el algoritmo estaba operando correctamente en lenguaje C, con ambos modos de inicialización se procedió a realizar evaluaciones del mismo con el problema PHI3, en un entorno de computación convencional. El tutorial de instalación de MinGW (compilador de GCC) y sus correspondiente pruebas están en el anexo 8.3. Luego se hizo el cambio del algoritmo con la librería MPI de C, se hicieron pruebas de ejecución en un cluster casero de dos computadores, migrando todo el software a Debian por facilidad, los tutoriales de instalación y configuración del cluster está en el anexo 8.4. Una vez se comprobó que las pruebas en MPI estaban operando correctamente se envió a producción al cluster **ZINE**.

En el *Cluster* de la Pontificia Universidad Javeriana **ZINE** del centro de alto rendimiento computacional, se realizaron pruebas con 240 procesadores, con los problemas PHI3 y PUJ3, pues son los problemas más complejos abordados por este trabajo de grado. Se hizo una comparación en las curvas de convergencia del algoritmo en ambos casos de inicialización, con las mismas características del protocolo de pruebas ya mencionadas.

Se agregó una comparación de los mejores resultados obtenidos por la función objetivo con 60 partículas con una malla que contrasta las celdas y los canales asignados y muestra las interferencias, en un entorno de computación convencional y en uno distribuido en paralelo.

6.2. PHI3: 21 celdas, 533 canales

La tabla 6.2 muestra las tasas de convergencia del algoritmo bajo los diferentes parámetros definidos para el protocolo de pruebas, con el problema PHI3 descrito en la sección 4.2.2. La tabla 6.3 muestra los tiempos de ejecución del algoritmo en un entorno convencional con 30 repeticiones y

en un entorno distribuido con 240 enjambres, los mismos resultados se encuentran comparados en la gráfica de la figura 6.6.

Entorno de prueba	Inicialización	Tamaño del enjambre	Prioridad	Tasas de convergencia
Convencional	Aleatorio	20	NA	0/30
Convencional	Aleatorio	40	NA	0/30
Convencional	Aleatorio	60	NA	0/30
Convencional	Intervalos & HS	60	1	0/30
Convencional	Intervalos & HS	60	4	0/30
Convencional	Intervalos & HS	60	8	0/30
Convencional	Generación aleatoria	NA	NA	0/30
Cluster	Aleatorio	20	NA	0/240
Cluster	Aleatorio	40	NA	0/240
Cluster	Aleatorio	60	NA	0/240
Cluster	Intervalos & HS	60	1	0/240
Cluster	Intervalos & HS	60	4	0/240
Cluster	Intervalos & HS	60	8	0/240

Cuadro 6.2: Tasas de convergencia del problema PHI3.

Características	Entorno convencional	Entorno distribuido
Aleatorio, 20 partículas	8m29.432s	0m27.523s
Aleatorio, 40 partículas	14m16.135s	1m33.772s
Aleatorio, 60 partículas	26m56.983s	1m19.175s
HotSpot, prioridad 1	24m16.254s	2m18.887s
HotSpot, prioridad 4	25m34.056s	2m16.678s
HotSpot, prioridad 8	21m14.730s	2m35.496s

Cuadro 6.3: Tiempos de ejecución de PHI3.

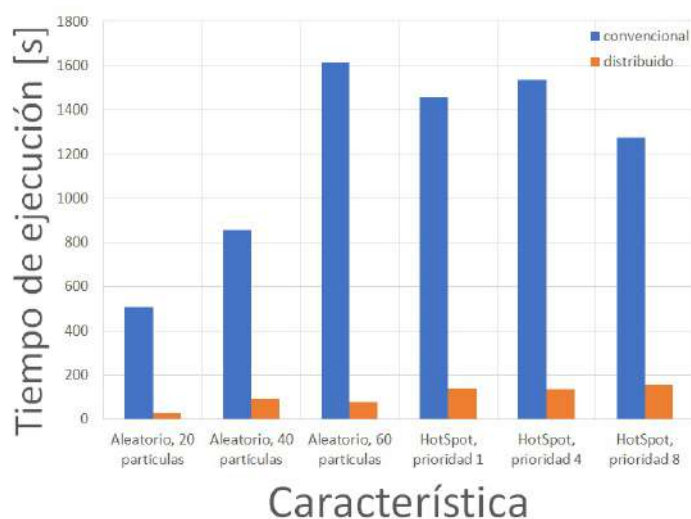


Figura 6.6: Comparación de los tiempos de ejecución para PHI3.

6.2.1. Inicialización aleatoria

En la figura 6.7a se muestran los resultados de la evaluación de la función objetivo para el problema PHI3, con inicialización aleatoria y un enjambre de 20 partículas, en una arquitectura convencional. y en la figura 6.7b se muestran los resultados de la convergencia en una arquitectura distribuida en paralelo.

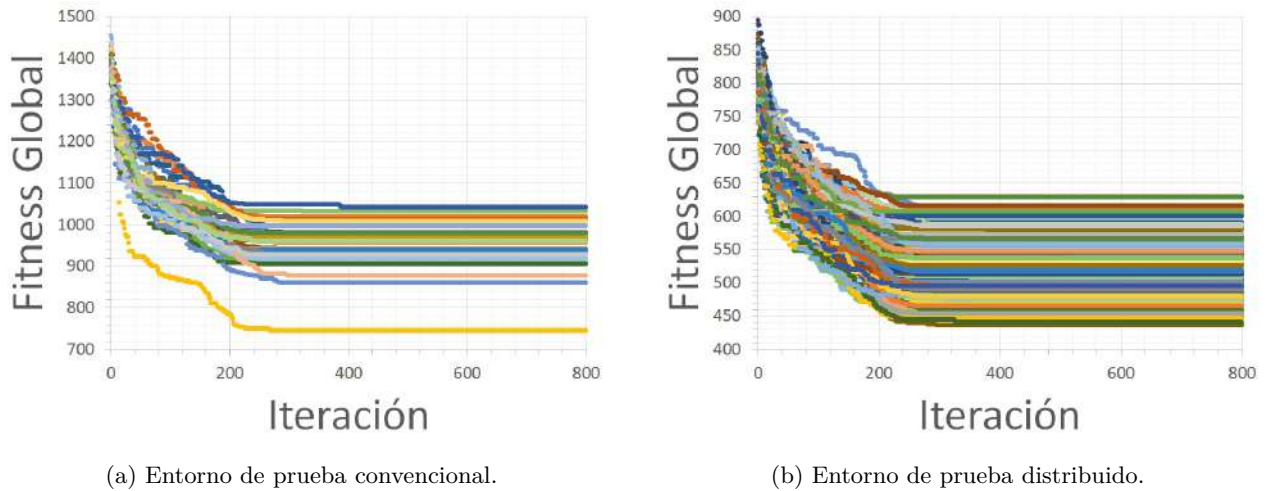


Figura 6.7: Curvas de convergencia para PHI3, inicialización aleatoria con 20 partículas.

Las figuras 6.8a y 6.8b, muestran las curvas de convergencia del problema PHI3 con inicialización aleatoria y un enjambre de 40 partículas, en un entorno de prueba convencional y uno distribuido en paralelo respectivamente.

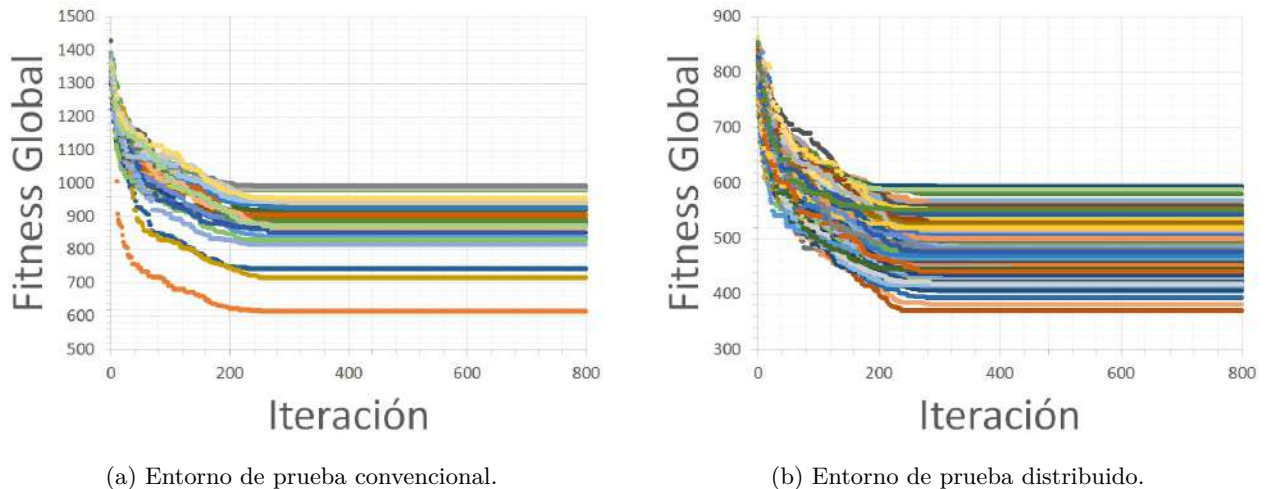
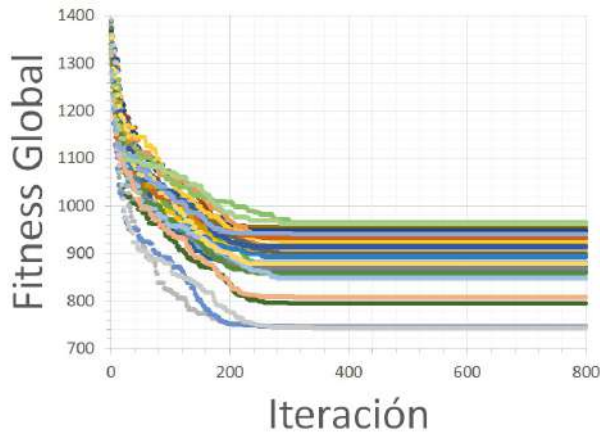
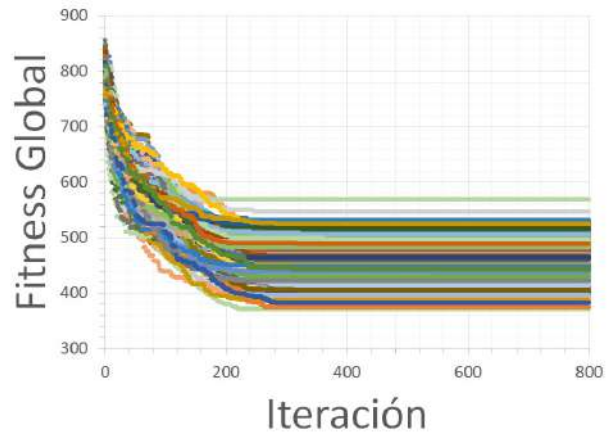


Figura 6.8: Curvas de convergencia para PHI3, inicialización aleatoria con 40 partículas.

Las pruebas de las figuras 6.9a y 6.9b, se realizaron para el problema PHI3 con inicialización aleatoria y una enjambre de 60 partículas, en un entorno de prueba convencional y uno distribuido en paralelo respectivamente.



(a) Entorno de prueba convencional.



(b) Entorno de prueba distribuido.

Figura 6.9: Curvas de convergencia para PHI3, inicialización aleatoria con 60 partículas.

La figura 6.10 muestra las curvas de los mejores resultados de convergencia en cada prueba realizada para PHI3 con inicialización aleatoria.

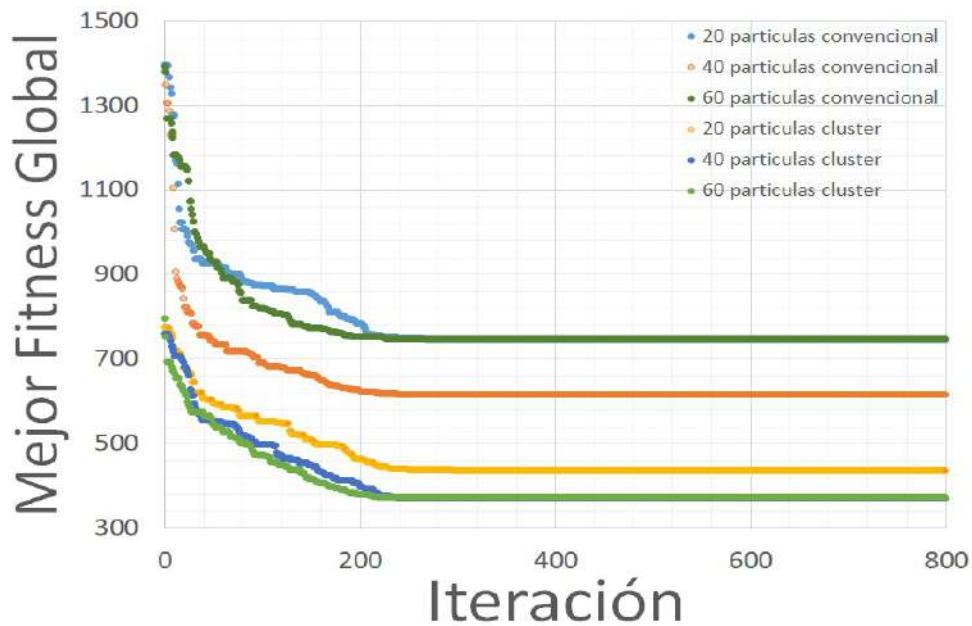
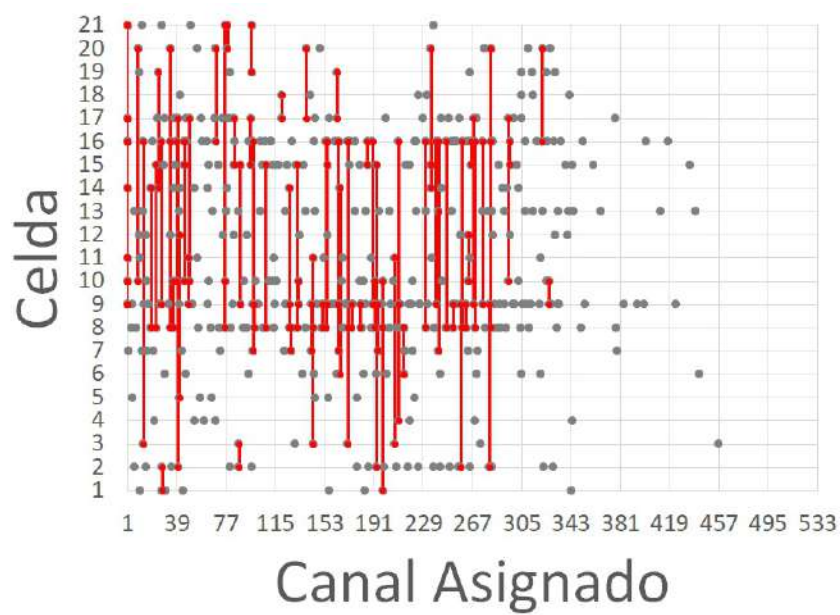


Figura 6.10: Curvas de las mejores convergencias para PHI3, inicialización aleatoria.

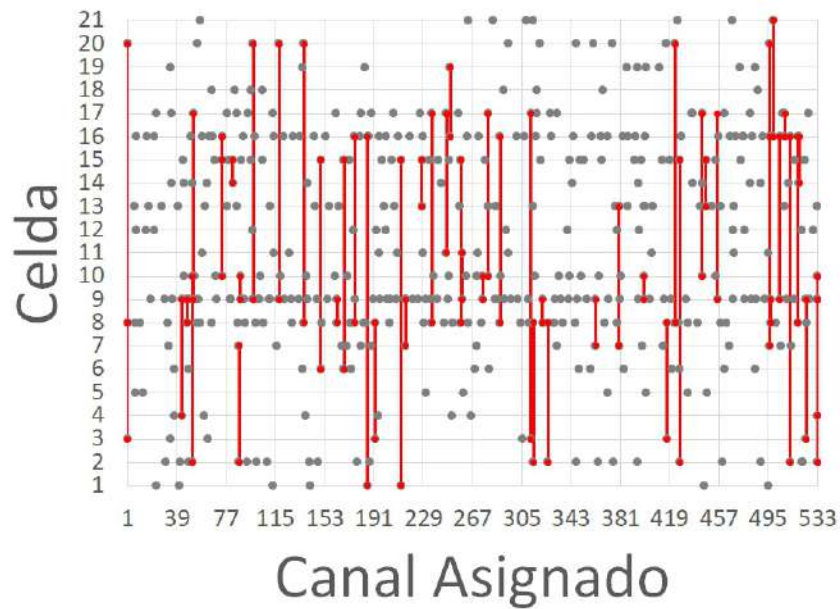
Como puede verse en la figura 6.10, los resultados obtenidos en ejecución en ZINE tiene una mejor respuesta a la convergencia que los resultados de un computador convencional, esto puede deberse a la cantidad de procesadores usados, pues permite tener mayor variedad en la ubicación inicial de las partículas sobre el espacio de búsqueda de manera aleatoria. Se puede notar que sobre la iteración 200 tiende a estabilizarse sobre un valor constante, esto debido a los parámetros cognitivos y sociales definidos por PSO lo que cambia el orden de cambio de posición y velocidad respectivamente.

Las figuras 6.11a y 6.11b son las mallas de asignación de canales y sus infracciones para las

partículas de mejor convergencia en los enjambres de 60 partículas, en cada entorno de prueba.



(a) Entorno de prueba convencional.



(b) Entorno de prueba distribuido.

Figura 6.11: Malla de asignación para PHI3, inicialización aleatoria con 60 partículas.

De acuerdo a las mallas de asignación de la figura 6.11, se ve que en un entorno de prueba convencional tiende a existir mayores infracciones de la matriz de compatibilidad (4.3) en celdas de alta demanda, y se encuentran concentradas en los primeros canales, mientras que en el cluster existe mayor distribución sobre los canales disponibles, y de igual forma las infracciones se esparcen sobre las asignaciones.

6.2.2. Generación aleatoria

Se realizaron 30 repeticiones de generaciones aleatorias de soluciones, en ningún caso estuvo cerca de converger, la figura 6.12 muestra los diferentes resultados obtenidos con la función *fitness*.

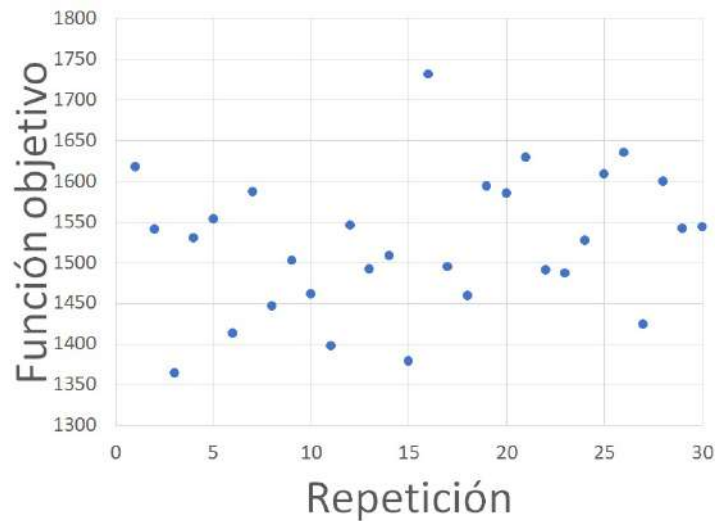


Figura 6.12: Función objetivo de las asignaciones aleatorias para PHI3.

6.2.3. Inicialización por intervalos y celdas *HotSpot*

En la figura 6.13a se muestran los resultados de la convergencia para el problema PHI3 con el método de inicialización por intervalos y celda *HotSpot* en un enjambre de 60 partículas y un nivel de prioridad igual a 1, en un entorno de prueba convencional y en la figura 6.13b están los resultados en un entorno distribuido bajo las mismas especificaciones de prueba.

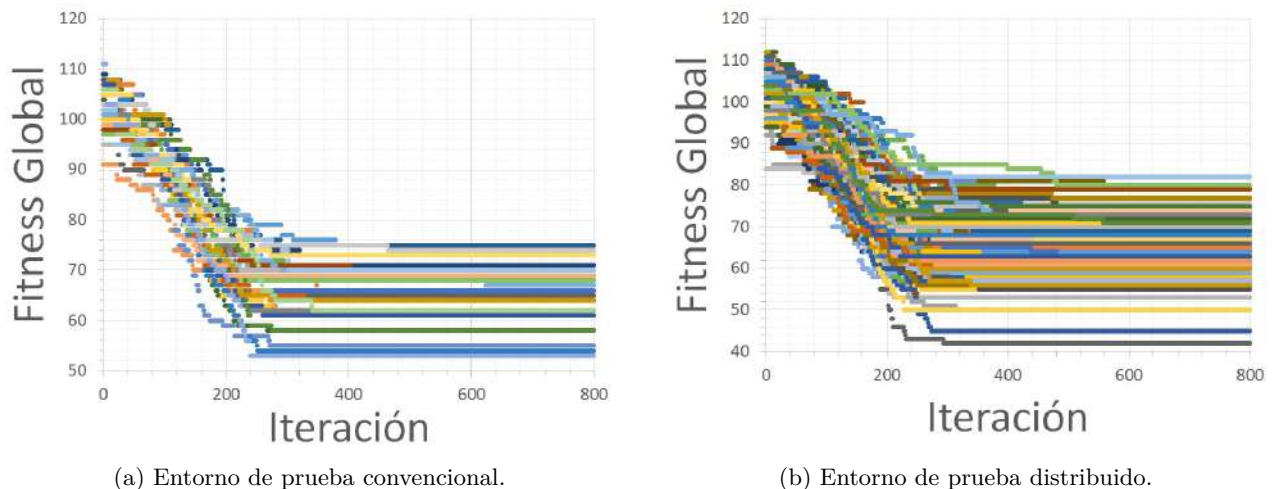
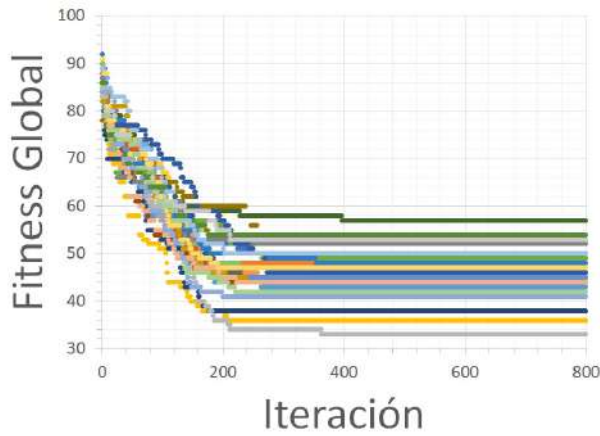
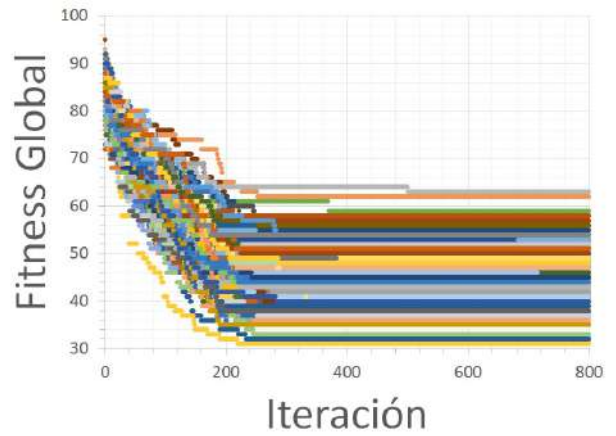


Figura 6.13: Curvas de convergencia para PHI3, inicialización con intervalos y Hotspot, prioridad 1.

Las figuras 6.14a y 6.14b, muestran los resultados de la evaluación de la función objetivo para el problema PHI3 con inicialización por intervalos y celdas *HotSpot* con nivel de prioridad 4, en un entorno de prueba convencional y distribuido respectivamente.



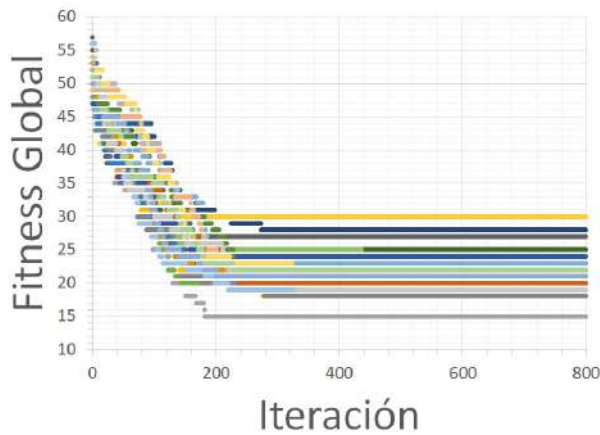
(a) Entorno de prueba convencional.



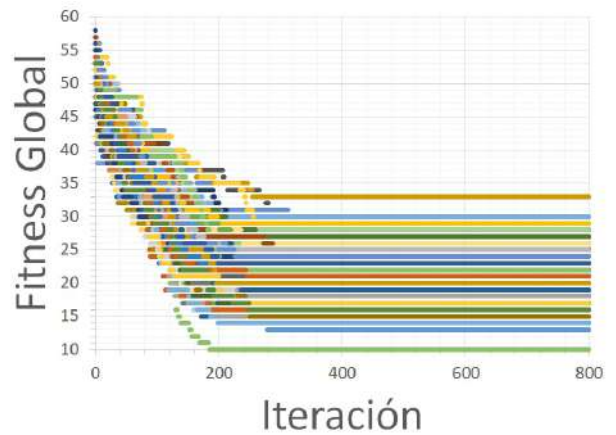
(b) Entorno de prueba distribuido.

Figura 6.14: Curvas de convergencia para PHI3, inicialización con intervalos y Hotspot, prioridad 4.

La figura 6.15a corresponde a las curvas de convergencia del problema PHI3 con inicialización por intervalos y celdas *HotSpot* con prioridad 8 en un entorno de prueba convencional, y la figura 6.15b muestra las curvas bajo el mismo esquema de pruebas en un entorno distribuido en paralelo.



(a) Entorno de prueba convencional.



(b) Entorno de prueba distribuido.

Figura 6.15: Curvas de convergencia para PHI3, inicialización con intervalos y Hotspot, prioridad 8.

La figura 6.16 muestra las curvas de convergencia de los mejores resultados en cada prueba variando el nivel de prioridad, según su entorno de prueba.

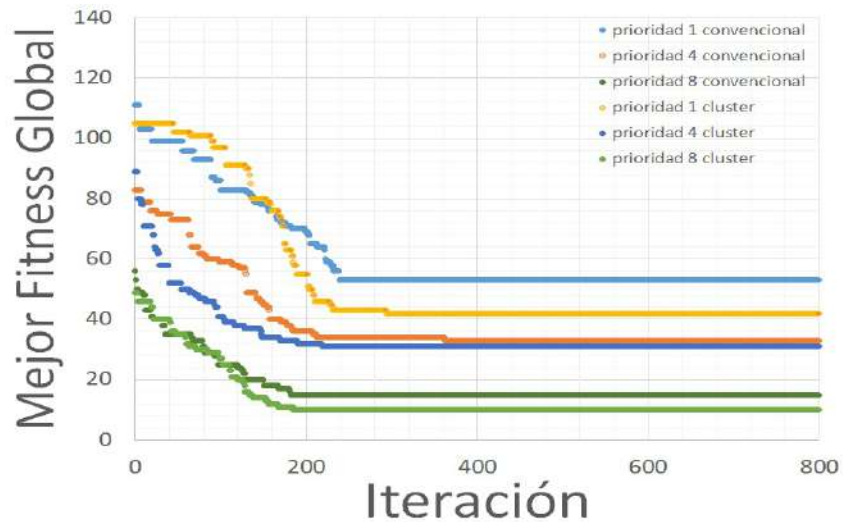
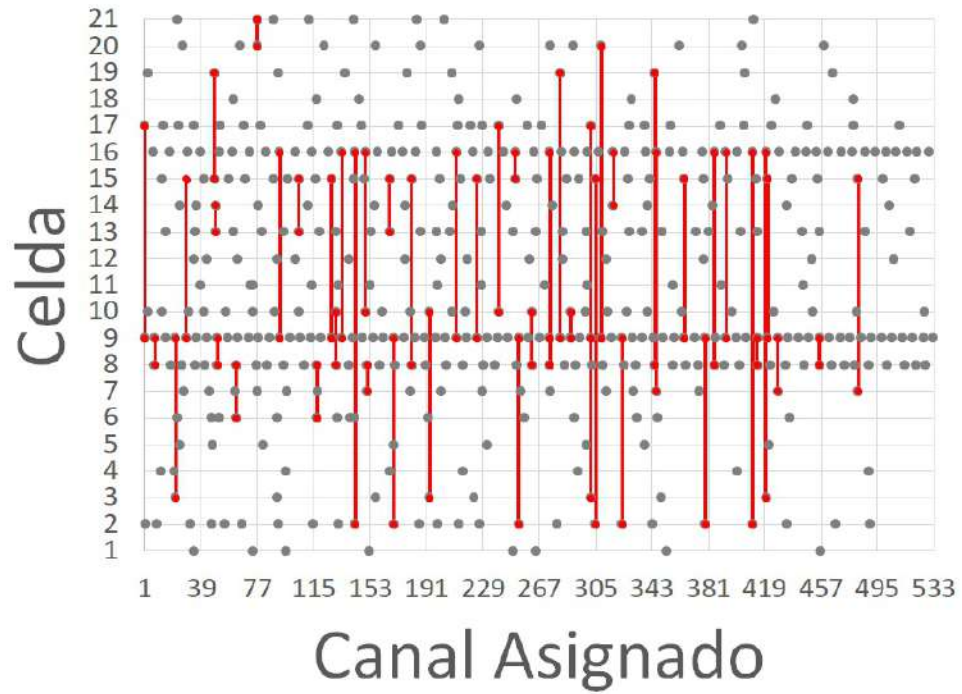


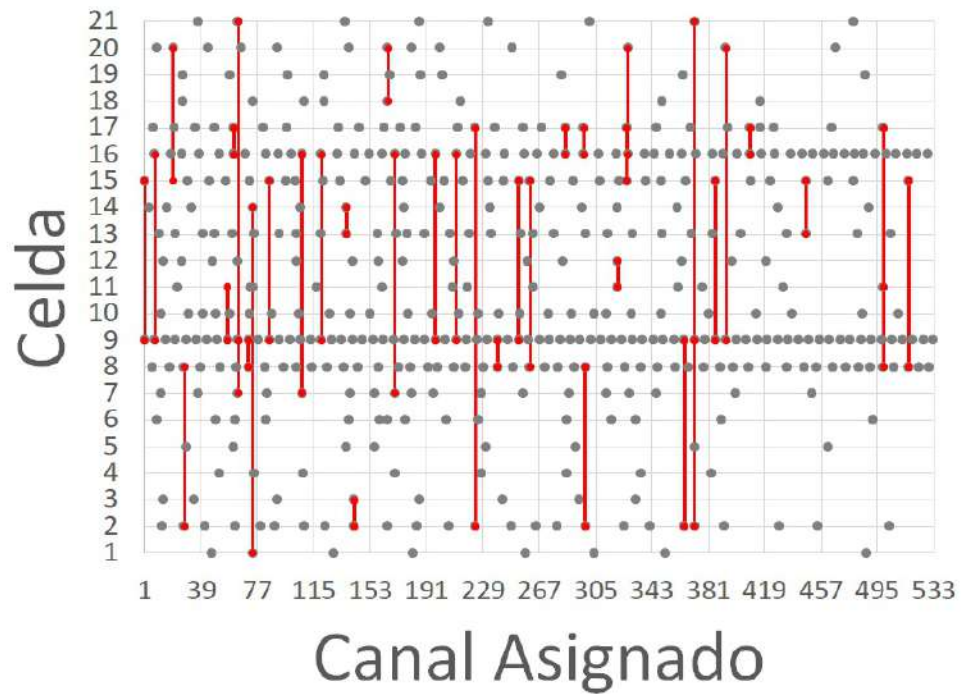
Figura 6.16: Curvas de las mejores convergencias para PHI3, inicialización con intervalos y HotSpot.

Como se ve en la figura 6.16, al aumentar la prioridad mejora la convergencia del algoritmo PSO con el problema PHI3, con los intervalos y la asignación «manual» de las celdas HotSpot prioritarias se ubican las partículas sobre un sub-óptimo del espacio de búsqueda, lo que mejora drásticamente los óptimos globales y locales, sus velocidades y posiciones. Adicionalmente, puede verse que las curvas de convergencia de los resultados del cluster siempre disminuyen sus valores de *Fitness* más rápidamente, nuevamente debido a la cantidad de procesadores que permite tener una asignación con mayor varianza.

La figura 6.17 define las mallas de asignación de canales y sus infracciones para las partículas de mejor convergencia con prioridad 1, en cada entorno de prueba.



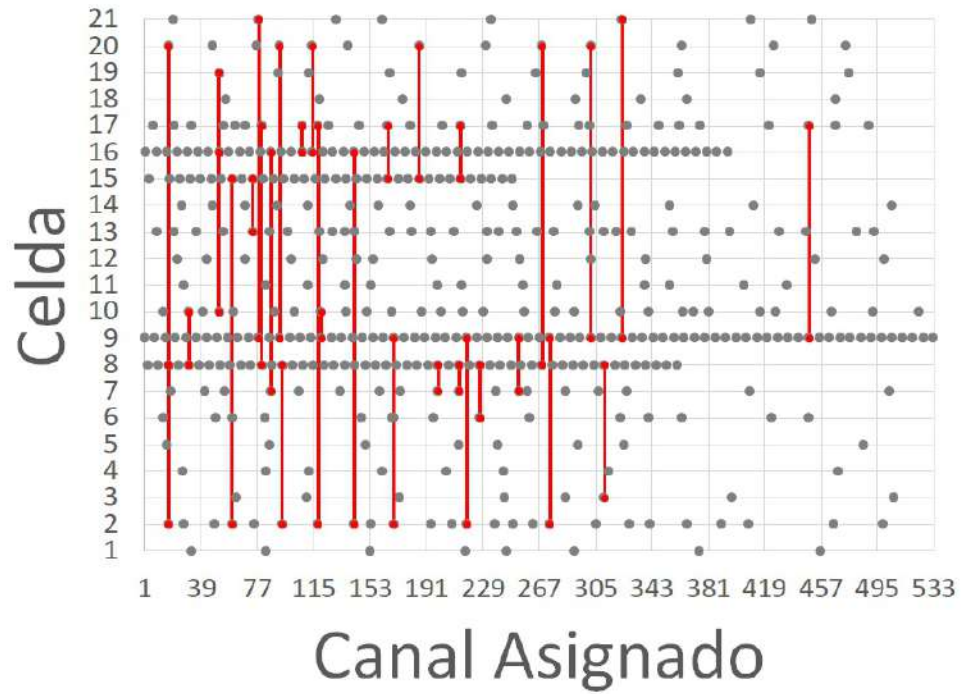
(a) Entorno de prueba convencional.



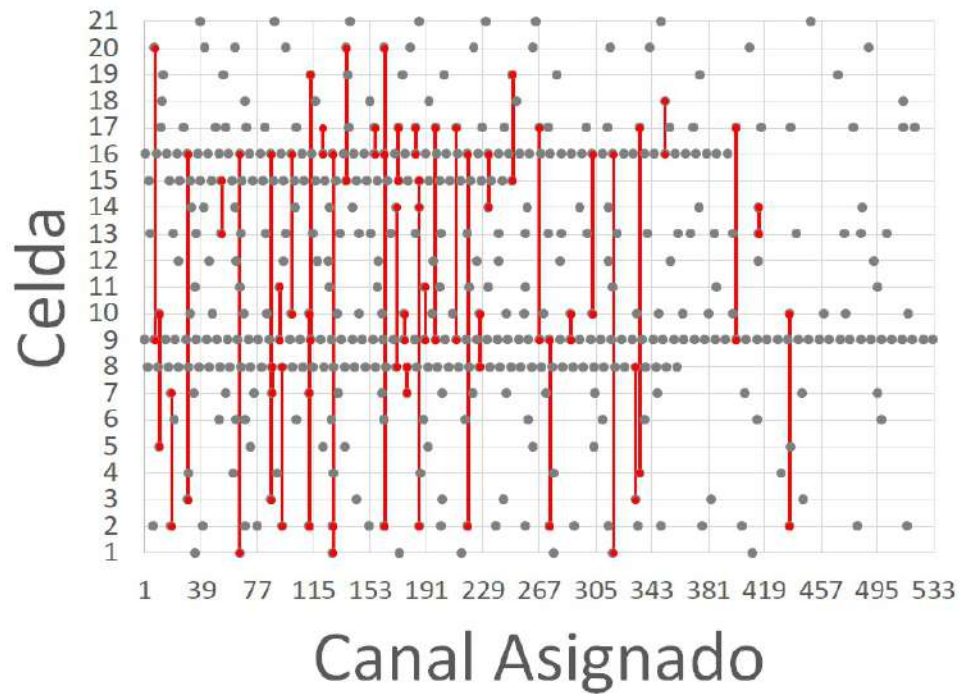
(b) Entorno de prueba distribuido.

Figura 6.17: Malla de asignación para PHI3, inicialización con intervalos y HotSpot con prioridad 1.

Las mallas de asignación de canales y sus infracciones de las partículas con mejor resultado en su evaluación de la función objetivo con prioridad 4, en un computador convencional y en una plataforma distribuida, se encuentran registrados en la figura 6.18.



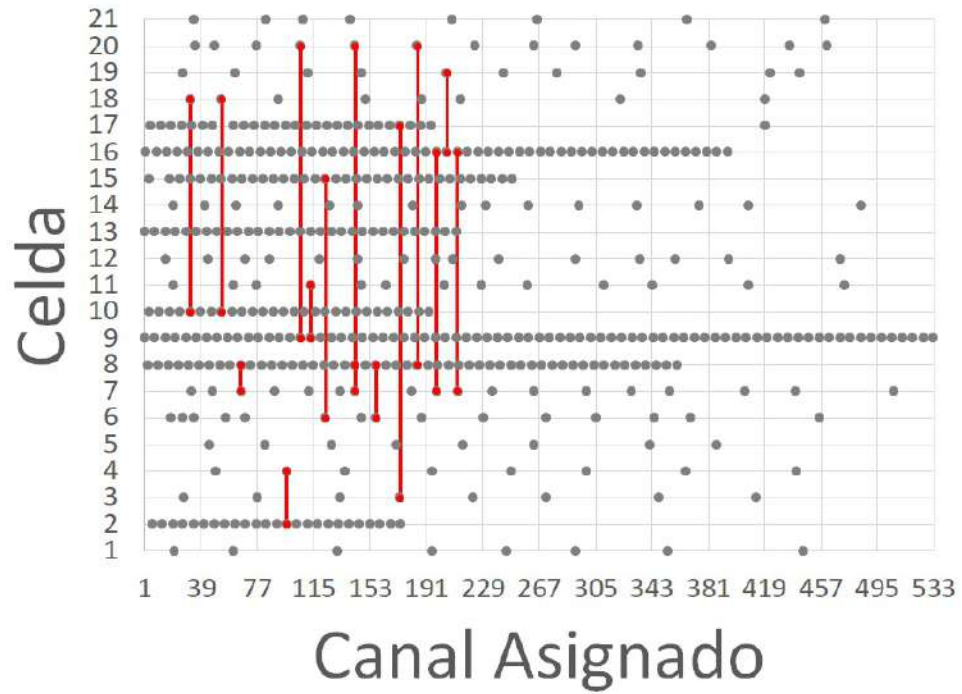
(a) Entorno de prueba convencional.



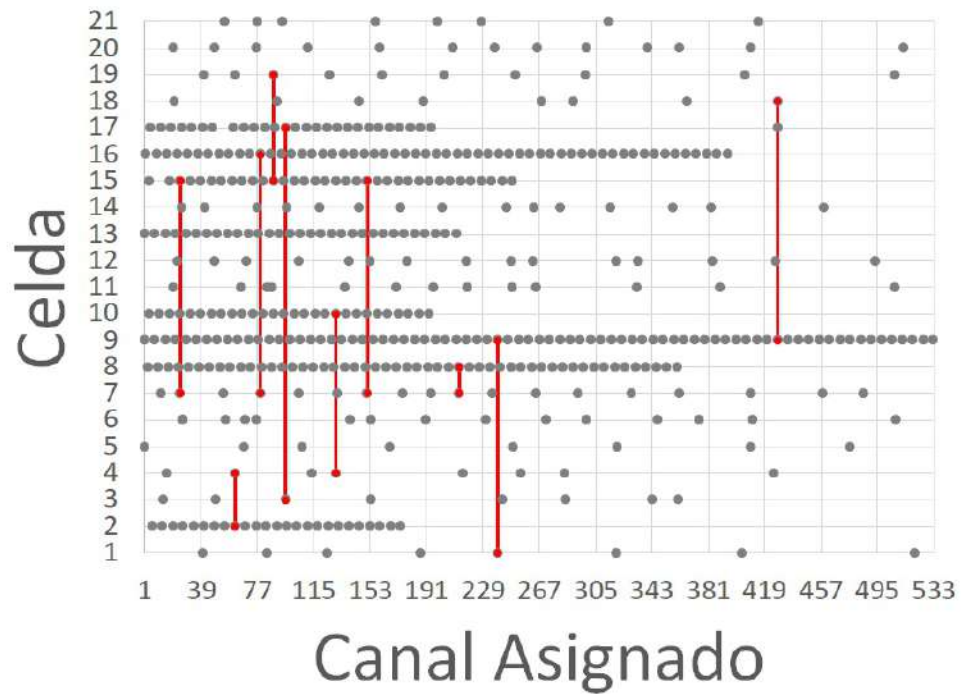
(b) Entorno de prueba distribuido.

Figura 6.18: Malla de asignación para PHI3, inicialización con intervalos y HotSpot con prioridad 4.

En la figura 6.19 se muestran las mallas de asignación de canales para las partículas con mejor convergencia ejecutadas con inicialización con intervalos y celdas *HotSpot* con prioridad 8, en ambos entorno de prueba.



(a) Entorno de prueba convencional.



(b) Entorno de prueba distribuido.

Figura 6.19: Malla de asignación para PHI3, inicialización con intervalos y HotSpot con prioridad 8.

En cada una de estas mallas de asignación puede verse el efecto de las celdas *HotSpot*, las cuales disminuyen significativamente las interferencias entre celdas, sin embargo puede notarse que algunas de las infracciones son fácilmente corregibles cambiando en uno o dos los canales asignados a las

celdas no prioritarias con conflicto, sin embargo el algoritmo no tiene la capacidad de realizar ese cambio de manera automática con los mecanismos de cambio de velocidad y posición de PSO, lo cual puede verse como algo a mejorar en el futuro.

6.3. PUI1: 11 celdas, 11 canales

La tabla 6.4 muestra las tasas de convergencia del algoritmo de acuerdo al protocolo de pruebas definidos al principio de este capítulo, con el problema PUI1 definido en la sección 4.3.2.

Entorno de prueba	Inicialización	Tamaño del enjambre	Prioridad	Tasas de convergencia
Convencional	Aleatorio	20	NA	30/30
Convencional	Aleatorio	40	NA	30/30
Convencional	Aleatorio	60	NA	30/30
Convencional	Generación aleatoria	NA	NA	16/30

Cuadro 6.4: Tasas de convergencia del problema PUI1.

6.3.1. Inicialización aleatoria

En las pruebas del algoritmo con el problema PUI1 con inicialización aleatoria, se logró convergencia en las 30 repeticiones con los diferentes tamaños de enjambre, en la primera iteración, es decir con la generación aleatoria se obtuvieron resultados que satisfacían las condiciones de la matriz (4.5). En las figuras 6.20, 6.21 y 6.22, se muestran las matrices de asignación del problema PUI1 con enjambres de 20, 40 y 60 partículas respectivamente.

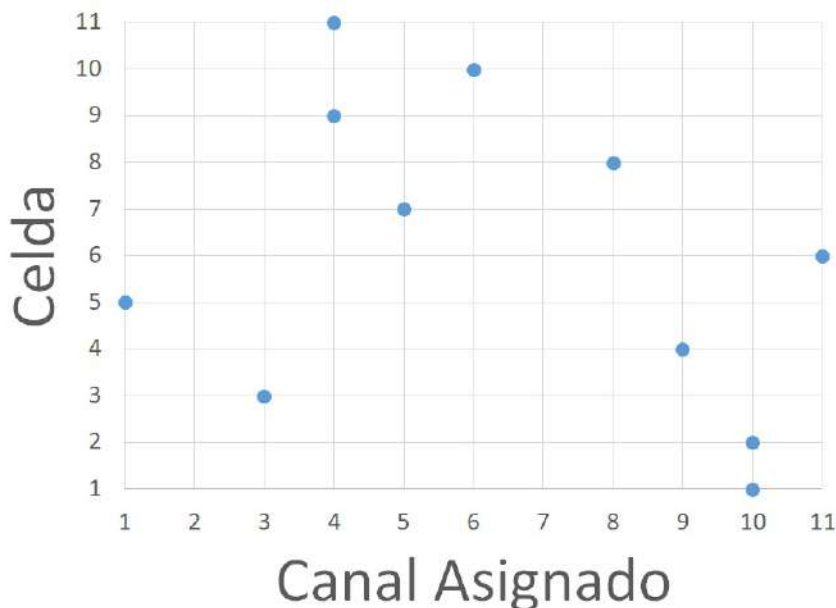


Figura 6.20: Malla de asignación para PUI1, inicialización aleatoria con 20 partículas, entorno de prueba convencional.

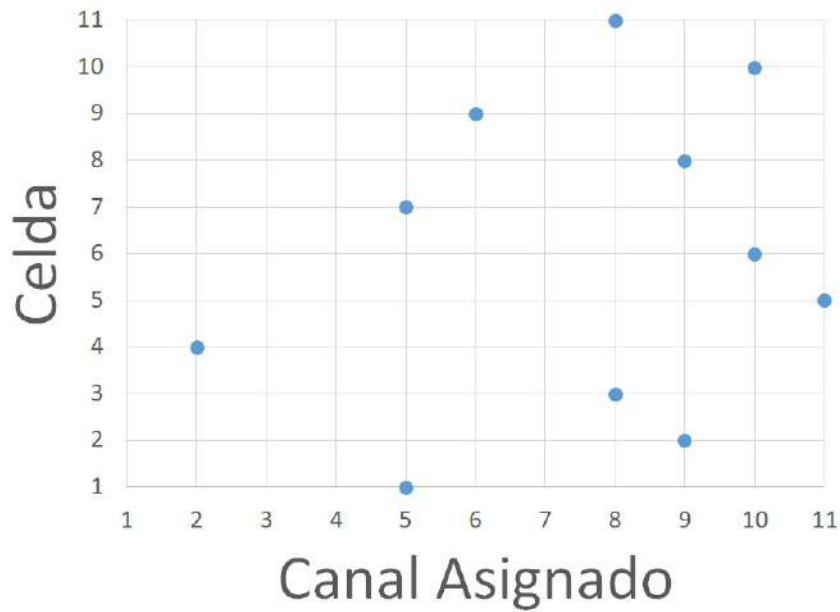


Figura 6.21: Malla de asignación para PUJ1, inicialización aleatoria con 40 partículas, entorno de prueba convencional.

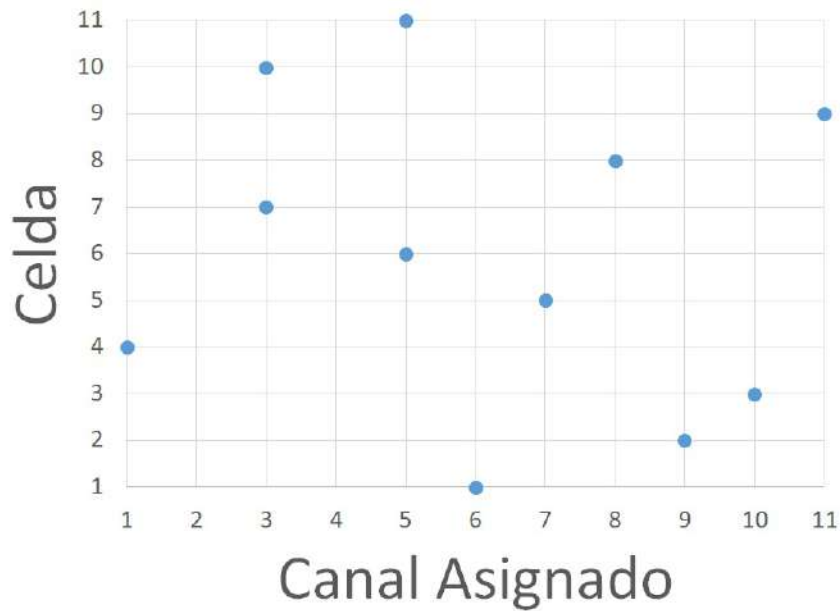


Figura 6.22: Malla de asignación para PUJ1, inicialización aleatoria con 60 partículas, entorno de prueba convencional.

Este problema converge rápidamente, independientemente del tamaño del enjambre, debido a las condiciones poco restrictivas existentes en la matriz definida en (4.5).

6.3.2. Generación aleatoria

Se realizaron 30 generaciones aleatorias de asignaciones en una arquitectura convencional, donde 16 de estas fueron exitosas, la figura 6.23 muestra los resultados obtenidos en cada repetición.

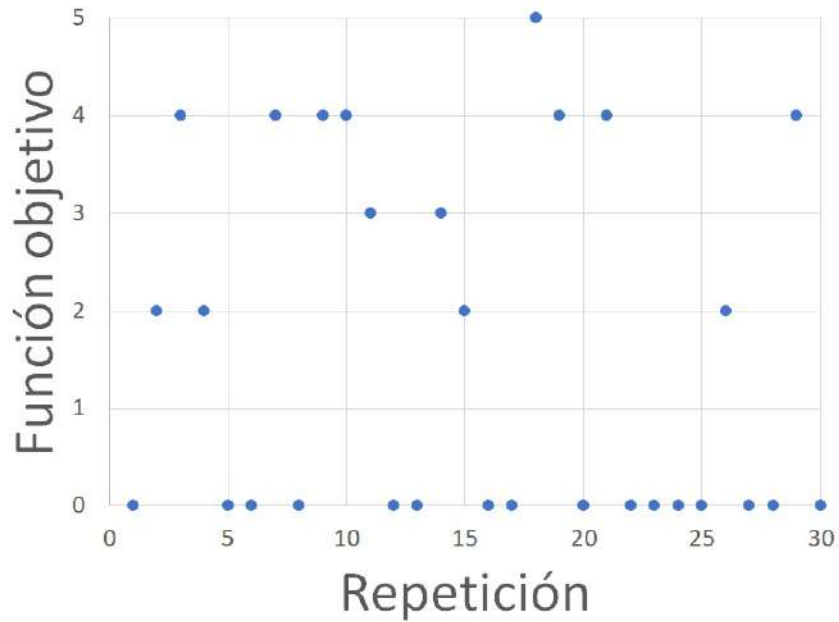


Figura 6.23: Función objetivo de las asignaciones aleatorias para PUJ1.

6.4. PUJ2: 11 celdas, 11 canales

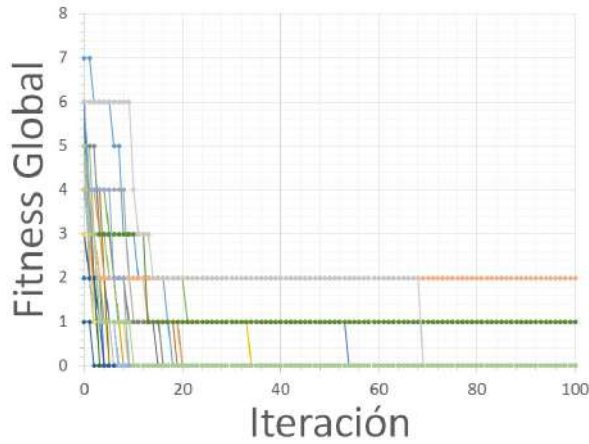
La tabla 6.5 muestra las tasas de convergencia del algoritmo, con el problema PUJ2 descrito en la sección 4.3.2.

Entorno de prueba	Inicialización	Tamaño del enjambre	Prioridad	Tasas de convergencia
Convencional	Aleatorio	20	NA	26/30
Convencional	Aleatorio	40	NA	28/30
Convencional	Aleatorio	60	NA	30/30
Convencional	Generación aleatoria	NA	NA	0/30

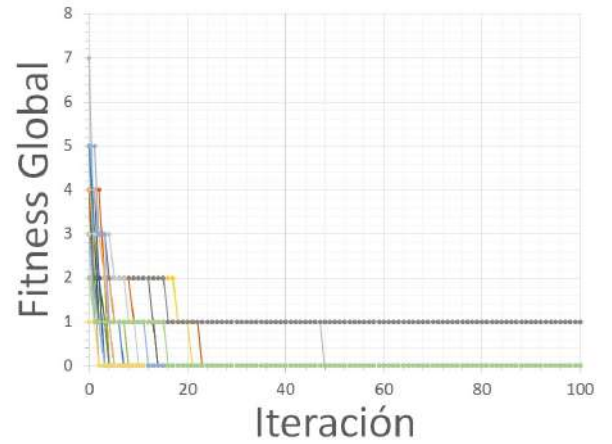
Cuadro 6.5: Tasas de convergencia del problema PUJ2.

6.4.1. Inicialización aleatoria

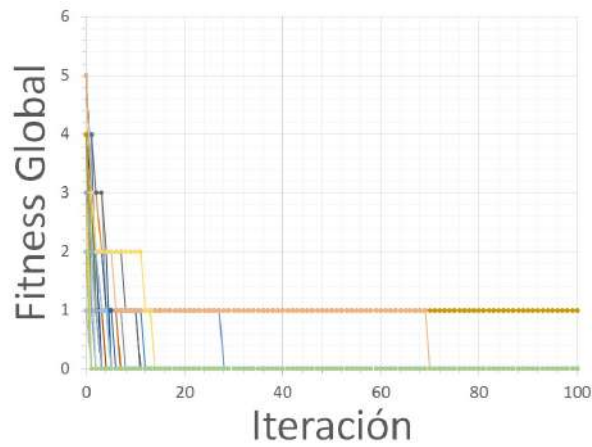
En las figuras 6.24a, 6.24b y 6.24c, se encuentran las curvas de convergencia del problema PUJ2, con inicialización aleatoria en un enjambre de 20, 40 y 60 partículas, en un entorno de pruebas convencional.



(a) 20 partículas.



(b) 40 partículas.



(c) 60 partículas.

Figura 6.24: Curvas de convergencia para PUF2, inicialización aleatoria, entorno de prueba convencional.

Se acotó el número de iteraciones a 100 para ayudar a la visualización de los resultados, pero el límite de iteraciones se mantuvo igual a las demás pruebas en 800. No se mostrarán las curvas de mejor convergencia pues existen resultados válidos con la generación aleatoria dando curvas de convergencia sobre cero.

Las figuras 6.25, 6.26 y 6.27, son las mallas de asignación de canales de los mejores resultados en cuanto a convergencia en cada caso, con inicialización aleatoria y 20, 40 o 60 partículas en el enjambre.

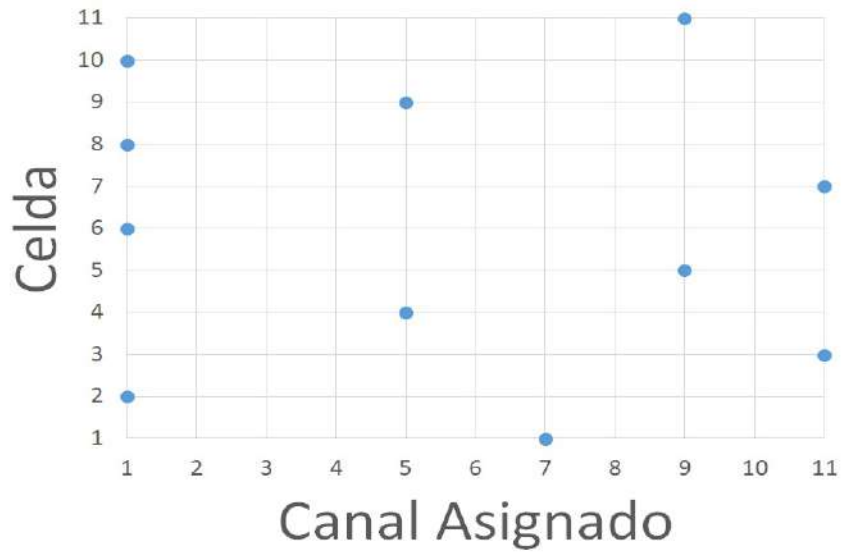


Figura 6.25: Malla de asignación para PUF2, inicialización aleatoria con 20 partículas, entorno de prueba convencional.

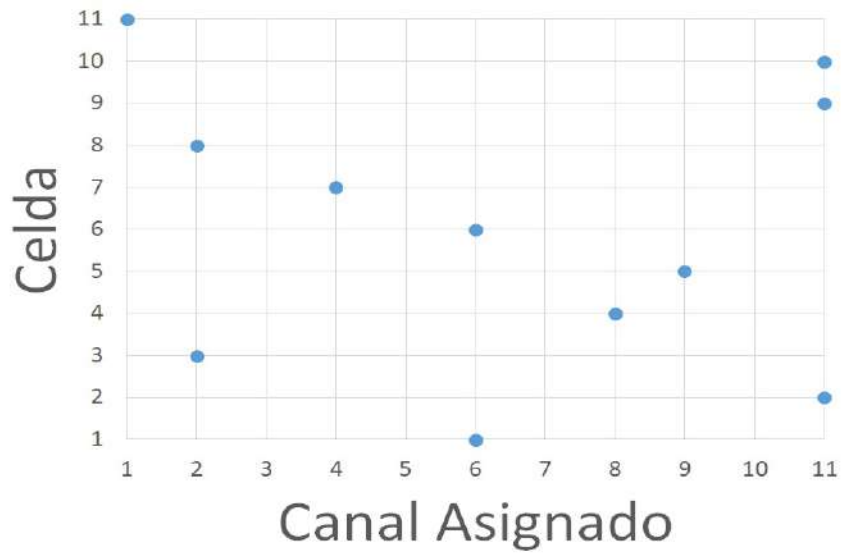


Figura 6.26: Malla de asignación para PUF2, inicialización aleatoria con 40 partículas, entorno de prueba convencional.

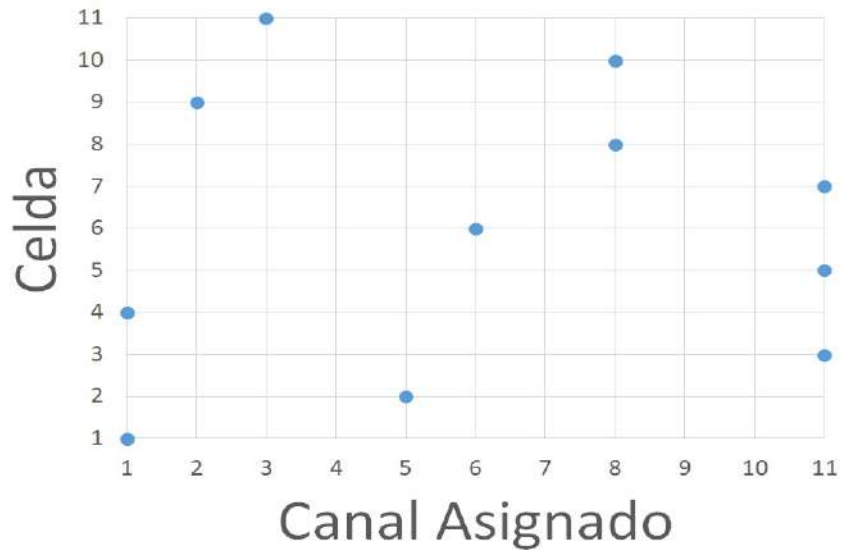


Figura 6.27: Malla de asignación para PUF2, inicialización aleatoria con 60 partículas, entorno de prueba convencional.

El problema PUF2 puede llegar a no converger al no tener una buena ubicación inicial de las partículas, por eso la tasa de convergencia disminuye en comparación del problema PUF1, esto debido a que la matriz (4.6) tiene condiciones más restrictivas para la asignación de canales.

6.4.2. Generación aleatoria

Con 30 asignaciones aleatorias en una arquitectura convencional no se presentó ningún caso con convergencia, los resultados obtenidos por repetición se encuentran en la figura 6.28.

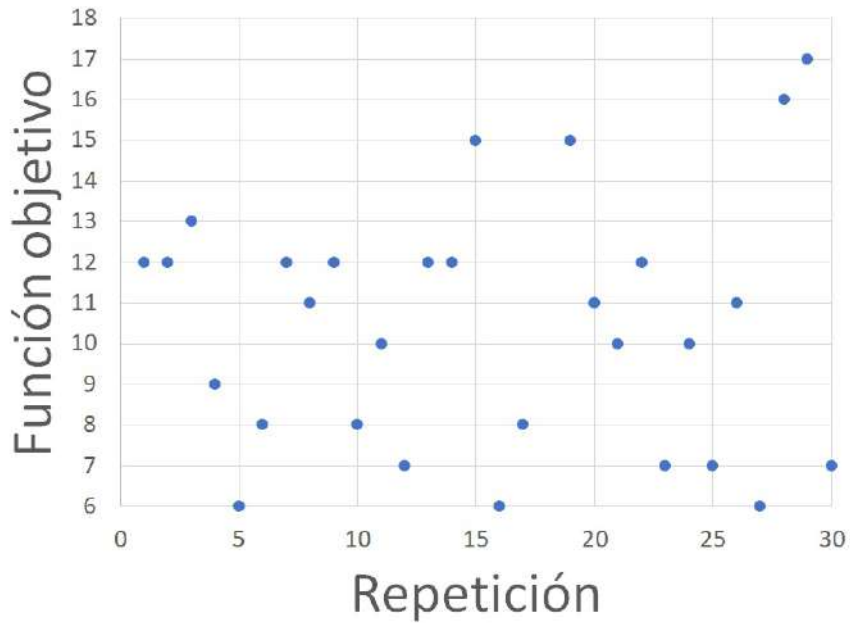


Figura 6.28: Función objetivo de las asignaciones aleatorias para PUF1.

6.5. PUJ3: 11 celdas, 11 canales

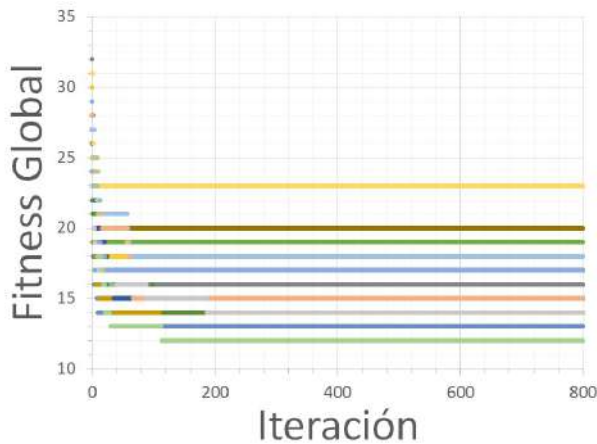
La tabla 6.6, muestra las tasas de convergencia del algoritmo bajo varios criterios de prueba, para el problema PUJ3 descrito en la sección 4.3.2.

Entorno de prueba	Inicialización	Tamaño del enjambre	Prioridad	Tasas de convergencia
Convencional	Aleatorio	20	NA	0/30
Convencional	Aleatorio	40	NA	0/30
Convencional	Aleatorio	60	NA	0/30
Convencional	Intervalos & HS	60	2	0/30
Convencional	Intervalos & HS	60	3	0/30
Convencional	Intervalos & HS	60	4	0/30
Convencional	Generación aleatoria	NA	NA	0/30
Cluster	Aleatorio	20	NA	0/240
Cluster	Aleatorio	40	NA	0/240
Cluster	Aleatorio	60	NA	0/240

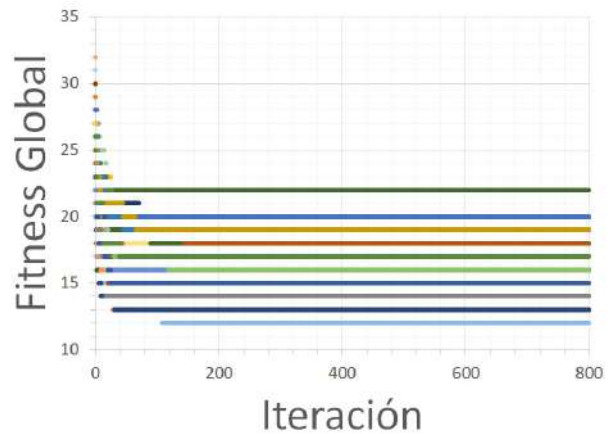
Cuadro 6.6: Tasas de convergencia del problema PUJ3.

6.5.1. Inicialización aleatoria

En las figuras 6.29a y 6.29b, se muestran las curvas de convergencia del problema PUJ3 con inicialización aleatoria, en un enjambre de 20 partículas, en un entorno de pruebas convencional y uno distribuido correspondientemente.



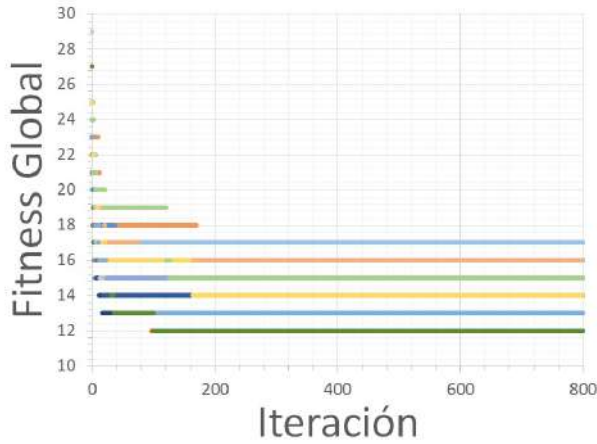
(a) Entorno de prueba convencional.



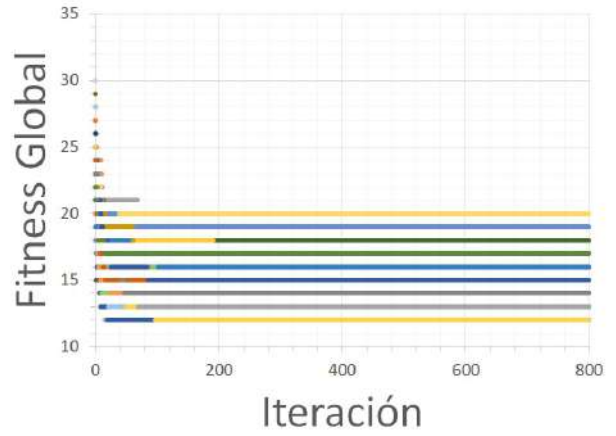
(b) Entorno de prueba distribuido.

Figura 6.29: Curvas de convergencia para PUJ3, inicialización aleatoria con 20 partículas.

Las curvas de convergencia del problema PUJ3, con inicialización aleatoria en un enjambre de 40 partículas en un entorno de pruebas convencional se registra en la figura 6.30a, y las curvas en un entorno de pruebas distribuido de mismas características en la figura 6.30b.



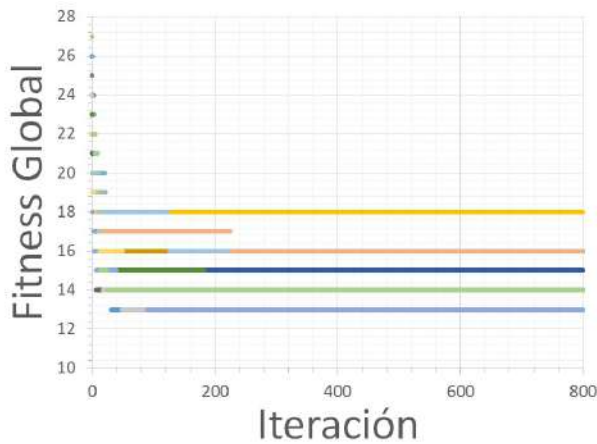
(a) Entorno de prueba convencional.



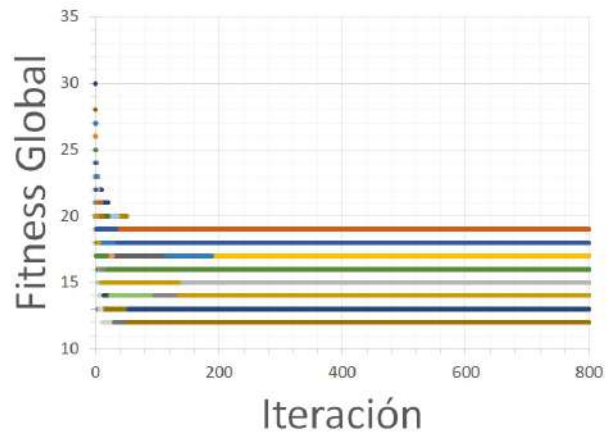
(b) Entorno de prueba distribuido.

Figura 6.30: Curvas de convergencia para PUJ3, inicialización aleatoria con 40 partículas.

La figura 6.31a, muestra las curvas de convergencia de PUJ3 con inicialización aleatoria en un enjambre de 60 partículas en un entorno de prueba convencional, mientras que la figura 6.31b, muestra las curvas bajo el mismo protocolo de pruebas en un entorno distribuido.



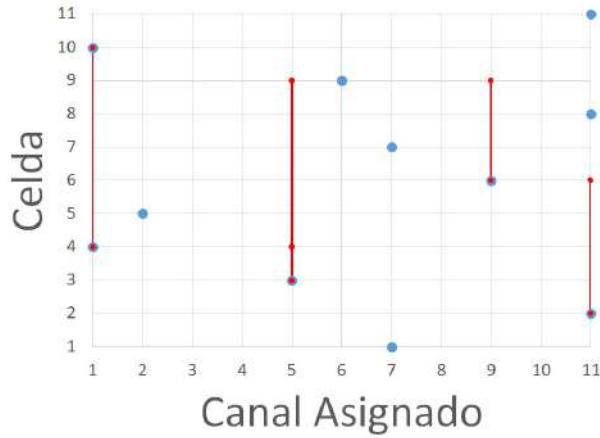
(a) Entorno de prueba convencional.



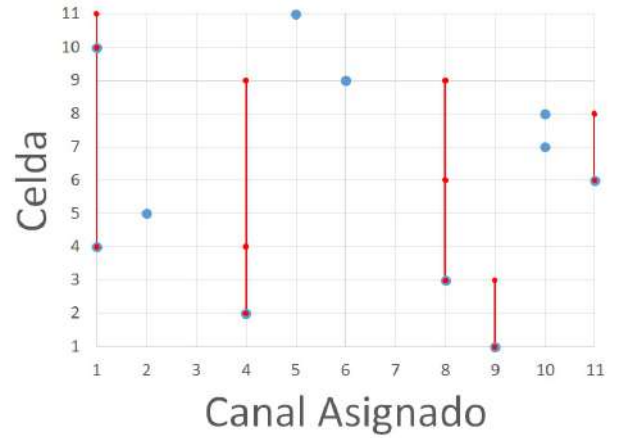
(b) Entorno de prueba distribuido.

Figura 6.31: Curvas de convergencia para PUJ3, inicialización aleatoria con 60 partículas.

La figura 6.32, muestra las mallas de asignación de canales del problema PUJ3 con inicialización aleatoria en un enjambre de 20 partículas, en un entorno de pruebas convencional y en uno distribuido.



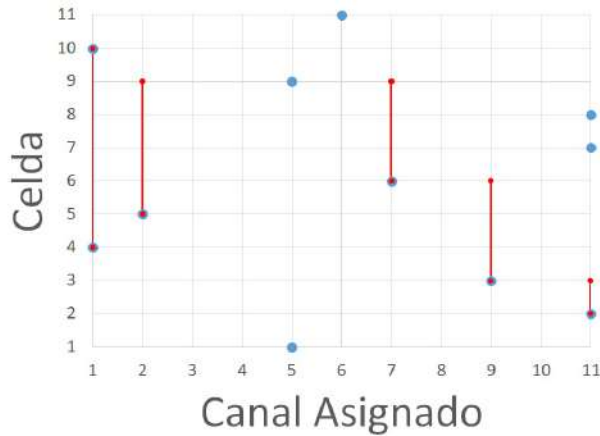
(a) Entorno de prueba convencional.



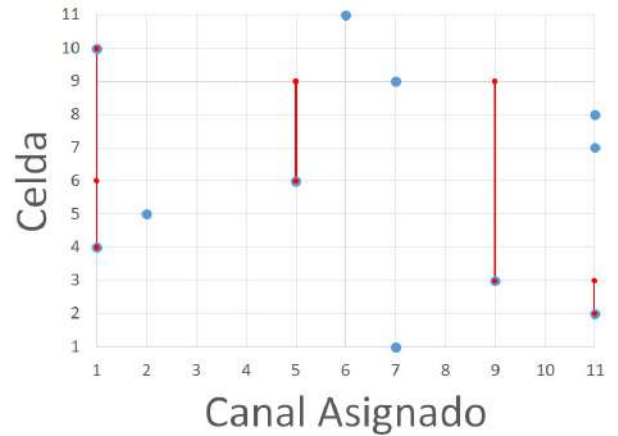
(b) Entorno de prueba distribuido.

Figura 6.32: Malla de asignación para PUJ3, inicialización aleatoria con 20 partículas.

Las mallas de asignación de canales de la figura 6.33, corresponden a las mejores soluciones encontradas para PUJ3 con inicialización aleatoria en un enjambre de 40 partículas, en un computador convencional y en el cluster ZINE.



(a) Entorno de prueba convencional.



(b) Entorno de prueba distribuido.

Figura 6.33: Malla de asignación para PUJ3, inicialización aleatoria con 40 partículas.

La figura 6.34, muestra las mallas de asignación definidas para el problema PUJ3 con inicialización aleatoria en un enjambre de 60 partículas, en ambos entornos de pruebas.

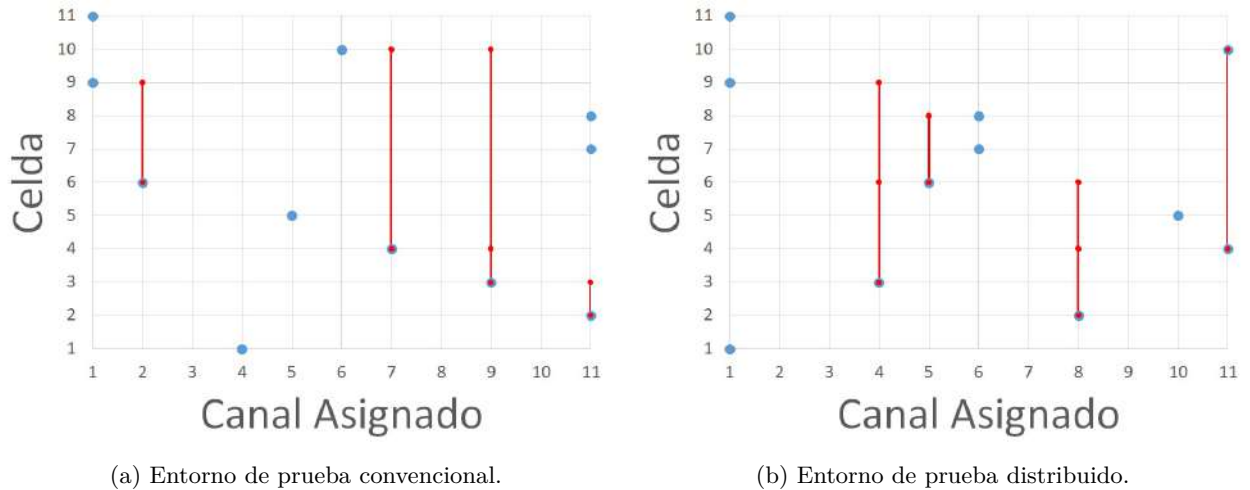


Figura 6.34: Malla de asignación para PUJ3, inicialización aleatoria con 60 partículas.

El problema PUJ3 tiene una tasa de convergencia nula con inicialización aleatoria, esto debido a la matriz de compatibilidad electromagnética diseñada (4.7), las condiciones son muy restrictivas y con una asignación aleatoria no se logra llegar a un resultado que cumpla con las características solicitadas. En la siguiente sección se formula un método que intentó ayudar a la convergencia de problemas con demanda unitaria para cada celda como son los problemas de asignación Wi-Fi de la Facultad de Ingeniería Electrónica de la Pontificia Universidad Javeriana.

6.5.2. Generación aleatoria

Realizando 30 asignaciones aleatorias en una arquitectura computacional convencional, no se obtuvo convergencia en ningún caso, los resultados por repetición se encuentran en la figura 6.35.

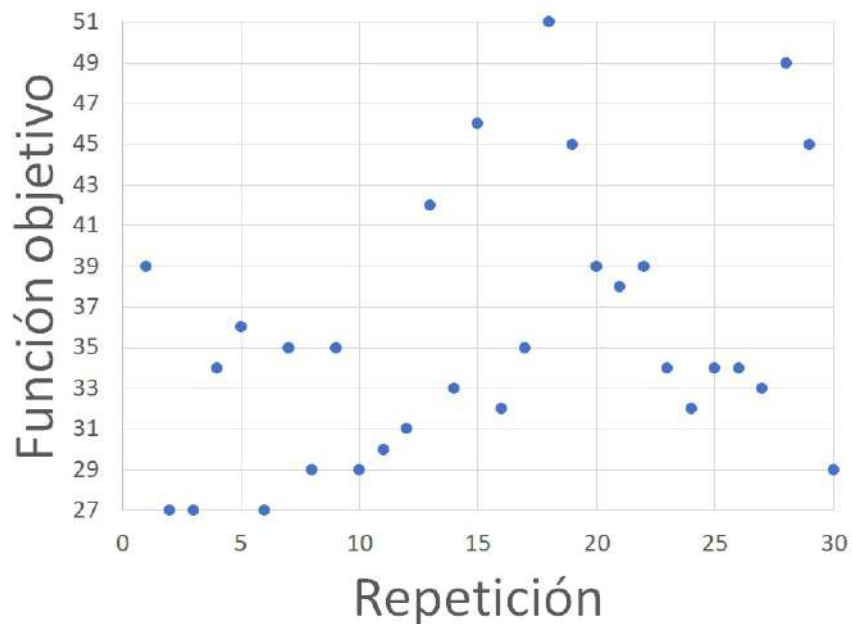


Figura 6.35: Función objetivo de las asignaciones aleatorias para PUJ1.

6.5.3. Inicialización por intervalos y celdas *HotSpot*

Para esta evaluación se realizó un cambio a la inicialización por intervalos y celdas *HotSpot* que se describió en la sección 5.2, como estos problemas son para asignaciones Wi-Fi donde solo se requiere un canal por celda, es necesario cambiar la forma de ubicación de las celdas *HotSpot*, pues la multiplicación de las restricciones co-Sitio de la matriz de compatibilidad electromagnética por la demanda de cada celda tiene el mismo valor. Por lo tanto, ahora las celdas *HotSpot* son las celdas que tienen mayor número de restricciones en la matriz C, definiendo entonces el vector *HotSpot* según la figura 6.36.

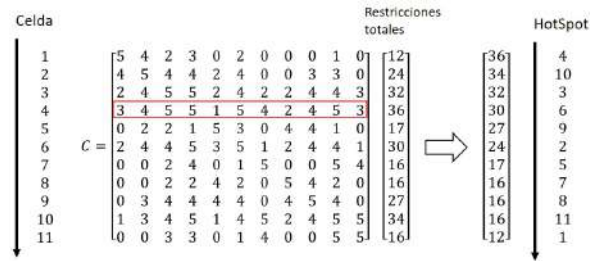


Figura 6.36: Cambio de método para la ubicación de la celda *HotSpot*.

La figura 6.37, muestra las curvas de convergencia del algoritmo con el problema PUJ3 usando la técnica modificada de intervalos y celdas *HotSpot*, variando el nivel de prioridad.

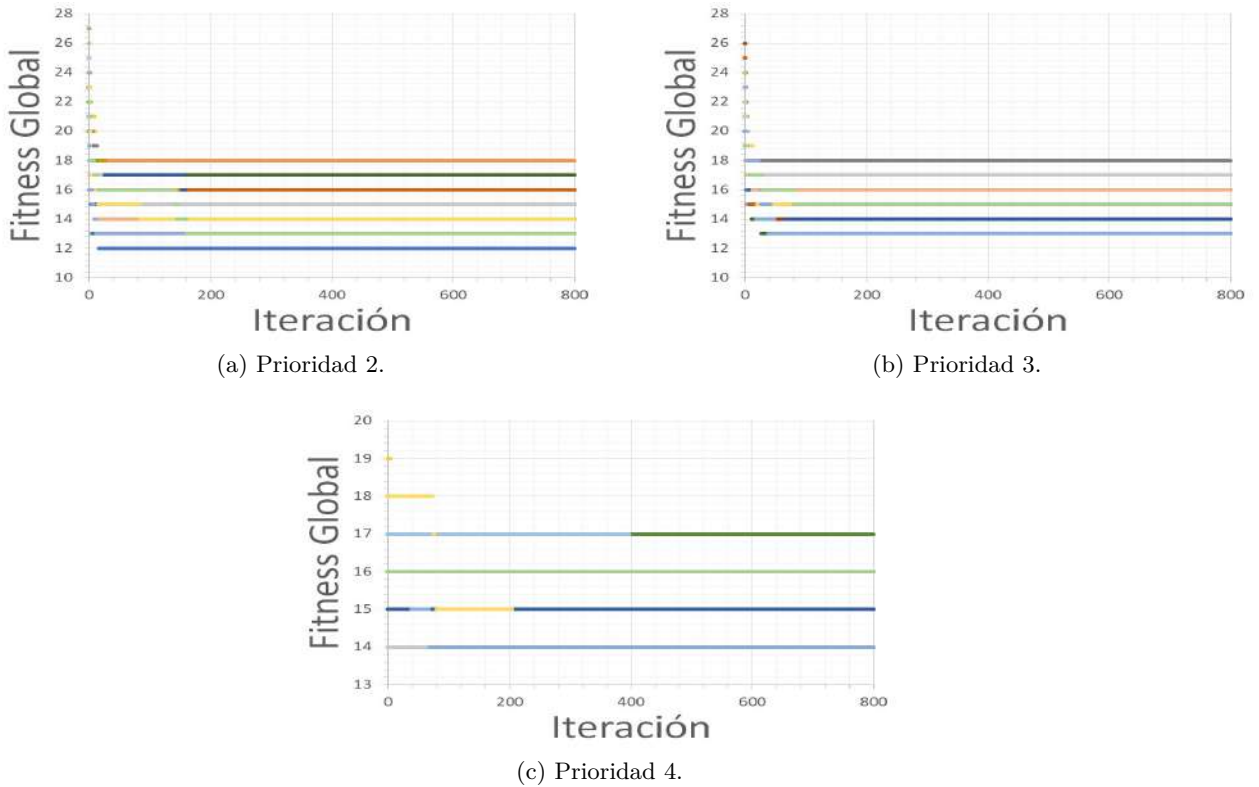


Figura 6.37: Curvas de convergencia para PUJ3, inicialización con intervalos y celdas *HotSpot*, entorno de prueba convencional.

La figura 6.38, son las curvas de convergencia de los 3 mejores resultados obtenidos por nivel de prioridad para el problema PUJ3.

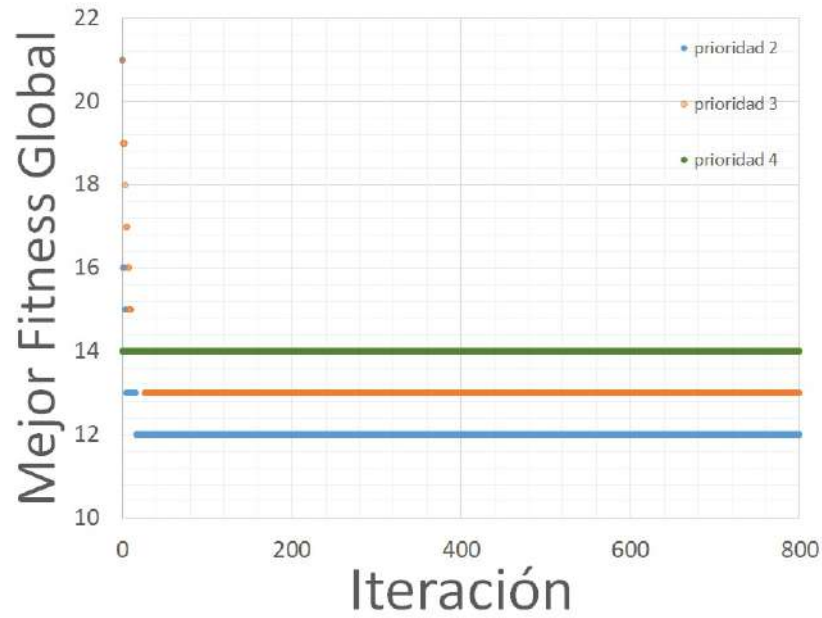


Figura 6.38: Curvas de las mejores convergencias para PUJ3, inicialización con intervalos y HotSpot.

Las figuras 6.39, 6.40 y 6.41, corresponden a las mallas de asignación de canales de los 3 mejores resultados obtenidos por nivel de prioridad con inicialización por intervalos y celdas *HotSpot* para el problema PUJ3.

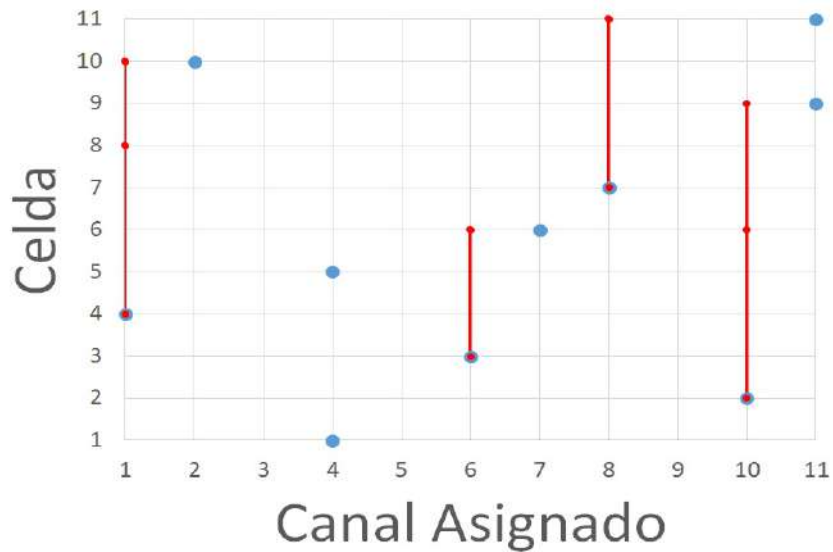


Figura 6.39: Malla de asignación para PUJ3, inicialización con intervalos y celdas *HotSpot*, prioridad 2.

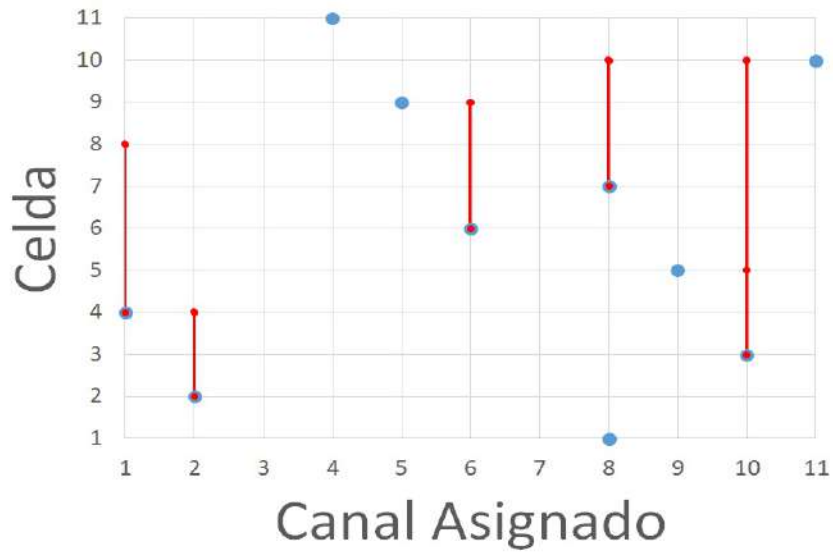


Figura 6.40: Malla de asignación para PUI3, inicialización con intervalos y celdas *HotSpot*, prioridad 3.

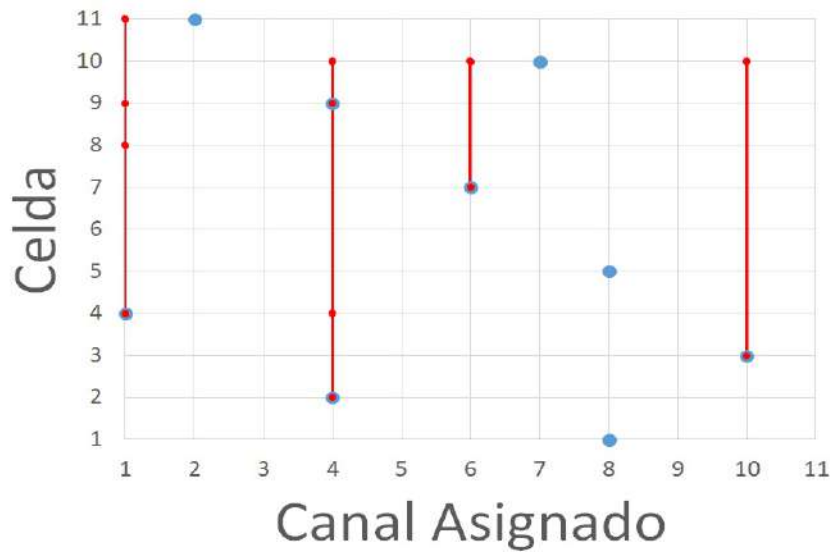


Figura 6.41: Malla de asignación para PUI3, inicialización con intervalos y celdas *HotSpot*, prioridad 4.

Contrario a lo esperado el nivel de prioridad otorgado no permite mejorar los resultados de la asignación, aunque en la inicialización se definen puntos fijos para las celdas 4, 10, 3 y 6, las asignaciones de los demás canales parecen interferir y por el mismo problema de cambio de posición del problema PHI3 no se puede corregir este error con PSO limpiamente, debe implementarse un método auxiliar que le permita salir de ese óptimo local.

6.6. Comparación de la asignación original y las asignaciones del trabajo de grado

La tabla 6.7 muestra los canales asignados a cada AP de la Facultad de Ingeniería de la Pontificia Universidad Javeriana. En las figuras 6.42, 6.43 y 6.44, se muestran las interferencias encontradas con dicha asignación de acuerdo a las 3 matrices planteadas con cada umbral de -45 dBm, -60 dBm y -70dBm, respectivamente.

ID	Dirección MAC	Canal
1	0XACA31E9EF200	1
2	0XACA31E9F0A40	1
3	0XD8C7C80D73B0	11
4	0XACA31E9EFD80	6
5	0XACA31E9F06E0	11
6	0XACA31E9F0AA0	6
7	0XD8C7C80D7310	11
8	0XACA31E9F04E0	6
9	0XACA31E9F0620	1
10	0XACA31E9F05E0	1
11	0XD8C7C80D5E00	1

Cuadro 6.7: Direcciones físicas de los AP y número de identificación.

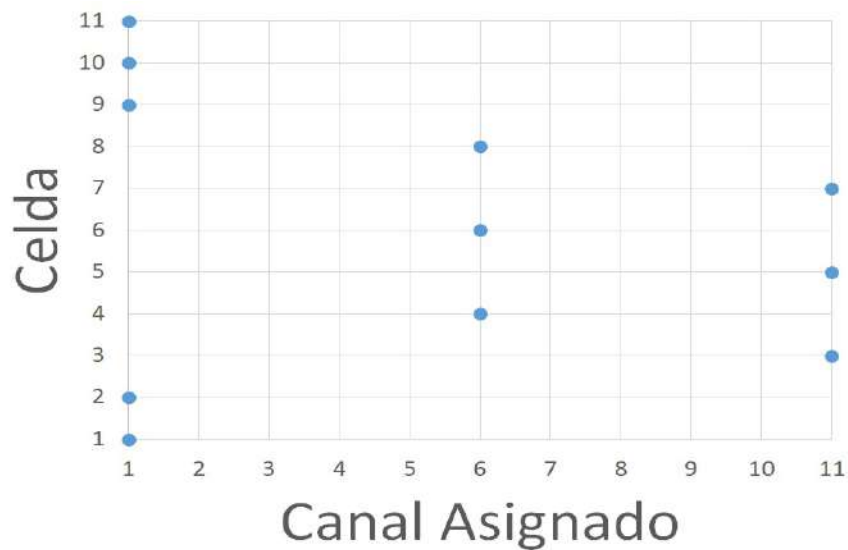


Figura 6.42: Asignación original de canales evaluados con PUJ1.

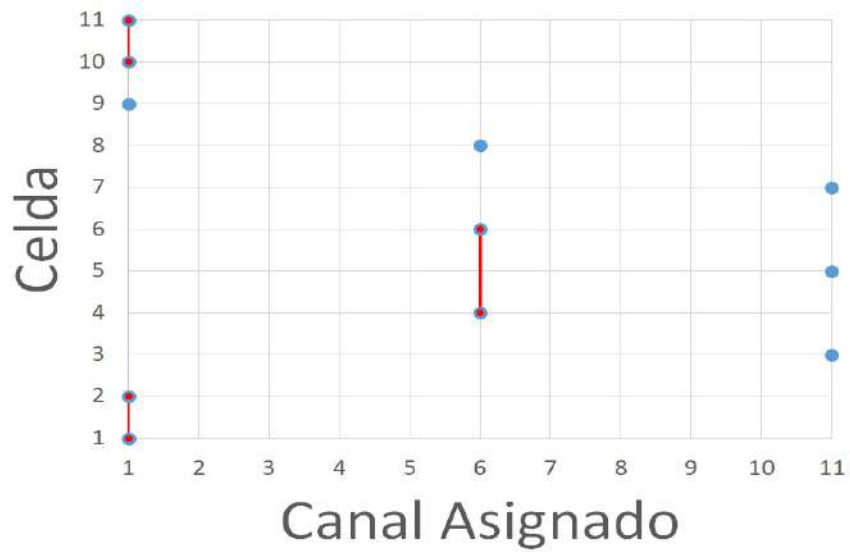


Figura 6.43: Asignación original de canales evaluados con PUF2.

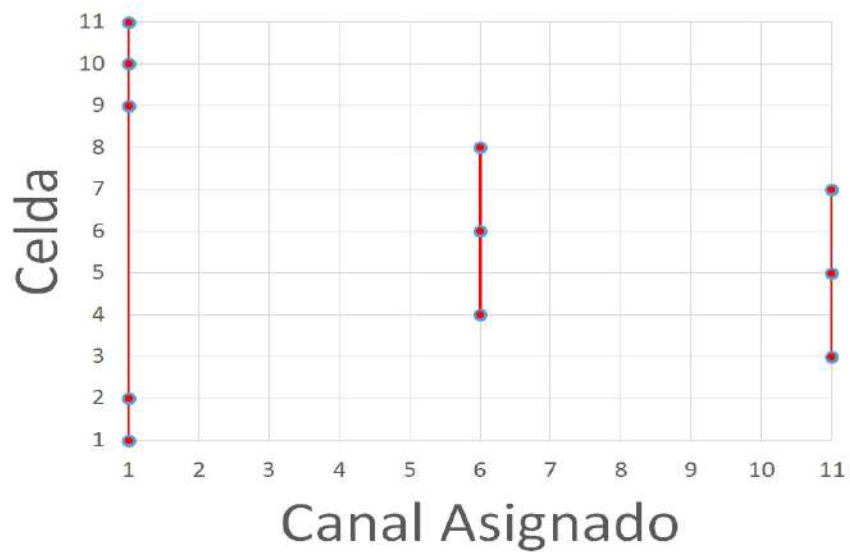


Figura 6.44: Asignación original de canales evaluados con PUF3.

La figura 6.45, muestra una comparación entre los resultados de la función objetivo de la asignación de canales originales de la Facultad, contra el mejor resultado obtenido para una asignación de generación aleatoria y contra el mejor obtenido de una asignación usando PSO en cualquiera de sus formas de inicialización, para cada uno de los problemas descritos por las matrices PUF1, PUF2 y PUF3.

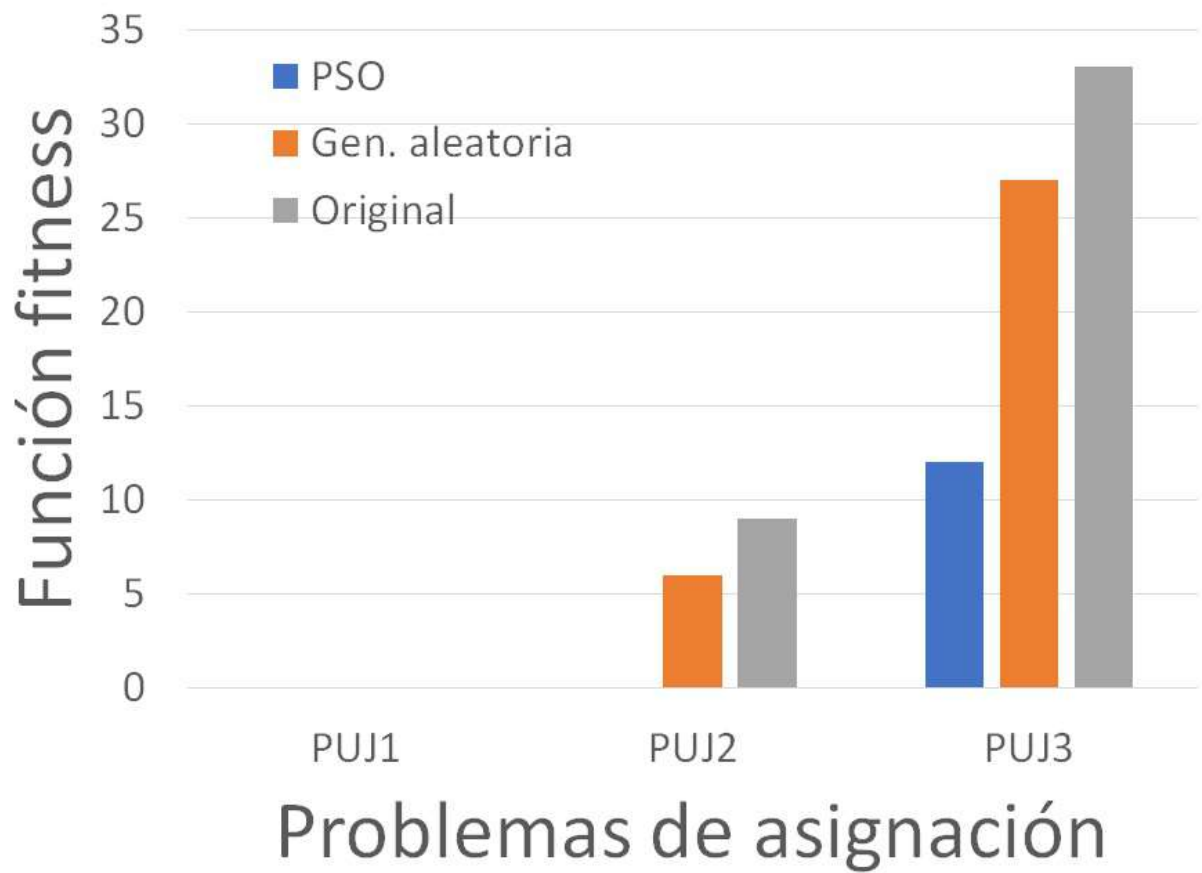


Figura 6.45: Comparación de los resultados de la evaluación de la función objetivo para el problema de asignación de la PUJ.

Capítulo 7

Conclusiones

Este trabajo de grado buscó mostrar el comportamiento del algoritmo PSO, en una plataforma de computación distribuida como lo es el cluster *ZINE*, del centro de alto rendimiento computacional Javeriano, y compararlo con el desempeño del mismo en un entorno convencional, como un computador de escritorio.

Con las primeras pruebas, se puede asegurar que el tiempo de ejecución disminuye significativamente en un factor de 1:20, lo cual es relevante para problemas más grandes en una implementación futura, pues mejora el rendimiento del algoritmo.

Los resultados de las pruebas, muestran que el la cantidad de procesadores del cluster otorga mayor varianza en la generación de las partículas, beneficiando la convergencia de los problemas, pues el enjambre esta mejor distribuido sobre el espacio de búsqueda. Adicionalmente, se debe mantener un enjambre grande para aumentar las probabilidades de encontrar una partícula que se aproximé a la solución óptima, pero por la limitación existente de memoria en las pruebas en un entorno convencional, no se podían operar con mayor cantidad de partículas. Aunque el entorno de pruebas distribuido no es suficiente para encontrar soluciones con interferencia nula, si se logra una disminución en los resultados obtenidos por la función objetivo diseñada.

Al implementar el algoritmo PSO con sus modificaciones propuestas, se mejoró significativamente el costo de la función fitness evaluada en este trabajo de grado, para los canales de la red de Access Points de la Facultad de Ingeniería de la Universidad Javeriana, en comparación con la evaluación de la asignación de frecuencias existente actualmente.

Se encontró una dificultad con la convergencia del algoritmo en problemas complejos de múltiples celdas y alta demanda, por lo que se diseñó una estrategia de inicialización, donde se tienen en cuenta las demandas y las restricciones co-sitio, para definir las celdas con mayor riesgo al incumplimiento de las condiciones dadas por las matrices de compatibilidad. Con este cambio se logró mejorar la convergencia, pero sigue siendo insuficiente, este error puede deberse al cambio de representación del problema de una manera binaria a entera (hipótesis). Como trabajo a futuro, se plantean dos modificaciones del algoritmo, la primera en cuanto a los cambios de posición de las partículas. Con este cambio, se buscaría que las celdas no prioritarias hicieran un desplazamiento lateral de canales, de manera ascendente o descendente, buscando evitar la infracción a las restricciones de compatibilidad electromagnética dictadas por cada matriz C , la cual es dependiente del problema. Y la segunda es cambiar el algoritmo y sus modificaciones a su representación binaria.

Capítulo 8

Anexos

8.1. Anexo 1

Código fuente en C del algoritmo PSO. <https://www.dropbox.com/s/bqinovk1jfgavwe/Codigo%20fuente%20PS0.c?dl=0>

8.2. Anexo 2

Código fuente en MPI (C) del algoritmo PSO en Paralelo. <https://www.dropbox.com/s/es416rne8q0423q/Codigo%20fuente%20PS0%20MPI.c?dl=0>

8.3. Anexo 3

Manual de instalación y pruebas de GCC. <https://www.dropbox.com/s/tqp1o1va3h49iai/PROCESO%20DE%20INSTALACI%C3%93N%20DE%20GCC.pdf?dl=0>

8.4. Anexo 4

Manual de instalación y pruebas de GCC, MPI y cluster (Debian-Linux). <https://www.dropbox.com/s/wx91w276wyon67y/GCCyMPIDebian.pdf?dl=0>

Bibliografía

- [1] 2014. <https://intothewave.wordpress.com/2014/11/06/which-mpi-implementation/>.
- [2] 2017. <http://ciencias.javeriana.edu.co/documents/3722984/3758121/Edificio+Ingenier>
- [3] Karen I. Aardal, Stan P. M. van Hoesel, Arie M. C. A. Koster, Carlo Mannino, and Antonio Sassano. Models and solution techniques for frequency assignment problems. *Annals of Operations Research*, 153(1):79–129, Sep 2007.
- [4] JosÁ© Miguel alonso. ProgramaciÃ³ndeaplicacionesparalelasconmpi(messagepassinginterface).
- [5] L. Benameur, J. Alami, and A. El Imrani. A new discrete particle swarm model for the frequency assignment problem. In *2009 IEEE/ACS International Conference on Computer Systems and Applications*, pages 139–144, May 2009.
- [6] Ing. Alexander Herrera Castro. Envío y monitoreo de trabajos en un cluster htcondor. Pontificia Universidad Javeriana, ZINE, Centro de alto rendimiento computacional, 2016, 2016.
- [7] M. Chakraborty, R. Chowdhury, J. Basu, R. Janarthanan, and A. Konar. A particle swarm optimization-based approach towards the solution of the dynamic channel assignment problem in mobile cellular networks. In *TENCON 2008 - 2008 IEEE Region 10 Conference*, pages 1–6, Nov 2008.
- [8] R. d. M. Calazan, N. Nedjah, and L. d. M. Mourelle. A cooperative parallel particle swarm optimization for high-dimension problems on gpus. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, pages 356–361, Sept 2013.
- [9] Jianxin Dai, Ming Chen, and Zhiliang Huang. A bi-objective multi-swarm particle swarm optimization algorithm for the frequency assignment problem. In *2011 IEEE 13th International Conference on Communication Technology*, pages 207–211, Sept 2011.
- [10] Universidad Carlos III de Madrid. <http://ocw.uc3m.es/ingenieria-informatica/arquitectura-de-computadores-ii/otros-recursos-1/or-f-001.-curso-de-mpi>.
- [11] I. Dolinska, A. Masiukiewicz, G. Rządowski, and M. Jakubowski. Algorithms for channels assignment in 802.11 networks. In *2016 International Conference on Information and Digital Technologies (IDT)*, pages 83–89, July 2016.
- [12] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, pages 39–43, Oct 1995.
- [13] H. M. Elkamchouchi, H. M. Elragal, and M. A. Makar. Channel assignment for cellular radio using particle swarm optimization. In *Proceedings of the Twenty Third National Radio Science Conference (NRSC'2006)*, volume 0, pages 1–9, March 2006.

- [14] K. Fallahi, H. Leung, and S. Chandana. An integrated aco-ahp approach for resource management optimization. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 4335–4340, Oct 2009.
- [15] S. Ghosh, A. Konar, and A. Nagar. Dynamic channel assignment problem in mobile networks using particle swarm optimization. In *2008 Second UKSIM European Symposium on Computer Modeling and Simulation*, pages 64–69, Sept 2008.
- [16] H. E. Hadji and M. Babes. Integrating tabu search in particle swarm optimization for the frequency assignment problem. *China Communications*, 13(3):137–155, March 2016.
- [17] Pontificia Universidad Javeriana, 2017. Centro de alto rendimiento computacional Javeriano. <http://zine.javeriana.edu.co/>.
- [18] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis. Particle swarm optimization for integer programming. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, volume 2, pages 1582–1587, 2002.
- [19] S. N. Ohatkar and D. S. Bormane. Hybrid channel allocation in cellular network based on genetic algorithm and particle swarm optimisation methods. *IET Communications*, 10(13):1571–1578, 2016.
- [20] Y. Rahmat-Samii, J. M. Kovitz, and H. Rajagopalan. Nature-inspired optimization techniques in communication antenna designs. *Proceedings of the IEEE*, 100(7):2132–2144, July 2012.
- [21] Yangyang Zhang and D. C. O’Brien. Fixed channel assignment in cellular radio networks using particle swarm optimization. In *Proceedings of the IEEE International Symposium on Industrial Electronics, 2005. ISIE 2005.*, volume 4, pages 1751–1756, June 2005.