



UNIVERSIDAD AUTÓNOMA METROPOLITANA

Unidad Iztapalapa

DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA

MAESTRÍA EN CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN

“ALGORITMOS GENÉTICOS PARALELOS PARA RESOLVER EL PROBLEMA DE RUTAS DE VEHÍCULOS CON VENTANAS DE TIEMPO”

Idónea comunicación de resultados que para obtener el grado de:

Maestra en Ciencias y Tecnologías de la Información

Presenta:

Ing. Maribel Flores González

Asesores:

Dr. Miguel Alfonso Castro García

Dr. Abel García Nájera

Jurado calificador:

Presidente: Dr. Santiago Domínguez Domínguez

Secretario: Dra. Graciela Román Alonso

Vocal: Dr. René Mackinney Romero

Vocal: Dr. Miguel Alfonso Castro García

México, Cd. de México a 5 de Septiembre de 2018

Resumen

El problema de rutas de vehículos (VRP por sus siglas en inglés), consiste en obtener las rutas de costo mínimo para la entrega de productos, a un conjunto de clientes que se encuentran dispersos geográficamente. Este problema es de gran interés por la comunidad científica por los beneficios que representa, y por los beneficios que brinda en el sector industrial, comercial y de servicio.

El VRP tiene variantes que se clasifican de acuerdo a ciertas restricciones, una de ellas es el problema de rutas de vehículos con ventanas de tiempo (VRPTW, por sus siglas en inglés), que es en el que enfocamos esta investigación.

En este trabajo se presenta un modelo paralelo de un algoritmo genético, para resolver casos de prueba del VRPTW. Este algoritmo realiza una exploración en el espacio de búsqueda para encontrar soluciones que minimizan el número de rutas y la distancia recorrida, el cual es el objetivo del problema. Para la parte del modelo paralelo se utilizó el paradigma de paso de mensajes mediante la biblioteca MPI (Message Passing Interface).

Los resultados obtenidos del algoritmo en paralelo para el VRPTW, se compararon con más de cincuenta casos de prueba disponibles públicamente (ver anexo tabla de resultados). Las soluciones obtenidas son comparables en términos de calidad de la solución, y tiempo computacional respecto al desempeño de la versión secuencial.

Agradecimientos

A Dios por darme la vida

A la Universidad Autónoma Metropolitana Iztapalapa, por darme la oportunidad de realizar mis estudios de posgrado en esta institución.

A mis asesores el Dr. Miguel Castro García y al Dr. Abel García Nájera, por su apoyo y paciencia que me brindaron durante la realización de este proyecto de investigación.

Al laboratorio de sistemas distribuidos por darme un espacio de trabajo y recursos para el desarrollo de la tesis, y en especial a los compañeros del laboratorio.

A todos mis compañeros de posgrado, por su amistad y apoyo.

A mi hermana Cristina, por su apoyo incondicional y paciencia que me demostró durante este tiempo.

A mi madre, que siempre será mi inspiración, por sus consejos, cariño y amor que me dio.

Contenido

Resumen	2
Agradecimientos.....	3
Lista de Figuras	6
Lista de Tablas	8
Acrónimos	9
1 Introducción.....	10
1.1 Antecedentes.....	11
1.2 Objetivo general	13
1.3 Objetivos específicos.....	13
1.4 Estructura de la tesis.....	13
2 VRPTW	14
2.1 El problema del VRP	14
2.2 Variantes del VRP.....	15
2.3 Problema de rutas de vehículos con ventanas de tiempo (VRPTW).....	16
2.4 Definición Formal del problema VRPTW.....	17
2.5 Metodologías para la solución del VRPTW	21
2.5.1 Métodos exactos	21
2.5.2 Heurísticas	21
2.5.3 Metaheurísticas.....	23
2.5.4 Algoritmos Genéticos	24
2.5.4.2 Algoritmos genéticos paralelos.....	29
2.6 Algoritmos genéticos multiobjetivo	31
3 Cómputo Paralelo	34
3.1 Cómputo paralelo	34
3.2 Arquitectura de sistemas paralelos	35
3.3 MPI.....	38
4 Algoritmo genético paralelo	42
4.1 Primera fase	42
4.2 Segunda fase	45

4.3 Modelo genético paralelo del algoritmo VRPTW	49
4.3.1 Primer modelo asíncrono paquete completo	51
4.3.2 Segundo modelo asíncrono paquete incompleto	52
4.3.3 Tercer modelo síncrono paquete completo	52
4.3.4 Cuarto modelo síncrono paquete incompleto	53
5 Resultados.....	54
5.1 Resultados de la implementación mediante JGAP	56
5.2 Resultados de la implementación del DrGA	57
5.3 Resultados de la implementación del DrGAP	58
6 Conclusiones.....	64
7. Referencias	67
8 ANEXOS (Algoritmos)	71
8 ANEXOS (Tabla de Resultados)	76

Lista de Figuras

1. Representación del problema VRP.....	15
2. Principales variantes del problema VRP	16
3. Ventana del tiempo	19
4. Dos rutas antes y después de ser unidas.	22
5. La forma general del AG.	24
6. Representación de un individuo	25
7. Cruza en un punto.....	26
8. Operador inversión	27
9. Operador Inserción	27
10. Operador intercambio	27
11. Operador basado en el desplazamiento.....	28
12. Algoritmo evolutivo paralelo maestro-esclavo.....	29
13. Algoritmo evolutivo paralelo modelo basado en Islas	30
14. Algoritmo evolutivo paralelo modelo celular.....	31
15. Ejemplo de óptimo de Pareto	32
16. Frente de Pareto	33
17. Suma de dos números	35
18. Arquitectura de un sistema con memoria compartida	37
19. Arquitectura de un sistema con memoria distribuida	37
20. Comunicador.....	40
21. Representación del cromosoma para JGAP.....	43
22. Una solución al VRPTW [30].....	46
23. Inserción	48
24. Intercambio.....	48
25. Inversión	48
26. Desplazamiento	49
27. Topología usada maestro – esclavo	50
28. Proceso 0 recibe Datos	51
29. Proceso 0 envía respuesta.....	501

30. Algoritmo paralelo.....	52
31. Proceso coordinador recibe de todos.....	53
32. Proceso coordinador envía la mejor solución.....	53
33. C1 y C2	54
34. R1 y R2	55
35. RC1 y RC2	55
36. Tiempo de ejecución del algoritmo en paralelo.....	62

Lista de Tablas

1. Comparación de la mejor solución y JGAP	56
2. Resultados del Algoritmo DrGA	57
3. Resultados de los Algoritmos en paralelo para 25 clientes	59
4. Resultados de los Algoritmos en paralelo para 50clientes	60
5. Resultados de los Algoritmos en paralelo para 100clientes	61
6. DrGA y DrGA paralelo.....	62

Acrónimos

VRP	Problema de rutas de vehículos o <i>Vehicle routing problem</i>
P	Tiempo polinomial o <i>Polynomial time</i>
NP	Polinomial no determinístico o <i>Nondeterministic polynomial time</i>
BBP	Problema de empaquetado en compartimientos o <i>Bin packing problem</i>
TSP	Problema del agente viajero o <i>Travelling salesman problem</i>
CVRP	VRP limitado en capacidad o <i>Capacitated vehicle routing problem</i>
MDVRP	VRP con múltiples depósitos o <i>Multi-depot vehicle routing problem</i>
SVRP	VRP estocástico o <i>Stochastic vehicle routing problem</i>
PVRP	VRP periódico o <i>Period vehicle routing problem</i>
VRPTW	VRP con ventanas de tiempo o <i>Vehicle routing problems with time windows</i>
AG	Algoritmo genético o <i>Genetic algorithm</i>
AE	Algoritmo evolutivo o <i>evolutionary algorithms</i>
EP	Programación evolutiva o <i>Evolutionary programming</i>
SISD	Flujo de instrucción único, flujo de datos único o <i>single instruction stream, single data stream</i>
SIMD	Flujo de instrucción único, flujo de datos múltiple o <i>single instruction stream, multiple data stream</i>
MIMD	Flujo de instrucción múltiple, flujo de datos múltiple o <i>multiple instruction stream, multiple data stream</i>
MISD	Flujo de instrucción múltiple, flujo de datos único o <i>Multiple instruction stream, single data stream</i>
UMA	Acceso uniforme a memoria o <i>Uniform memory access</i>
MPI	Interface para el paso de mensajes o <i>Message passing interface</i>
GALIB	Librerías de algoritmos genéticos o <i>Genetic algorithms library</i>
JAGA	Librerías para algoritmos genéticos en java o <i>Java API for genetic algorithms</i>
ECJ	Sistema de cómputo evolutivo en java o <i>Java evolutionary computation</i>
JGAP	Paquete de algoritmos genéticos en java o <i>Genetic algorithms and genetic programming package written in java</i>
DrGA	Algoritmo genético de densidad restringida o <i>Density restricted genetic algorithm</i>

Capítulo 1

Introducción

El principal objetivo del problema de rutas de vehículos VRP, consiste en reducir el costo de distribuir bienes a un conjunto de clientes dispersos geográficamente, donde el costo puede ser la longitud de rutas, número de vehículos utilizados, gasolina, tiempo, entre otros.

El VRP es un problema central en sistemas de distribución y transporte, que se dedican a la entrega o recolección de bienes, productos o servicios. El interés que se tiene en resolver este tipo de problemas es el ahorro, ya que, al disponer de técnicas de optimización para una correcta planificación y gestión, genera ahorros significativos en el costo total del producto. Toth y Vigo [1] aseguran que en un sistema de producción el costo del transporte va del 10% al 20% del costo final de los bienes.

Algunas aplicaciones del VRP son las siguientes:

- Problemas de preparación de pedidos en un almacén
- Planificación de transporte urbano
- Planificación de transporte escolar
- Problemas de reparto o distribución
- Sistemas de navegación

En el VRP existen condiciones que se deben tomar en cuenta en problemas de transporte como son: múltiples almacenes, tiempos de envío, capacidad de los vehículos, ventanas de tiempo, etc. Para estas condiciones, las cuales afectan la calidad del servicio prestado [2], se han desarrollado variantes que proponen restricciones adicionales al VRP,

por ejemplo:

- VRP limitado en capacidad (CVRP)
- VRP con múltiples depósitos (MDVRP)
- VRP estocástico (SVRP)
- VRP periódico (PVRP)
- VRP con ventanas de tiempo (VRPTW)

Al tomar en cuenta estas restricciones, la dificultad del problema se incrementa, ya que se toman en cuentas condiciones que se deben cumplir para satisfacer el objetivo de la variante del VRP. Esto hace difícil la búsqueda de una solución factible, por esto se considera de mucho interés para la comunidad científica, por el reto que presenta poder encontrar esa solución, a pesar de la complejidad que tiene, y además el beneficio que aporta.

1.1 Antecedentes

En 1959 se realiza el primer estudio sobre el problema de rutas de vehículos, Dantzing y Ramser [3] introdujeron el VRP, ellos proponen una formulación matemática a una aplicación real, un algoritmo que determina la ruta óptima de una flota de camiones para reparto de gasolina, entre una terminal (almacén) y un número de estaciones de servicio proporcionadas por la terminal.

Posteriormente Clarke y Wright en 1964 [4], propusieron un algoritmo para resolver el VRP, conocido como el algoritmo de ahorros, que resultó ser efectivo. Sin embargo, el principal defecto es su elevado costo computacional cuando el número de clientes es superior a una centena, por lo que su tiempo de ejecución resulta elevado para problemas grandes [5].

A partir de esta investigación se han desarrollado algoritmos para la solución del VRP, algunos autores han utilizado métodos exactos, métodos de aproximación y métodos

heurísticos como búsqueda tabú, algoritmos genéticos y colonia de hormigas.

El VRP se puede analizar como una combinación del TSP (problema del agente viajero) y del BBP (problema de empaquetado en compartimientos). El problema del TSP, consiste en localizar un recorrido que conecte todas las ciudades a visitar, de tal manera que solo las visite una vez y regrese al punto de partida. El objetivo es minimizar la distancia recorrida. El segundo problema se trata de colocar en un contenedor, un conjunto de paquetes de tal forma que se aproveche todo el espacio. Como el TSP y el BBP son problemas NP-duros, entonces, igualmente lo es para el VRP y no se conoce aún un algoritmo eficiente que resuelva todos los casos del problema de forma exacta.

Por lo tanto, a medida que crece el tamaño del problema a resolver, también lo hace el tiempo de ejecución. Dicho de otra manera, no se garantiza encontrar la mejor solución en un tiempo razonable. Por esta razón se utilizan heurísticas que permiten encontrar buenas soluciones.

En esta investigación, nos enfocaremos en el problema de rutas de vehículos con ventanas de tiempo. Para poder resolver este problema complejo, además de utilizar las técnicas mencionadas, en los trabajos más recientes recurren a técnicas de cómputo paralelo para resolver el problema del VRPTW [32], [33], [34], [36],[37] y otras variantes [38]. Estos han desarrollado algoritmos que encuentran buenas soluciones en tiempos de ejecución aceptables. Actualmente se tienen varios diseños de algoritmos paralelos, entre los cuales se distinguen por la forma de comunicación, por ejemplo maestro – esclavo, islas, e incluso hay modelos híbridos que combinan los diseños anteriores.

Por otro lado, una de las maneras en que se lleva a cabo el trabajo paralelo, es través del paso de mensajes, lo cual permite la comunicación entre los distintos procesadores, donde cada uno ejecuta una tarea distinta y posteriormente comparten datos por medio del paso de mensajes. Para lo anterior, se han utilizado un conjunto de librerías, para sistemas paralelos, conocidas como: OpenMP [39], MPI [23], (descrito con más detalle en el Capítulo 3) y donde combinan ambas [34]. Estas aplicaciones se pueden ejecutar en *clústers* o servidores de alto rendimiento.

En esta propuesta se hace uso de técnicas de cómputo paralelo a través del paso de mensajes, para obtener mejores soluciones del problema VRPTW, haciendo uso de un algoritmo genético que resuelve el problema en forma secuencial. Los resultados que se obtuvieron se comparan con los mejores conocidos y se describen con más detalle en el Capítulo 5.

1.2 Objetivo general

El objetivo principal es diseñar un algoritmo genético que, mediante herramientas de cómputo paralelo, se consiga buenos resultados para el VRPTW en términos de eficiencia y tiempo de respuesta.

1.3 Objetivos específicos

- Revisar literatura de trabajos, que han resuelto problemas de tipo multiobjetivo para el VRPTW.
- Revisar literatura relacionada con el análisis de los métodos de cómputo paralelo.
- Desarrollar una adaptación paralela a un algoritmo evolutivo para resolver el VRPTW.
- Evaluar los resultados obtenidos.

1.4 Estructura de la tesis

A continuación, se describe como está estructurado este trabajo: en el Capítulo 2 se presenta la definición formal del VRPTW y se presentan las metodologías propuestas para su solución. En el Capítulo 3 se exponen las características del cómputo paralelo y la herramienta utilizada para el diseño del algoritmo en paralelo. En el Capítulo 4 se presenta la metodología empleada para la solución con métodos genéticos paralelos. En el Capítulo 5 se analizan los resultados y por último en el Capítulo 6 se presentan las conclusiones y recomendaciones para trabajo futuro.

Capítulo 2

VRPTW

2.1 El problema del VRP

Como ya se mencionó en el Capítulo anterior, el VRP es un problema que aparece en la logística de distribución; consiste en abastecer a un conjunto de clientes que están geográficamente dispersos, mediante una flota de vehículos, con un costo mínimo.

El costo mínimo puede ser: el tiempo de reparto, el uso de combustible en los vehículos, la cantidad de vehículos de reparto, etc. Los datos del problema más importantes para la solución del VRP son: clientes, almacén, vehículos.

- a) Clientes: Cada cliente tiene una demanda que debe ser satisfecha por algún vehículo. La demanda es un bien que ocupa un lugar en los vehículos. Los clientes tienen restricciones relativas a su horario de servicio. Estas restricciones se expresan en forma de intervalos (ventanas) en los que se puede atender al cliente.
- b) Vehículos: Es el medio para transportar la mercancía a cada uno de los clientes. La flota puede ser homogénea (capacidad y costo iguales) dependiendo del tipo de problema. Comúnmente un mismo vehículo no pueda satisfacer la demanda de todos los clientes y un vehículo sólo puede visitar a un cliente una vez.
- c) Almacén: Es el punto de inicio y de terminación del recorrido de un vehículo, es aquí donde se encuentran los productos a repartir. Hay aplicaciones donde existen múltiples depósitos y los vehículos pueden iniciar y terminar su ruta en cualquiera de ellos. El almacén, al igual que los clientes, podrían tener ventanas de tiempo.

El objetivo principal del VRP es utilizar la menor cantidad de vehículos y minimizar la distancia recorrida. Por lo tanto, una solución al VRP es determinar las rutas para cada uno de los vehículos de la flota, cumpliendo con las restricciones y minimizando los costos.

En la Figura 1 se muestra una representación sencilla del VRP, en donde cada punto representa un cliente y las líneas que los unen son las rutas, que inician y terminan en el almacén.

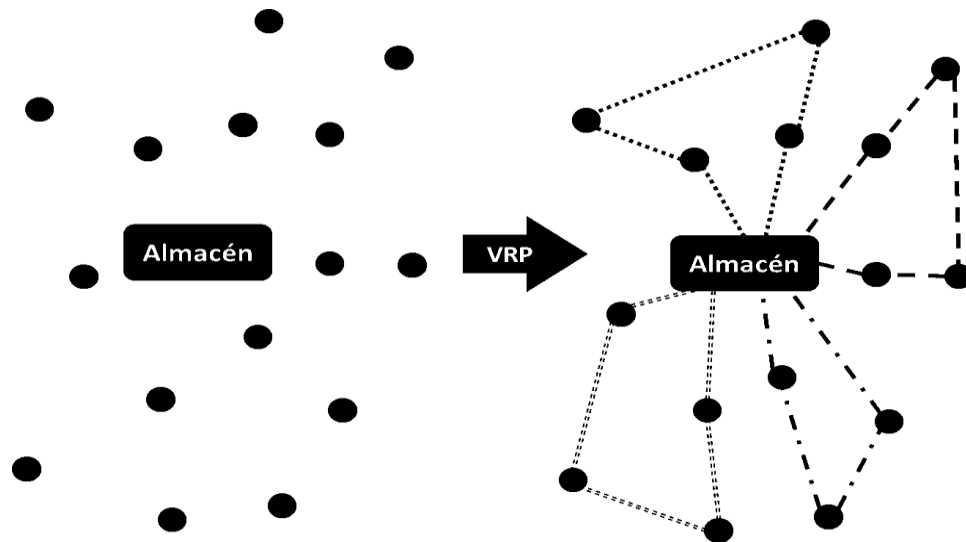


Figura 1. Representación del problema VRP

2.2 Variantes

En la literatura existen muchas variantes del VRP, las cuales difieren en las restricciones que consideran, entre las que encontramos las siguientes:

a) VRP limitado en capacidad (CVRP)

En este problema se consideran vehículos homogéneos, es decir, la flota tiene una capacidad determinada e igual para todos ellos y la suma de las demandas de los clientes que se visitan en cada ruta no puede superar la capacidad máxima de los vehículos.

b) VRP con múltiples depósitos (MDVRP)

En este problema, como su nombre lo indica, existen varios depósitos (almacenes) para abastecer a los clientes.

c) VRP estocástico (SVRP)

En este caso, algunos valores como el número de clientes, la demanda de cada cliente, el tiempo de servicio y el costo de traslado son estocásticos, es decir, se desconocen con cierta probabilidad.

d) VRP periódico (PVRP)

En esta variante, la demanda de los clientes incluye una periodicidad, la cual puede ser mayor a un día, es decir, el período de planificación se extiende a n días.

e) VRP con ventanas de tiempo (VRPTW)

Cada cliente tiene que ser atendido dentro de una ventana de tiempo u horario.

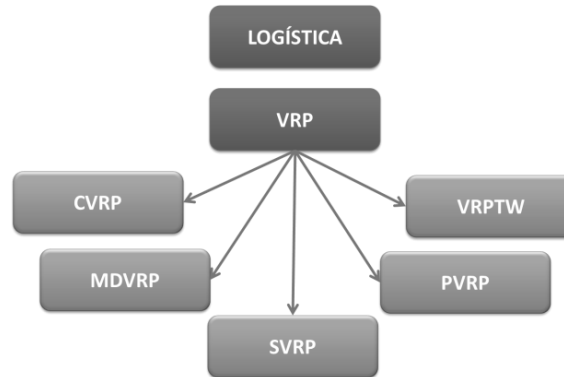


Figura 2. Principales variantes del problema VRP

En la Figura 2 se muestran las variantes del VRP, que se han descrito en esta sección. A continuación, se presentan las características del VRPTW, que es la variante bajo estudio en esta investigación.

2.3 Problema de rutas de vehículos con ventanas de tiempo (VRPTW)

El objetivo del problema de rutas de vehículos con ventanas de tiempo es encontrar el conjunto de rutas de menor costo para entregar mercancía a un número de clientes, que están dispersos geográficamente y darles el servicio de acuerdo a sus ventanas de tiempo. Una ventana de tiempo es el rango de tiempo establecido para la entrega de productos a cada cliente.

En este problema, cada cliente tiene una demanda de mercancía y una ventana de tiempo durante la cual debe ser abastecido, un tiempo de servicio necesario para descargar sus mercancías y cada cliente es atendido por solo un vehículo. Algunas restricciones para el VRPTW se mencionan a continuación:

- En el almacén está una flota de vehículos disponibles y es el punto de inicio y final de cada ruta.
- El almacén tiene un horario establecido, por lo que las rutas no pueden exceder ese horario.
- Cada vehículo tiene asignado un conjunto de clientes que tiene que abastecer, por lo tanto, la suma de las demandas de los clientes no debe exceder la capacidad del vehículo. El vehículo no puede llegar después de que la ventana de tiempo haya terminado. En este problema es posible que el vehículo espere a un cliente, en caso que llegue antes de que el cliente pueda atenderlo.

2.4 Definición Formal del problema VRPTW

La siguiente descripción formal, pertenece a cada una de las variables y restricciones del problema VRPTW.

- Un conjunto de clientes $C = \{c_1, \dots, c_n\}$, con demandas conocidas $d_i > 0, \forall i \in \{1, \dots, n\}$, cada cliente c_i debe ser visitado únicamente por un vehículo.
- Un nodo llamado almacén $d_0 = 0$, en donde están estacionados los vehículos, que es el punto de inicio y fin de cada ruta.
- $[b_0, e_0]$ es la ventana de tiempo del almacén.
- Un costo equilibrado w_{ij} , que es asignado a la distancia dos clientes o entre el almacén y cualquier cliente. $\forall i, j \in \{0, \dots, n\}, i < j$,
- Cada cliente tiene una ventana de tiempo $[b_i, e_i]$, que es el tiempo en que tiene que ser abastecido.
- Un tiempo de servicio s_i , que es el tiempo que tiene cada cliente en descargar, para cada cliente, el servicio debe comenzar dentro de su ventana de tiempo.
- Un conjunto de vehículos idénticos con capacidad limitada $q \geq \max \{d_i : i \in$

$\{1, \dots, n\}$ es decir, la suma de las demandas de los clientes en una ruta no puede sobrepasar la capacidad del vehículo.

También se consideran los siguientes parámetros

- Los índices de los vehículos son: $k \in [1 \dots m]$
- Los índices de los clientes es $i \in [1 \dots n]$
- El índice del almacén es $i = 0$
- La cantidad de demanda del cliente i es q_i $i \in [1 \dots n]$
- La capacidad del vehículo k es q_k
- La distancia desde el cliente i (o almacén 0) a j es c_{ij} , $i, j \in [0 \dots n]$
- La distancia del recorrido desde el cliente i a j es t_{ij} , $i, j \in [0 \dots n]$
- La ventana de tiempo del cliente i , o del almacén es $[b_i, e_i]$, $i \in [1 \dots n]$, donde b_i es el inicio y e_i es el fin de la ventana de tiempo.

La fórmula para el problema multiobjetivo del VRPTW se puede representar de la siguiente forma.

$$\text{Minimizar} \quad \sum_{k=1}^m \sum_{i=0}^n \sum_{j=0}^n C_{ij} x_{ij}^k \quad (1)$$

Sujeto a

$$\sum_{k=1}^m \sum_{i=0}^n x_{ij}^k = 1, \quad j \in [1 \dots n] \quad (2)$$

$$\sum_{k=1}^m \sum_{i=0}^n x_{ji}^k = 1, \quad j \in [1 \dots n] \quad (3)$$

$$\sum_{i=1}^n x_{0i}^k = 1, \quad k \in [1 \dots m] \quad (4)$$

$$\sum_{i=1}^n x_{i0}^k = 1, \quad k \in [1 \dots m] \quad (5)$$

$$r_0^k \leq q_k, \quad k \in [1 \dots m] \quad (6)$$

$$b_i \leq s_i^k, \quad i \in [1 \dots n], \quad k \in [1 \dots m] \quad (7)$$

$$s_i^k + u_i \leq l_i, \quad i \in [1 \dots n], \quad k \in [1 \dots m] \quad (8)$$

La función objetivo representada en (1), representa el tiempo o distancia total, que se requiere para abastecer a todos los clientes, las variables x_{ij}^k toma el valor de 1 si el vehículo k va del cliente i al cliente j , y es 0 en otro caso. El almacén se representa como $i = 0$ o $i = 1 + n$.

Las restricciones que se deben tomar en cuenta (2) y (3), que indican que cada cliente es abastecido por un solo vehículo. La restricción (4) y (5), establecen que cada vehículo inicia y termina su ruta en el almacén, garantizando que ningún vehículo es usado más de una vez. La restricción (6) señala que para todo vehículo k , la suma de las demandas de todos los clientes idénticos no debe exceder la capacidad de este, donde la variable r representa la carga del vehículo.

Por ultimo las restricciones (7) y (8) establecen que cada cliente tiene que ser atendido dentro de la ventana de tiempo, donde s_i^k representa el tiempo en el que el vehículo inicia el servicio a cada cliente i

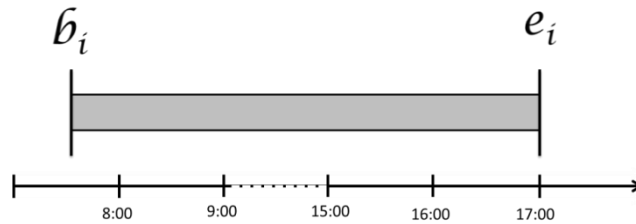


Figura 3. Ventana del tiempo

Una forma de representar la ventana de tiempo es como la que se muestra en la Figura 3, donde b_i es el inicio de la ventana de tiempo y e_i el final.

Para calcular la distancia de los n clientes, cuyas coordenadas son (i, j) , se calcula usando la distancia euclidiana(9) [30] donde t_{ij} y d_{ij} , se refieren a los costos de la ruta, que pueden ser reflejado en tiempo o distancia.

$$t_{ij} = d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (9)$$

La distancia total, se refiere a la suma de todas las distancias que recorren todos los vehículos desde que salen del almacén hasta su regreso, y el número de rutas que se requiere para abastecer a todos los clientes.

Para calcular el número de rutas, se tiene que conocer del cliente: el inicio y fin de la ventana de tiempo, la demanda y el tiempo de servicio para calcular el número de rutas. También es necesario calcular: la distancia entre el cliente visitado y el anterior, el tiempo de espera, el tiempo de arribo y el tiempo de salida.

Por ejemplo; cuando se visita a cada cliente, se determina un tiempo de arribo a_i , que es el tiempo al llegar a este punto. Al iniciar el recorrido todos los vehículos salen del almacén al tiempo 0 y para el i -ésimo cliente, el tiempo de arribo está representado por la ecuación (10).

$$a_i = l_{i-1} + t_{i-1,i} \quad (10)$$

En donde l_{i-1} es el tiempo de salida del cliente $i-1$, y $t_{i-1,i}$ es el tiempo de traslado del cliente $i-1$ al i . Es importante considerar que un vehículo no puede llegar después de finalizada la ventana de tiempo del cliente. El vehículo puede llegar antes, en cuyo caso tendrá que esperar y este tiempo de espera está representado por w_i , que es el tiempo que el vehículo tiene que esperar para estar dentro del intervalo de la ventana de tiempo del cliente i . Así el tiempo de salida l_i se representa con la ecuación (11) y el tiempo de espera por la ecuación (12).

$$l_i = a_i + w_i + s_i \quad (11)$$

$$w_i = \begin{cases} 0 & \text{si } a_i \geq b \\ b_i - a_i & \text{en otro caso} \end{cases} \quad (12)$$

2.5 Metodologías para la solución del VRPTW

Algunos métodos de solución que han sido útiles, se pueden clasificar de acuerdo a las siguientes categorías: Métodos exactos, Heurísticas, Metaheurísticas. Sin embargo, la complejidad de los problemas de optimización, plantea un reto difícil para su resolución mediante algoritmos exactos.

2.5.1 Métodos exactos

Los métodos exactos son aquellos que nos dan la solución óptima a un problema. Estos métodos funcionan correctamente para un problema pequeño, por ejemplo, para un caso del VRP con 50 clientes se puede utilizar un método exacto para su solución. Sin embargo, para un caso con mayor número de clientes este problema llega a complicarse en tiempo y recurso computacional. En estos métodos se pueden mencionar a ramificación y acotamiento y a ramificación y corte, los cuales están basados en relajaciones de programación lineal [9].

2.5.2 Heurísticas

Estos son algoritmos que proporcionan una buena solución a un problema específico. Este tipo de técnicas no garantizan una solución exacta, pero en general proveen buenas soluciones. En contraste con los anteriores, su costo en recursos de tiempo es bajo y son específicos para cada problema [10]. Las heurísticas se pueden clasificar como: constructivas y de mejora.

- Constructivas:

Construyen una solución progresivamente, tomando en cuenta una solución parcial, y van haciendo cambios para llegar a una solución final, aunque no siempre encuentran la solución óptima. Por ejemplo, una heurística constructiva bien conocida para el VRP es el algoritmo de ahorros de Clarke y Wright [43], y el vecino más cercano.

- Algoritmo de ahorros:

El algoritmo de Clarke y Wright, calcula el ahorro en términos de distancia, si en una solución dos rutas diferentes $(0, \dots, i)$ y $(0, j, \dots, 0)$, pueden ser combinadas formando una nueva ruta $(0, \dots, i, j, \dots, 0)$ como se muestra en la Figura 4. El ahorro (distancia) que se obtiene por la unión es $s_{ij} = c_{0i} + c_{0j} - c_{ij}$.

Para obtener mayores ahorros, parte de una solución inicial y se realizaron las uniones siempre que no quebrante las restricciones del problema. Para este algoritmo hay una versión paralela que trabaja sobre todas las rutas simultáneamente y una secuencial que construye las rutas una cada vez [42].

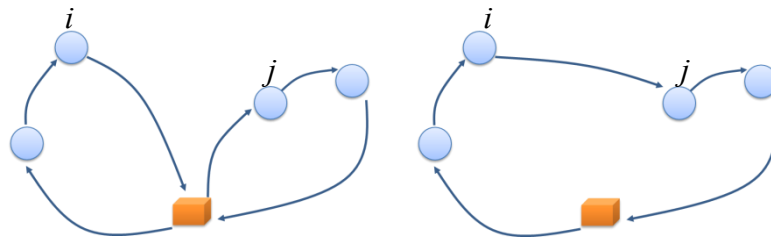


Figura 4. Dos rutas antes y después de ser unidas.

- Vecino más cercano

Este algoritmo fue propuesto por Rosenkrantz, Stearns y Lewis [45] construye una solución desde cero, ruta por ruta, una cada nodo tomando en cuenta la menor distancia entre uno y los demás nodos, parte de un nodo inicial (puede ser el almacén) incorporando el nodo más cercano de su posición, después de ser incluido se inicia la búsqueda del siguiente nodo a ser incluido con la menor distancia al anterior nodo incluido.

Para la incorporación de los nodos, puede tomar en cuenta las restricciones; como la capacidad del vehículo, tiempo de servicio y número de rutas. La inserción de nodos se realiza de manera secuencial hasta que se hayan visitado todos.

- De mejora

Este tipo de heurísticas trabajan sobre una solución factible, y comienza una búsqueda local mediante modificaciones de la solución actual.

2.5.3 Metaheurísticas

Son métodos generales que se pueden aplicar a diferentes problemas multiobjetivo. Una metaheurística consiste en procedimientos iterativos que guían una heurística subordinada, combinando distintos conceptos para explorar adecuadamente el espacio de búsquedas. Este tipo de métodos pueden ser aplicados a problemas complejos para los cuales no se puede llegar a la solución de forma exacta.

Algunos de estos métodos, se basan en la imitación de la naturaleza, la evolución biológica o procesos físicos asociados a la manufactura.

- Colonia de hormigas

Esta metaheurística está basada en el proceso de alimentación de las hormigas. Cuando una hormiga encuentra un camino hacia alguna fuente de alimento, deja un rastro de feromonas. La cantidad de feromonas que deja depende de la longitud del camino y de la cantidad del alimento encontrado. De esta forma, si una hormiga detecta las feromonas decidirá qué camino elegir, el cual tenga la mayor cantidad de sustancia, lo que provocará un aumento en la cantidad de feromonas en ese trayecto. En caso de que una hormiga no detecte la presencia de feromonas, se mueve aleatoriamente hacia otras direcciones, para ser exploradas [11].

- Búsqueda tabú

Esta técnica fue propuesta por Glover [10], la cual pertenece a la clase de métodos de búsqueda local. Consiste en explorar el espacio de soluciones para resolver problemas de manera eficiente, para esto utiliza información de los movimientos realizados con anterioridad, mediante el uso de estructuras de memoria. Una vez que una solución es encontrada e se marca como “tabú”, de manera que el algoritmo no vuelve a visitar esa posible solución. Así, al tener un historial de los caminos recorridos y las soluciones encontradas, se puede evitar permanecer en las mismas regiones y recorrer regiones nuevas para encontrar otras posibles soluciones.

2.5.4 Algoritmos Genéticos

Los algoritmos genéticos AG, son métodos de búsqueda, que se basan en la teoría de la evolución de Darwin, por su imitación de la evolución natural. Se basa en los mecanismos de selección que utiliza la naturaleza, mediante los cuales los individuos más aptos de una población son los que tienen mayor posibilidad de sobrevivir y cada individuo de esta población representa una solución a un problema.

Los algoritmos genéticos simulan el proceso de evolución como un proceso de cambios. El ciclo de trabajo (generación) de estos algoritmos es el siguiente: medir la aptitud de los individuos en la población, crear una nueva población mediante la selección, cruzamiento y mutación de individuos y la selección de los individuos que sobreviven.

Este proceso se muestra en el algoritmo que presenta la Figura 5.

-
1. Número de población de soluciones iniciales
 2. Calcular la aptitud de estas soluciones
 3. **REPETIR**
 4. Seleccionar dos padres
 5. Combinar ambos padres y crear uno o dos hijos
 6. Realizar la Mutación en los hijos
 7. Calcular la aptitud de los hijos
 8. Seleccionar los sobrevivientes para la nueva generación
 9. **HASTA** criterio de terminación.
-

Figura 5. La forma general del AG.

Para la construcción de un AG para resolver el VRP se deben de tomar en cuenta los siguientes parámetros:

A. Representación

La población es representada por un conjunto de cromosomas o individuos, en donde cada cromosoma representa una solución posible del problema. La representación es una permutación de n números que constituyen la población inicial. A su vez, cada cromosoma contiene varios genes. La Figura 7 es un ejemplo de un individuo con 10 genes.

3	5	9	7	1	6	2	4	8	10
---	---	---	---	---	---	---	---	---	----

Figura 6. Representación de un individuo

B. Función de aptitud

Para poder evaluar cada solución, es importante establecer una función de aptitud, la cual sirve para conocer la calidad relativa de las soluciones. Esta función mide la adaptación del individuo y se aplica sobre los cromosomas para evaluar la calidad del gen (aptitud). Esta evaluación sirve para darles una puntuación con el fin de saber que tan buena o mala es la solución respecto a las demás.

C. Operador de selección:

Este operador selecciona un individuo para ser sometido al operador de cruza. Los individuos más aptos deben tener mayor probabilidad de ser seleccionados, sin embargo, los individuos con baja aptitud también pueden ser elegidos. Existen varias técnicas para la selección de padres, de las cuales se describen dos de las más utilizadas a continuación.

- Selección por ruleta: Este método utiliza una forma de selección como un juego de ruleta, donde cada individuo tiene una sección de la misma, pero los más aptos obtienen mayores secciones, que los menos aptos. Luego la ruleta se hace girar, y cada vez se elige al individuo que tenga la sección en la que se pare la ruleta.
- Selección por torneo: Este método consiste en elegir un conjunto de individuos de la

población para que compitan y elegir al mejor individuo del grupo para reproducirse.

D. Operador de cruce:

Este operador es análogo a la reproducción que a partir de dos cromosomas genera nuevos descendientes, en los que se combinan características de ambos padres. Con ello se obtienen nuevas soluciones al problema. Las formas más comunes de cruce son: uso de un punto de cruce y uso de dos puntos de cruce.

- **Cruza en un punto:** Se escoge un punto para dividir los genes de ambos padres y, una vez divididos, se cruza la parte inicial del padre A con la parte final del padre B y viceversa para formar los hijos, como se muestra en la Figura 7.

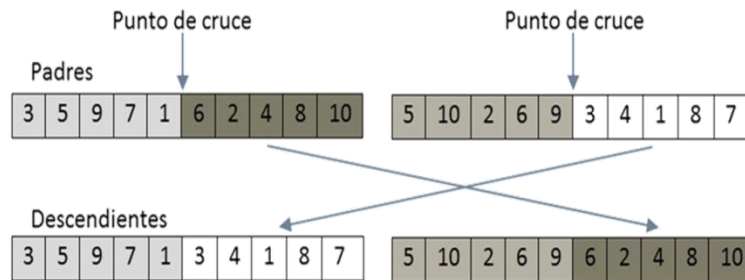


Figura 7. Cruza en un punto

Normalmente la operación de cruce en un algoritmo genético se maneja con un porcentaje que indica con qué frecuencia se efectuará.

E. Operador de mutación:

Mediante esta operación, los nuevos descendientes pueden modificar alguno de sus genes. Este operador permite mantener diversidad en la población. Se han propuesto muchas técnicas para aplicar este operador, entre las cuales podemos mencionar inversión, inserción, intercambio, remplazo, desplazamiento.

- **Inversión:** Se eligen al azar dos posiciones dentro de un cromosoma y se invierte el orden de recorrido entre los puntos elegidos. Este operador preserva la mayor parte de la información sobre las adyacencias, pero altera significativamente la información sobre el orden, como se ilustra en la Figura 8.

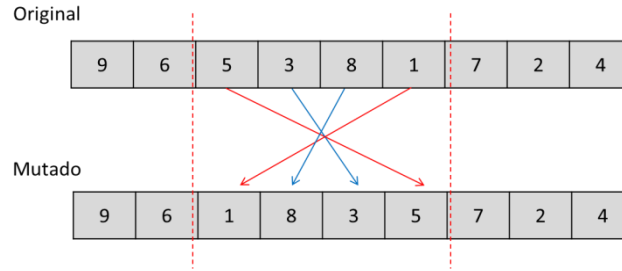


Figura 8. Operador inversión

- Inserción: Se selecciona una posición al azar y se inserta en otra aleatoria. Este operador preserva la mayor parte del orden que existe entre los valores y de la información sobre los contiguos, como se ilustra en la Figura 9.

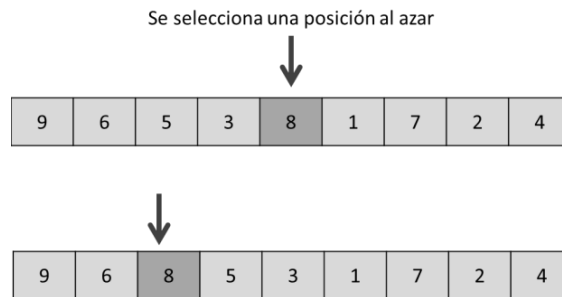


Figura 9. Operador Inserción

- Intercambio: Se eligen dos posiciones al azar y se intercambian los valores. Este operador preserva la mayor parte de la información sobre los contiguos, pero el orden cambia significativamente. como se muestra en la Figura 10.

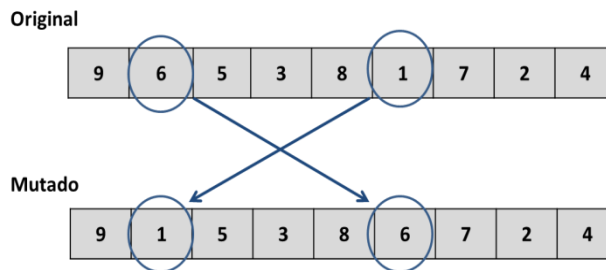


Figura 10. Operador intercambio

- Desplazamiento: Se elige una subsecuencia al azar y se inserta en otra posición escogida aleatoriamente y las demás se desplazan como se muestra en la Figura 11.

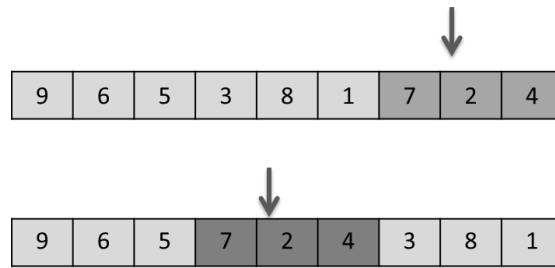


Figura 11. Operador basado en el desplazamiento

2.5.4.1 Coevolución

La coevolución es un término de la biología, se le atribuye a Ehrlich y Raven en 1964, en su artículo donde muestran los resultados de la evolución paralela entre mariposas y plantas, en donde el término coevolución se utilizó para expresar la relación evolutiva entre éstas [46].

Otra definición fue dada por Janzen en 1980 [46] “coevolución *es aquel proceso por el cual dos o más organismos ejercen presión de selección mutua y sincrónica (en tiempo geológico) que resulta en adaptaciones específicas recíproca*”.

Por lo anterior, coevolución se utiliza para describir los casos en los que dos o más especies influyen mutuamente para su evolución, es decir, se refiere a un cambio evolutivo bilateral entre especies que interactúan entre sí [15] [12].

Una característica importante como resultado de la coevolución es el cambio que existe en las especies involucradas. Así la primera desarrollará una mejora y la segunda, por selección natural, desarrollará otra adaptación. Por ejemplo, en el caso de una presa que es perseguida por un depredador, la primera desarrollará mejor rapidez y destreza para ocultarse y la segunda, en respuesta, mejorará su estrategia de ataque.

Este concepto se ha utilizado en el área de inteligencia computacional para desarrollar algoritmos con enfoque coevolutivos [35]. La mayoría son algoritmos genéticos en donde cada población trabaja de manera independiente y la aptitud de un individuo

depende de los individuos en otras poblaciones. Podemos encontrar un tipo de coevolución cooperativa, en donde la aptitud de un individuo es el resultado de una colaboración con individuos de otras especies o poblaciones.

El concepto de coevolución cooperativa se ha utilizado normalmente para problemas de optimización [12] [13], incluyendo el VRP [14], el cual se ha considerado como un problema multiobjetivo.

2.5.4.2 Algoritmos genéticos paralelos

Las técnicas de procesamiento paralelo se aplican en AG, con el objetivo de obtener eficiencia computacional y mejorar en la calidad de la búsqueda genética. Estos algoritmos introducen un modelo de evolución diferente al secuencial, permitiendo explorar de forma simultánea varias poblaciones distribuidas para resolver el mismo problema. La manera en que se distribuye la población, constituye el principal criterio para clasificar los modelos. A continuación, se mencionan tres modelos más comunes de los Algoritmos Evolutivos Paralelos.

A. Modelo maestro esclavo:

Este modelo consiste en la implementación de un componente llamado maestro que se encarga de recolectar la información procesada por cada esclavo y de distribuirla en su totalidad a cada uno de ellos, esto se puede hacer en cada generación establecida según el AG. En la figura 12 se muestra gráficamente este modelo.

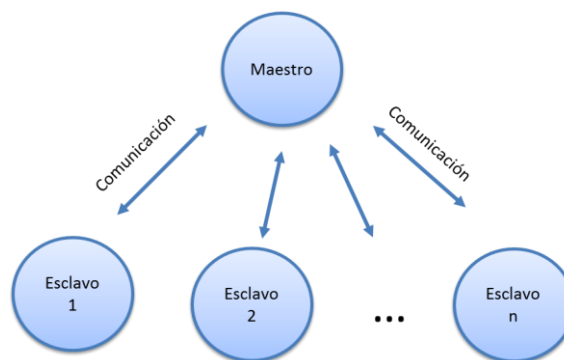


Figura 12. Algoritmo evolutivo paralelo maestro-esclavo

Se pueden tener las versiones síncrona y asíncrona, para coordinar las poblaciones antes de iniciar una nueva generación. Algunas implementaciones de este modelo para variantes del VRPTW [37].

B. Modelo basado en Islas:

Este modelo se caracteriza al trabajar con un conjunto de subpoblación (islas) donde cada una de ellas ejecuta en forma paralela un AG secuencial, y solo se puede interactuar entre los mismos individuos de una misma isla, hasta cierta generación. Solo un proceso especial llamado migración puede realizar intercambios ocasionales de individuos entre islas, según una cantidad de generaciones preestablecidas y parámetros determinados. La figura 13 muestra gráficamente el modelo de islas.

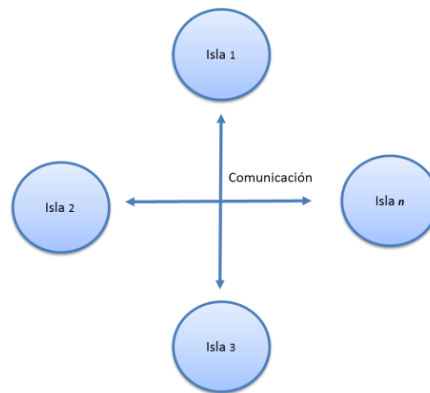


Figura 13. Algoritmo evolutivo paralelo modelo basado en Islas

C. Modelo celular:

Se caracteriza por que la población se distribuye en una malla bidimensional, un individuo por cada celda, en el que se define una estructura de vecindario, tiene un modelo de propagación de características de individuos denominados *difusión*, la forma de comunicación está definida por la topología de interconexión de elementos de procesamiento [38]. La figura 14 muestra gráficamente el modelo celular.

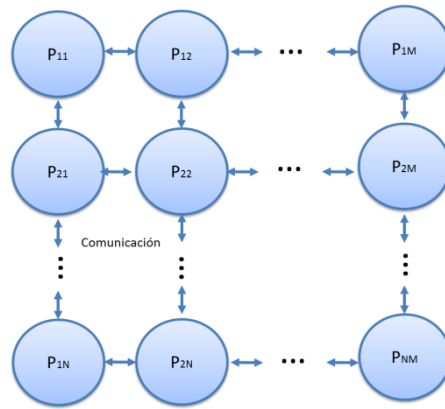


Figura 14. Algoritmo evolutivo paralelo modelo celular

Estos modelos tienen diferentes variantes, y existen modelos híbridos que combinan las características de dos o más de los modelos presentados [38], [44], celular y de islas [45].

2.6 Algoritmos genéticos multiobjetivo

Un problema multiobjetivo es aquel que suele tener más de un objetivo, que plantea la optimización (minimación o maximización) de un conjunto de funciones posiblemente en conflicto entre sí [46],[47]. Consiste en encontrar un vector de variables de decisión que satisfaga las restricciones del problema y optimice al mismo tiempo las funciones objetivo [6].

$$\begin{array}{ll}
 \text{Minimizar / Maximizar} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})) \\
 \text{sujeto a} & \mathbf{G}(\mathbf{x}) \leq 0 \\
 & \mathbf{H}(\mathbf{x}) = 0
 \end{array} \tag{1}$$

El problema (1) muestra el modelo general de un problema de optimización multiobjetivo. La solución a este tipo de problemas corresponde a un vector de variables de decisión $\mathbf{x} = (x_1, x_2, \dots, x_n)$ que satisfaga las restricciones impuestas por los vectores de

funciones G y H, ofreciendo valores que representen una compensación entre los valores de la funciones a optimizar f_1, f_2, \dots, f_n [7].

Este conjunto de soluciones que satisface las restricciones, se caracterizan generalmente, ya que entran en conflicto al tratar de buscar el óptimo [41]. Para comparar dos elementos del espacio de soluciones se establece el concepto de dominancia de Pareto, una solución S_1 domina a otra S_2 , cuando la primera mejora a la segunda en al menos un objetivo y no es peor en los otros. En cambio, se consideran indiferentes ($S_1 \sim S_2$) cuando ni S_1 domina a S_2 , ni S_2 domina a S_1 [8].

Algunos algoritmos multiobjetivos hacen uso del concepto conocido como Pareto óptimo que define, x^* es Pareto-óptima, si y solo si no existe otro vector x talque $z = f(x)$ domine a $y = f(x^*)$. En decir que x^* es un óptimo de Pareto si no existe un vector x que haga mejorar algunos de los objetivos sin que empeore de forma simultánea alguno de los otros.

Por ejemplo, la Figura 3 muestra la relación entre dos funciones f_1 y f_2 , la primera representa el número de vehículos y la segunda la distancia. La solución representada por Q indica que mejora a R y S en ambos objetivos, distancia y número de vehículos, y es mejor que P sólo en cuanto a número de vehículos, sin embargo, el punto P es mejor que Q en cuanto a distancia.

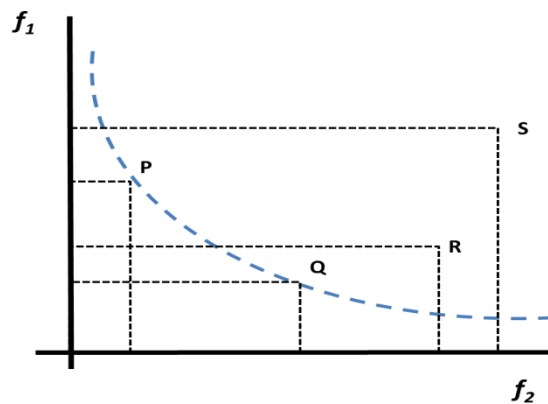


Figura 15. Ejemplo de óptimo de Pareto

Para cualquier problema multiobjetivo, la meta es encontrar el frente de Pareto o un conjunto de soluciones que representen la mejor aproximación posible, esta debe estar lo más cerca del frente de Pareto y que las soluciones estén lo mejor distribuidas [42].

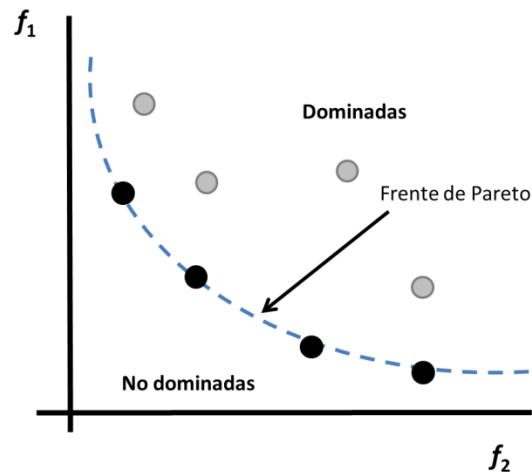


Figura 16. Frente de Pareto

A la solución formada por el conjunto de todos los vectores que son no – dominados se les conoce como, el conjunto no dominado o frente de Pareto.

En la Figura 16 se muestran varias soluciones, en donde los puntos más oscuros representan a las soluciones que forman el frente de Pareto.

Para mejorar la precisión de la aproximación del frente de Pareto y así mantener la diversidad de soluciones, es decir, conseguir un conjunto de soluciones no dominadas lo más grande posible, ya que entre mayor sea la diversidad, mejor será la información sobre las posibles soluciones a este problema, se utilizan diferentes métodos [30], algunos de ellos son los siguientes.

- a) Dominación en profundidad: Implica dividir la población en diferentes frentes no dominados.
- b) Dominación en clasificación: Número de soluciones por las cuales una determinada solución es dominada.
- c) Dominación en contador: Se calcula de acuerdo al número de soluciones que domina cierta solución.

Capítulo 3

Cómputo Paralelo

3.1 Cómputo paralelo

El avance en la tecnología de procesadores ha mejorado la capacidad de cómputo. Por ejemplo, algunos problemas pueden requerir días o meses de procesamiento en una computadora con un solo procesador, en cambio, en una computadora paralela se puede realizar en fracciones de horas. La reducción de los tiempos de respuesta es una ventaja que ofrece el procesamiento en paralelo, gracias al uso de la memoria compartida, velocidad de procesamiento, número de procesadores y el tipo de interconexión que existe entre estos.

El cómputo paralelo es la manera de dividir una tarea entre varios procesos que trabajan simultáneamente, es decir, partir la tarea en varias subtareas de manera que cada procesador trabaja en cada una de ellas. El objetivo es reducir el tiempo total de cómputo distribuyendo la carga de trabajo entre los procesadores disponibles. Una definición de [15] nos dice que: “Es una colección de elementos de procesamiento que se comunican y cooperan entre sí para resolver problemas grandes de manera rápida”. Esencialmente un sistema en paralelo es un conjunto de procesadores capaces de cooperar en la solución de un problema.

Muchas aplicaciones científicas requieren una cantidad considerable de tiempo de cálculo, la forma en que se puede resolver este problema es mediante su distribución entre varios procesadores de manera eficiente [16], para lo cual se requiere de programación en paralelo.

Actualmente se cuenta con métodos de programación en paralelo que funcionan en máquinas multi-core, sistemas de memoria compartida y distribuida.

3.2 Arquitectura de sistemas paralelos

Michael J. Flynn [17], propuso una clasificación de las computadoras basándose en el número de instrucciones y los flujos de datos que procesa. El flujo de instrucciones se refiere a la secuencia de instrucciones que son ejecutadas por la computadora, y el flujo de datos a la secuencia, de los datos sobre los cuales operan las instrucciones. Se pueden encontrar secuencias de instrucciones sencillas o múltiples y secuencias de datos sencillas o múltiples. Flynn distinguen cuatro grupos de computadoras, de las cuales dos se aplican a las computadoras paralelas [15].

- A. Single Instruction stream, Single Data stream (SISD)
- B. Single Instruction stream, Multiple Data stream (SIMD)
- C. Multiple Instruction stream, Single Data stream (MISD)
- D. Multiple Instruction stream, Multiple Data stream (MIMD)

A. Single Instruction stream, Single Data stream (SISD)

Este tipo de sistemas se caracterizan por tener un solo flujo de instrucciones sobre un único flujo de datos realiza una instrucción por ciclo, es decir una cadena de instrucciones es ejecutada, una tras otra sobre una secuencia de datos extraídos de la memoria. Por ejemplo, en la Figura 17 se necesita cinco ciclos para completar la tarea.

Este sistema no tiene posibilidades de procesamiento en paralelo, ya que esta arquitectura es el tipo de computadora secuencial, de acuerdo con el modelo de Von Neumann, cuya característica principal es que dispone de una única memoria principal, donde se almacenan los datos y las instrucciones, un camino sobre el cual fluyen, y un mecanismo de sincronización.

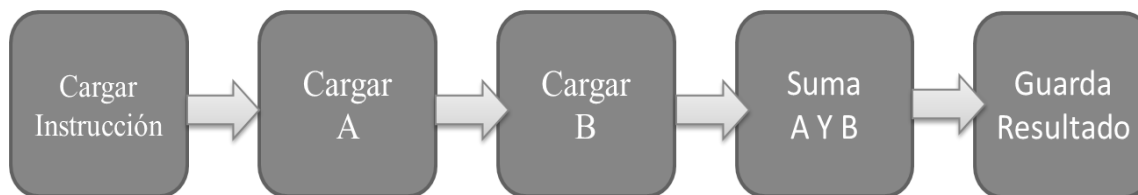


Figura 17. Suma de dos números

B. Single Instruction stream, Multiple Data stream (SIMD)

Estos sistemas tienen un flujo de control para múltiples datos. Este tipo de máquinas tienen varios procesadores para llevar a cabo una misma instrucción sobre un conjunto de datos. Para aplicaciones con un alto grado de datos en paralelo, esta arquitectura es muy eficiente [15]. Algunos ejemplos de estos sistemas son aplicaciones multimedia, algoritmos de gráficos para generar vistas tridimensionales.

C. Multiple Instruction stream, Single Data stream (MISD)

Múltiples instrucciones pasan a través de un número de procesadores, cada uno con su propia unidad de control, pero compartiendo una memoria común. Esta arquitectura es solo de interés teórico porque no se ha construido ningún sistema que utilice esta estructura. [18].

D. Multiple Instruction stream, Multiple Data stream (MIMD)

Este tipo de sistemas son los que trabajan con más de un procesador, es asíncrono, no tiene reloj central, cada procesador es independiente, ejecuta su propia secuencia de instrucciones y por lo mismo hacen instrucciones diferentes.

Los procesadores operan de forma asíncrona, es decir pueden estar haciendo diferentes tareas al mismo tiempo. Estos tipos de sistemas a su vez se clasifican en: sistemas de memoria compartida y sistemas de memoria distribuida.

a. Sistema de memoria compartida:

Cada procesador tiene acceso a toda la memoria, tienen el mismo espacio de direcciones. En este sistema se debe de asegurar que los procesadores no accedan de forma simultánea a las mismas regiones de memoria ya que provocaría que ocurra alguna falla. La Figura 18 hace referencia a este tipo de sistema.

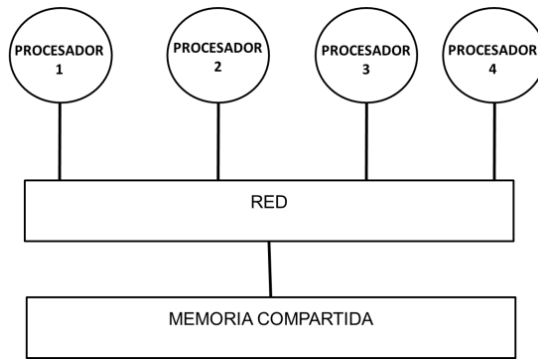


Figura 18. Arquitectura de un sistema con memoria compartida

Este tipo de MIMD con memoria compartida es conocido como sistemas de multiprocesamiento simétrico o multiprocesadores.

b. Sistema de memoria distribuida

Este tipo de sistemas se puede ver como una computadora paralela, donde cada procesador tiene su propia memoria local, pero sin un canal compartido. Para poder transferir datos es necesario que exista una red de conexión, como se muestra en la Figura 19. Este tipo de sistemas son más económicos y son fáciles de escalar, ya que, si la demanda crece, se pueden agregar más memoria y procesadores.

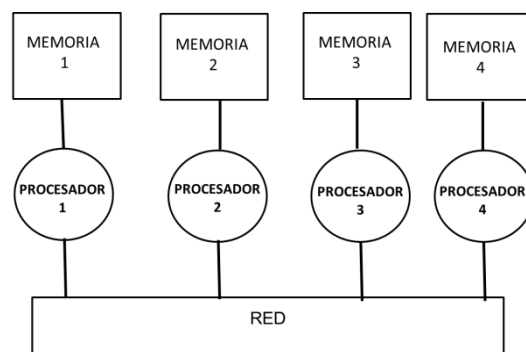


Figura 19. Arquitectura de un sistema con memoria distribuida

A estas también se les conoce como multicomputadoras o clúster. Un clúster puede estar formado por varios equipos interconectados de tal manera que trabajan en conjunto,

distribuyéndose la tarea entre ellos. Estos se clasifican en dos tipos según sus características físicas:

- Clúster homogéneo, se considera dentro de esta clasificación, si están formados por equipos de la misma arquitectura y recursos similares.
- Clúster heterogéneo, se considera dentro de esta clasificación si están formados por nodos con distinta capacidad de cómputo.

La descripción de esta clasificación de computadoras, da un panorama de la forma en que trabajan los procesadores, en especial las arquitecturas paralelas. Uno de los objetivos del paralelismo es ejecutar un programa en el menor tiempo posible haciendo uso de varios procesadores.

El cálculo de operaciones paralelas en un sistema de memoria distribuido, por ejemplo, se realiza dividiendo la tarea en varias porciones, distribuyéndolas en cada uno de los nodos para que ejecuten estas subtarefas. Para realizar esto se debe conocer cómo se realiza la comunicación entre los procesos, por lo que es necesario conocer los lenguajes de programación paralela, la mayoría de los cuales ya tienen diseñadas primitivas para el manejo de procesos de manera asíncrona o síncrona.

Algunas herramientas del paralelismo que facilitan la forma en que se va a distribuir la carga de trabajo son: MPI, POSIX. A continuación, describiremos que es MPI y las características más importantes de este conjunto de librerías para sistemas paralelos.

3.3 MPI

La interfaz de paso de mensajes MPI es utilizado en las arquitecturas de memoria distribuida como protocolo de comunicación. MPI permite la comunicación entre procesos a través de mensajes que contienen datos. El intercambio de datos debe ser realizado por paso de mensajes, esto permite la sincronización entre procesos y evitar la exclusión mutua.

Además, cada proceso tiene una memoria local, a la que tiene acceso exclusivo, no hay memoria global. Cada proceso puede acceder a sus datos locales y puede intercambiar información y datos con otros procesos por medio del envío y recepción de mensajes. Los elementos principales en este proceso son el proceso que envía, el que recibe y el mensaje, es decir similar al procedimiento conocido como cliente-servidor.

Para facilitar el diseño del programa, se asume que cada uno de los procesos ejecutan el mismo programa SPMD [15]. Pero no es una restricción. Ya que cada proceso puede ejecutar diferentes partes del programa, únicamente se tiene que especificar el rank (identificador de cada proceso). De esta manera, cada proceso ejecuta el mismo programa, pero hará diferentes instrucciones dependiendo en que proceso se ejecuta.

3.3.1 Funcionamiento

MPI cuenta con bibliotecas que proporcionan rutinas para el manejo de paso de mensajes, estas permiten entre otras cosas: Inicializar, gestionar y finalizar procesos, comunicar datos, realizar operaciones de comunicación entre grupos de procesos. Las funciones básicas de MPI son las siguientes:

- `MPI_INIT`: Inicializa el proceso en MPI, bloquea la ejecución hasta que todos los procesos hayan sido inicializados.
- `MPI_COMM_SIZE`: Obtiene el número de procesos ejecutados en el grupo.
- `MPI_COMM_RANK`: Obtiene el número de rango del proceso actual.
- `MPI_FINALIZE`: Termina la conexión MPI, bloquea la ejecución hasta que todos los procesos terminen.
- `MPI_SEND`: Permite el envío del mensaje, especificando, tamaño, tipo, un identificador del mensaje y al proceso.
- `MPI_RECV`: Recibe mensaje.

Para que exista la comunicación entre los procesos y que puedan transferir datos locales, es necesario que exista un comunicador. Este se define como un grupo de procesos,

en donde cada proceso tendrá un rango (rank) que lo identifica. Un caso simple puede ser, una transferencia punto a punto, por ejemplo, de un proceso *A*, a un proceso *B*, donde *A* llama una operación de envío, y *B* llama a una operación de recepción. Cada mensaje que se envía se identifica por una etiqueta (tag) que permite identificar al mensaje y en cada mensaje enviado especifica quien lo envía, el tipo y número de datos que se envían, el comunicador al que pertenece y el estatus de la recepción.

El comunicador por defecto se llama `MPI_COMM_WORLD`, aunque se pueden establecer más comunicadores. La Figura 20 muestra la representación de un comunicador, por medio de un semicírculo, que contiene un conjunto de procesos, representados con círculos y cada uno tiene un identificador (rank), ese identificador comienza en 0.

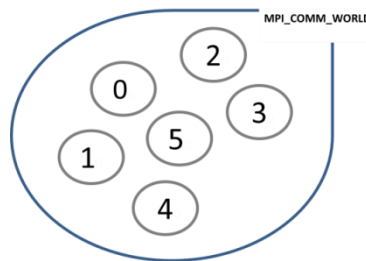


Figura 20. Comunicador.

Para realizar el proceso de intercambio de mensajes y de información entre procesos, MPI maneja funciones de comunicación que son de punto a punto, colectivas y con tipos de datos, las cuales se explican a continuación.

a) De punto a punto:

Esta es la forma de comunicación más sencilla, dos procesos participan en esta operación de comunicación, consiste en el envío de un mensaje desde un proceso a otro. Para que esto se lleve a cabo se utilizan las funciones `MPI_Send` y `MPI_Recv`. Las dos maneras de comunicación en un tipo punto a punto son de tipo bloqueante y síncrona.

- Bloqueante (síncrono): El mensaje se envía, pero la rutina no termina hasta que el mensaje es recibido por el proceso receptor.

- No Bloqueante (asíncrono): El mensaje se envía o se recibe inmediatamente, cada proceso puede seguir trabajando y posteriormente se puede verificar si se recibió el mensaje.

b) De tipo colectivas:

Esta se utiliza cuando se quiere lograr la comunicación entre varios grupos de procesadores (comunicadores). Para ello se tienen rutinas que permiten la comunicación colectiva que son: MPI_Bcast, MPI_Scatter, MPI_Gather, MPI_Reduce.

c) Con tipos de datos:

Esta es una forma de enviar información entre los procesos cuando la información consta de diferentes tipos de datos, por ejemplo, una estructura de datos. En este caso, los datos se pueden mandar de forma separada en varios mensajes, pero esto es ineficiente, en lugar de esto se puede enviar un submensaje del tipo MPI_Type_struct, el cual contenga todos los datos.

3.4 Ventajas de MPI

- Escalabilidad, mientras que la demanda de recursos crece, se puede agregar más memoria y procesadores.
- Mejor capacidad de proceso
- Permite mejorar la calidad y fiabilidad de los resultados
- Menor tiempo de proceso

Algunos trabajos que utilizaron herramientas paralelas como MPI, aplicado a problemas multiobjetivo, para la solución de problemas VRPTW, indican que, los algoritmos presentan un buen rendimiento en términos de calidad de las soluciones y esfuerzo computacional [32] [34], [39]. Así como lograr una alta precisión de las soluciones a través de la cooperación de los procesos [40].

Capítulo 4

Algoritmo genético paralelo

En esta investigación se realizaron dos fases para resolver el problema, en la primera se desarrolla un algoritmo genético que resuelve el problema del VRPTW. Se realiza una exploración de las herramientas disponibles para resolver problemas multiobjetivo, los resultados se muestran en el capítulo 5. La segunda fase involucra el diseño en paralelo de un algoritmo genético que resuelve el VRPTW, la cual se tomó del trabajo realizado por García-Nájera y Bulimia [30] para resolver el VRPTW, utilizando métodos específicos de AG. El cual permite la incorporación de técnicas de paralelismo. A continuación se describe cada una de estas propuestas.

4.1 Primera fase

En la actualidad, para el desarrollo de AG existe una gama de bibliotecas disponibles en lenguajes diferentes, las cuales permiten su rápido desarrollo. Por ejemplo, *EADG* [19] para Matlab, *GALIB* [20] para c/c++, *GA - playground* [21], *JAGA* [22] y *JGAP* [23] para java. La mayoría de estos marcos de trabajo son adecuados para hacer frente a problemas que utilizan el enfoque evolutivo.

Para resolver el VRPTW mediante AG, se propone utilizar un framework con métodos evolutivos y poder adaptarlo al VRPTW. Para esto se utiliza JGAP (Java Genetic Algorithms Package), ya que provee un conjunto de librerías que proporciona operadores de cruce, mutación y selección más utilizados para el desarrollo de AG. JGAP es una aplicación orientada a objetos, que ofrece clases e interfaces para representar: genes, cromosomas, individuos, población, la aptitud y operadores genéticos.

Además, simula el proceso de selección, permite crear individuos de forma aleatoria, a los cuales se les aplican los operadores genéticos, se evalúa cada nueva solución y al final devuelve el individuo mejor adaptado.

La idea del JGAP es utilizar los métodos que proporciona, para la aplicación rápida de AG y resolver problemas de optimización, tomando en cuenta las restricciones, los parámetros y variables que ya se han mencionado correspondientes al VRPTW. Para la implementación de JGAP, se deben de realizar cinco pasos básicos que se describen a continuación:

1) Cromosomas

El cromosoma representa una solución, y este se divide en múltiples genes [22]. Un cromosoma representa una secuencia de clientes y cada gen representa a un cliente. Para este problema utilizaremos una población compuesta por m arreglos, donde cada arreglo es una permutación de n números enteros y cada número representa el identificador de un cliente (por esto tiene que ser único). Esta permutación corresponde a un individuo (cromosoma) de la población. La Figura 21 muestra un ejemplo de un individuo.

3	5	9	7	1	6	2	4	8	10
---	---	---	---	---	---	---	---	---	----

Figura 21. Representación del cromosoma para JGAP

2) Implementación de la función aptitud

La función aptitud, en JGAP permite valorar la aptitud de los individuos. Este parámetro se tiene que implementar, el cual tiene que calcular la distancia total, y el número de rutas, de cada cromosoma, considerando las restricciones de ventanas de tiempo, para esto se emplea las fórmulas 9 al 11 descritas en el capítulo 2.

3) Población

La población es el conjunto de soluciones posibles para el VRPTW. Cada solución

está representada por un cromosoma, a la población de cromosomas se le llama genotipo en JGAP y la clase para definir la población es *Genotype*. Se crea la población con el método *RandomInitialGenotype*, para generar una población al azar.

Una vez hecha la configuración de la población y creada la función de aptitud, se deben de establecer los operadores que actuarán sobre la población. Estos son los operadores genéticos, que permiten mantener la diversidad genética, para llevar a cabo el proceso de evolución de la población, hasta obtener el individuo mejor adaptado. Para lograr esto se debe de crear un objeto de configuración.

4) Objeto de configuración:

JGAP cuenta con una clase *DefaultConfiguration*, que tiene los ajustes para realizar el proceso evolutivo, para esto necesitamos proporcionarle la siguiente información: la función de aptitud, el orden de los cromosomas y el número de cromosomas, los cromosomas, es una permutación de n números que constituyen la población y una solución posible al problema.

El método de selección preestablecido es de elitismo, ya que asegura que el individuo con mayor aptitud sobrevivirá para la siguiente generación. El método de cruce es de un punto y el de mutación es el de intercambio.

5) Evolucionar la población

Una vez que se ha terminado la configuración, se tiene que hacer evolucionar la población un número determinado de veces, hasta encontrar el individuo con mayor aptitud. El método para hacer este proceso es *population.evolve*. La implementación con JGAP es útil, ya que nos permite seleccionar los operadores genéticos sin necesidad de desarrollarlos. Además, la plataforma de desarrollo es accesible, así como la sencillez de explicar cada método. Los resultados obtenidos se presentan en el capítulo 5.

La implementación de JGAP, en el diseño de algoritmos multiobjetivo, tiene ventajas en cuanto al uso de métodos genéticos, las herramientas que se puede utilizar de forma

gratuita sin tener que diseñar. Sin embargo una de las desventajas que se observa es la dificultad en la modificación de métodos, que permitan mejorar la diversidad de la población, e identificar las soluciones no dominadas. Además dificulta la posibilidad de implementarlo en paralelo utilizando las herramientas propuestas para este proyecto. Por lo tanto se sugiere el uso de un algoritmo multiobjetivo que resuelve el problema del VRPTW, y facilita la implementación en paralelo, así como ofrece buenas soluciones de forma secuencial, en seguida se describe la segunda fase de este proyecto.

4.2 Segunda fase

A continuación se presenta el modelo DrGA (Algoritmo genético de densidad restringida) [30], propuesto por García-Nájera y Bulimia, que ofrece una solución para el VRPTW como problema multiobjetivo. Se toma esta solución ya que los resultados obtenidos son buenos. Además, nos facilita la incorporación de métodos de paralelismo en lenguaje C para la implementación de MPI, que es una de las contribuciones principales de este trabajo.

El DrGA (por sus siglas en inglés, density restricted genetic algorithm) es un algoritmo genético que resuelve el VRPTW como problema multiobjetivo que, además, restringe la densidad de soluciones como una forma sencilla para preservar la diversidad, con el objetivo de obtener una mejor exploración del espacio de búsqueda. A este algoritmo se le introdujo un parámetro en relación de la densidad (cantidad de soluciones concentradas en una zona) [24]. A continuación se describen las características principales de este algoritmo.

1) Representación de una solución

En este caso, el cromosoma se divide en tres partes, como se muestra en la Figura 22. El primer gen del cromosoma representa el número de rutas de la solución. Los siguientes n genes son una permutación de n clientes, cuya secuencia indica el orden en que se atienden los clientes. Los últimos genes nos indican la posición de los genes que corresponde al

último cliente que visita en cada ruta. De esta forma una solución nos indica el número de rutas, y los clientes que visitó cada ruta.

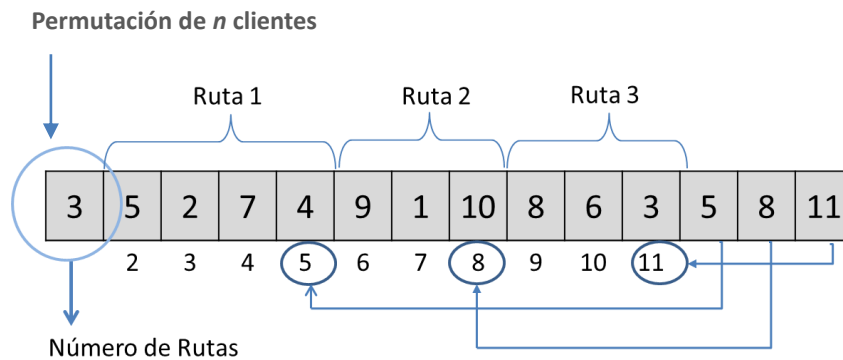


Figura 22. Una solución al VRPTW [30]

2) Población inicial

La población inicial contiene soluciones generadas de forma aleatoria. La manera en cómo se crea es de la siguiente forma; Primero un cliente es seleccionado de forma aleatoria y se coloca como el primero en visitar, en esa ruta, posteriormente otro cliente es elegido de la misma forma si cumple las restricciones de capacidad y tiempo se coloca después del anterior cliente. Si no se cumplen dichas restricciones, se crea una nueva ruta y este cliente será el primer cliente para visitar en la nueva ruta, este proceso se repite hasta que todos los clientes estén asignados a una ruta.

3) Función aptitud

En problemas multiobjetivo, no hay una única solución que optimice todos los objetivos (distancia y rutas), pero se pueden encontrar, resultados óptimos, en un conjunto de soluciones llamadas no dominadas o frente de Pareto (ver sección 2.6).

Para mejorar la precisión de la aproximación del frente de Pareto y mantener su diversidad, al aplicar AG en este tipo de problemas García- Nájera [32] propone 2 objetivos; el primero consiste asignar un valor escalar a la aptitud de cada individuo, y el segundo se

refiere a evitar las soluciones iguales en el conjunto no dominado, para esto hace uso de los siguientes métodos: dominación en profundidad, dominación en clasificación, dominación en contador.

Mantener la diversidad en la aproximación de Pareto es importante, porque de esta manera se puede asegurar que las soluciones sean diferentes. Pero la preservación de la diversidad se relaciona directamente con la densidad, ya que la última proporciona información para mejorar la diversidad. Para medir la estimación de la densidad de las soluciones García – Nájera [30], utiliza tres métodos, kernel, vecino más cercano, histograma los cuales se describen a continuación.

- Kernel: Se aplica una función kernel a la distancia entre una solución y todas las demás, la densidad será la suma de estas evaluaciones.
- Vecino más cercano: Este método calcula la distancia entre un punto dado y su *k*-ésimo vecino para estimar la densidad.
- Histograma: Este método, realiza cuadrículas *k*- dimensionales en el espacio *k*-dimensional. El número de individuos en la misma área de la cuadrícula, se toma como la solución para la estimación de la densidad.

Después del proceso de selección, se aplican los operadores genéticos, para el desarrollo del DrGA, se aplicaron: cruce y mutación. El operador de cruce permitirá crear nuevos individuos con mejor aptitud, como resultado de la selección de los mejores padres. La mutación se aplica a cada individuo después de la cruce y consiste principalmente en el intercambio de genes.

a) Cruce:

El operador de cruce elige un número aleatorio de rutas del primer padre y las copias en el hijo. Luego se copian aquellas rutas del segundo padre que no incluyen ningún cliente ya copiado del primer padre. Si quedan clientes sin asignar, estos son asignados en el orden que aparecen a cualquiera de las rutas existentes en el hijo. Si después de la inserción de un cliente se genera una solución no factible, se crea una nueva ruta.

b) Mutación: Para la mutación García- Nájera [30] considera algunos operadores que se describen a continuación: Inserción, intercambio, inversión.

- Mutación por inserción: Una sub-ruta es elegida de forma aleatoria, y estos clientes se insertan en otra ruta. En la Figura 23 se muestra la forma en que trabaja.

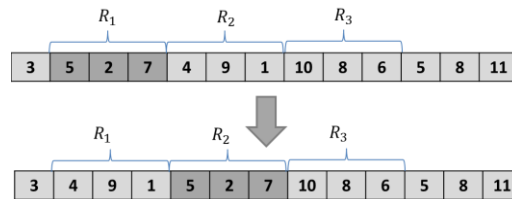


Figura 23. Inserción

- Mutación por intercambio: dos sub-rutas son seleccionados al azar y sus clientes se intercambian, como se representa en la Figura 24.

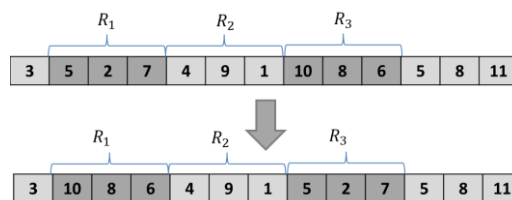


Figura 24. Intercambio

- Mutación por inversión: Una sub-ruta es elegida al azar y los clientes se invierten. Como se observa en la figura 25.

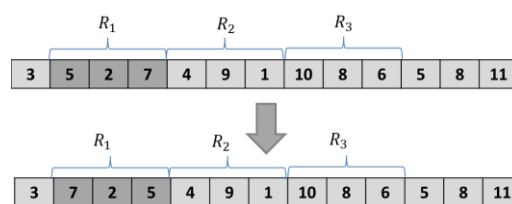


Figura 25. Inversión

- Mutación por desplazamiento: Una sub - ruta es seleccionada al azar y los clientes se desplazan una posición. En la figura 26, se muestra la forma en que trabaja este operador.

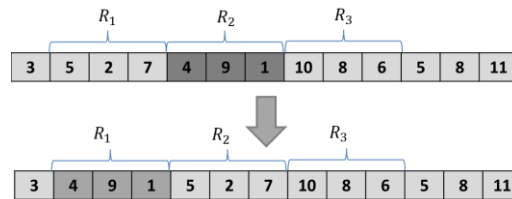


Figura 26. Desplazamiento

La falta de diversidad genética conduce a que todos los individuos de la población sean muy parecidos. Por esto es importante verificar si después de las etapas de cruce y mutación, la densidad de las soluciones iguales, son más del máximo permitido. Para evitar la falta de diversidad García – Nájera [30], introduce un parámetro, al que nombra *diversity ratio* δ . Este parámetro permite eliminar soluciones similares, dejando solo un porcentaje de la población, cuando después de las etapas de cruce y mutación la densidad de las soluciones ha crecido más del máximo permitido.

4.3 Modelo paralelo del algoritmo VRPTW

Las ventajas de utilizar el cómputo paralelo son, entre otras: dividir una tarea entre varios procesos, intercambiar información entre ellos y obtener mejoras en el tiempo de ejecución [33], [36]. Esto da soporte para modelar en paralelo el algoritmo DrGA, haciendo uso de funciones que facilitan el proceso de comunicación e intercambio de información para mejorar la solución y promover la exploración de más soluciones factibles [3].

La idea general de esta implementación, es ejecutar en paralelo el algoritmo DrGA utilizando procesos que trabajan simultáneamente y obtener soluciones de mayor calidad sin incrementar el tiempo de ejecución [33]. Para todos los algoritmos que se describen, se utiliza una topología similar al de maestro-esclavo, como se muestra en la Figura 27, en la cual hay un proceso coordinador (maestro) que recibe información de los demás procesos participantes (esclavos).

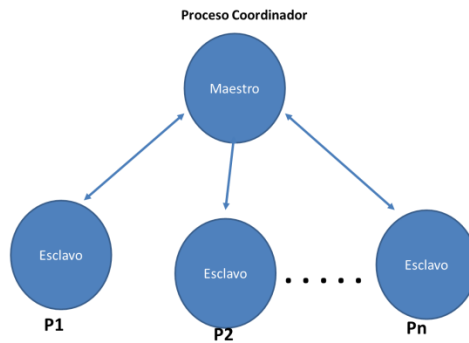


Figura 27. Topología usada maestro – esclavo

La información que cada proceso intercambia para este algoritmo, es una solución que guarda los siguientes datos: número de rutas y distancia recorrida, así como algunos parámetros que se requieren para su evolución. También se utilizan las soluciones no dominadas para conocer qué soluciones tienen la distancia más corta o el menor número de rutas.

Para todos los algoritmos, se realiza lo siguiente: se comparan dos soluciones no dominadas, cada n generaciones y se toman aquellas soluciones que mejoran en algún objetivo, la menor distancia recorrida o el menor número de rutas. De esta manera guardamos dos soluciones, una que contiene la distancia más corta pero que no mejora el número de rutas y otra que tiene menor número de rutas, pero quizás no mejora en distancia.

Todos los procesos evolucionan de forma paralela, mediante la comunicación entre varios procesos que trabajan de forma síncrona y asíncrona, por medio del intercambio de información se permite conocer las soluciones no dominadas de cada proceso participante. Cuando el proceso coordinador obtiene las soluciones de cada proceso esclavo, las evalúa, y las renvía a cada proceso para que formen parte de su población principal, las cuales se someten al proceso evolutivo de forma paralela hasta cierta generación y se vuelve a realizar el mismo procedimiento. Ahora se presentan los cuatro modelos que se emplearon con este algoritmo.

4.3.1 Primer modelo asíncrono paquete completo

En este algoritmo se intenta asemejar al proceso de coevolución, en donde cada proceso coopera intercambiando información con los demás procesos que también están evolucionando, dicha información se utiliza para generar una nueva población con individuos más aptos. De esta manera, cada población trabaja de forma independiente, pero coopera para generar mejores individuos. En la Figura 28, se ilustra la forma es que trabaja este algoritmo. El proceso 0 es el coordinador, el cual no evoluciona, solo recibe información de cada uno de los demás procesos, el orden en que llegan los procesos es de acuerdo a la forma que evolucionan, no de acuerdo a su identificador.

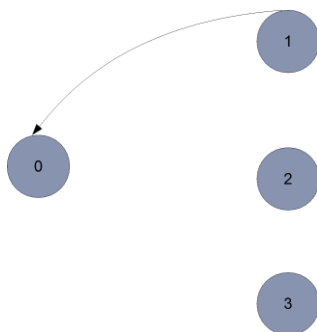


Figura 28. Proceso 0 recibe Datos

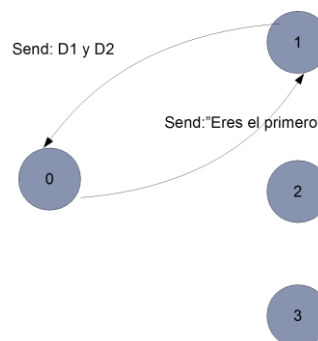


Figura 29. Proceso 0 envía respuesta

En la Figura 29, el proceso 0, inmediatamente envía una respuesta al proceso emisor, indicando si éste fue el primero en enviarle la información, ya que será el primer parámetro a comparar. Si no es el primero, el proceso 0 compara el valor en rutas y distancia que tiene con los datos que recibe, elige la que tiene la mejor solución y la envía nuevamente al que le envió el dato para que éste intercambie la solución recibida con el peor individuo de su población.

Una vez que los procesos reciben un mensaje del proceso 0, continúan evolucionando con la información recibida. Así continúa hasta que todos los procesos intercambien la mejor solución, el pseudocódigo de este procedimiento se encuentra en el anexo (ver anexo algoritmos). En la Figura 30 se muestra la forma en que trabaja este proceso.

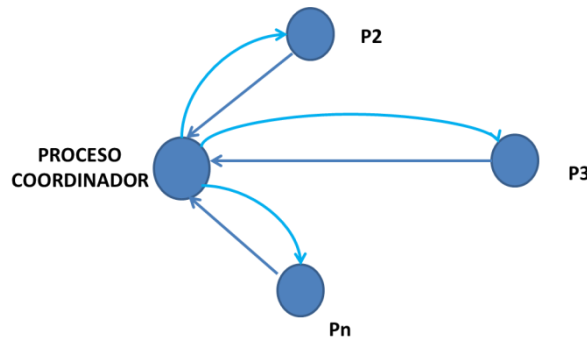


Figura 30. Algoritmo paralelo

Lo que envía cada proceso contiene la información completa para el algoritmo DrGA, es decir, una estructura de datos conformada por el número de rutas y distancia del recorrido, tiempos de espera, capacidad del vehículo, clientes, así como algunos parámetros que se requieren para su evolución.

4.3.2 Segundo modelo asíncrono paquete incompleto

Este algoritmo trabaja de manera similar al anterior, pero con la diferencia que no envía la información completa, sino solo dos datos (distancia y el número de rutas). La forma de trabajo es asíncrona, porque el proceso coordinador, recibe las mejores soluciones de cualquier proceso, no de acuerdo a su número (rango) si no en el orden que llega, el pseudocódigo de este procedimiento se encuentra en el anexo algoritmos.

4.3.3 Tercer modelo síncrono paquete completo

Este modelo se dice que trabaja de manera síncrona, porque la comunicación entre los procesos es en el orden de su identificador, es decir, el proceso coordinador, tiene un identificador 0 , y espera recibir las mejores soluciones del proceso $1, 2 \dots n$. Compara las mejores soluciones y se queda con la mejor, hasta haber recibido de los n procesos participantes, el pseudocódigo de este procedimiento se encuentra en el anexo algoritmos.

Cuando el coordinador tiene la mejor solución, la reenvía a todos los procesos para que la sustituyan por la que tienen y puedan continuar con el proceso de evolución. Así continúa hasta que se terminan las n generaciones. En la Figura 31 se ilustra el proceso de recepción y la Figura 32 el de envío.

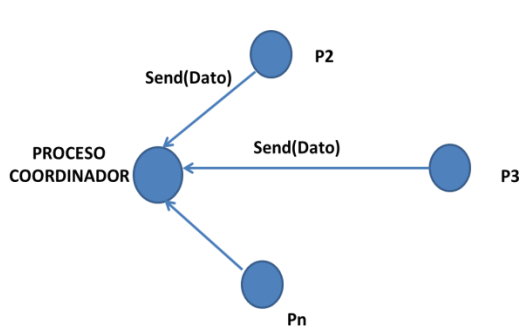


Figura 31. Proceso coordinador recibe de todos

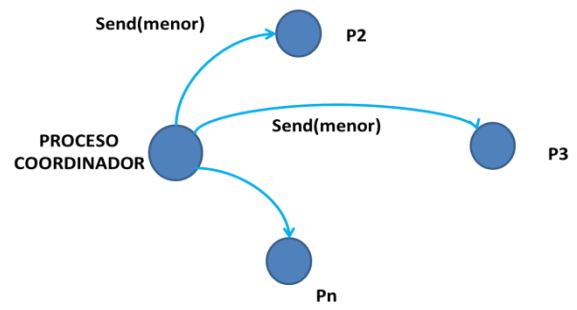


Figura 32. Proceso coordinador envía la mejor solución

4.3.4 Cuarto modelo síncrono paquete incompleto

Este modelo trabaja de la misma manera que el anterior, pero ahora el proceso coordinador solicita únicamente una solución con dos datos; el número de rutas y distancia, el orden de la comunicación de los procesos es de acuerdo a su identificador (rank). El trabajo del coordinador, es verificar si un proceso tiene una mejor solución, el coordinador le pide que le envíe la solución completa, el pseudocódigo de este procedimiento se encuentra en el anexo algoritmos.

Capítulo 5

Resultados

En este Capítulo analizaremos los casos de prueba para la experimentación. Se utilizaron casos de prueba del VRPTW diseñados por Solomon [25]. Los datos que se incluyen en estos casos de prueba son la capacidad del vehículo y una lista de los clientes, donde se proporciona la siguiente información: el identificador del cliente, su ubicación geográfica, su demanda total, tiempo de inicio y fin de la ventana de tiempo, y el tiempo de servicio.

Existen tres conjuntos de casos de prueba, los cuales consideran 25, 50 y 100 clientes, respectivamente. Cada conjunto contiene 56 instancias, que por su ventana de tiempo y su distribución geográfica se identifican por la siguiente notación: C1 (nueve casos de prueba), C2 (ocho casos de prueba), R1 (doce casos de prueba), R2 (once casos de prueba), RC1 (ocho casos de prueba), RC2 (ocho casos de prueba). Tomando en cuenta lo anterior, se dividen en tres grupos, que se describen a continuación.

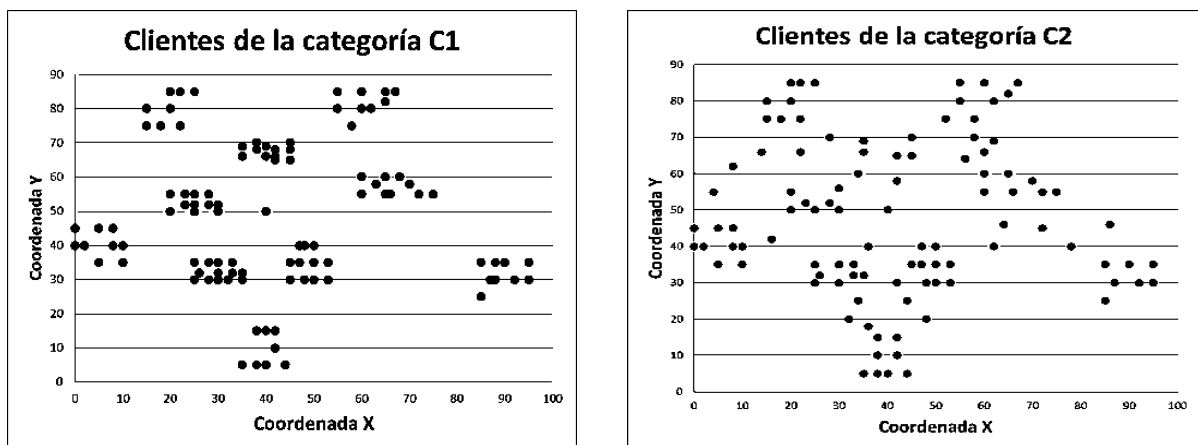


Figura 33. C1 y C2

- a) C1 y C2. En estas categorías los clientes se encuentran aglomerados. La Figura 33 muestra un ejemplo de la distribución de los clientes para estas categorías.
- b) R1 y R2. En estas categorías los clientes se encuentran distribuidos de forma aleatoria y con distribución uniforme. La Figura 34 muestra un ejemplo de la distribución de los clientes de estas categorías.

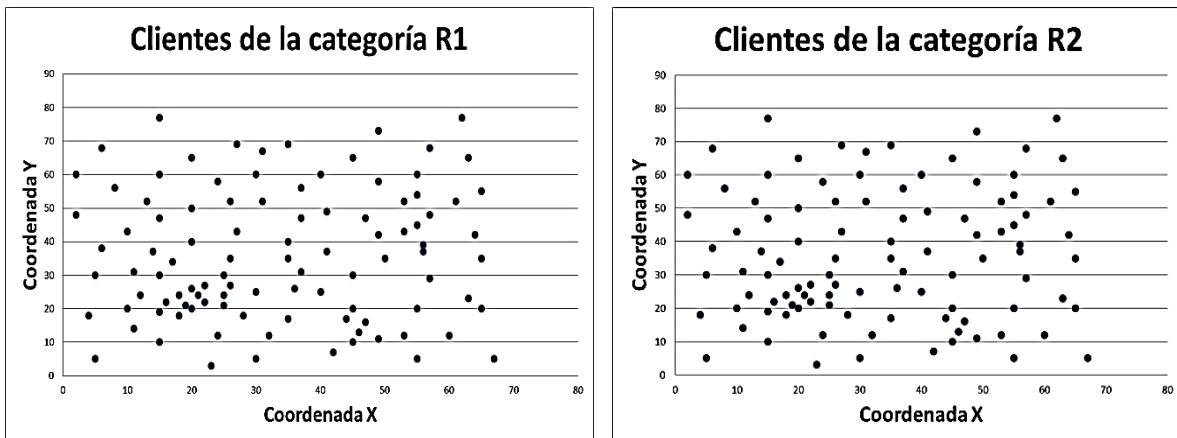


Figura 34. R1 y R2

- c) RC1 y RC2. Son una combinación de clientes ubicados aleatoriamente y por aglomerados. En la Figura 35 se muestra un ejemplo de estas categorías.

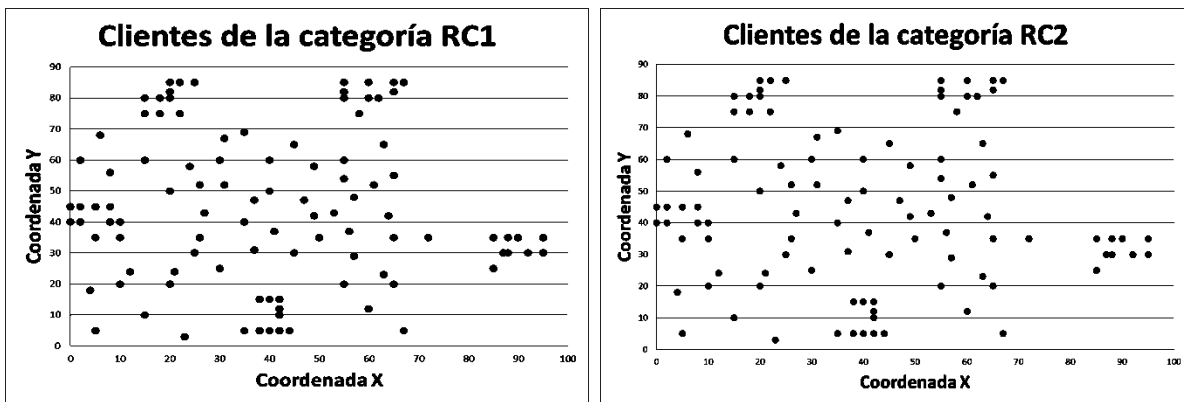


Figura 35. RC1 y RC2

Los conjuntos de casos de prueba R1, C1, y RC1 se diferencian de los demás por tener un horizonte de planificación corto, es decir, tiene ventanas de tiempo pequeñas. En contraste, los conjuntos R2, C2 y RC2 tienen un horizonte largo. Esta característica, junto con la capacidad del vehículo, permite que muchos clientes puedan ser atendidos por el mismo vehículo.

5.1 Resultados de la implementación mediante JGAP

Para realizar las pruebas de este algoritmo secuencial, se hace uso de los casos de prueba para 25 clientes, con una tasa de mutación del 15%, en proporción al número de cromosomas. El tamaño de la población fue 150 y las generaciones permitidas fueron 500. En la Tabla 1 se observa el resumen de los resultados obtenidos para los casos de prueba utilizados que fueron C101, C201 y RC101. El algoritmo se ejecutó diez veces y se obtuvo el promedio de la distancia y del número de rutas.

Una vez realizadas las pruebas, se comparan con los mejores resultados de [25]. La Tabla 1 se explica de la siguiente manera: en la primera columna se presenta el nombre del caso de prueba, las siguientes dos columnas son las mejores soluciones, luego los resultados del algoritmo hecho en JGAP y por último la diferencia porcentual en distancia y rutas.

Problema	Distancia (Mejor Solución)	Rutas (Mejor Solución)	Distancia JGAP	Rutas JGAP	%Dif. Distancia	%Dif Rutas
C101	191.3	3	283.21	6.3	48	110
C201	214.7	2	561.94	12.4	162	520
RC101	461.1	4	741.95	14.3	61	258

Tabla 1. Comparación de la mejor solución [25] y JGAP

En los resultados obtenidos se muestran que los objetivos (distancia y rutas) no están cercanos a las mejores soluciones conocidas, entonces se requiere explorar mejores soluciones. En este caso, se tiene que cambiar la configuración de los operadores genéticos para mejorar la diversidad de las soluciones, así como analizar las soluciones no dominadas.

De acuerdo a lo anterior y debido a la dificultad que se presenta replantear este algoritmo, se ve la necesidad de utilizar otra alternativa, que nos ofrezca soluciones cercanas al óptimo. Por ello se utiliza DrGA, el cual se describió anteriormente.

A continuación, se describen los resultados de las pruebas realizadas con este algoritmo. Dichas pruebas se realizaron en Pacífico, que es un clúster ubicado en el

Laboratorio de Sistemas Distribuidos de la UAM Iztapalapa, el cual cuenta con una arquitectura heterogénea y 10 nodos con cuatro procesadores cada uno (es decir 40 unidades de procesamiento).

5.2 Resultados de la implementación del DrGA

Para analizar el algoritmo se utilizaron los casos de prueba de Solomon [25], C101, C201, con 25 clientes y RC101 para 100 clientes. Para ejecutar el algoritmo se establecieron los siguientes parámetros, tamaño de la población 200, tasa de cruce 0.8, tasa de mutación (inserción 0.3, swap 0.3, inversión 0.2, displace 0.15, split 0.05). Este algoritmo se ejecutó diez veces secuencialmente, para obtener un parámetro preciso del valor de la solución, y solo se tomaron las mejores soluciones que se muestran en la Tabla 2.

Problema	Distancia (Mejor Solución)	Rutas	Distancia DrGA	Rutas DrGA	% Diferencia Distancia	% Diferencia Rutas
C101	191.3	3	191.81	3	0.3%	0.0%
C201	214.7	2	215.54	2	0.4%	0.0%
RC101	1619.8	15	1635.11	15	0.9%	0.0%

Tabla 2. Resultados del Algoritmo DrGA

Los resultados obtenidos mejoran considerablemente a los que se obtuvieron mediante el algoritmo JGAP. La diferencia porcentual, nos indica lo lejana que está a la mejor solución. Ahora se puede implementar el diseño en paralelo para seguir buscando mejorar los resultados presentados con este algoritmo.

Se realizaron cuatro variantes en paralelo del algoritmo DrGA [30]. Una característica especial de estos modelos ya descritos en el capítulo anterior, es la forma en que trabajan (síncrona y asíncrona) y la comunicación para el envío y recepción de información que serán parte del proceso de evolución de estos algoritmos, para ofrecer las mejores soluciones.

Las variantes en paralelo, es la propuesta presentada en el capítulo anterior, los cuales se comparan para conocer la forma en que trabajan de acuerdo a su diseño. Las pruebas se ejecutaron en un clúster homogéneo llamado Aitzalooa, ubicado en el Laboratorio de Supercómputo y Visualización en Paralelo de la UAM Iztapalapa, el cual tiene 270 nodos (2160 unidades de procesamiento), 2 procesadores por nodo (Intel Quad_core Xeon 3Ghz, 16 GB RAM) y sistema operativo Linux Centos 5.2.

5.3 Resultados de la implementación del DrGAP

Con los cuatro algoritmos paralelos, se realizaron pruebas para cada conjunto de instancias mencionados anteriormente, es decir 56 casos de prueba para cada instancia (ver anexo tabla de resultados). Para la ejecución de los algoritmos, se utilizaron 32 unidades de procesamiento en Aitzalooa. La siguiente tabla 3, 4 y 5 de este apartado, muestran los resultados obtenidos para estos algoritmos, y se describe de la siguiente manera: las siglas de la primera columna, corresponden a las siglas de los algoritmos paralelos que se describen en el Capítulo 4. Las segunda y tercera columnas indican el número de rutas y la distancia de las mejores soluciones conocidas, cada par de columnas siguientes, corresponden al número de rutas y distancias de la solución que tiene la menor distancia y la solución que mejora en rutas, estas soluciones se toman del frente no dominado.

Además, se incluyen dos columnas que indican un porcentaje de diferencia en rutas y distancia, las cuales se obtienen del promedio de los casos de prueba. Para obtener esta diferencia se tomaron los valores de la mejor solución conocida y aquella donde la distancia es menor, y para obtener la diferencia en rutas se toma la solución donde el número de rutas es menor. La columna %Dif (D), indica la diferencia en distancia que hay entre la mejor solución y la de los algoritmos que se diseñaron, y la columna %Dif(R) la diferencia en rutas.

Se ejecutaron los cuatro algoritmos evolutivos en paralelo, haciendo uso de 32 procesos, con los mismos parámetros utilizados para el DrGA. Se obtuvo el promedio de las mejores soluciones de los casos de prueba de 25 clientes, y el promedio de cada algoritmo

DrGAP para los mismos casos de Prueba. En la tabla 3 se observan los resultados.

Algoritmo	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C1								
Apc	3.0	190.6	191.1	3.0	191.1	3.0	0.26%	0.00%
Api	3.0	190.6	191.1	3.0	191.1	3.0	0.26%	0.00%
Spc	3.0	190.6	191.1	3.0	191.1	3.0	0.26%	0.00%
Spi	3.0	190.6	191.1	3.0	191.1	3.0	0.26%	0.00%
C2								
Apc	2.0	214.5	215.5	1.9	245.4	1.1	0.51%	43.75%
Api	2.0	214.5	215.3	1.9	245.4	1.1	0.39%	43.75%
Spc	2.0	214.5	215.3	1.9	245.4	1.1	0.39%	6.25%
Spi	2.0	214.5	216.1	1.8	243.5	1.3	0.76%	12.50%
R1								
Apc	4.9	463.4	466.1	5.1	471.2	4.8	0.59%	3.39%
Api	4.9	463.4	464.4	5.1	469.6	4.8	0.23%	3.39%
Spc	4.9	463.4	464.4	5.1	469.6	4.8	0.23%	3.39%
Spi	4.9	463.4	464.5	5.1	469.6	4.8	0.24%	3.39%
R2								
Apc	2.7	382.1	383.8	2.7	428.2	1.3	0.45%	53.33%
Api	2.7	382.1	383.8	2.6	428.2	1.3	0.44%	53.33%
Spc	2.7	382.1	383.4	2.6	427.9	1.3	0.34%	3.33%
Spi	2.7	382.1	386.2	2.5	397.2	1.3	1.06%	6.67%
RC1								
Apc	3.3	347.0	351.1	3.3	351.1	3.3	1.19%	0.00%
Api	3.3	347.0	351.1	3.3	351.1	3.3	1.19%	0.00%
Spc	3.3	347.0	351.1	3.3	351.1	3.3	1.19%	0.00%
Spi	3.3	347.0	351.1	3.3	351.1	3.3	1.19%	0.00%
RC2								
Apc	2.9	319.3	319.8	2.8	422.5	1.3	0.15%	56.52%
Api	2.9	319.3	320.1	2.9	425.1	1.3	0.25%	56.52%
Spc	2.9	319.3	320.1	2.9	424.0	1.3	0.25%	0.00%
Spi	2.9	319.3	323.5	2.6	403.2	1.3	1.33%	8.70%

Tabla 3. Resultados de los Algoritmos en paralelo para 25 clientes

Las variantes del algoritmo DrGAP, apc(asíncrono paquete completo), api(asíncrono paquete incompleto), spc (síncrono paquete completo), y spi(síncrono paquete incompleto), marcan una diferencia porcentual con respecto al promedio de la mejor solución en distancia del .5%, es decir que no alcanza a igualar a las mejores soluciones conocidas (MSD). Pero los valores de la columna R2, marcan una diferencia en algunos casos mayor del 50%, a las mejores soluciones. Estos resultados nos muestran que la segunda solución es mejor en número de rutas, se utilizan menos de rutas, pero es mayor la distancia del recorrido.

Los resultados de la Tabla 3, nos muestra el promedio de todas las instancias de los 56 casos de prueba, para 25 clientes. Se observa que, en algunos casos, ofrece una solución

buena en rutas en comparación a la mejor solución, sin embargo, en distancia no. Esto nos muestra que cuando se trata de problemas multiobjetivo, podemos encontrar soluciones que mejoren un objetivo, pero no garantiza que ofrezca una buena solución en el otro. Por ello se queda a consideración cual es la solución que se puede tomar; la que mejora en distancias, pero no en rutas o viceversa.

En la tabla 4 se observan los resultados de las cuatro variantes del algoritmo DrGAP, para los casos de prueba de 50 clientes. Los resultados de las instancias RC2 muestran una mejoría en las soluciones en distancia con una diferencia porcentual de 5% y en rutas 40% con respecto a los resultados de las mejores soluciones conocidas.

Algoritmo	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C1								
Apc	5.0	361.7	361.0	5.0	361.0	5.0	0.20%	0.00%
Api	5.0	361.7	361.5	5.0	361.5	5.0	0.04%	0.00%
Spc	5.0	361.7	361.4	5.0	361.4	5.0	0.07%	0.00%
Spi	5.0	361.7	361.6	5.0	365.4	5.0	0.03%	0.00%
C2								
Apc	2.8	357.5	359.1	2.8	401.7	2.0	0.45%	27.27%
Api	2.8	357.5	359.1	2.8	408.3	2.0	0.45%	27.27%
Spc	2.8	357.5	359.1	2.8	399.1	2.0	0.45%	27.27%
Spi	2.8	357.5	359.1	2.8	379.1	2.0	0.45%	27.27%
R1								
Apc	7.75	766.13	772.44	8.00	782.81	7.58	0.82%	2.15%
Api	7.75	766.13	773.40	7.92	788.39	7.58	0.95%	2.15%
Spc	7.75	766.13	771.27	8.00	787.55	7.50	0.67%	3.23%
Spi	7.75	766.13	773.56	7.75	784.66	7.42	0.97%	4.30%
R2								
Apc	3.4	518.8	629.0	3.2	669.1	2.0	21.26%	40.54%
Api	3.4	518.8	628.2	3.2	676.5	2.0	21.10%	40.54%
Spc	3.4	518.8	629.0	3.2	669.1	2.0	21.26%	40.54%
Spi	3.4	518.8	640.9	2.8	680.7	2.1	23.55%	37.84%
RC1								
Apc	6.5	730.3	731.9	6.5	731.9	6.5	0.21%	0.00%
Api	6.5	730.3	731.7	6.5	731.7	6.5	0.18%	0.00%
Spc	6.5	730.3	731.9	6.5	731.9	6.5	0.21%	0.00%
Spi	6.5	730.3	731.9	6.5	731.9	6.5	0.21%	0.00%
RC2								
Apc	4.3	615.6	579.2	4.3	683.8	2.5	5.91%	41.18%
Api	4.3	615.6	573.5	4.3	694.2	2.4	6.85%	44.12%
Spc	4.3	615.6	574.0	4.3	692.5	2.3	6.76%	47.06%
Spi	4.3	615.6	579.9	3.9	699.9	2.3	5.80%	47.06%

Tabla 4. Resultados de los Algoritmos en paralelo para 50 clientes

Los resultados para los problemas con 100 clientes se muestran en la Tabla 5. Los

valores que arrojan los algoritmos paralelos, no supera a las mejores soluciones conocidas, pero la segunda solución en rutas ofrece buenos resultados comparados a las mejores conocidas, ya que la diferencia porcentual va del 1% al 40%.

Algoritmo	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C1								
Apc	10.00	826.70	826.1	10.0	826.1	10.0	0.07%	0.00%
Api	10.00	826.70	828.5	10.0	836.9	9.9	0.21%	1.11%
Spc	10.00	826.70	826.4	10.0	826.4	9.9	0.04%	1.11%
Spi	10.00	826.70	827.8	10.0	826.7	10.0	0.13%	0.00%
C2								
Apc	3.00	587.38	589.9	3.0	589.9	3.0	0.42%	0.00%
Api	3.00	587.38	589.9	3.0	589.9	3.0	0.42%	0.00%
Spc	3.00	587.38	592.1	3.0	592.1	3.0	0.80%	0.00%
Spi	3.00	587.38	590.6	3.0	590.6	3.0	0.56%	0.00%
R1								
Apc	11.58	1016.88	1207.1	12.9	1214.7	12.8	18.71%	10.07%
Api	11.58	1016.88	1230.5	13.2	1239.3	12.6	21.01%	8.63%
Spc	11.58	1016.88	1213.0	13.1	1227.3	12.6	19.28%	8.63%
Spi	11.58	1016.88	1248.4	13.1	1253.8	12.7	22.76%	9.35%
R2								
Apc	4.55	893.03	914.7	4.2	968.9	3.1	2.4%	32.07%
Api	4.55	893.03	940.1	4.5	992.6	3.1	5.27%	32.07%
Spc	4.55	893.03	924.6	3.8	960.2	3.1	3.54%	32.07%
Spi	4.55	893.03	943.1	3.7	982.4	3.1	5.60%	32.07%
RC1								
Apc	13.00	1384.95	1353.1	12.6	1366.1	12.1	2.30%	6.73%
Api	13.00	1384.95	1361.9	12.9	1384.4	12.1	1.67%	6.73%
Spc	13.00	1384.95	1361.1	12.6	1369.0	12.3	1.72%	5.77%
Spi	13.00	1384.95	1388.1	12.8	1421.9	12.8	0.23%	1.92%
RC2								
Apc	5.63	1025.31	1040.7	5.4	1184.2	3.5	1.50%	37.83%
Api	5.63	1025.31	1046.8	5.1	1176.3	3.4	2.09%	40.05%
Spc	5.63	1025.31	1078.2	4.6	1155.2	3.4	5.16%	40.05%
Spi	5.63	1025.31	1074.7	4.8	1210.5	3.4	4.82%	40.05%

Tabla 5. Resultados de los Algoritmos en paralelo para 100clientes

También se realiza un análisis de las variantes del DrGA en paralelo con relación al secuencial de García-Nájera [30], donde se muestran los resultados del algoritmo DrGA, para las instancias con 100 clientes. Lo anterior con la finalidad de verificar si la versión paralela mejora las soluciones del DrGA. En la tabla 6 se muestran los resultados, donde la primera columna corresponde al nombre de la instancia, la segunda y tercera los resultados del algoritmo DrGA, y en los pares siguientes los resultados de una de las variantes del algoritmo DrGA paralelo.

Instancia	Rutas	Distancia	R1	D1	R2	D2	%D	%R
C1	10.00	857.49	10.00	826.12	10.00	826.12	3.66%	0%
C2	3.00	591.03	3.00	589.86	3.00	589.86	0.20%	0%
R1	13.17	1248.55	12.92	1207.15	12.75	1214.69	3.32%	3%
R2	3.64	954.47	4.18	914.70	3.09	968.90	4.17%	15%
RC1	13.00	1413.95	12.63	1353.11	12.13	1366.11	4.30%	7%
RC2	4.38	1099.09	5.38	1040.69	3.50	1184.16	5.31%	20%

Tabla 6. DrGA y DrGA paralelo

Los resultados que se comparan con el DrGA, es el algoritmo asíncrono paquete completo, en la tabla 6 se observa que las soluciones en distancia tienen una diferencia porcentual del 3% con respecto a las del DrGA. En cuanto a rutas, la diferencia porcentual va del 3% al 15%, Estos resultados indican que el trabajo del algoritmo en paralelo ofrece resultados eficientes, aunque no mejora a las mejores soluciones conocidas, salvo los casos de RC2 para 50 clientes, como se observó en las tablas anteriores. Se pueden observar todos los resultados de cada algoritmo en paralelo, para cada caso de prueba e instancias correspondientes, en las tablas del anexo (ver anexo tabla de resultados).

A continuación, se presentan dos gráficas del tiempo de ejecución de cada variante del algoritmo DrGAP, para las instancias C2 y R2 para 25 clientes. El propósito es el análisis de la eficiencia en tiempo computacional de estos algoritmos paralelos, para considerar la eficacia del trabajo en paralelo de cada proceso.

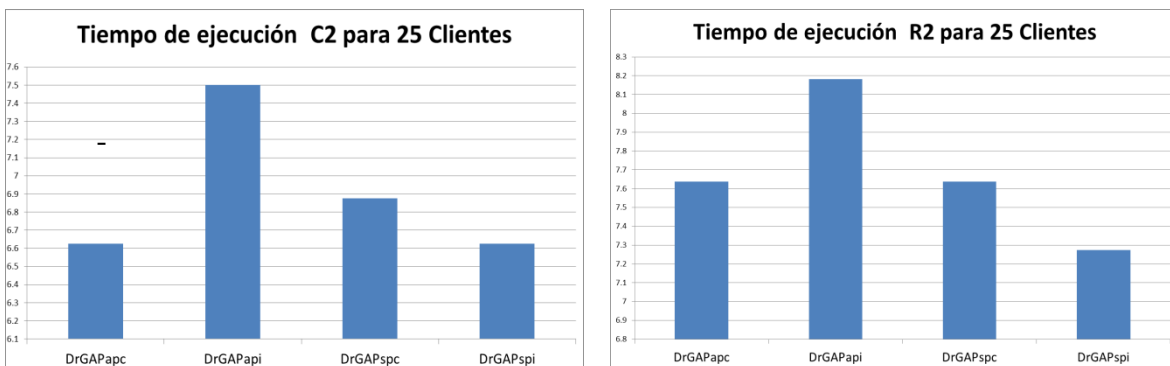


Figura 36. Tiempo de ejecución del algoritmo en paralelo

La figura 36 muestra dos gráficas que presentan el promedio del tiempo en segundos, al ejecutar los algoritmos en paralelo con las instancias C2 y R2 para 25 clientes. Para la instancia C2, el algoritmo asíncrono paquete incompleto, presenta mayor costo computacional con relación a los otros, es decir cuatro décimas de segundos de diferencia.

Para la instancia R2, el tiempo de ejecución del algoritmo síncrono paquete incompleto es el menor, con una diferencia de ocho décimas de segundos en relación al mayor.

Esto nos demuestra que el diseño de cada variante del algoritmo DrGAP, no incrementa de forma considerable el costo computacional de los algoritmos en paralelo. Sin embargo, si se compara con el costo computacional de la versión del DrGA al ejecutarlo treinta dos veces, que es el número de los procesos que se utilizaron para la ejecución de los algoritmos paralelos, es notable la gran diferencia.

Capítulo 6

Conclusiones

En este trabajo se realizó un análisis del VRPTW, que es un problema cuyo objetivo es minimizar el costo de abastecer a un conjunto de clientes, dispersos geográficamente. Este problema es de gran interés por la complejidad que presenta, así como los beneficios que se obtienen en problemas de logística y transporte.

Se realizó una descripción del VRP, sus variantes, para introducir la definición formal del VRPTW, además, se analizó porque es un problema multiobjetivo, y se presentaron algunas herramientas utilizadas para su solución.

También se efectuó una investigación del cómputo paralelo, cuyo objetivo principal, es la reducción de tiempo de ejecución y la optimización de los procesos. Se analizaron las diferentes arquitecturas paralelas, para entender la forma en que trabaja. Asimismo se presentaron las características del MPI, una herramienta de cómputo paralelo, que se utilizó para el diseño del algoritmo DrGA paralelo, que son las propuestas presentadas en esta investigación.

El objetivo general de esta investigación, se mencionó en el capítulo uno, que fue diseñar un algoritmo genético usando cómputo paralelo que resuelva el problema VRPTW, obteniendo buenos resultados comparables con los mejores publicados en un tiempo de respuesta aceptable.

Para resolver el problema VRPTW paralelo, se realizaron dos fases para su diseño, como primera fase experimental, fue el diseño de un algoritmo utilizando un framework en java, que consiste en una librería de métodos genéticos para la implementación de problemas multi-objetivo. El algoritmo hecho en JGAP secuencial se comprobó con algunos de los 56

casos de prueba para el VRPTW, y se compararon con los mejores resultados publicados.

Los resultados obtenidos fueron soluciones muy lejanas al óptimo. La estructura del framework, así como los resultados obtenidos no permitieron continuar con la implementación en paralelo. Tomando en cuenta esto, se concluye que si existen muchas herramientas de AG útiles, pero dificulta la modificación de su diseño para la incorporación de otros mecanismos de programación.

Como segunda fase se utilizó el algoritmo DrGA [30], la cual ofreció buenas soluciones, y además nos facilitó el diseño en paralelo al utilizar las librerías MPI. Se realizaron cuatro variantes del DrGA en paralelo, las cuales se diseñaron para que cada proceso evolucione al mismo tiempo, pero a su vez puedan comunicarse en determinado tiempo, para intercambiar la mejor solución y de esta manera mejorar la población que se tiene hasta ese momento, para posteriormente seguir evolucionando hasta terminar. Los datos que envían y reciben son solo una parte, o la información completa de una solución del VRPTW.

Se presentaron los resultados de la experimentación de estos algoritmos, para esto se tomaron en cuenta los 56 casos de prueba diseñados por Solomon [25]. Estos resultados mostraron una aproximación a las mejores soluciones publicadas. Además, se incluye dos soluciones del VRPTW, una donde se tiene la distancia más corta, y otra solución donde se utilizan menos rutas pero la distancia total recorrida es mayor, ambas soluciones se obtienen de los valores que se guardan del frente no dominado. Los resultados se incluyen al final de este documento en el anexo tablas de resultados, donde se comparan los 56 casos de prueba para cada categoría, los cuales se encuentran en la parte de anexos (ver anexo tabla de resultados).

Además de los cuatro modelos propuestos en paralelo se concluye que, su desempeño en tiempo de ejecución, se ve afectado por su modelo asíncrono o síncrono, pero no de forma considerable, es decir, es mínima la diferencia en cada uno de ellos. En cuanto a las soluciones que se obtuvieron con cada uno de ellos, se concluye que cada modelo presento buenas soluciones, pero en especial el modelo asíncrono paquete completo,

presento en algunas pruebas mayores de 50 y 100 clientes, mejores soluciones en comparación con los otros tres, sin embargo los resultados arrojados, también demostraron que los otros modelos también presentaron buenas soluciones.

De acuerdo a los resultados obtenidos, el trabajo en paralelo permitió encontrar buenas soluciones, las cuales son comparables en términos de calidad de la solución y tiempo computacional respecto al desempeño de la versión secuencial. Además, se concluye que en un problema multiobjetivo podemos encontrar soluciones que mejoren un objetivo pero no garantizan buenos resultados en el otro, y queda a consideración cual es la solución que se puede tomar; la que mejora en distancias pero no en rutas o viceversa.

Los resultados obtenidos sugieren que el diseño en paralelo del algoritmo propuesto podría ser aplicado a otros problemas similares. Algunos trabajos futuros propuestos son los siguientes: modificar el diseño paralelo en un modelo en islas, las cuales evolucionen de forma independiente y en determinadas generaciones intercambiar las soluciones, similar al modelo de coevolución. Considerar la implementación de este algoritmo con alguna instancia real, de alguna empresa encargada de distribución de algún bien o servicio.

Referencias

- [1] S. M. M., “Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints,” *Operations Research*, vol. 35, pp. 254 - 265, 1987.
- [2] D. Vigo, P. Toth, “An Overview of Vehicle Routing Problems,” *SIAM*, pp. 1–26, 2002.
- [3] Y. Medina, “Capacitance vehicle routing problem with soft time windows (CVPSTW),” *European Journal of Marketing*, pp. 1-9, Abril 2000.
- [4] G. B. Dantzig, J. H. Ramser, “The truck dispatching problem,” *Management Science*, Vol. 6, p. 80–91, 1959.
- [5] G. Clarke and J.W. Wright, “Scheduling of vehicles from a central depot to a number of delivery points,” *Operations Research*, vol. 12, p. 568–581, 1964.
- [6] J.J. Novelle, J. Campos, A. Formella, P. Carrión and J.F. Gálvez, “Adaptación y optimización del algoritmo clarke&wright a un problema de distribución multirrestringido,” in *IX Congreso Gallego de Estadística e Investigación de Operaciones Ourense, España, November 2000*.
- [7] E. El-Gharably, S. El-Kilany, and E. El-Sayed “Optimization Using Simulation of the Vehicle Routing Problem,” *International Science Index, Industrial and Manufacturing Engineering*, vol. 7, 2013.
- [8] C. Coello, G. Lamont, D. Van Veldhuizen, “Evolutionary Algorithms for Solving Multi-Objective Problems,” *New York: Springer*, 2007.
- [9] S. Nasmachnow, “Parallel multiobjective evolutionary algorithms for batch scheduling in heterogeneous computing and grid systems,” *Computational Optimization and Applications an International Journal*, vol. 55, no. 2, pp. 515–544, June 2013.
- [10] A. Fernández, C. Gil, R.Baños, M.G.Montoya , “A parallel multi-objective algorithm for two-dimensional bin packing with rotations and load balancing,” *Computational*

Optimization and Applications an International Journal, vol. 40, no. 13, pp. 5169–5180, October 2013.

- [11] A.Konak, D.W. Coit, A.E.Smith, “Multi-objective optimization using genetic algorithms: A tutorial,” Reliability Engineering & System Safety, Vol.91, Issue 9, pp. 992-1007 . September 2006,
- [12] J. Francois, and C. G. Laporte, “A unified tabu search heuristic for vehicle routing problems with time windows,” in Journal of the Operational Research Society, Canada, 2011.
- [13] Y. Bin, Y. Zhong-Zhen, and Y. Baozhen, “An improved ant colony optimization for vehicle routing problem,” European Journal of Operational Research, vol. 196, Pages 171-176, July 2009.
- [14] M. Potter, and K. De Jong, “A Cooperative Coevolutionary Approach to Function Optimization,” in International Conference on Evolutionary Computation The Third Conference on Parallel Problem Solving from Nature Jerusalem, Israel, October 9 – 14, 1994.
- [15] M. M. Reyes, and C. A. Coello, “Un Algoritmo Coevolutivo para Optimización Multiobjetivo Basado en Clustering,” in Congreso de Computación Evolutiva, Aguascalientes, 2005.
- [16] L. Polat, and A. Acan, “Cooperative Coevolutionary Algorithms for Fuzzy Vehicular,” in IEEE Congress on Evolutionary Computation, 2007.
- [17] T. Rouber, G. Rüniger, “Parallel Programing: for Multicore and Cluster Systems,” Springer, 2010.
- [18] T. Wittwer, An Introduccion to Parallel Programing, C.A, VSSD, Verschijningsjaar: 2a. Ed. 2009.
- [19] M. Flynn, “Some computer organizations and their efectiveness,” IEEE Transactions on Computers, vol 21, pp. 948-960, September 1972.
- [20] M. Mano, Arquitectura de Computadoras, Mexico: Prentice Hall Hispanoamericana s.a.
- [21] MPICH, <http://www.mpich.org/downloads/>, Last accessed on December 20, 2017.
- [22] LAM/MPI Parallel Computing, <http://www.lam-mpi.org>, Last accessed on January 13, 2018.

- [23] Open MPI: Open Source High Performance Computing, <http://www.open-mpi.org/>, Last accessed on December 5, 2017.
- [24] A. Garcia, and J. A. Bullinaria, "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows," *Computers & Operations Research*, Vol. 38, no. 1, January 2011, Pages 287-300.
- [25] Marius M. Solomon, <http://w.cba.neu.edu/~msolomon/problems.htm>, Last accessed on May 30, 2018.
- [26] GALib, <http://lancet.mit.edu/ga/>, Last accessed on June 24, 2017.
- [27] A. Dolan, *Artificial life and other experiments*, <http://www.aridolan.com/default.aspx>, Last accessed on January 13, 2018.
- [28] JAGA, <http://www.jaga.org/about.html>, Last accessed on January 20, 2017.
- [29] JGAP, <http://jgap.sourceforge.net/>, Last accessed on June 20, 2017.
- [30] A. Garcia, and J. A. Bullinaria, "A multi-Objective Density Restricted Generic Algorithm the vehicle Routing problem with time windows," in *Workshop on Computational Intelligence*, 2009.
- [31] A. Garcia, and J. A. Bullinaria, "Bi-objective Optimization for the Vehicle Routing Problem with Time Windows: Using Route Similarity to Enhance Performance," in *International Conference on Evolutionary Multi-Criterion Optimization EMO 2009: Evolutionary Multi-Criterion Optimization*, Nantes, France, pp. 275-289.
- [32] M. Blocho, and Z. J. Czech, "A Parallel Algorithm for Minimizing the Number of Routes in the Vehicle Routing Problem with Time Windows," in *International Conference on Parallel Processing and Applied Mathematics*, Torun, Poland, 2011, pp. 255-265.
- [33] J. Nalepa, and Z. J. Zech, "Adaptive threadsco-operation schemesina parallel heuristic algorithm for the vehiclerouting problema withtime Windows," in *Theoretical and Applied Informatics*, Gliwice, Poland, 2012, pp. 44-100.
- [34] M. Blocho, and Z. J. Czech, "A Parallel EAX-based Algorithm for Minimizing the Number of Routes in the Vehicle Routing Problem with Time Windows," in *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, Liverpool, UK, 2012.

- [35] F.Berdardes, R. Enayatifar, H. Javedani, F. Gadelha, and J. Potvin, “A Cooperative Coevolutionary Algorithm for the Multi-Depot Vehicle Routing Problem,” in *Journal Expert Systems with Applications: An International Journal archive*, Tarrytown, NY, pp. 117-130, January 2016.
- [36] J. Berger, and M. Barkaoui, “A parallel hybrid genetic algorithm for the vehicle routing problem with time Windows,” *Computers & Operations Research*, vol. 31, pp. 2037-2053, October 2004.
- [37] O. M. González, and C.Segura, “A Parallel Memetic Algorithm to Solve the Capacitated Vehicle Routing Problem with Time Windows,” in *Journal of Combinatorial Optimization Problems and Informatics*, Vol. 9, January 2018, pp. 35-45.
- [38] J. Jin, T. Gabriel, and A. Lokketangena, “A cooperative parallel metaheuristic for the capacitated vehicle routing problem,” in *Computers & Operations Research*, vol. 44, April 2014, pp. 33-41.
- [39] A. Debudaj, and R. Rabenseifner, “Nesting OpenMP in MPI to Implement a Hybrid Communication Method of Parallel Simulated Annealing on a Cluster of SMP Nodes,” in *Euro PVM/MPI*, Berlin, Heidelberg, pp.18-25, 2005.
- [40] Z. J. Czech, W. Mikanik, and R. Skinderowicz “Implementing a Parallel Simulated Annealing Algorithm,” in *PAM:Parallel Processing and Applied Mathematics*, Heidelberg, pp 146-15, 2009.
- [41] C. A. García, E. García, and F. Villada, “Implementación del Algoritmo Evolutivo Multi-Objetivo de Frente de Pareto (SPEA) para la Planeación de Sistemas Eléctricos de Distribución incluyendo Huecos de Voltaje,” in *Información tecnológica*, Vol.26, 2015.
- [42] M.A. Cruz, and A. Martinez, “Feasible Initial Population with Genetic Diversity for a Population-Based Algorithm Applied to the Vehicle Routing Problem with Time Windows,” in *Mathematical Problems in Engineering*, Morelos Mexico, 2016, pp.11.
- [43] G. U. Clarke, and J.W. Wright, Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, Vol. 4, 1964, pp. 568–581.
- [44] K. Sörensen, F.Arnold, and C. Palhazi “A critical analysis of the “improved Clarke and Wright savings algorithm,” in *International Transactions in Operational Research*, 2017.

- [45] D. Rosenkrantz, R. E. Stearns and P. M. Lewis, “Analysis of several heuristics for the traveling salesman problem,” in SIAM journal on computing, Vol 6, 1977, pp. 563–581.
- [46] M. Soler, Coevolución, <http://sesbe.org/sites/sesbe.org/files/recursos-sesbe/coevol.pdf>, Last accessed on December 20, 2017.
- [47] R. Baños, J. Ortega, C. Gil, A. Fernández, and F. de Toro, “A Simulated Annealing-based parallel multi-objective approach to vehicle routing problems with time windows,” in Expert Systems with Application, 2013, pp. 1696-1707.
- [48] K.Ghosserri, and S. F. Ghannadpour, “Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm,” in Applied Soft Computing, 2010, pp. 1096 – 1107.

ANEXOS (Algoritmos)

1. Primer modelo DrGAP apc (asíncrono paquete Completo)

```
1. Si rank == 0 //proceso coordinador
2.     al recibir SOLUCION1 desde proceso i efectúa
3.         si proceso _i es primero
4.             menor1= SOLUCION1
5.             Send("Primero", tag=1)
6.         Si no
7.             Si (SOLUCION1.distancia < menor1.distancia)
8.                 menor1= SOLUCION1
9.                 Send("menor", tag=2) //tiene la mejor solución
10.            Si (SOLUCION1== menor1)
11.                Send("Iguales", tag=3)
12.            Si no
13.                Send(menor1,tag=4) //envía la mejor solución
14.    al recibir SOLUCION2 desde proceso i efectúa
15.        si proceso _i es primero en enviar
16.            Menor2= SOLUCION2
17.            Send("Primero", tag=1)
18.        Si no
19.            Si (SOLUCION2.rutas < menor2.rutas)
20.                Menor2= SOLUCION2
21.                Send("menor", tag=2)
22.            Si (SOLUCION2== menor2)
23.                Send("Iguales", tag=3)
24.            Si no
25.                Send("menor",tag=5)
26.
27. Si rank!=0
28.     Obtener SOLUCION1 y SOLUCION2
29.     Send(SOLUCION1,)
30.     Send(SOLUCION2,)
31.     En caso de recibir tag
32.         tag=1 Recv("primero")
33.         tag=2 Rev("menor")
34.         tag=3 Rev("iguales")
35.         tag=4 Rev(menor1)
36.         Reemplazar PEOR_SOLUCION1 por menor1
37.         tag=5 Rev(menor2)
38.         Reemplazar PEOR_SOLUCION2 por menor2
```

2. Segundo modelo DrGAP api(asíncrono paquete incompleto)

```
1. Si rank == 0 //proceso coordinador
2.     al recibir Datos desde proceso i efectúa
3.         d1= Datos[0], d2= Datos[1], r1=Dato [2], r2= Datos [3],
4.     si proceso _i es el primero
5.         Send("primero", tag=1)
6.         Recv(COMPLETO1)
7.         Menor1= COMPLETO1
8.         Recv(COMPLETO2)
9.         Menor= COMPLETO2
10.    si no
11.        Si (d1 > menor1.distancia)
12.            Send("mayor", tag=2) , Send(menor)
13.        Si (d1 == menor.distancia)
14.            Send("iguales", tag=3)
15.        Si no
16.            Send("menor", tag=4) Recv(menor1)
17.            Menor1= menor1
18.        Si (r2 < menor2.rutas)
19.            Send("menor", tag=5) Recv(menor2)
20.            Menor2= menor2
21.        Si (r2.distancia== menor2.distancia)
22.            Send("iguales", tag=3)
23.        Si (r2> menor2.distancia)
24.            Send("mayor", tag=6) Send(menor2)
25. Si rank !=0
26.     Obtener SOLUCION1 y SOLUCION2
27.     Datos[0]= SOLUCION1.distancia, Datos[1]= SOLUCION1.rutas
28.     Datos[2]= SOLUCION2.distancia, Datos[2]= SOLUCION2.rutas
29.     Send(Datos)
30.     En caso de recibir tag
31.         tag=1
32.             Send(SOLUCION1), Send(SOLUCION2)
33.         tag=2
34.             Recv (menor)
35.             Reemplazar PEOR_SOLUCION1 por menor
36.         tag=3
37.             Recv("iguales")
38.         tag=4
39.             Send(SOLUCION1)
40.         tag=5
41.             Send(SOLUCION2)
42.         tag=6
43.             Recv(menor2)
44.             Reemplazar PEOR_SOLUCION2 por menor2
```

3. Tercer modelo DrGAP spc(síncrono paquete completo)

```
1.      Si rank == 0 //proceso coordinador
2.          al recibir SOLUCION1 desde proceso i efectúa
3.              si proceso _i es primero
4.                  menor1= SOLUCION1
5.                  proceso_i= es_menor;
6.              si no, si (menor1.distancia < SOLUCION1.distancia)
7.                  menor1=menor1
8.                  si no menor1= SOLUCION1
9.
10.         al recibir SOLUCION2 desde proceso i efectúa
11.             si proceso _i es primero en enviar
12.                 menor2= SOLUCION2
13.                 proceso_i= es_menor2;
14.             si no, si (menor2.distancia < SOLUCION2.distancia)
15.                 menor2=menor2
16.                 si no menor2= SOLUCION2
17.         Para cada proceso_i efectua
18.             Send(menor1,tag=1)
19.             Send(menor2,tag=1)
20.
21.     Si rank !=0
22.         Obtener SOLUCION1 y SOLUCION2
23.         Send(SOLUCION1,)
24.         Send(SOLUCION2,)
25.         Si tag=1
26.             Recibo SOLUCION1 y SOLUCION2
27.             Si SOLUCION1 != mySOLUCION
28.                 Reemplazar PEOR_SOLUCION1 por SOLUCION1
29.             Si SOLUCION2 != mySOLUCION
30.                 Reemplazar PEOR_SOLUCION2 por SOLUCION2
```

4. Cuarto modelo DrGAP spi (síncrono paquete incompleto)

```
1. Si rank == 0 //proceso coordinador
2.     al recibir Datos desde proceso i efectúa
3.         d=Datos[0]
4.         si proceso _i es primero
5.             Send("primero", tag=1)
6.             Recv(SOLUCION1)
7.             menor1= SOLUCION1
8.         si no
9.             si (d1 < menor1.distancia)
10.                Send("menor", tag=1)
11.                Recv(SOLUCION1)
12.                menor1=SOLUCION1
13.         si no menor1= menor1
14.     al recibir Datos2 desde proceso i efectúa
15.         r=Datos2[0]
16.         si proceso _i es primero
17.             Send("primero", tag=2)
18.             Recv(SOLUCION2)
19.             menor2= SOLUCION2
20.         si no
21.             si (r < menor1.distancia)
22.                Send("menor", tag=2)
23.                Recv(SOLUCION2)
24.                menor2=SOLUCION2
25.         si no menor2= menor2
26.     Para cada proceso_i efectua
27.         Send(menor1,tag=5)
28.         Send(menor2,tag=5)
29. Si rank !=0
30.     Obtener SOLUCION1 y SOLUCION2
31.     Datos[0]= SOLUCION1.distancia, Datos[1]= SOLUCION1.rutas
32.     Datos[2]= SOLUCION2.distancia, Datos[2]= SOLUCION2.rutas
33.     En caso de tag
34.         tag=1
35.             Send(SOLUCION1,)
36.         tag=2
37.             Send(SOLUCION2,)
38.         tag=5
39.             Recv(SOLUCION1)
40.             Remplazar PEOR_SOLUCION1 por SOLUCION1
41.             Recv(SOLUCION2)
42.             Remplazar PEOR_SOLUCION2 por SOLUCION2
```

ANEXOS (Tablas de Resultados)

Resultados del algoritmo asíncrono paquete completo para los casos de prueba con 25 clientes.

Caso de Prueba	Mejor solución		Soluciones Obtenidas				Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	3	191.3	191.81	3	191.81	3	0.27%	0.00%
C102	3	190.3	190.74	3	190.74	3	0.23%	0.00%
C103	3	190.3	190.74	3	190.74	3	0.23%	0.00%
C104	3	186.9	187.45	3	187.45	3	0.29%	0.00%
C105	3	191.3	191.81	3	191.81	3	0.27%	0.00%
C106	3	191.3	191.81	3	191.81	3	0.27%	0.00%
C107	3	191.3	191.81	3	191.81	3	0.27%	0.00%
C108	3	191.3	191.81	3	191.81	3	0.27%	0.00%
C109	3	191.3	191.81	3	191.81	3	0.27%	0.00%
PROMEDIO	3	190.58	191.08	3	191.08	3	0.26%	0.00%
C201	2	214.7	215.54	2	215.54	2	0.39%	0.00%
C202	2	214.7	215.54	2	223.31	1	0.39%	50.00%
C203	2	214.7	215.54	2	223.31	1	0.39%	50.00%
C204	2	213.1	215.54	1	213.93	1	1.15%	50.00%
C205	2	214.7	215.54	2	297.45	1	0.39%	50.00%
C206	2	214.7	215.54	2	285.39	1	0.39%	50.00%
C207	2	214.5	215.54	2	274.78	1	0.48%	50.00%
C208	2	214.5	215.54	2	229.84	1	0.48%	50.00%
PROMEDIO	2	214.45	215.54	1.875	245.44	1.125	0.51%	43.75%
R101	8	617.1	618.33	8	618.33	8	0.20%	0.00%
R102	7	547.1	548.11	7	548.11	7	0.18%	0.00%
R103	5	454.6	455.7	5	473.39	4	0.24%	20.00%
R104	4	416.9	417.96	4	417.96	4	0.25%	0.00%
R105	6	530.5	531.54	6	556.72	5	0.20%	16.67%
R106	3	465.4	466.48	5	466.48	5	0.23%	66.67%
R107	4	424.3	425.27	4	425.27	4	0.23%	0.00%
R108	4	397.3	417.94	4	417.94	4	5.20%	0.00%
R109	5	441.3	442.63	5	460.52	4	0.30%	20.00%
R110	4	444.1	445.18	5	445.8	4	0.24%	0.00%
R111	5	428.8	429.7	4	429.7	4	0.21%	20.00%
R112	4	393	394.1	4	394.1	4	0.28%	0.00%
PROMEDIO	4.9	463.3	466.0	5.0	471.1	4.75	0.59%	3.39%
R201	4	463.3	464.37	5	524.59	2	0.23%	50.00%
R202	4	410.5	411.49	4	457.74	2	0.24%	50.00%
R203	3	391.4	392.33	3	400.4	2	0.24%	33.33%
R204	2	355	358.58	2	389.81	1	1.01%	50.00%
R205	3	393	394.06	3	503.66	1	0.27%	66.67%
R206	3	374.4	375.48	3	413.18	1	0.29%	66.67%
R207	3	361.60	362.63	3	398.04	1	0.28%	66.67%
R208	1	328.2	329.33	1	329.33	1	0.34%	0.00%
R209	2	370.7	371.56	2	418.25	1	0.23%	50.00%
R210	3	404.6	410.6	2	509.73	1	1.48%	66.67%

R211	2	350.9	351.91	2	365.06	1	0.29%	50.00%
PROMEDIO	2.73	382.14	383.84	2.72	428.16	1.2	0.45%	53.33%
RC101	4	461.1	462.16	4	462.16	4	0.23%	0.00%
RC102	3	351.8	352.74	3	352.74	3	0.27%	0.00%
RC103	3	306.6	333.92	3	333.92	3	8.91%	0.00%
RC104	3	306.6	307.14	3	307.14	3	0.18%	0.00%
RC105	4	411.3	412.38	4	412.38	4	0.26%	0.00%
RC106	3	345.5	346.51	3	346.51	3	0.29%	0.00%
RC107	3	298.3	298.95	3	298.95	3	0.22%	0.00%
RC108	3	294.5	294.99	3	294.99	3	0.17%	0.00%
PROMEDIO	3.25	346.9625	351.09875	3.25	351.09875	3.25	1.19%	0.00%
RC201	3	360.2	361.24	3	432.3	2	0.29%	33.33%
RC202	3	338	338.82	3	548.42	1	0.24%	66.67%
RC203	3	326.9	327.69	3	432.95	1	0.24%	66.67%
RC204	3	299.7	297.7	2	309.6	1	0.67%	66.67%
RC205	3	338	338.93	3	386.15	2	0.28%	33.33%
RC206	3	324	325.1	3	482.02	1	0.34%	66.67%
RC207	3	298.3	298.95	3	478.97	1	0.22%	66.67%
RC208	2	269.1	269.57	2	309.62	1	0.17%	50.00%
PROMEDIO	2.88	319.275	319.75	2.75	422.50375	1.25	0.15%	56.52%

Resultados del algoritmo asíncrono paquete incompleto para los casos de prueba con 25 clientes.

Caso de Prueba	Mejor solución		Soluciones Obtenidas				Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C102	3	190.3	190.74	3	190.74	3	0.23%	0.0%
C103	3	190.3	190.74	3	190.74	3	0.23%	0.0%
C104	3	186.9	187.45	3	187.45	3	0.29%	0.0%
C105	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C106	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C107	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C108	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C109	3	191.3	191.81	3	191.81	3	0.27%	0.0%
PROMEDIO	3	190.58	191.08	3	191.08	3	0.26%	0.0%
C201	2	214.7	215.54	2	215.54	2	0.39%	0.0%
C202	2	214.7	215.54	2	223.31	1	0.39%	50.0%
C203	2	214.7	215.54	2	223.31	1	0.39%	50.0%
C204	2	213.1	213.93	1	213.93	1	0.39%	50.0%
C205	2	214.7	215.54	2	297.45	1	0.39%	50.0%
C206	2	214.7	215.54	2	285.39	1	0.39%	50.0%
C207	2	214.5	215.34	2	274.78	1	0.39%	50.0%
C208	2	214.5	215.37	2	229.84	1	0.41%	50.0%
PROMEDIO	2	214.45	215.29	1.875	245.44	1.125	0.39%	43.8%
R101	8	617.1	618.33	8	618.33	8	0.20%	0.0%
R102	7	547.1	548.11	7	548.11	7	0.18%	0.0%
R103	5	454.6	455.7	5	473.39	4	0.24%	20.0%
R104	4	416.9	417.94	4	417.94	4	0.25%	0.0%
R105	6	530.5	531.54	6	556.72	5	0.20%	16.7%

R106	3	465.4	466.48	5	466.48	5	0.23%	66.7%
R107	4	424.3	425.27	4	425.27	4	0.23%	0.0%
R108	4	397.3	398.29	4	398.29	4	0.25%	0.0%
R109	5	441.3	442.63	5	460.52	4	0.30%	20.0%
R110	4	444.1	445.18	5	445.8	4	0.24%	0.0%
R111	5	428.8	429.7	4	429.7	4	0.21%	20.0%
R112	4	393	394.1	4	394.1	4	0.28%	0.0%
PROMEDIO	4.92	463.36	464.43	5.083	469.55	4.75	0.23%	3.4%
R201	4	463.3	464.37	4	524.59	2	0.23%	50.0%
R202	4	410.5	411.49	4	455.53	2	0.24%	50.0%
R203	3	391.4	392.33	3	406.24	2	0.24%	33.3%
R204	2	355	355.89	2	389.47	1	0.25%	50.0%
R205	3	393	394.06	3	503.66	1	0.27%	66.7%
R206	3	374.4	377.81	3	413.18	1	0.91%	66.7%
R207	3	361.60	362.63	3	398.04	1	0.28%	66.7%
R208	1	328.2	329.33	1	329.33	1	0.34%	0.0%
R209	2	370.7	371.56	2	418.25	1	0.23%	50.0%
R210	3	404.6	410.6	2	509.73	1	1.48%	66.7%
R211	2	350.9	351.91	2	361.69	1	0.29%	50.0%
PROMEDIO	2.73	382.14	383.81	2.63	428.15	1.27	0.44%	53.3%
RC101	4	461.1	462.16	4	462.16	4	0.23%	0.0%
RC102	3	351.8	352.74	3	352.74	3	0.27%	0.0%
RC103	3	306.6	333.92	3	333.92	3	8.91%	0.0%
RC104	3	306.6	307.14	3	307.14	3	0.18%	0.0%
RC105	4	411.3	412.38	4	412.38	4	0.26%	0.0%
RC106	3	345.5	346.51	3	346.51	3	0.29%	0.0%
RC107	3	298.3	298.95	3	298.95	3	0.22%	0.0%
RC108	3	294.5	294.99	3	294.99	3	0.17%	0.0%
PROMEDIO	3.25	346.96	351.09	3.25	351.09	3.25	1.19%	0.0%
RC201	3	360.2	361.24	3	434.23	2	0.29%	33.3%
RC202	3	338	338.82	3	548.42	1	0.24%	66.7%
RC203	3	326.9	327.69	3	432.55	1	0.24%	66.7%
RC204	3	299.7	300.23	3	329.16	1	0.18%	66.7%
RC205	3	338	338.93	3	386.15	2	0.28%	33.3%
RC206	3	324	325.1	3	482.02	1	0.34%	66.7%
RC207	3	298.3	298.95	3	478.97	1	0.22%	66.7%
RC208	2	269.1	269.57	2	309.62	1	0.17%	50.0%
PROMEDIO	2.88	319.27	320.06	2.875	425.14	1.25	0.25%	56.5%

Resultados del algoritmo síncrono paquete completo para los casos de prueba con 25 clientes.

Caso de Prueba	Mejor solución		Soluciones Obtenidas				Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C102	3	190.3	190.74	3	190.74	3	0.23%	0.0%
C103	3	190.3	190.74	3	190.74	3	0.23%	0.0%
C104	3	186.9	187.45	3	187.45	3	0.29%	0.0%
C105	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C106	3	191.3	191.81	3	191.81	3	0.27%	0.0%

C107	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C108	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C109	3	191.3	191.81	3	191.81	3	0.27%	0.0%
PROMEDIO	3	190.58	191.08	3	191.087	3	0.26%	0.0%
C201	2	214.7	215.54	2	215.54	2	0.39%	0.0%
C202	2	214.7	215.54	2	223.31	1	0.39%	0.0%
C203	2	214.7	215.54	2	223.31	1	0.39%	0.0%
C204	2	213.1	213.93	1	213.93	1	0.39%	50.0%
C205	2	214.7	215.54	2	297.45	1	0.39%	0.0%
C206	2	214.7	215.54	2	285.39	1	0.39%	0.0%
C207	2	214.5	215.34	2	274.78	1	0.39%	0.0%
C208	2	214.5	215.37	2	229.84	1	0.41%	0.0%
PROMEDIO	2	214.45	215.29	1.875	245.44	1.12	0.39%	6.3%
R101	8	617.1	618.33	8	618.33	8	0.20%	0.0%
R102	7	547.1	548.11	7	548.11	7	0.18%	0.0%
R103	5	454.6	455.7	5	473.39	4	0.24%	0.0%
R104	4	416.9	417.96	4	417.96	4	0.25%	0.0%
R105	6	530.5	531.54	6	556.72	5	0.20%	0.0%
R106	3	465.4	466.48	5	466.48	5	0.23%	66.7%
R107	4	424.3	425.27	4	425.27	4	0.23%	0.0%
R108	4	397.3	398.29	4	398.29	4	0.25%	0.0%
R109	5	441.3	442.63	5	460.52	4	0.30%	0.0%
R110	4	444.1	445.18	5	445.8	4	0.24%	25.0%
R111	5	428.8	429.7	4	429.7	4	0.21%	20.0%
R112	4	393	394.1	4	394.1	4	0.28%	0.0%
PROMEDIO	4.92	463.36	464.44	5.083	469.55	4.75	0.23%	3.4%
R201	4	463.3	464.37	4	524.59	2	0.23%	0.0%
R202	4	410.5	411.49	4	458.41	2	0.24%	0.0%
R203	3	391.4	392.33	3	400.4	2	0.24%	0.0%
R204	2	355	358.58	2	389.47	1	1.01%	0.0%
R205	3	393	394.06	3	503.66	1	0.27%	0.0%
R206	3	374.4	375.48	3	413.18	1	0.29%	0.0%
R207	3	361.60	363.28	2	398.04	1	0.46%	33.3%
R208	1	328.2	329.33	1	329.33	1	0.34%	0.0%
R209	2	370.7	371.56	2	418.25	1	0.23%	0.0%
R210	3	404.6	405.48	3	509.73	1	0.22%	0.0%
R211	2	350.9	351.91	2	361.69	1	0.29%	0.0%
PROMEDIO	2.73	382.14	383.44	2.63	427.88	1.27	0.34%	3.3%
RC101	4	461.1	462.16	4	462.16	4	0.23%	0.0%
RC102	3	351.8	352.74	3	352.74	3	0.27%	0.0%
RC103	3	306.6	333.92	3	333.92	3	8.91%	0.0%
RC104	3	306.6	307.14	3	307.14	3	0.18%	0.0%
RC105	4	411.3	412.38	4	412.38	4	0.26%	0.0%
RC106	3	345.5	346.51	3	346.51	3	0.29%	0.0%
RC107	3	298.3	298.95	3	298.95	3	0.22%	0.0%
RC108	3	294.5	294.99	3	294.99	3	0.17%	0.0%
PROMEDIO	3.25	346.96	351.09	3.25	351.09	3.25	1.19%	0.0%
RC201	3	360.2	361.24	3	432.3	2	0.29%	0.0%
RC202	3	338	338.82	3	548.42	1	0.24%	0.0%
RC203	3	326.9	327.69	3	432.55	1	0.24%	0.0%
RC204	3	299.7	300.23	3	327.33	1	0.18%	0.0%

RC205	3	338	338.93	3	386.15	2	0.28%	0.0%
RC206	3	324	325.1	3	476.77	1	0.34%	0.0%
RC207	3	298.3	298.95	3	478.97	1	0.22%	0.0%
RC208	2	269.1	269.57	2	309.6	1	0.17%	0.0%
PROMEDIO	2.88	319.27	320.06	2.87	424.01	1.25	0.25%	0.0%

Resultados del algoritmo síncrono paquete incompleto para los casos de prueba con 25 clientes.

Caso de Prueba	Mejor solución			Soluciones Obtenidas			Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C102	3	190.3	190.74	3	190.74	3	0.23%	0.0%
C103	3	190.3	190.74	3	190.74	3	0.23%	0.0%
C104	3	186.9	187.45	3	187.45	3	0.29%	0.0%
C105	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C106	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C107	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C108	3	191.3	191.81	3	191.81	3	0.27%	0.0%
C109	3	191.3	191.81	3	191.81	3	0.27%	0.0%
PROMEDIO	3	190.58	191.08	3	191.08	3	0.26%	0.0%
C201	2	214.7	215.54	2	215.54	2	0.39%	0.0%
C202	2	214.7	215.54	2	223.31	1	0.39%	0.0%
C203	2	214.7	223.31	1	223.31	1	4.01%	50.0%
C204	2	213.1	213.93	1	213.93	1	0.39%	50.0%
C205	2	214.7	215.54	2	297.45	1	0.39%	0.0%
C206	2	214.7	215.54	2	285.39	1	0.39%	0.0%
C207	2	214.5	215.34	2	274.78	1	0.39%	0.0%
C208	2	214.5	213.93	2	213.93	2	0.27%	0.0%
PROMEDIO	2	214.45	216.08	1.75	243.455	1.25	0.76%	12.5%
R101	8	617.1	618.33	8	618.33	8	0.20%	0.0%
R102	7	547.1	548.11	7	548.11	7	0.18%	0.0%
R103	5	454.6	455.7	5	473.39	4	0.24%	0.0%
R104	4	416.9	417.96	4	417.96	4	0.25%	0.0%
R105	6	530.5	531.54	6	556.72	5	0.20%	0.0%
R106	3	465.4	466.48	5	466.48	5	0.23%	66.7%
R107	4	424.3	425.27	4	425.27	4	0.23%	0.0%
R108	4	397.3	398.29	4	398.29	4	0.25%	0.0%
R109	5	441.3	442.63	5	460.52	4	0.30%	0.0%
R110	4	444.1	445.65	5	445.8	4	0.35%	25.0%
R111	5	428.8	429.7	4	429.7	4	0.21%	20.0%
R112	4	393	394.1	4	394.1	4	0.28%	0.0%
PROMEDIO	4.92	463.36	464.48	5.08	469.55	4.75	0.24%	3.4%
R201	4	463.3	464.37	4	526.84	2	0.23%	0.0%
R202	4	410.5	411.49	4	460.11	2	0.24%	0.0%
R203	3	391.4	392.33	3	419.43	2	0.24%	0.0%
R204	2	355	358.58	2	388.58	1	1.01%	0.0%
R205	3	393	394.06	3	502.39	1	0.27%	0.0%
R206	3	374.4	376.86	3	429.24	1	0.66%	0.0%
R207	3	361.60	386.86	2	398.04	1	6.99%	33.3%
R208	1	328.2	329.33	1	329.33	1	0.34%	0.0%
R209	2	370.7	371.56	2	18.25	1	0.23%	0.0%
R210	3	404.6	410.6	2	534.92	1	1.48%	33.3%

R211	2	350.9	351.91	2	361.69	1	0.29%	0.0%
PROMEDIO	2.73	382.2	386.17	2.54	397.16	1.27	1.06%	6.7%
RC101	4	461.1	462.16	4	462.16	4	0.23%	0.0%
RC102	3	351.8	352.74	3	352.74	3	0.27%	0.0%
RC103	3	306.6	333.92	3	333.92	3	8.91%	0.0%
RC104	3	306.6	307.14	3	307.14	3	0.18%	0.0%
RC105	4	411.3	412.38	4	412.38	4	0.26%	0.0%
RC106	3	345.5	346.51	3	346.51	3	0.29%	0.0%
RC107	3	298.3	298.95	3	298.95	3	0.22%	0.0%
RC108	3	294.5	294.99	3	294.99	3	0.17%	0.0%
PROMEDIO	3.25	346.9	351.09	3.25	351.09	3.25	1.19%	0.0%
RC201	3	360.2	361.24	3	432.3	2	0.29%	33.3%
RC202	3	338	338.82	3	376.12	1	0.24%	66.7%
RC203	3	326.9	327.69	3	432.55	1	0.24%	66.7%
RC204	3	299.7	327.77	1	327.77	1	9.37%	66.7%
RC205	3	338	338.93	3	386.15	2	0.28%	33.3%
RC206	3	324	325.1	3	482.02	1	0.34%	66.7%
RC207	3	298.3	298.95	3	478.97	1	0.22%	66.7%
RC208	2	269.1	269.57	2	309.6	1	0.17%	50.0%
PROMEDIO	2.88	319.27	323.50	2.625	403.18	1.25	1.33%	8.7%

Resultados del algoritmo asíncrono paquete completo para los casos de prueba con 50 clientes.

Caso de Prueba	Mejor solución			Soluciones Obtenidas			Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	5	362.4	363.25	5	363.25	5	0.23%	0.00%
C102	5	361.4	362.17	5	362.17	5	0.21%	0.00%
C103	5	361.4	358.42	5	358.42	5	0.82%	0.00%
C104	5	358	351.13	5	351.13	5	1.92%	0.00%
C105	5	362.4	362.05	5	362.05	5	0.10%	0.00%
C106	5	362.4	363.25	5	363.25	5	0.23%	0.00%
C107	5	362.4	362.05	5	362.05	5	0.10%	0.00%
C108	5	362.4	363.25	5	363.25	5	0.23%	0.00%
C109	5	362.4	363.25	5	363.25	5	0.23%	0.00%
PROMEDIO	5	361.68	360.98	5	360.98	5	0.20%	0.00%
C201	3	360.2	361.8	3	444.96	2	0.44%	33.33%
C202	3	360.2	361.8	3	403.81	2	0.44%	33.33%
C203	3	359.8	361.41	3	402.52	2	0.45%	33.33%
C204	2	350.1	351.72	2	351.71	2	0.46%	0.00%
C205	3	359.8	361.41	3	444.08	2	0.45%	33.33%
C206	3	359.8	361.41	3	416.23	2	0.45%	33.33%
C207	3	359.6	361.21	3	398.33	2	0.45%	33.33%
C208	2	350.5	352.12	2	352.12	2	0.46%	0.00%
PROMEDIO	2.75	357.5	359.11	2.75	401.72	2	0.45%	27.27%
R101	12	1044	1046.7	12	1046.7	12	0.26%	0.00%
R102	11	909	911.44	11	924.19	10	0.27%	9.09%
R103	9	772.9	780.77	9	784.56	8	1.02%	11.11%
R104	6	625.4	631.32	6	631.32	6	0.95%	0.00%
R105	9	899.3	914.31	9	914.31	9	1.67%	0.00%

R106	5	793	795.25	8	845.58	7	0.28%	40.00%
R107	7	711.1	713.5	7	749.05	6	0.34%	14.29%
R108	6	617.7	631.32	6	631.32	6	2.20%	0.00%
R109	8	786.8	794.34	8	816.41	7	0.96%	12.50%
R110	7	697	699.38	7	699.38	7	0.34%	0.00%
R111	7	707.2	709.53	7	709.53	7	0.33%	0.00%
R112	6	630.2	641.37	6	641.37	6	1.77%	0.00%
PROMEDIO	7.75	766.13	772.43	8	782.81	7.58	0.82%	2.15%

R201	6	791.9	807.59	5	953.29	2	1.98%	66.67%
R202	5	698.5	718.77	4	825.16	2	2.90%	60.00%
R203	5	605.3	618.46	4	679.05	2	2.17%	60.00%
R204	2	506.4	509.25	2	509.25	2	0.56%	0.00%
R205	4	690.1	701.24	4	755.44	2	1.61%	50.00%
R206	4	632.4	647.61	3	660.52	2	2.41%	50.00%
R207			587.04	2	587.04	2		
R208			496.94	2	496.94	2		
R209	4	600.6	614.9	4	659.54	2	2.38%	50.00%
R210	4	645.6	672.64	2	672.64	2	4.19%	50.00%
R211	3	535.5	544.92	3	560.91	2	1.76%	33.33%
PROMEDIO	3.36	518.75	629.03	3.18	669.07	2	21.26%	40.54%

RC101	8	944	945.58	8	945.58	8	0.17%	0.00%
RC102	7	822.5	823.97	7	823.97	7	0.18%	0.00%
RC103	6	710.9	712.56	6	712.56	6	0.23%	0.00%
RC104	5	545.8	546.51	5	546.51	5	0.13%	0.00%
RC105	8	855.3	856.97	8	856.97	8	0.20%	0.00%
RC106	6	723.2	724.65	6	724.65	6	0.20%	0.00%
RC107	6	642.7	645.58	6	645.58	6	0.45%	0.00%
RC108	6	598.1	599.17	6	599.17	6	0.18%	0.00%
PROMEDIO	6.5	730.31	731.87	6.5	731.87	6.5	0.21%	0.00%

RC201	5	684.8	686.31	5	845.35	3	0.22%	40.00%
RC202	5	613.6	615.04	5	867.83	2	0.23%	60.00%
RC203	4	555.3	556.5	4	556.5	4	0.22%	0.00%
RC204	3	444.2	490.5	3	506.09	2	10.42%	33.33%
RC205	5	630.2	631.98	5	761.91	3	0.28%	40.00%
RC206	5	610	611.68	5	755.13	2	0.28%	60.00%
RC207	4	558.6	559.91	4	670.51	2	0.23%	50.00%
RC208	3	828.14	481.93	3	506.69	2	41.81%	33.33%
PROMEDIO	4.25	615.60	579.23	4.25	683.75	2.5	5.91%	41.18%

Resultados del algoritmo asíncrono paquete incompleto para los casos de prueba con 50 clientes.

Caso de Prueba	Mejor solución			Soluciones Obtenidas			Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	5	362.4	363.25	5	363.25	5	0.23%	0%
C102	5	361.4	362.17	5	362.17	5	0.21%	0%
C103	5	361.4	362.17	5	362.17	5	0.21%	0%
C104	5	358	352.35	5	352.35	5	1.58%	0%
C105	5	362.4	362.05	5	362.05	5	0.10%	0%
C106	5	362.4	363.25	5	363.25	5	0.23%	0%

C107	5	362.4	362.05	5	362.05	5	0.10%	0%
C108	5	362.4	363.25	5	363.25	5	0.23%	0%
C109	5	362.4	363.25	5	363.25	5	0.23%	0%
PROMEDIO	5	361.68	361.53	5	361.53	5	0.04%	0%

C201	3	360.2	361.8	3	444.96	2	0.44%	33%
C202	3	360.2	361.8	3	403.81	2	0.44%	33%
C203	3	359.8	361.41	3	402.52	2	0.45%	33%
C204	2	350.1	351.72	2	351.71	2	0.46%	0%
C205	3	359.8	361.41	3	444.08	2	0.45%	33%
C206	3	359.8	361.41	3	441.15	2	0.45%	33%
C207	3	359.6	361.21	3	426.24	2	0.45%	33%
C208	2	350.5	352.12	2	352.12	2	0.46%	0%
PROMEDIO	2.75	357.5	359.11	2.75	408.32	2	0.45%	27%

R101	12	1044	1046.7	12	1046.7	12	0.26%	0%
R102	11	909	920.17	11	924.19	10	1.23%	9%
R103	9	772.9	784.56	8	784.56	8	1.51%	11%
R104	6	625.4	631.32	6	631.32	6	0.95%	0%
R105	9	899.3	916.59	9	916.59	9	1.92%	0%
R106	5	793	795.25	8	864.27	7	0.28%	40%
R107	7	711.1	723.46	7	723.46	7	1.74%	0%
R108	6	617.7	620.67	6	620.67	6	0.48%	0%
R109	8	786.8	795.66	8	815.14	7	1.13%	13%
R110	7	697	699.38	7	699.38	7	0.34%	0%
R111	7	707.2	709.53	7	796.85	6	0.33%	14%
R112	6	630.2	637.52	6	637.52	6	1.16%	0%
PROMEDIO	7.75	766.13	773.40	7.91	788.3875	7.58	0.95%	2%

R201	6	791.9	802.07	5	996.24	2	1.28%	67%
R202	5	698.5	723.61	4	862.71	2	3.59%	60%
R203	5	605.3	610.75	4	685.58	2	0.90%	60%
R204	2	506.4	509.25	2	509.25	2	0.56%	0%
R205	4	690.1	708.01	4	740.73	2	2.60%	50%
R206	4	632.4	658.04	2	658.04	2	4.05%	50%
R207			584.57	3	595.7	2		
R208			500.3	2	500.3	2		
R209	4	600.6	603.1	4	659.72	2	0.42%	50%
R210	4	645.6	672.64	2	672.64	2	4.19%	50%
R211	3	535.5	537.98	3	560.91	2	0.46%	33%
PROMEDIO	3.37	518.75	628.21	3.18	676.52	2	21.10%	41%

RC101	8	944	945.58	8	945.58	8	0.17%	0%
RC102	7	822.5	823.97	7	823.97	7	0.18%	0%
RC103	6	710.9	712.56	6	712.56	6	0.23%	0%
RC104	5	545.8	546.51	5	546.51	5	0.13%	0%
RC105	8	855.3	856.97	8	856.97	8	0.20%	0%
RC106	6	723.2	724.65	6	724.65	6	0.20%	0%
RC107	6	642.7	643.86	6	643.86	6	0.18%	0%
RC108	6	598.1	599.17	6	599.17	6	0.18%	0%
PROMEDIO	6.5	730.31 25	731.65	6.5	731.65	6.5	0.18%	0%

RC201	5	684.8	686.31	5	845.34	3	0.22%	40%
RC202	5	613.6	615.04	5	880.6	2	0.23%	60%

RC203	4	555.3	556.5	4	675.29	2	0.22%	50%
RC204	3	444.2	444.97	3	480.41	2	0.17%	33%
RC205	5	630.2	631.98	5	761.91	3	0.28%	40%
RC206	5	610	611.68	5	772.15	2	0.28%	60%
RC207	4	558.6	559.91	4	656.3	2	0.23%	50%
RC208	3	828.14	481.25	3	481.25	3	41.89%	0%
PROMEDIO	4.25	615.60	573.45	4.25	694.15	2.375	6.85%	44%

Resultados del algoritmo síncrono paquete completo para los casos de prueba con 50 clientes.

Caso de Prueba	Mejor Solución		Soluciones Obtenidas				Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	5	362.4	363.25	5	363.25	5	0.23%	0%
C102	5	361.4	362.17	5	362.17	5	0.21%	0%
C102	5	361.4	358.42	5	358.42	5	0.82%	0%
C103	5	358	355.21	5	355.21	5	0.78%	0%
C104	5	362.4	362.05	5	362.05	5	0.10%	0%
C105	5	362.4	363.25	5	363.25	5	0.23%	0%
C106	5	362.4	362.05	5	362.05	5	0.10%	0%
C107	5	362.4	363.25	5	363.25	5	0.23%	0%
C108	5	362.4	363.25	5	363.25	5	0.23%	0%
PROMEDIO	5	361.68	361.43	5	361.43	5	0.07%	0%
C201	3	360.2	361.8	3	444.96	2	0.44%	33%
C202	3	360.2	361.8	3	403.81	2	0.44%	33%
C203	3	359.8	361.41	3	402.52	2	0.45%	33%
C204	2	350.1	351.72	2	351.72	2	0.46%	0%
C205	3	359.8	361.41	3	430.03	2	0.45%	33%
C206	3	359.8	361.41	3	409.61	2	0.45%	33%
C207	3	359.6	361.21	3	398.33	2	0.45%	33%
C208	2	350.5	352.12	2	352.12	2	0.46%	0%
PROMEDIO	2.75	357.5	359.11	2.75	399.13	2	0.45%	27%
R101	12	1044	1046.7	12	1046.7	12	0.26%	0%
R102	11	909	911.44	11	924.19	10	0.27%	9%
R103	9	772.9	779.87	9	786.83	8	0.90%	11%
R104	6	625.4	632.21	6	632.21	6	1.09%	0%
R105	9	899.3	914.31	9	914.31	9	1.67%	0%
R106	5	793	799.33	8	863.63	7	0.80%	40%
R107	7	711.1	713.5	7	753.96	6	0.34%	14%
R108	6	617.7	620.26	6	620.26	6	0.41%	0%
R109	8	786.8	796.01	8	801.97	7	1.17%	13%
R110	7	697	699.38	7	699.38	7	0.34%	0%
R111	7	707.2	709.53	7	774.47	6	0.33%	14%
R112	6	630.2	632.72	6	632.72	6	0.40%	0%
PROMEDIO	7.75	766.13	771.27	8	787.55	7.5	0.67%	3%
R201	6	791.9	807.59	5	953.29	2	1.98%	67%
R202	5	698.5	718.77	4	825.16	2	2.90%	60%
R203	5	605.3	618.46	4	679.05	2	2.17%	60%
R204	2	506.4	509.25	2	509.25	2	0.56%	0%
R205	4	690.1	701.24	4	755.44	2	1.61%	50%
R206	4	632.4	647.61	3	660.52	2	2.41%	50%
R207			587.04	2	587.04	2		
R208			496.94	2	496.94	2		

R209	4	600.6	614.9	4	659.54	2	2.38%	50%
R210	4	645.6	672.64	2	672.64	2	4.19%	50%
R211	3	535.5	544.92	3	560.91	2	1.76%	33%
PROMEDIO	3.36	518.75	629.03	3.18	669.07	2	21.26%	41%

RC101	8	944	945.58	8	945.58	8	0.17%	0%
RC102	7	822.5	823.97	7	823.97	7	0.18%	0%
RC103	6	710.9	712.56	6	712.56	6	0.23%	0%
RC104	5	545.8	546.51	5	546.51	5	0.13%	0%
RC105	8	855.3	856.97	8	856.97	8	0.20%	0%
RC106	6	723.2	724.65	6	724.65	6	0.20%	0%
RC107	6	642.7	645.58	6	645.58	6	0.45%	0%
RC108	6	598.1	599.17	6	599.17	6	0.18%	0%
PROMEDIO	6.5	730.31	731.87	6.5	731.87	6.5	0.21%	0%

RC201	5	684.8	686.31	5	838.54	3	0.22%	40%
RC202	5	613.6	615.04	5	867.02	2	0.23%	60%
RC203	4	555.3	556.5	4	674.42	2	0.22%	50%
RC204	3	444.2	444.97	3	481.71	2	0.17%	33%
RC205	5	630.2	631.98	5	761.38	3	0.28%	40%
RC206	5	610	611.68	5	755.13	2	0.28%	60%
RC207	4	558.6	559.91	4	655.81	2	0.23%	50%
RC208	3	828.14	485.37	3	505.85	2	41.39%	33%
PROMEDIO	4.25	615.60	573.97	4.25	692.48	2.25	6.76%	47%

Resultados del algoritmo síncrono paquete incompleto para los casos de prueba con 50 clientes.

Caso de Prueba	Mejor Solución			Soluciones Obtenidas			Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	5	362.4	363.25	5	363.25	5	0.23%	0%
C102	5	361.4	362.17	5	362.17	5	0.21%	0%
C103	5	361.4	362.17	5	362.17	5	0.21%	0%
C104	5	358	353.96	5	388.39	5	1.13%	0%
C105	5	362.4	362.05	5	362.05	5	0.10%	0%
C106	5	362.4	363.25	5	363.25	5	0.23%	0%
C107	5	362.4	362.05	5	362.05	5	0.10%	0%
C108	5	362.4	363.25	5	363.25	5	0.23%	0%
C109	5	362.4	362.05	5	362.05	5	0.10%	0%
PROMEDIO	5	361.68	361.57	5	365.40	5	0.03%	0%

C201	3	360.2	361.8	3	444.96	2	0.44%	33%
C202	3	360.2	361.8	3	213.93	2	0.44%	33%
C203	3	359.8	361.41	3	402.52	2	0.45%	33%
C204	2	350.1	351.72	2	351.71	2	0.46%	0%
C205	3	359.8	361.41	3	444.08	2	0.45%	33%
C206	3	359.8	361.41	3	424.76	2	0.45%	33%
C207	3	359.6	361.21	3	398.33	2	0.45%	33%
C208	2	350.5	352.12	2	352.12	2	0.46%	0%
PROMEDIO	2.75	357.5	359.11	2.75	379.05	2	0.45%	27%

R101	12	1044	1046.7	12	1046.7	12	0.26%	0%
R102	11	909	923.71	10	923.71	10	1.62%	9%
R103	9	772.9	775.65	9	785.04	8	0.36%	11%
R104	6	625.4	631.32	6	631.32	6	0.95%	0%
R105	9	899.3	914.31	9	914.31	9	1.67%	0%

R106	5	793	795.25	8	888.61	7	0.28%	40%
R107	7	711.1	725.74	7	753.93	6	2.06%	14%
R108	6	617.7	620.26	4	620.26	4	0.41%	33%
R109	8	786.8	799.67	8	801.97	7	1.64%	13%
R110	7	697	699.38	7	699.38	7	0.34%	0%
R111	7	707.2	713.01	7	713.01	7	0.82%	0%
R112	6	630.2	637.71	6	637.71	6	1.19%	0%
PROMEDIO	7.75	766.13	773.55	7.75	784.66	7.41	0.97%	4%

R201	6	791.9	817.19	4	996.09	2	3.19%	67%
R202	5	698.5	754.04	3	891.5	2	7.95%	60%
R203	5	605.3	666.15	3	683.75	2	10.05%	60%
R204	2	506.4	511.07	2	511.07	2	0.92%	0%
R205	4	690.1	726.45	4	749.7	3	5.27%	25%
R206	4	632.4	646.85	3	665.04	2	2.28%	50%
R207			592.77	2	592.77	2		
R208			498.22	2	498.22	2		
R209	4	600.6	623.52	3	669.61	2	3.82%	50%
R210	4	645.6	673.81	2	673.81	2	4.37%	50%
R211	3	535.5	540.06	3	555.61	2	0.85%	33%
PROMEDIO	3.36	518.75	640.92	2.81	680.65	2.09	23.55%	38%

RC101	8	944	945.58	8	945.58	8	0.17%	0%
RC102	7	822.5	823.97	7	823.97	7	0.18%	0%
RC103	6	710.9	712.56	6	712.56	6	0.23%	0%
RC104	5	545.8	546.51	5	546.51	5	0.13%	0%
RC105	8	855.3	856.97	8	856.97	8	0.20%	0%
RC106	6	723.2	724.65	6	724.65	6	0.20%	0%
RC107	6	642.7	645.58	6	645.58	6	0.45%	0%
RC108	6	598.1	599.17	6	599.17	6	0.18%	0%
PROMEDIO	6.5	730.31	731.87	6.5	731.87	6.5	0.21%	0%

RC201	5	684.8	714.97	4	838.72	3	4.41%	40%
RC202	5	613.6	615.04	5	891.8	2	0.23%	60%
RC203	4	555.3	557.8	4	687.36	2	0.45%	50%
RC204	3	444.2	444.97	3	479.87	2	0.17%	33%
RC205	5	630.2	631.98	5	764.33	3	0.28%	40%
RC206	5	610	619.24	4	764.29	2	1.51%	60%
RC207	4	558.6	574.03	3	667.75	2	2.76%	50%
RC208	3	828.14	481.25	3	505.09	2	41.89%	33%
PROMEDIO	4.25	615.60	579.91	3.875	699.90	2.25	5.80%	47%

Resultados del algoritmo asíncrono paquete completo para los casos de prueba con 100 clientes.

Caso de Prueba	Mejor solución		Soluciones Obtenidas				Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	10	828.94	828.94	10	828.94	10	0.00%	0.00%
C102	10	828.94	828.25	10	828.25	10	0.08%	0.00%
C103	10	828.06	821.06	10	821.06	10	0.85%	0.00%
C104	10	824.78	821.77	10	821.77	10	0.36%	0.00%
C105	10	828.94	828.94	10	828.94	10	0.00%	0.00%
C106	10	828.94	828.94	10	828.94	10	0.00%	0.00%
C107	10	828.94	822.42	10	822.42	10	0.79%	0.00%
C108	10	828.94	825.86	10	825.86	10	0.37%	0.00%
C109	10	828.94	828.94	10	828.94	10	0.00%	0.00%
PROMEDIO	10	828.38	826.12	10.00	826.12	10	0.27%	0.00%
C201	3	591.56	591.56	3	591.56	3	0.00%	0.00%
C202	3	591.56	591.56	3	591.56	3	0.00%	0.00%
C203	3	591.17	591.17	3	591.17	3	0.00%	0.00%
C204	3	590.6	590.60	3	590.60	3	0.00%	0.00%
C205	3	588.88	588.88	3	588.88	3	0.00%	0.00%
C206	3	588.49	588.49	3	588.49	3	0.00%	0.00%
C207	3	588.29	588.29	3	588.29	3	0.00%	0.00%
C208	3	588.32	588.32	3	588.32	3	0.00%	0.00%
PROMEDIO	3	589.9	516.0	3.0	516.0	3.0	12.52%	0.00%
R101	19	1645.79	1648.12	20	1655.76	19	0.14%	5.26%
R102	17	1486.12	1495.29	17	1495.29	17	0.62%	0.00%
R103	13	1292.68	1232.52	14	1232.52	14	4.65%	7.69%
R104	9	1007.24	1016.54	10	1016.54	10	0.92%	11.11%
R105	14	1377.11	1380.37	15	1380.37	15	0.24%	7.14%
R106	12	1251.98	1264.6	13	1347.46	12	1.01%	8.33%
R107	10	1104.66	1113.82	11	1113.82	11	0.83%	10.00%
R108	9	960.88	986.03	10	986.03	10	2.62%	11.11%
R109	11	1194.73	1189.94	13	1189.94	13	0.40%	18.18%
R110	10	1118.59	1101.44	11	1101.44	11	1.53%	10.00%
R111	10	1096.72	1081.07	11	1081.07	11	1.43%	10.00%
R112	9	982.14	976.05	10	976.05	10	0.62%	11.11%
PROMEDIO	11.92	1209.9	1207.1	12.9	1214.7	12.8	0.23%	8.39%
R201	4	1252.37	1180.89	6	1279.22	4	5.71%	50.00%
R202	3	1191.7	1090.02	5	1118.26	4	8.53%	66.67%
R203	3	939.54	923.21	4	985.29	3	1.74%	33.33%
R204	2	825.52	743.9	4	755.01	3	9.89%	100.00%
R205	3	994.42	972.39	5	1080.14	3	2.22%	66.67%
R206	3	906.14	922.35	4	997.92	3	1.79%	33.33%
R207	2	893.33	840.49	3	840.49	3	5.91%	50.00%
R208	2	726.75	730.56	3	786.4	2	0.52%	50.00%
R209	3	909.16	911.05	4	953.63	3	0.21%	33.33%
R210	3	939.34	963.61	4	992.93	3	2.58%	33.33%
R211	2	892.7	783.3	4.0	868.7	3.0	12.26%	100.00%
PROMEDIO	2.727	951.9	914.7	4.2	968.9	3.1	3.91%	53.33%
RC101	14	1696.94	1637	15	1697.94	14	3.53%	7.14%
RC102	12	1554.75	1491.15	14	1491.15	14	4.09%	16.67%

RC103	11	1261.67	1270.57	11	1270.57	11	0.71%	0.00%
RC104	10	1135.48	1126.35	10	1126.35	10	0.80%	0.00%
RC105	13	1629.44	1519.26	15	1547.48	14	6.76%	15.38%
RC106	11	1424.73	1399.97	13	1401.87	12	1.74%	18.18%
RC107	11	1230.48	1213.65	12	1226.56	11	1.37%	9.09%
RC108	10	1139.82	1166.94	11	1166.94	11	2.38%	10.00%
PROMEDIO	11.5	1384.2	1353.1	12.6	1366.1	12.1	2.24%	9.78%
RC201	4	1406.91	1292.1	7	1488.49	4	8.16%	75.00%
RC202	3	1367.09	1123.23	7	1306.1	4	17.84%	133.33%
RC203	3	1049.62	964.84	4	1155.5	3	8.08%	33.33%
RC204	3	798.41	813.52	4	908.13	3	1.89%	33.33%
RC205	4	1297.19	1204.06	7	1204.06	4	7.18%	75.00%
RC206	3	1146.32	1089.13	5	1500.28	3	4.99%	66.67%
RC207	3	1061.14	980.78	5	1028.94	4	7.57%	66.67%
RC208	3	828.14	857.86	4	881.81	3	3.59%	33.33%
PROMEDIO	3.25	1119.35	1040.69	5.38	1184.16	3.50	7.03%	65.38%

Resultados del algoritmo Segundo modelo asíncrono paquete incompleto para los casos de prueba con 100 clientes.

Caso de Prueba	Mejor solución		Soluciones Obtenidas				Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	10	828.94	828.94	10	828.94	10	0.00%	0.00%
C102	10	828.94	821.93	10	821.93	10	0.85%	0.00%
C103	10	828.06	821.06	10	821.06	10	0.85%	0.00%
C104	10	824.78	819.16	10	894.66	9	0.68%	0.00%
C105	10	828.94	828.94	10	828.94	10	0.00%	0.00%
C106	10	828.94	828.94	10	828.94	10	0.00%	0.00%
C107	10	828.94	824.66	10	824.66	10	0.52%	0.00%
C108	10	828.94	827.76	10	827.76	10	0.14%	0.00%
C109	10	828.94	854.78	10	854.78	10	3.12%	0.00%
PROMEDIO	10	828.38	828.46	10.0 0	836.85	9.88	0.01%	0.00%
C201	3	591.56	591.56	3	591.56	3	0.00%	0.00%
C202	3	591.56	591.56	3	591.56	3	0.00%	0.00%
C203	3	591.17	591.17	3	591.17	3	0.00%	0.00%
C204	3	590.6	590.6	3	590.6	3	0.00%	0.00%
C205	3	588.88	588.88	3	588.88	3	0.00%	0.00%
C206	3	588.49	588.49	3	588.49	3	0.00%	0.00%
C207	3	588.29	588.29	3	588.29	3	0.00%	0.00%
C208	3	588.32	588.32	3	588.32	3	0.00%	0.00%
PROMEDIO	3	589.9	589.9	3.0	589.9	3.0	0.00%	0.00%
R101	19	1645.79	1650.8	19	1650.8	19	0.30%	0.00%
R102	17	1486.12	1499.85	17	1499.85	17	0.92%	0.00%
R103	13	1292.68	1246.41	14	1246.41	14	3.58%	7.69%
R104	9	1007.24	1020.11	11	1024.27	10	1.28%	22.22%
R105	14	1377.11	1399.06	15	1408.74	14	1.59%	7.14%
R106	12	1251.98	1290.28	12	1290.28	12	3.06%	0.00%
R107	10	1104.66	1134.47	12	1137.28	11	2.70%	20.00%
R108	9	960.88	1141.23	11	1171.65	10	18.77%	22.22%

R109	11	1194.73	1171.09	13	1172.34	12	1.98%	18.18%
R110	10	1118.59	1112.63	12	1137.31	11	0.53%	20.00%
R111	10	1096.72	1097.95	11	1097.95	11	0.11%	10.00%
R112	9	982.14	1002.63	11	1034.92	10	2.09%	22.22%
PROMEDIO	11.92	1209.9	1230.5	13.2	1239.3	12.6	1.71%	10.49%
R201	4	1252.37	1168.09	7	1335.15	4	6.73%	75.00%
R202	3	1191.7	1192.96	6	1295	4	0.11%	100.00%
R203	3	939.54	932.73	4	1016.2	3	0.72%	33.33%
R204	2	825.52	763.21	3	763.21	3	7.55%	50.00%
R205	3	994.42	1009.6	5	1064.59	3	1.53%	66.67%
R206	3	906.14	1021.7	4	961.95	3	12.75%	33.33%
R207	2	893.33	839.26	3	839.26	3	6.05%	50.00%
R208	2	726.75	749.56	4	835.92	2	3.14%	100.00%
R209	3	909.16	923.64	4	986.02	3	1.59%	33.33%
R210	3	939.34	964.78	5	1011.29	3	2.71%	66.67%
R211	2	892.7	775.1	4.0	809.9	3.0	13.18%	100.00%
PROMEDIO	2.727	951.9	940.1	4.5	992.6	3.1	1.25%	63.33%
RC101	14	1696.94	1637.87	15	1637.87	15	3.48%	7.14%
RC102	12	1554.75	1477.5	15	1512.57	13	4.97%	25.00%
RC103	11	1261.67	1302.58	11	1302.58	11	3.24%	0.00%
RC104	10	1135.48	1175.18	11	1232.36	10	3.50%	10.00%
RC105	13	1629.44	1554.77	15	1617.6	14	4.58%	15.38%
RC106	11	1424.73	1400.08	13	1405.43	12	1.73%	18.18%
RC107	11	1230.48	1213.39	12	1232.82	11	1.39%	9.09%
RC108	10	1139.82	1133.59	11	1133.59	11	0.55%	10.00%
PROMEDIO	11.5	1384.2	1361.9	12.9	1384.4	12.1	1.61%	11.96%
RC201	4	1406.91	1290.28	7	1428.44	4	8.29%	75.00%
RC202	3	1367.09	1135.79	6	1279.86	4	16.92%	100.00%
RC203	3	1049.62	925.74	4	1046.48	3	11.80%	33.33%
RC204	3	798.41	833.09	3	833.09	3	4.34%	0.00%
RC205	4	1297.19	1203.43	8	1466.1	4	7.23%	100.00%
RC206	3	1146.32	1094.73	5	1311.56	3	4.50%	66.67%
RC207	3	1061.14	1060.67	4	1128.68	3	0.04%	33.33%
RC208	3	828.14	830.28	4	915.82	3	0.26%	33.33%
PROMEDIO	3.25	1119.35	1046.75	5.13	1176.25	3.38	6.49%	57.69%

Resultados del algoritmo Tercer modelo síncrono paquete completo para los casos de prueba con 100 clientes.

Caso de Prueba	Mejor solución		Soluciones Obtenidas				Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	10	828.94	828.94	10	828.94	10	0.00%	0.00%
C102	10	828.94	828.94	10	828.94	10	0.00%	0.00%
C103	10	828.06	828.06	10	828.06	10	0.00%	0.00%
C104	10	824.78	816.75	10	816.75	9	0.97%	0.00%
C105	10	828.94	824.04	10	824.04	10	0.59%	0.00%
C106	10	828.94	828.94	10	828.94	10	0.00%	0.00%
C107	10	828.94	824.66	10	824.66	10	0.52%	0.00%
C108	10	828.94	828.94	10	828.94	10	0.00%	0.00%
C109	10	828.94	828.3	10	828.3	10	0.08%	0.00%
PROMEDIO	10	828.38	826.40	10.00	826.40	9.88	0.24%	0.00%

C201	3	591.56	591.56	3	591.56	3	0.00%	0.00%
C202	3	591.56	591.56	3	591.56	3	0.00%	0.00%
C203	3	591.17	600.21	3	600.21	3	1.53%	0.00%
C204	3	590.6	599.29	3	599.29	3	1.47%	0.00%
C205	3	588.88	588.88	3	588.88	3	0.00%	0.00%
C206	3	588.49	588.49	3	588.49	3	0.00%	0.00%
C207	3	588.29	588.29	3	588.29	3	0.00%	0.00%
C208	3	588.32	588.32	3	588.32	3	0.00%	0.00%
PROMEDIO	3	589.9	592.1	3.0	592.1	3.0	0.38%	0.00%
R101	19	1645.79	1654.06	19	1654.06	19	0.50%	0.00%
R102	17	1486.12	1482.8	18	1499.7	17	0.22%	5.88%
R103	13	1292.68	1249.3	14	1324.39	13	3.36%	7.69%
R104	9	1007.24	1025.68	10	1025.68	10	1.83%	11.11%
R105	14	1377.11	1390	16	1427.47	14	0.94%	14.29%
R106	12	1251.98	1268.89	13	1268.89	13	1.35%	8.33%
R107	10	1104.66	1093.93	11	1093.93	11	0.97%	10.00%
R108	9	960.88	1000.77	10	1000.77	10	4.15%	11.11%
R109	11	1194.73	1169.7	13	1182.05	12	2.10%	18.18%
R110	10	1118.59	1123.18	12	1153.18	11	0.41%	20.00%
R111	10	1096.72	1081.63	11	1081.63	11	1.38%	10.00%
R112	9	982.14	1015.62	10	1015.62	10	3.41%	11.11%
PROMEDIO	11.91	1209.9	1213.0	13.1	1227.3	12.6	0.25%	9.79%
R201	4	1252.37	1200.96	6	1307.62	4	4.11%	50.00%
R202	3	1191.7	1045.17	5	1113.87	4	12.30%	66.67%
R203	3	939.54	1002.12	3	1002.12	3	6.66%	0.00%
R204	2	825.52	763.95	3	763.95	3	7.46%	50.00%
R205	3	994.42	967.04	5	1055.21	3	2.75%	66.67%
R206	3	906.14	944.42	3	944.42	3	4.22%	0.00%
R207	2	893.33	839.52	3	839.52	3	6.02%	50.00%
R208	2	726.75	713.63	3	765.23	2	1.81%	50.00%
R209	3	909.16	914.8	4	965.78	3	0.62%	33.33%
R210	3	939.34	980.42	3	980.42	3	4.37%	0.00%
R211	2	892.7	798.7	4.0	823.9	3.0	10.53%	100.00%
PROMEDIO	2.72	951.9	924.6	3.8	960.2	3.1	2.87%	40.00%
RC101	14	1696.94	1638.79	15	1638.79	15	3.43%	7.14%
RC102	12	1554.75	1496.59	14	1532.39	13	3.74%	16.67%
RC103	11	1261.67	1288	11	1288	11	2.09%	0.00%
RC104	10	1135.48	1157.07	10	1157.07	10	1.90%	0.00%
RC105	13	1629.44	1528.76	15	1549.43	14	6.18%	15.38%
RC106	11	1424.73	1438.47	14	1443.83	13	0.96%	27.27%
RC107	11	1230.48	1210.66	12	1230.08	11	1.61%	9.09%
RC108	10	1139.82	1130.14	10	1112.3	11	0.85%	0.00%
PROMEDIO	11.5	1384.2	1361.1	12.6	1369.0	12.3	1.67%	9.78%
RC201	4	1406.91	1464.73	4	1464.73	4	4.11%	0.00%
RC202	3	1367.09	1164.55	6	1245.53	4	14.82%	100.00%
RC203	3	1049.62	965.63	4	1117.42	3	8.00%	33.33%
RC204	3	798.41	830.15	3	830.15	3	3.98%	0.00%
RC205	4	1297.19	1165.34	7	1386.76	4	10.16%	75.00%
RC206	3	1146.32	1110.66	5	1204.91	3	3.11%	66.67%
RC207	3	1061.14	1066.58	4	1110.41	3	0.51%	33.33%
RC208	3	828.14	857.86	4	881.81	3	3.59%	33.33%
PROMEDIO	3.25	1119.35	1078.19	4.63	1155.22	3.38	3.68%	42.31%

Resultados del algoritmo Cuarto modelo síncrono paquete incompleto para los casos de prueba con 100 clientes.

Caso de Prueba	Mejor solución		Soluciones Obtenidas				Diferencia	
	MS R	MS D	D1	R1	D2	R2	% Dif(D)	%Dif(R)
C101	10	828.94	828.94	10	828.94	10	98.79%	0.00%
C102	10	828.94	828.02	10	828.02	10	98.79%	0.11%
C103	10	828.06	828.06	10	828.06	10	98.79%	0.00%
C104	10	824.78	828.06	10	828.06	10	98.79%	0.40%
C105	10	828.94	825.86	10	825.86	10	98.79%	0.37%
C106	10	828.94	828.94	10	828.94	10	98.79%	0.00%
C107	10	828.94	825.31	10	825.31	10	98.79%	0.44%
C108	10	828.94	828.3	10	828.3	10	98.79%	0.08%
C109	10	828.94	828.3	10	828.3	10	98.79%	0.08%
PROMEDIO	10	828.38	827.75	10.00	827.75	10	0.08%	0.00%
C201	3	591.56	591.56	3	591.56	3	99.49%	0.00%
C202	3	591.56	591.56	3	591.56	3	99.49%	0.00%
C203	3	591.17	591.17	3	591.17	3	99.49%	0.00%
C204	3	590.6	596.89	3	596.89	3	99.50%	1.07%
C205	3	588.88	588.88	3	588.88	3	99.49%	0.00%
C206	3	588.49	588.49	3	588.49	3	99.49%	0.00%
C207	3	588.29	588.29	3	588.29	3	99.49%	0.00%
C208	3	588.32	588.32	3	588.32	3	99.49%	0.00%
PROMEDIO	3	589.9	590.6	3.0	590.6	3.0	0.13%	0.00%
R101	19	1645.79	1658.13	19	1658.13	19	98.85%	0.75%
R102	17	1486.12	1516.41	18	1507.2	17	98.81%	1.42%
R103	13	1292.68	1264.5	14	1264.5	14	98.89%	2.18%
R104	9	1007.24	1054.1	10	1054.1	10	99.05%	4.65%
R105	14	1377.11	1397.25	15	1424.85	14	98.93%	3.47%
R106	12	1251.98	1314.91	13	1314.91	13	99.01%	5.03%
R107	10	1104.66	1133.46	11	1133.46	11	99.03%	2.61%
R108	9	960.88	1048.73	10	1048.73	10	99.05%	9.14%
R109	11	1194.73	1183.27	13	1199.82	12	98.90%	0.43%
R110	10	1118.59	1165.08	12	1176.63	11	98.97%	5.19%
R111	10	1096.72	1170.71	11	1170.71	11	99.06%	6.75%
R112	9	982.14	1073.97	11	1092.56	10	98.98%	11.24%
PROMEDIO	11.916	1209.9	1248.4	13.1	1253.8	12.7	3.18%	9.79%
R201	4	1252.37	1214.81	5	1309.09	4	99.59%	4.53%
R202	3	1191.7	1122.69	5	1141.58	4	99.55%	4.21%
R203	3	939.54	931.99	4	1049.99	3	99.57%	11.76%
R204	2	825.52	781.11	3	781.11	3	99.62%	5.38%
R205	3	994.42	999.81	4	1066.6	3	99.60%	7.26%
R206	3	906.14	939.84	3	939.84	3	99.68%	3.72%
R207	2	893.33	850.3	3	850.3	3	99.65%	4.82%
R208	2	726.75	770.23	3	820.84	2	99.61%	12.95%
R209	3	909.16	899.33	4	960.71	3	99.56%	5.67%
R210	3	939.34	1026.32	3	1026.32	3	99.71%	9.26%
R211	2	892.7	837.4	4.0	859.6	3.0	99.52%	3.71%
PROMEDIO	2.72	951.9	943.1	3.7	982.4	3.1	0.93%	36.67%
RC101	14	1696.94	1629.33	15	1629.33	15	99.08%	3.98%
RC102	12	1554.75	1483.41	14	1490.66	13	99.06%	4.12%
RC103	11	1261.67	1297.1	11	1297.1	11	99.15%	2.81%

RC104	10	1135.48	1206.66	11	1206.66	11	99.09%	6.27%
RC105	13	1629.44	1572.55	15	1572.55	15	99.05%	3.49%
RC106	11	1424.73	1399.48	12	1655.27	14	99.14%	16.18%
RC107	11	1230.48	1335.79	13	1342.83	12	99.03%	9.13%
RC108	10	1139.82	1180.4	11	1180.4	11	99.07%	3.56%
PROMEDIO	11.5	1384.2	1388.1	12.8	1421.9	12.8	0.28%	10.87%
RC201	4	1406.91	1303.45	6	1495.69	4	99.54%	6.31%
RC202	3	1367.09	1196.59	5	1237.31	4	99.58%	9.49%
RC203	3	1049.62	993.46	4	1144.05	3	99.60%	9.00%
RC204	3	798.41	822.44	4	833.86	3	99.51%	4.44%
RC205	4	1297.19	1252.48	6	1416.88	4	99.52%	9.23%
RC206	3	1146.32	1087.76	5	1450.51	3	99.54%	26.54%
RC207	3	1061.14	1083.65	4	1223.56	3	99.63%	15.31%
RC208	3	828.14	857.86	4	881.81	3	99.53%	6.48%
PROMEDIO	3.25	1119.35	1074.71	4.75	1210.46	3.38	3.99%	46.15%



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

ACTA DE EXAMEN DE GRADO

No. 00073

Matrícula: 2113802290

ALGORITMOS GENÉTICOS
PARALELOS PARA RESOLVER EL
PROBLEMA DE RUTAS DE
VEHÍCULOS CON VENTANAS DE
TIEMPO

En la Ciudad de México, se presentaron a las 11:00 horas del día 5 del mes de septiembre del año 2018 en la Unidad Iztapalapa de la Universidad Autónoma Metropolitana, los suscritos miembros del jurado:

DR. SANTIAGO DOMINGUEZ DOMINGUEZ
DR. MIGUEL ALFONSO CASTRO GARCIA
DR. RENE MACKINNEY ROMERO
DRA. GRACIELA ROMAN ALONSO



MARIBEL FLORES GONZALEZ
ALUMNA

Bajo la Presidencia del primero y con carácter de Secretaria la última, se reunieron para proceder al Examen de Grado cuya denominación aparece al margen, para la obtención del grado de:

MAESTRA EN CIENCIAS (CIENCIAS Y TECNOLOGIAS DE LA INFORMACION)

DE: MARIBEL FLORES GONZALEZ

y de acuerdo con el artículo 78 fracción III del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

Aprobar

REVISÓ

LIC. JULIO CESAR DE LARA ISASSI
DIRECTOR DE SISTEMAS ESCOLARES

Acto continuo, el presidente del jurado comunicó a la interesada el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.

DIRECTOR DE LA DIVISIÓN DE CBI

DR. JESUS ALBERTO OCHOA TAPIA

PRESIDENTE

DR. SANTIAGO DOMINGUEZ DOMINGUEZ

VOCAL

DR. MIGUEL ALFONSO CASTRO GARCIA

VOCAL

DR. RENE MACKINNEY ROMERO

SECRETARIA

DRA. GRACIELA ROMAN ALONSO