



**UNIVERSIDAD NACIONAL DEL SUR**

TESIS DE DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

“METAHEURÍSTICAS APLICADAS A PROBLEMAS DE TRANSPORTE”

Diego Alejandro Rodriguez

[drodriguez@plapiqui.edu.ar](mailto:drodriguez@plapiqui.edu.ar)

BAHIA BLANCA

ARGENTINA

2015



## **PREFACIO**

Esta Tesis se presenta como parte de los requisitos para optar al grado Académico de Doctor en Ciencias de la Computación, de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otra. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el ámbito del Departamento de Ciencias e Ingeniería de la Computación durante el período comprendido entre el 12 de Octubre de 2012 y el 12 de Marzo de 2015, bajo la dirección de la Dra. Nélide Beatriz Brignole, Profesora Adjunta del Dpto. de Ciencias en Ingeniería de la Computación e Investigadora Independiente del CONICET.

Diego Alejandro Rodriguez



UNIVERSIDAD NACIONAL DEL SUR

Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el ....../...../..... , mercedo la calificación de .....(.....)



## **AGRADECIMIENTOS**

En primer instancia deseo agradecerle a la Dra. Nélidea Beatriz Brignole, por su inmensa contribución en el desarrollo de esta tesis. Su experiencia y predisposición sirvieron para formarme como investigador y direccionar con éxito el presente trabajo. En este sentido deseo agradecerle a la Dra. Ana Carolina Olivera, quien desinteresadamente siempre estuvo a nuestra disposición para colaborar en lo que necesitemos. A mis compañeros y personal de PLAPIQUI, por el apoyo constante y las hermosas experiencias compartidas.

Un reconocimiento a las instituciones que asistieron y me facilitaron los medios para efectuar las tareas a lo largo de mi carrera como profesional. El Consejo Nacional de Investigaciones Científicas y Técnicas; la Planta Piloto de Ingeniería Química y el Departamento de Computación de la Universidad Nacional del Sur en Bahía Blanca; la Facultad de Ciencias Exactas en la Universidad Nacional de Salta y el Colegio Secundario Alberto Juan Lacroze N° 5032, de mi ciudad natal, Colonia Santa Rosa en la provincia de Salta.

Un especial agradecimiento a Claudia, la mujer que desde su amor, apoyo y comprensión fue capaz de acompañarme y alentarme para hacer posible la realización de mi carrera. Ella es el sostén principal de una familia maravillosa. Mis hijos Martín, Franco y Thiago, las personas que hacen que cualquier esfuerzo valga la pena.

Por último, mi gratitud al apoyo de mis padres, de ellos recibí una formación como persona que me enriqueció enormemente. A mis hermanos, mi suegra y demás familiares por su afecto y colaboración constante.

*A mis hijos  
A mi esposa  
A mis padres*

## **RESUMEN**

La idea central de esta tesis consiste en aprovechar las técnicas metaheurísticas para resolver problemas reales asociados al transporte. Por ende, en esta investigación hemos incluido el desarrollo de algoritmos metaheurísticos, su diseño, aplicación y testeo del software.

Los problemas de transporte se caracterizan por su difícil tratamiento, con respecto a su modelado. En particular, en Argentina, la calidad del servicio de transporte público ha sido considerada históricamente como inadecuada, encontrándose típicamente asociada a un reclamo continuo por parte de la sociedad.

Por lo tanto, la principal línea de investigación aquí tratada se relaciona con el transporte público de pasajeros, particularmente concentrándose en las localizaciones de las paradas de colectivos y en el establecimiento de las rutas que servirán para prestar un mejor servicio, minimizando los costos globales.

Como última parte de nuestras investigaciones, creamos una hiperheurística eficiente basada en el análisis de desempeño de distintas metaheurísticas, cuyos algoritmos individuales desarrollamos y testeamos en primera instancia. Se consideraron las siguientes metaheurísticas: Algoritmos Genéticos, Recocido Simulado y Optimización por Colonia de Hormigas.

Como resultado hemos obtenido una técnica hiperheurística que resuelve exitosamente el problema de localización y ruteo para el transporte público. El algoritmo propuesto selecciona y aplica la metaheurística apropiada para cada momento, a fin de obtener un proceso de búsqueda exitoso. Comparaciones con procedimientos bien establecidos reflejan el impacto de un procedimiento hiperheurístico sobre la optimización completa.

Las principales contribuciones de esta tesis se pueden resumir como sigue:

- Nuevas propuestas para el tratamiento de un problema que aqueja a gran parte de la población mundial como lo es el transporte público.
- Aplicación exitosa del enfoque en problemas de transporte no relacionado a las personas, tal como la localización de cañerías.
- Técnicas que reducen el número de evaluaciones improductivas mediante la implementación de una memoria interna.
- Novedosa selección de vecinos que acelera el proceso de búsqueda.
- La capacidad de enlazar nuestra propuesta con software externo.
- Estrategia paralela que ofrece notables mejoras en cuanto al tiempo de cómputo a la vez que optimiza los recursos disponibles.



## **ABSTRACT**

The thesis statement concerns taking advantage of metaheuristic techniques as a means to solve realistic problems associated to transport. Then, in this research we have included the development of metaheuristic algorithms, their design, application and software testing.

Transport problems are characterized by their difficult treatment, as regards their modeling. In particular, in Argentina, the quality of public transport service has historically been deemed inadequate, being typically marked by continual social demands.

Therefore, the main points that this research covered is related to public transport of passengers, particularly concentrating on bus-stop locations and bus-route design so as to offer a better service while minimizing global costs.

As the last part of our research work, an efficient hyper-heuristics was created on the basis of the performance analysis of various meta-heuristics, whose individual algorithms were firstly developed and tested. The following meta-heuristics were considered: Genetic Algorithms, Simulated Annealing and Ant Colony Optimization.

As a result, we have obtained a hyperheuristic technique that succeeds in solving the location and routing problem for the public transport. The proposed algorithm chooses and applies the most adequate metaheuristic at any time instance, in order to obtain a successful search process. The comparisons with the well-established procedures reflect the impact of a hyperheuristic procedure on the optimization as a whole.

The main contributions of this thesis can be summarized as follows:

- New proposals for the treatment of a problem that concerns a wide amount of the world population, like the public transport.

- Successful application of this approach to transport problems unrelated to persons, like pipeline routing.
- Techniques that reduce the number of unproductive evaluations by means of an internal memory implementation.
- Novel neighbour choice that speeds up the search process.
- The capacity to link our proposal with external software.
- Parallel strategy that offers noteworthy improvements as regards computing times, while optimizing the available resources.

# ÍNDICE

## CONTENIDO

INDICE DE FIGURAS.....	xii
INDICE DE TABLAS.....	xiii
INDICE DE ALGORITMOS.....	xiii
<b>CAPÍTULO 1 .....</b>	<b>1</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
1.1    MOTIVACIÓN.....	2
1.2    OBJETIVOS.....	3
1.3    METODOLOGÍA GENERAL .....	4
1.4    ORGANIZACIÓN DE LA TESIS .....	6
<b>CAPÍTULO 2 .....</b>	<b>9</b>
<b>PROBLEMA DE TRANSPORTE.....</b>	<b>9</b>
2.1    INTRODUCCIÓN.....	10
2.2    TRANSPORTE PÚBLICO.....	11
2.2.1    PARADAS .....	14
2.2.2    RECORRIDOS .....	15
2.2.3    FRECUENCIAS .....	15
2.3    MODELO GENERAL DE TRANSPORTE .....	15
<b>CAPÍTULO 3 .....</b>	<b>21</b>
<b>METAHEURÍSTICAS .....</b>	<b>21</b>
3.1    METODOS HEURÍSTICOS .....	22
3.2    METAHEURISTICAS ADOPTADAS .....	26
3.3    EVALUACION COMPARATIVA.....	34
3.4    HIBRIDACIÓN.....	36
3.5    ESTRATEGIAS PARALELAS.....	37
<b>CAPÍTULO 4 .....</b>	<b>39</b>
<b>METAHEURÍSTICAS APLICADAS AL IPTP .....</b>	<b>39</b>
4.1    INTRODUCCIÓN.....	40
4.2    IPTP: PLANTEAMIENTO DEL PROBLEMA.....	42

4.2.1	IPTP: REPRESENTACION DE LA SOLUCIÓN .....	42
4.2.2	IPTP: FUNCIÓN DE APTITUD.....	43
4.3	IPTP: LÍNEA INTERURBANA BAHÍA BLANCA – PUNTA ALTA.....	44
4.4	EVALUACIÓN COMPARATIVA DE GA Y SA.....	45
4.4.1	RESULTADOS Y CONCLUSIONES DE GA vs. SA.....	46
4.5	ADAPTACIÓN DE UN MODELO SA.....	49
4.5.1	RESULTADOS Y CONCLUSIONES DE ASA .....	52
4.6	ESTRATEGIA PARALELA DE SA.....	54
4.6.1	RESULTADOS Y CONCLUSIONES DE SA PARALELO .....	55
<b>CAPÍTULO 5 .....</b>		<b>59</b>
<b>METAHEURÍSTICAS PARA PROBLEMAS DE LOCALIZACIÓN Y RUTEO .....</b>		<b>59</b>
5.1	INTRODUCCIÓN.....	60
5.2	CASO DE ESTUDIO: TRANSPORTE PÚBLICO .....	61
5.2.1	IMPLEMENTACIÓN .....	64
5.2.2	RESULTADOS Y CONCLUSIONES.....	67
5.3	CASO DE ESTUDIO: RED DE CAÑERIAS .....	70
5.3.1	METODOLOGÍAS DE RESOLUCIÓN .....	73
5.3.2	DISCUSIÓN DE RESULTADOS.....	76
<b>CAPÍTULO 6 .....</b>		<b>79</b>
<b>HIPERHEURÍSTICAS .....</b>		<b>79</b>
6.1	INTRODUCCIÓN.....	80
6.2	CLASIFICACIÓN.....	82
6.3	HIPERHEURÍSTICA PARALELA .....	83
<b>CAPÍTULO 7 .....</b>		<b>87</b>
<b>DISEÑO DE UNA HIPERHEURÍSTICA PARA EL LRP.....</b>		<b>87</b>
7.1	INTRODUCCIÓN.....	88
7.2	MODELO PROPUESTO DE HIPERHEURISTICA PARA LRP .....	90
7.3	IMPLEMENTACIÓN PARALELA.....	93
7.4	EVALUACIÓN DE RESULTADOS.....	94
<b>CAPÍTULO 8 .....</b>		<b>101</b>
<b>CONCLUSIONES.....</b>		<b>101</b>
8.1	CONCLUSIONES.....	102
8.2	PRINCIPALES CONTRIBUCIONES CIENTIFICAS .....	106

8.3 TRABAJOS FUTUROS .....	107
<b>REFERENCIAS.....</b>	<b>111</b>

**INDICE DE FIGURAS**

Figura 2.1: Importancia del transporte público.....	13
Figura 2.2: Representación del modelo con un grafo dirigido.....	19
Figura 3.1: Diagrama de un GA Canónico.....	27
Figura 4.1: Mapa en OpenStreetMap y el resultado obtenido en SUMO.....	41
Figura 4.2: Representación de la solución para el IPTP.....	43
Figura 4.3: Evolución del Algoritmo Genético (a) y Simulated Annealing (b).....	47
Figura 4.4: Diagrama de caja con los resultados obtenidos por GA y SA.....	48
Figura 4.5: Un ejemplo del método de selección de vecinos.....	50
Figura 4.6: Diagrama de flujo de nuestro proceso ASA.....	51
Figura 4.7: Evolución del algoritmo ASA.....	52
Figura 4.8: Llamadas evitadas al simulador.....	53
Figura 4.9: Topología Maestro-Esclavo.....	55
Figura 4.10: Evolución del algoritmo Paralelo y Secuencial.....	57
Figura 5.1: Representación de una instancia de este modelo.....	63
Figura 5.2: Inspiración y comportamiento de la metaheurística ACO.....	66
Figura 5.3: Evolución de ACO en una ejecución completa.....	69
Figura 5.4: Progreso de las iteraciones para la instancia 1.....	69
Figura 5.5: Esquema del problema.....	71
Figura 5.6: Diagrama SA - GAMS.....	74
Figura 5.7: Modificaciones de la matriz de costos cuando se descarta el nodo N2.....	75
Figura 5.8: Convergencia del Algoritmo SA - GAMS.....	77
Figura 7.1: Representación de un prototipo.....	90
Figura 7.2: Modelo de la hiperheurística.....	93
Figura 7.3: Evolución de los algoritmos.....	97
Figura 7.4: Aceleración de la convergencia (a) – Escape de óptimos locales (b).....	98
Figura 7.5: Soluciones encontradas por las distintas técnicas.....	99

**INDICE DE TABLAS**

Tabla 3.1: Clasificación de las técnicas metaheurísticas.....	26
Tabla 4.1: Configuración de parámetros para el Algoritmo Genético.....	46
Tabla 4.2: Configuración de parámetros para Simulated Annealing.....	46
Tabla 4.3: Media, mediana y desviación estándar para los valores de aptitud obtenidos.....	48
Tabla 4.4: Valores de los parámetros del algoritmo.....	55
Tabla 4.5: Comparación del algoritmo secuencial con el paralelo.....	56
Tabla 5.1: Restricciones del modelo.....	65
Tabla 5.2: Parámetros utilizados durante las simulaciones.....	66
Tabla 5.3: Resultados computacionales para las instancias consideradas.....	67
Tabla 5.4: Resultados computacionales para la instancia 3, con autobuses de 38 y 45 pasajeros..	70
Tabla 5.5: Resultados obtenidos para los dos ejemplos presentados.....	76
Tabla 6.1: Taxonomía de Flynn.....	84
Tabla 7.1: Resultados computacionales para los casos estudiados.....	95
Tabla 7.2: Comparación de Algoritmos.....	95
Tabla 7.3: Comparación de Algoritmos.....	96

**INDICE DE ALGORITMOS**

Algoritmo 3.1: Pseudocódigo GA.....	29
Algoritmo 3.2: Pseudocódigo SA.....	31
Algoritmo 3.3: Pseudocódigo ACO.....	33
Algoritmo 7.1: Pseudocódigo HHP.....	92





# CAPÍTULO 1

## INTRODUCCIÓN

---

### *Contenidos de este capítulo*

En este capítulo se presentan las motivaciones que dieron origen a las investigaciones sobre metaheurísticas aplicadas a problemas de transporte. Se realiza una introducción sobre la complejidad de dichos problemas y se justifica la necesidad del presente trabajo.

Aquí se definen los objetivos que se pretenden conseguir, la metodología y los medios utilizados para obtenerlos. Finalmente, se muestra la forma en que está organizada esta tesis, con una breve explicación de cada capítulo.

## 1.1 MOTIVACIÓN

El presente trabajo surge ante la necesidad de generar técnicas que puedan aplicarse a problemas de transporte. Su aplicación está orientada a casos de gran interés y relevancia para la comunidad. Nuestra investigación se enfoca en proponer métodos basados en metaheurísticas que posteriormente pasan a ser parte de una hiperheurística. La hiperheurística es capaz de obtener soluciones satisfactorias adaptándose fácilmente a distintos problemas y puede ser puesta en práctica rápidamente.

Las metaheurísticas son técnicas modernas, de reciente aparición con resultados exitosos comprobados. Para nuestras investigaciones se eligió trabajar con: Recocido Simulado (SA: Simulated Annealing), Algoritmo Genético (GA: Genetic Algorithm) y Optimización basada en Colonia de Hormigas (ACO: Ant Colony Optimization). Una vez estudiados los beneficios y falencias de estos métodos se pretende obtener una técnica general, que aproveche el conocimiento adquirido en su ejecución. Como estrategia se adopta el concepto de hiperheurística a los fines de que mediante la colaboración de las metaheurísticas se obtengan mejores resultados.

En primera instancia, se desarrollaron técnicas para ser aplicadas al Problema de Ruteo de Vehículos (VRP: Vehicle Routing Problem). Utilizamos una variante a dicho problema que se denomina Problema de Transporte Público Interurbano (IPTP: Intercity Public Transport Problem). Finalmente, se trabajó sobre el Problema de Localización y Ruteo (LRP: Location-Routing Problem).

Dentro de las investigaciones en el campo de la optimización computacional, la generación de técnicas que resuelvan de manera eficiente problemas complejos de optimización es una de las inquietudes que ha cobrado mayor importancia en los últimos

tiempos. El principal reto en este sentido se presenta en la necesidad de contar con soluciones apropiadas en tiempos razonables.

Cuando se trabaja con problemas reales, la dimensión de los mismos resulta un obstáculo para la utilización de un método exacto. En muchos casos obtener la mejor solución, e incluso soluciones aceptables, resulta temporalmente imposible. En estos casos, la aplicación de técnicas metaheurísticas, aunque no siempre aseguran la obtención de la solución óptima, se considera una alternativa válida y satisfactoria. A lo largo de nuestras investigaciones, se lograron progresos significativos con respecto a diferentes metaheurísticas gracias a la utilización del paralelismo intrínseco en los métodos desarrollados, el empleo de memorias auxiliares, entre otros.

Las técnicas analizadas, diseñadas y desarrolladas aquí se enfocan en el contexto de la resolución de problemas asociados al transporte, especialmente al transporte público de pasajeros. Tanto el VRP como el LRP resultan de especial interés debido a su importancia para la sociedad en su conjunto y además por su complejidad computacional. Debido a la gran cantidad de variantes asociadas a estos reconocidos problemas, surge la necesidad de contar con una técnica lo suficientemente general para que se adapte a cualquiera de estos problemas sin mayores inconvenientes. La generalidad de nuestra propuesta es una contribución interesante de nuestro desarrollo en la resolución de los problemas de transporte.

## **1.2 OBJETIVOS**

El objetivo principal de esta tesis doctoral consiste en generar técnicas basadas en metaheurísticas que resuelvan con éxito problemas asociados al transporte. Para cumplir con el objetivo propuesto se plantearon los siguientes objetivos específicos y se establecieron los beneficios que proporcionan nuestras investigaciones:

- **Estudio, definición y formulación de los problemas VRP y LRP.** Estos problemas son de gran relevancia social. En la actualidad, se invierten muchos recursos para mejorar tanto el transporte de personas como el transporte de mercadería, un progreso en este sentido es beneficioso para el gobierno, las industrias y la sociedad en su conjunto.
- **Análisis, diseño e implementación de técnicas basadas en metaheurísticas.** Desde el punto de vista científico, profundizar en el diseño de técnicas que resuelven problemas que no pueden ser abordados por técnicas exactas implica acercarnos a mejores soluciones que las actuales. Comparar y analizar las ventajas y desventajas de las metaheurísticas estudiadas sirve como preludeo para la obtención de mejores técnicas.
- **Diseño de estrategias que permitan ejecuciones paralelas.** El estudio de distintas maneras de llevar a cabo ejecuciones paralelas a los fines de obtener algoritmos con mejores soluciones en tiempos de cómputos más reducidos.
- **Consecución de algoritmos que optimicen los recursos y se adapten a las características del problema.** El rendimiento de los algoritmos es superior cuando hacemos un uso eficiente de los recursos de hardware disponibles. Además, es deseable que la implementación sea capaz de adaptarse a características propias de cada problema.

### 1.3 METODOLOGÍA GENERAL

La metodología de trabajo propuesta sigue las bases de la computación científica. Conformamos un grupo interdisciplinario que pretende integrar la investigación científica con los procesos ingenieriles para la conformación de modelos precisos y de fácil interpretación.

Además, aprovechar los recursos de software y hardware existentes a los fines de obtener soluciones competitivas.

Las investigaciones realizadas incluyen las siguientes fases elementales que se desarrollaron en cada uno de los trabajos presentados.

- i. Creación de modelos matemáticos que representan aspectos de un sistema trascendente.
- ii. Generación de algoritmos que resuelven el modelo planteado.
- iii. Implementación de programas de computadoras que ejecutan los algoritmos propuestos, y se complementan con otros programas informáticos.
- iv. Representar los resultados de manera clara, utilizando herramientas gráficas y elementos estadísticos para facilitar la comprensión.
- v. Interpretar y validar los resultados obtenidos, en vistas a perfeccionar el modelo actual.

Estas fases forman parte de un ciclo en donde se procuró perfeccionar la búsqueda de una solución con estrategias a corto y largo plazo.

Generamos un modelo asociado al VRP para encontrar las mejores rutas en un problema de transporte público. Luego, un modelo asociado al LRP donde se incorpora la necesidad de encontrar las mejores ubicaciones para la instalación de las paradas de autobuses del mismo sistema de transporte.

Desarrollamos algoritmos empleando lenguajes y herramientas de desarrollo modernas y amigables como C++ y Java. Utilizamos equipos informáticos estándares y otros con procesadores multinúcleo con el objeto de maximizar los rendimientos mediante implementaciones paralelas. Los algoritmos desarrollados están basados en las metaheurísticas

GA, SA y ACO. Finalizamos con la implementación de una hiperheurística que reúne las mejores condiciones de las tres técnicas mencionadas.

Complementamos nuestros algoritmos con otros programas que aportan beneficios para el progreso del modelo. MALLBA (Alba et al., 2007) es una biblioteca de esqueletos algorítmicos para optimización combinatoria que simplifican la implementación de aspectos predeterminados de un algoritmo. SUMO (Krajzewicz et al., 2002) nos permite simular un escenario de transporte, contemplando características que son difíciles de incorporar en un modelo matemático. GAMS-CPLEX (Brooke et al., 1998) es una plataforma que nos permite ejecutar modelos de optimización para nuestros problemas de transporte.

Los programas SUMO y GAMS fueron empleados para evaluar la calidad de las soluciones propuestas por los distintos algoritmos. Se puede considerar la utilización de un software distinto para la realización de dicha tarea, como por ejemplo un simulador de transporte como MATSim (<http://www.matsim.org/>). Por esta razón, los tiempos de cómputos presentados en este trabajo se corresponden con la ejecución de los distintos algoritmos, sin considerar el tiempo que implica obtener el valor de aptitud de las distintas soluciones. Al momento de elegir el software de simulación, se debe tener en cuenta la manera en que se proyecta el costo de evaluar las soluciones sobre el tiempo total de la ejecución.

El proceso de desarrollo de las fases mencionadas fue efectuado de manera iterativa hasta obtener un método con la robustez y la eficiencia deseada. Adicionalmente, dejamos planteadas cuestiones que podrían contribuir a perfeccionar el modelo final propuesto en esta tesis.

## **1.4 ORGANIZACIÓN DE LA TESIS**

El trabajo de esta tesis doctoral está estructurado en 8 capítulos de la siguiente forma:

**Capítulo 1. *Introducción*:** En el presente capítulo se exponen las principales razones que motivaron el desarrollo de esta tesis. Se explica la metodología de trabajo y se plantean los objetivos que se pretenden alcanzar con esta tesis y se expone un resumen de cada capítulo.

**Capítulo 2. *Problemas de Transporte*:** Se enmarcan los problemas de transporte dentro de los problemas combinatorios. Se destacan aspectos como la complejidad y el tiempo de cómputo asociado. En particular, para el transporte público de pasajeros se especifican detalles relacionados a las paradas, los recorridos y las frecuencias de los autobuses. Finalmente se plantea un modelo general que representa la base para los problemas tratados.

**Capítulo 3. *Metaheurísticas*:** Se describen los orígenes de las metaheurísticas y sus fundamentos, se muestra un estudio de las tres técnicas (GA, SA y ACO) que fueron utilizadas, una evaluación comparativa sobre las mismas y una explicación de las acciones que suelen llevarse a cabo para obtener mejores resultados, como la hibridación y el paralelismo.

**Capítulo 4. *Metaheurísticas aplicadas al IPTP*:** Se discute el problema de transporte público interurbano junto a la implementación de una técnica basada en Algoritmos Genéticos y otro basada en Simulated Annealing. De una comparación entre las dos opciones se resuelve la utilización de SA para las siguientes investigaciones. Al SA canónico le introducimos varias mejoras con el objetivo de obtener mejores soluciones y disminuir los tiempos de cómputo.

**Capítulo 5. *Metaheurísticas para problemas de localización y ruteo*:** Se explica el problema de localización y ruteo. Se elige la metaheurística basada en Optimización por Colonia de Hormigas para resolver dos instancias asociadas al problema: a) La localización de paradas de autobuses y la selección de los recorridos para una línea de transporte público y b) La localización de almacenes intermedios y el diseño de una red de cañerías para el traslado de gas natural desde distintos yacimientos hacia plantas de procesamiento.

**Capítulo 6. *Hiperheurísticas*:** Se presentan los orígenes de las hiperheurísticas, distintas formas de clasificarlas, los beneficios que proporcionan y las nuevas tendencias para su utilización.

**Capítulo 7. *Diseño de una hiperheurística para IPTP*:** Se describe un algoritmo hiperheurístico basado en metaheurísticas. El uso de una técnica en el momento apropiado nos permite aprovechar las ventajas propias de cada método y contrarrestar las falencias observadas. El uso inteligente de distintos métodos, que además pueden ser ejecutados en paralelo, permite obtener una técnica novedosa y eficiente. Además, es fácil de implementar, ya que se adapta fácilmente a distintos problemas sin demasiadas modificaciones.

**Capítulo 8. *Conclusiones*:** Se exponen las conclusiones principales de la tesis, una lista de las publicaciones presentadas y se establecen líneas de investigación que podrían ser resueltas en un trabajo futuro.



# CAPÍTULO 2

## PROBLEMA DE TRANSPORTE

---

### *Contenidos de este capítulo*

Se enuncian los lineamientos generales del problema de transporte y su relación a los problemas combinatorios. Nos enfocamos en la complejidad de estos tipos de problemas, prestando especial interés en los tiempos de cómputos que insumen los algoritmos que lo resuelven.

Se describen las características especiales que incorpora un problema de transporte cuando efectúa el traslado de personas. Se remarcan conceptos importantes para el transporte público como lo son: las paradas para subidas y bajadas de pasajeros, el recorrido establecido para cada uno de los autobuses y la frecuencia determinada de los mismos.

Se plantea un modelo general para el problema del transporte, que sirve de base para las investigaciones desarrolladas en esta tesis. Dicho modelo se complementa al incorporar las variables relacionadas al transporte público y la posibilidad de localizaciones intermedias.

## 2.1 INTRODUCCIÓN

El problema de transporte ha motivado distintas investigaciones en todo el mundo en los últimos años. Recibe este nombre debido a que muchas de sus aplicaciones involucran determinar la manera óptima de transportar bienes o personas, desde distintos lugares, llamados orígenes, hacia otras partes, llamadas destinos, de tal manera que se minimicen los costos totales de distribución. Las distintas características de lo que se transporta, la demanda, la existencia de almacenes, los vehículos utilizados, la estipulación de horarios a cumplir, las restricciones sobre las rutas y otras cuestiones dan lugar a una gran variedad de problemas asociados (Desrochers et al. 1990).

El problema se aborda con un enfoque de optimización combinatoria, que implica en una primera instancia, la formulación de un modelo de optimización y posteriormente su resolución algorítmica. En un problema de optimización, el objetivo es encontrar la mejor solución entre todas las posibles. Cuando se dispone de un conjunto finito de variables, los mejores valores para las variables conforman la solución óptima, el problema es combinatorio si las variables pueden adoptar valores pertenecientes a un conjunto finito. Entonces, en un problema combinatorio se busca establecer un orden específico sobre un conjunto de elementos discreto (Grotschel y Lovász, 1995).

Por lo dicho anteriormente, un problema de optimización combinatoria se define como el conjunto de todas las instancias del problema, cada una de las instancias  $I$  puede ser representada por el par  $(F, c)$ , donde  $F$  es el espacio de búsqueda (el conjunto de todas las soluciones posibles) y  $c$  es un valor asociado a la performance de la solución (Steiglitz y Papadimitriou, 1982). Dicho valor se calcula empleando una función de costo cuyos detalles dependen del problema a resolver y a la que denominamos función de aptitud. Lógicamente, cuando el espacio de soluciones crece, también aumenta la dificultad para resolver el problema.

La complejidad del problema es un factor muy importante que requiere una elaboración especial. En un trabajo científico, tradicionalmente se construye un modelo capaz de simplificar la realidad para posteriormente resolverlo encontrando una solución precisa (exacta). Sin embargo, esta solución se corresponde con el modelo, no con el problema real.

Las ciencias de la complejidad tienen una idea contrapuesta. Consisten en construir un modelo cercano a la realidad pensando en un conjunto de problemas, para que empleando técnicas aproximadas, como por ejemplo las metaheurísticas, sea posible obtener varias soluciones referidas al problema real conformando el espacio de soluciones (Maldonado y Gómez Cruz, 2010). Con este enfoque, es posible clasificar a los problemas que tienen una representación algorítmica en aquellos que pueden resolverse en un tiempo polinomial (problemas P), y aquellos que pueden resolverse de forma aproximada con un algoritmo no determinista en un tiempo igualmente polinomial (problemas NP). Por último, los problemas más difíciles de resolver en NP se denominan NP-Complejos (Cook, 1971).

## **2.2 TRANSPORTE PÚBLICO**

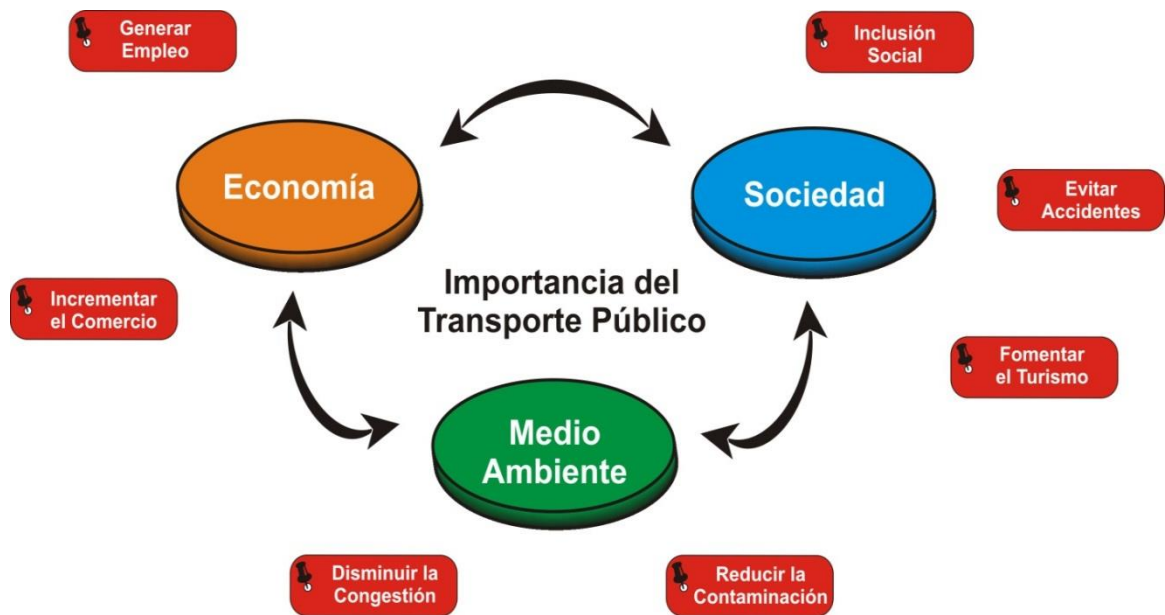
Cuando se trasladan personas de un lugar a otro en forma colectiva, estamos en presencia de un servicio de transporte público. La principal diferencia respecto al transporte privado es que los pasajeros no son propietarios del medio de transporte. Además, los usuarios deben adaptarse a los lugares de paradas, las rutas que recorren y las frecuencias en que pasan los vehículos. Muchas veces existe una regulación que fija valores de algunos componentes del sistema para asegurar el servicio social del transporte, en este caso las autoridades juegan un rol fundamental en la planificación.

El funcionamiento del sistema de transporte público impacta directamente en el desarrollo de una región. Es un desafío permanente para las autoridades encargadas de su buen funcionamiento puesto que esta actividad afecta directamente en la economía, la sociedad y el

medio ambiente. La Figura 2.1 presenta los principales factores que se ven afectados con el rendimiento del transporte público. La expansión urbana, el aumento del número de vehículos particulares y la falta de alternativas a la utilización de estos presentan un panorama desfavorable.

Entendiendo que el traslado de personas en auto particulares es perjudicial porque genera gran cantidad de contaminantes, accidentes, congestión, etc. Consideramos el transporte público como esencial para la generación de empleo, fomentar el turismo e incrementar el comercio. Se considera necesario establecer políticas de funcionamiento, que contribuyan a una solución sobre el complejo problema del transporte en una gran ciudad.

Por esto, vale la pena mejorar este servicio a la vez que se reducen los costos de operación, representando una ventaja para el operador y los usuarios. Además, cuando las medidas se vuelven obsoletas e ineficientes, es necesario reconsiderarlas. Las nuevas formas de movilidad, eficiencia, costo y conveniencia, deben contemplarse a fin de satisfacer las demandas y necesidades actuales. En nuestras investigaciones se realizó un análisis sistemático para definir las frecuencias de los vehículos, la identificación de las rutas y la selección de las paradas, con el objetivo de lograr un sistema de transporte efectivo y económico.



*Figura 2.1: Importancia del transporte público.*

La importancia asociada al transporte público genera la necesidad de desarrollar técnicas y estrategias que proporcionen soluciones competitivas resolviendo el problema en tiempos razonables. En este sentido, hay algunas contribuciones importantes que merecen destacarse: Yan y Chen (2002) han propuesto una red espacio-tiempo que refleja el movimiento de vehículos y pasajeros. Mauttone y Urquhart (2009) han desarrollado un algoritmo heurístico para generar soluciones iniciales que mejoren el proceso de solución del problema de diseño de red de tránsito. En Olivera et al. (2009) han presentado un nuevo método con el fin de evaluar las posibles soluciones al problema de programación de la red de autobuses. Xinchao (2011) ha abordado el diseño de rutas y el ajuste de las frecuencias de forma simultánea mediante el uso de un algoritmo genético hibridado con una heurística de búsqueda por vecindario. Finalmente, en Cipriani et al. (2012) se ha propuesto un conjunto de heurísticas que resuelven un problema de diseño de la red de autobuses en una zona con un sistema de transporte multimodal.

En Ortúzar y Willumsen (2011) se presenta una introducción a los modelos de transporte de uso general en los escenarios urbanos. En particular, se discute la zonificación y la construcción de redes de transporte. En Mauttone (2005) se hace hincapié en que el nivel de detalle adoptado es una de las decisiones más importantes que se debe tomar al inicio del proceso de modelado. Hay algunos estándares ampliamente aceptados para dividir el área de estudio en zonas. Estas normas se centran en los modelos de transporte de propósito general, que por lo general trabajan sobre la base de varias etapas: generación, distribución, división modal y asignación. La zonificación representa una entrada para los modelos de transporte de propósito general y también participa en la recolección y procesamiento de los datos de transporte. Por otra parte, en la mayoría de los estudios de planificación del transporte, uno de los primeros pasos es la definición de un esquema de zonificación con el fin de dividir la zona de estudio y, en consecuencia discretizar el espacio correspondiente.

### **2.2.1 PARADAS**

Determinar el número de paradas a utilizar y sus localizaciones es crucial para la planificación del transporte público. Influye directamente en la accesibilidad, desempeño y nivel de servicio. Una mala elección en este sentido puede derivar en un incremento en el tiempo de viaje y mayores congestiones en las zonas donde están las paradas.

Murray (2003) estudia dos variaciones de este problema. En la primera parte, considera la localización de las paradas en una red existente dentro del objetivo de minimizar el número de paradas. Un problema de cubrimiento y un problema de maximización del cubrimiento son utilizadas en el modelo. La segunda parte, trata de la localización óptima de las paradas para crear o extender la red. Dando un número fijo de paradas a localizar, el objetivo es maximizar el acceso al servicio en las áreas no cubiertas.

### **2.2.2 RECORRIDOS**

Las rutas disponibles para una línea de transporte público es la entrada principal para constituir los recorridos. Los períodos planificados de servicio (tiempo del primer y último viaje y el tiempo de recorrido) también son necesarios para computar el cronograma. Cuestiones como el nivel de congestionamiento y el estado de las rutas, así como la incorporación de los nuevos barrios influyen en el establecimiento de los recorridos. También es importante contar con información sobre la demanda, conocer las necesidades de transporte de pasajeros es necesaria para dimensionar el servicio y hacer un uso eficiente de los recursos disponibles.

### **2.2.3 FRECUENCIAS**

La frecuencia indica el período de tiempo entre las salidas de los autobuses para iniciar el recorrido de una línea de transporte público. Existen distintas maneras de obtener la frecuencia a utilizar. Tras el estudio a un recorrido en particular, la determinación de la frecuencia puede establecerse en función a información relacionada al caso de estudio, como el tamaño de la flota, la capacidad de los vehículos, los viajes realizados o la demanda existente. Por ejemplo, la raíz cuadrada del total de viajes, la demanda del arco más cargado, dividido por la capacidad, etc. (Jansson, 1998 y Groneau, 2000).

Una vez establecidas las frecuencias, se confeccionan las tablas de tiempos para planificar las salidas de los autobuses que brinden un servicio controlado y ordenado, que maximice la sincronización de los autobuses y así minimizar costos operativos.

## **2.3 MODELO GENERAL DE TRANSPORTE**

El modelo general de transporte es un caso especial de programación lineal. Su objetivo consiste en determinar un esquema para el envío de los productos desde sus fuentes hacia sus destinos, a los fines de minimizar el costo total del transporte y que a la vez se satisfagan los

límites de la oferta y la demanda. Existen múltiples problemas en el ámbito comercial, industrial, etc. que no tienen relación al transporte pero que pueden ser modelados de la misma manera. En este contexto, podemos representar la existencia de  $m$  orígenes,  $n$  destinos y las siguientes variables:

$s_i$  = Cantidad de oferta en el origen  $i$ .

$d_j$  = Cantidad de demanda en el destino  $j$ .

$c_{ij}$  = Costo de transporte unitario entre el origen  $i$  y el destino  $j$ .

$x_{ij}$  = Cantidad transportada desde el origen  $i$  al destino  $j$

La formulación matemática del problema de transporte en su forma estándar es la siguiente:

$$\text{Min } Z = \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} c_{ij} x_{ij} \quad (1)$$

El objetivo consiste en determinar la cantidad transportada desde los orígenes ( $i$ ) a los destinos ( $j$ ) a los fines de minimizar el costo total del transporte. Satisfaciendo las restricciones de oferta y demanda establecidas.

Restricciones en el origen:

$$\sum_{j=1}^{j=n} x_{ij} = s_{ij} \quad (i = 1, 2, \dots, m) \quad (2)$$

Se debe transportar la totalidad de la oferta existente en cada uno de los orígenes ( $s_i$ )

Restricciones en el destino:

$$\sum_{i=1}^{i=m} x_{ij} = d_{ij} \quad (j = 1, 2, \dots, n) \quad (3)$$

Se debe satisfacer la demanda existente en cada uno de los destinos ( $d_j$ )



Y además

$$x_{ij} \geq 0 \quad (i = 1, 2, \dots, m) / (j = 1, 2, \dots, n) \quad (4)$$

Se establecen condiciones de no negatividad para  $x_{ij} \geq 0$  Para todo  $i, j$ .

Cuando se debe determinar un conjunto de rutas para una flota de vehículos que parten de uno o más depósitos para satisfacer la demanda de clientes dispersos geográficamente, estamos en presencia de la clase de problemas denominada VRP. El VRP fue introducido en Dantzig y Ramser (1959), es uno de los problemas de optimización combinatoria más estudiados en los últimos años. Múltiples métodos fueron propuestos para resolver problemas relacionados con el VRP (Maffioli, 2003; Dondo et al., 2003; Giaglis et al., 2004; Sa'adah y Paecher, 2004; Dooley et al., 2005; Alvarenga et al., 2007).

Una de las variantes más utilizadas consiste en el empleo de vehículos con una capacidad finita, la misma se denomina Problema de Ruteo de Vehículos Capacitados (CVRP: Capacitated Vehicle Routing Problem). En este caso, se puede expresar matemáticamente como un grafo dirigido  $G = (V, E)$ , donde  $V$  es el conjunto de nodos que representan las ciudades o clientes y  $E$  es el conjunto de arcos que los conectan, relacionados con la matriz de costos  $C = (c_{ij})$ , de tamaño  $N \times N$ , de modo que cada arco tiene asignado un costo  $c_{ij}$ . Un arreglo de la forma  $(Q_k)$  que contiene los datos de capacidad máxima de los vehículos que integran una flota compuesta por  $M$  unidades, es decir,  $1 \leq k \leq M$ . Entonces, la variable  $x_{jik}$  representa la cantidad transportada desde el origen  $i$  al destino  $j$ , mediante el vehículo  $k$ .

Así, el problema puede expresarse como:

$$\text{Min } Z = \sum_{i=0}^{i=n} \sum_{j=0}^{j=n} \sum_{k=1}^{k=m} c_{ij} x_{ijk} \quad (5)$$

Sujeto a:

$$\sum_{k=1}^{k=m} \sum_{j=1}^{j=n} x_{ijk} \leq M \quad i = 0 \quad (6)$$

$$\sum_{k=1}^{k=m} \sum_{j=0}^{j=n} x_{ijk} = 1 \quad \forall i \in [1, N] \quad (7)$$

$$\sum_{j=1}^{j=n} x_{ijk} = \sum_{i=1}^{i=n} x_{ijk} \quad \forall k \in \{1, M\} \quad i = 0 \quad (8)$$

$$\sum_{i=0}^{i=n} \sum_{j=0}^{j=n} q_i x_{ijk} \leq Q_k \quad \forall k \in \{1, M\} \quad (9)$$

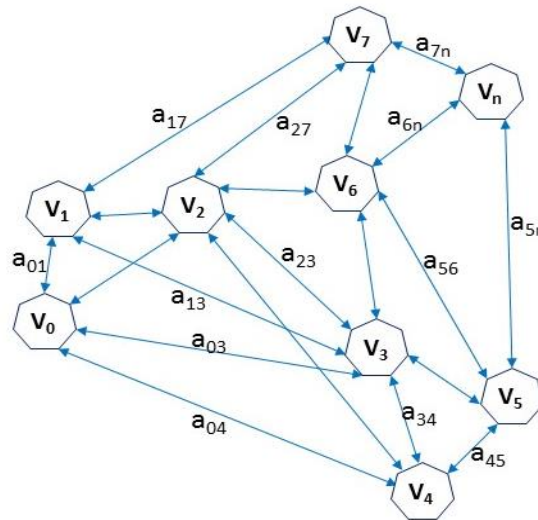
$$\sum_{\substack{i=1 \\ i \in S}}^{i=n} \sum_{\substack{j=1 \\ j \in S}}^{j=n} x_{ijk} \leq |S| - 1 \quad \forall S \subseteq (V - \{0\}) \\ |S| \geq 2 \\ k \in \{1, M\} \\ \forall i, j \in [1, N] \quad (10)$$

$$x_{ijk} \in \{0, 1\} \quad \forall k \in \{1, M\} \quad (11)$$

Con la restricción (6) se limita a que salgan desde el depósito inicial un máximo de  $M$  vehículos. Las restricciones (7) y (8) indican que un y sólo un vehículo visite y abandone cada cliente. La restricción (9) impide sobrepasar la capacidad vehicular. Finalmente los conjuntos de restricciones (10) y (11) imposibilitan la generación de sub-rutas inconexas y los valores admisibles para las variables de decisión, respectivamente.

Si consideramos el transporte de personas en vehículos especiales (autobuses) provistos por un operador determinado, se debe considerar un escenario con objetivos contrapuestos. El modelo general de optimización combinatoria que establece los recorridos de las líneas de transporte busca cumplir con los objetivos de los usuarios y los operadores. Los usuarios requieren minimizar los tiempos de los viajes, estos incluyen el tiempo de espera en las paradas y el viaje en el autobús. Los operadores buscan reducir el tamaño de la flota, producto de los costos asociados a su puesta en marcha.

En la estructura modelada mediante grafos, el conjunto de vértices representan a las paradas de autobuses, es el lugar de atracción para los pasajeros que existen en la zona. Los arcos representan un segmento de la ruta que une dos paradas cualesquiera (Figura 2.2).



**Figura 2.2:** Representación del modelo con un grafo dirigido.

En particular, consideramos el Problema de Transporte Público Interurbano como una variante del VRP (Rodríguez et al., 2011). Una solución al modelo es el conjunto de secuencias de paradas que conforman las líneas de transporte sobre la cual opera los autobuses. Durante la búsqueda de las soluciones se pretende minimizar el costo total, comprendido por los costos asociados a los usuarios y el costo asociado a los operadores. Aquí, se debe establecer la importancia relativa para cada uno de los términos que intervienen en la solución objetivo ( $\alpha$ ). Debido a que generalmente esta es una decisión política, surge la participación de un nuevo actor, la administración pública. Su actividad consiste en proporcionar directivas que regulen el servicio, las cuales deben ser incorporadas al modelo.

Entonces, se incorpora al modelo un término que esté asociado a los tiempos de los viajes ( $t_{ij}$ ) representando el tiempo de viaje de un usuario desde la ubicación  $i$  hasta la ubicación  $j$  (Ecuación 12).

$$\text{Min } Z = \alpha_1 \sum_{\substack{i,j \in N \\ k \in K}} c_{ijk} x_{ijk} + \alpha_2 \sum_{i,j \in N} t_{ij} \quad (12)$$

Cuando el planteamiento del problema no supone los nodos como preestablecidos, sino que considera la posibilidad de elegir su instalación en múltiples lugares de la región en estudio, estamos en presencia de un problema de localización y ruteo. En estos casos se debe contemplar el costo de instalar las distintas combinaciones de nodos posibles y evaluar los costos asociados. Este contexto se corresponde con un modelo de Programación Entera Mixta (PEM). Es un híbrido entre la Programación Lineal (PL) y la Programación Entera (PE), es decir, un modelo de programación lineal donde algunas de las variables adoptan valores enteros o binarios.

En el caso del transporte público de pasajeros, este costo está asociado a la instalación de un subconjunto de potenciales paradas ( $H$ ). Entonces, se procede a incorporar al modelo un término relacionado al costo de las paradas instaladas ( $p_h$ ) y la variable  $y_h$  será igual a 1 cuando se elige instalar la parada  $h$  y 0 en caso contrario (Ecuación 13).

$$\text{Min } Z = \alpha_1 \sum_{\substack{i,j \in N \\ k \in K}} c_{ijk} x_{ijk} + \alpha_2 \sum_{i,j \in N} t_{ij} x_{ijk} + \alpha_3 \sum_{h \in H} p_h y_h \quad (13)$$

La generalidad de las formulas presentadas permite la incorporación inmediata de algunos factores que puedan llegar a ser necesarios. El manejo de restricciones se lleva a cabo mediante factores de penalización que impiden la aceptación de soluciones infactibles. El nivel de detalle con el que se construye el modelo afecta directamente en el rendimiento del algoritmo y la calidad de las soluciones. Modelos simples se ejecutan rápidamente, pero difícilmente sean aplicables a un caso real y modelos muy complejos requieren excesivos tiempos de cómputo para obtener soluciones satisfactorias.

# CAPÍTULO 3

## METAHEURÍSTICAS

---

### *Contenidos de este capítulo*

Presentamos el concepto de heurística como método de resolución de problemas combinatorios. Fundamentamos su existencia, así como los beneficios e inconvenientes asociados. Además, establecemos una clasificación que relaciona las metaheurísticas más conocidas con las principales características de búsqueda.

Se explica el funcionamiento de las tres metaheurísticas empleadas como base para la generación de nuestros algoritmos (GA, SA y ACO) y se realiza una evaluación comparativa sobre estas técnicas.

Exponemos los conceptos de hibridación y paralelismo, como alternativas diferentes para la obtención de mejores resultados.

### 3.1 METODOS HEURÍSTICOS

El término “heurística” proviene de la palabra griega eureka cuya exclamación se atribuye a Arquímedes. La Real Academia lo registró en 1936, con la definición: “Arte de inventar”, para luego extenderlo a otras definiciones. Son métodos sencillos y rápidos que resuelven un problema de manera inteligente empleando la información disponible.

Reeves (1993) define un método heurístico como una técnica que encuentra buenas soluciones a un costo computacional razonable sin garantizar la optimalidad. Las heurísticas se empezaron a usar a partir de los años 60, se fundamentan en su capacidad práctica para solucionar problemas combinatorios reales de difícil resolución. Merece destacarse la forma en que ha crecido el uso de procedimientos heurísticos en los últimos años.

Su utilización se recomienda cuando, por la naturaleza del problema, no existe un método exacto de resolución o si bien existe, éste emplea un tiempo excesivo de cómputo para ofrecer la solución. También puede utilizarse como complemento a un método exacto, por ejemplo cuando se necesita de una buena solución inicial. En otros casos, se requiere incorporar a un modelo algunas condiciones que son difíciles de hacerlo en un método exacto. Entonces, se aprovecha la flexibilidad de los métodos heurísticos y se resigna una solución óptima por una aproximada a cambio de una versión más realista del modelo.

El obtener soluciones aproximadas en lugar de la óptima no es el único inconveniente de los métodos heurísticos. Una situación usual es que son muy dependientes del problema para el que fue diseñado. De manera que cuando cambian las condiciones del problema, muchas veces no se obtiene el mismo comportamiento y se deben hacer modificaciones al método. Además, muchas veces ofrecen soluciones localmente óptimas, las cuales pueden estar muy alejadas del óptimo global. Justamente para superar esta limitación surgen las técnicas metaheurísticas.

El término metaheurística se obtiene de anteponer a heurística el sujeto meta, que significa más allá o a un nivel superior. Consiste en una estrategia de alto nivel que sirve de guía a otras heurísticas para orientar la búsqueda. Han tenido un gran desarrollo a partir de la década del 80, son métodos robustos y eficientes, muy utilizados en el área de investigación operativa.

Las metaheurísticas exploran aquellas regiones donde sea factible encontrar buenas soluciones y aplican procedimientos especiales que les permiten escapar de óptimos locales. Así, alcanzan una mayor eficiencia y efectividad en la exploración del espacio de búsqueda (Glover y Kochenberger, 2003). En este sentido, los operadores asociados a cada una de las técnicas cumplen un rol importante. La aplicación de un operador hace que se modifique la dirección de la búsqueda hacia otra vecindad y así el algoritmo puede salir de un óptimo local. Entonces, con la aplicación reiterada de operadores idóneos es más factible obtener buenos resultados, ya que difícilmente se encuentre el mismo óptimo local en distintas estructuras de vecindades.

Una característica que debería ser común a toda buena metaheurística consiste en un balance entre un proceso de intensificación para explotar el espacio de búsqueda en las cercanías de una solución de buena calidad y un proceso de diversificación para explorar diversas regiones del espacio de búsqueda. El balance entre exploración y explotación afectará directamente la calidad de la solución encontrada y el tiempo utilizado para obtenerla.

Otra cuestión importante está asociada a la factibilidad de las soluciones proporcionadas por los métodos. Estos pueden realizar movimientos que siempre generan soluciones factibles o en algunas circunstancias generar una solución que no sea factible, en ese caso se debe proveer un procedimiento que repare esa solución convirtiéndola en factible o directamente descartar dichas soluciones. Muchas veces se suele utilizar un esquema de

penalizaciones para considerar soluciones infactibles durante la búsqueda (Gendreau et al., 1994).

También puede haber variantes en cuanto al criterio de finalización de la búsqueda. En general suelen contemplar opciones como por ejemplo:

- a) Un número máximo de iteraciones o un tiempo máximo de ejecución del algoritmo.
- b) Hasta obtener una solución de suficiente calidad. Sucede cuando la solución se considera aceptable para el problema en cuestión.
- c) No se obtiene mejoras en un número determinado de iteraciones. Esto implica un estancamiento en la búsqueda.

La primera opción es la que se usa con más frecuencia, ya que requiere de menos dificultad para establecer los valores límites. Sin embargo, se observa que la combinación de distintos criterios de paradas induce a un equilibrio en la búsqueda que contribuye al uso eficiente de los recursos.

Existe una amplia gama de metaheurísticas distintas generadas. Es posible hacer mención a un grupo reducida de ellas como las técnicas más relevantes hasta la actualidad. Este grupo podría estar integrado por las tres técnicas que utilizamos en esta tesis (GA, SA y ACO) y además: Búsqueda Adaptativa Aleatoriamente Voraz (GRASP: Greedy Randomized Adaptive Search Procedure) (Feo y Resende, 1995), Búsqueda Tabú (TS: Tabu Search) (Glover, 1990) y Optimización por Cúmulo de Partículas (PSO: Particle Swarm Optimization) (Kennedy, 2010). La elección se basa en la cantidad de veces que fueron elegidas como método de resolución y por los resultados obtenidos durante su ejecución.

En este trabajo se establece una clasificación de estas metaheurísticas a partir de:



- a) *La trayectoria considerada para la búsqueda:* Estas técnicas inician en un punto inicial. Luego proceden con la actualización de la mejor solución encontrada mediante la exploración del vecindario generando una trayectoria en la búsqueda.
- b) *El uso de poblaciones de soluciones:* Son técnicas que manipulan un conjunto de soluciones en cada iteración.
- c) *Su fuente de inspiración.* Para diferenciar aquellas metaheurísticas que se inspiran en la naturaleza para resolver el problema.
- d) *La utilización de memoria:* Hay técnicas que emplean información sobre características de soluciones de buena calidad o información obtenida durante el proceso de búsqueda. Se pretende evitar la ejecución de procesos innecesarios.
- e) *La cantidad de vecindarios evaluados:* El vecindario de una solución se define como el conjunto de soluciones que se pueden construir a partir de ella, realizando solo un movimiento de búsqueda. La posibilidad de contar con múltiples vecindarios implica la ventaja de poder buscar en distintos lugares del espacio de soluciones total.
- f) *La construcción de la solución:* Las metaheurísticas constructivas inician con una solución vacía e iterativamente van incorporando elementos hasta representar una solución completa.

En la Tabla 3.1 se puede apreciar la clasificación establecida para las principales metaheurísticas en función de las diferentes estrategias de búsqueda, considerando sus versiones canónicas. En el transcurso de los últimos años, muchos autores desarrollaron con éxito versiones de metaheurísticas más sofisticadas, incorporando características distintas a las canónicas consiguiendo mejorar los resultados obtenidos hasta ese entonces (Blum et al., 2011; Drexler et al. 2013 y Vidal et al., 2013).

Meta-heurísticas	Trayectoria (I)	Población (II)	Naturaleza (III)	Uso de memoria (IV)	Múltiple Vecindario (V)	Construc-tiva (VI)
ACO		✓	✓	✓		✓
GRASP	✓					✓
SA	✓		✓			
TS	✓			✓		
GA		✓	✓	✓	✓	
PSO		✓	✓	✓		

**Tabla 3.1:** Clasificación de las técnicas metaheurísticas.

Debemos reconocer que existe mucho material sobre metaheurísticas, incluso con clasificaciones más rigurosas a la presentada. El lector puede consultar en (Blum y Roli, 2003) para más información.

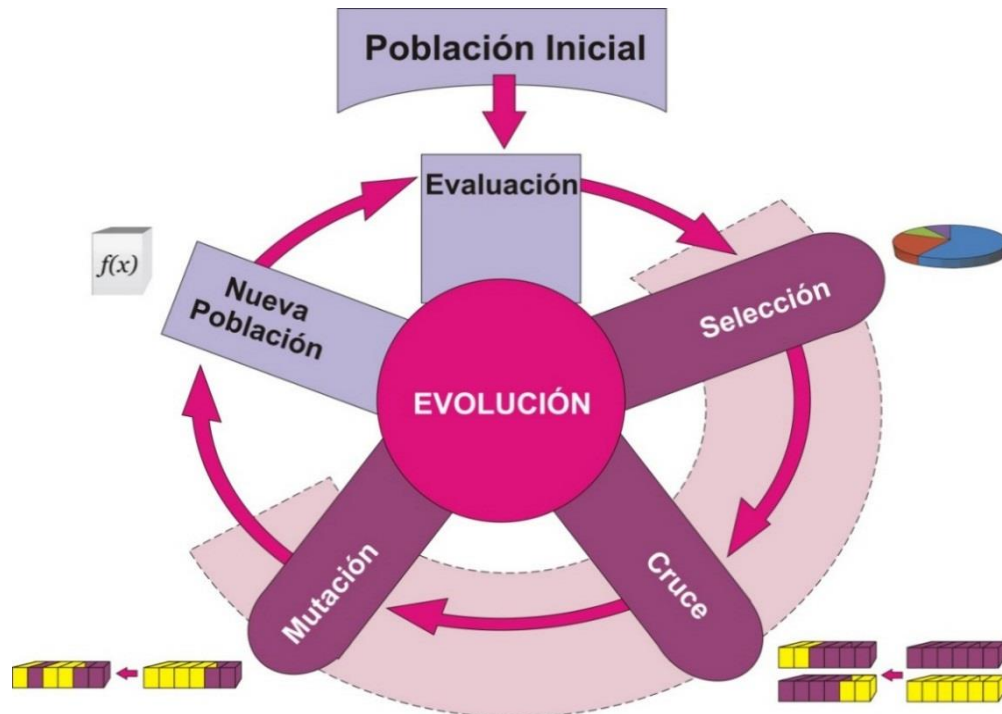
### 3.2 METAHEURISTICAS ADOPTADAS

A continuación se explican las características más importantes de las tres metaheurísticas empleadas en este trabajo: Algoritmos Genéticos, Recocido Simulado y Optimización por Colonia de Hormigas. Indicamos el origen y los principios de cada técnica, los parámetros que utilizan, sus estrategias de búsqueda y una descripción de cada uno de los algoritmos canónicos.

#### Algoritmo Genético

Los Algoritmos Genéticos son algoritmos de búsqueda basados en el principio de la evolución biológica de Darwin (Darwin, 1859). Para alcanzar la solución al problema se parte de un conjunto inicial de individuos generados de manera aleatoria. Cada uno de estos individuos representa una posible solución al problema. El conjunto de todos los individuos constituyen la población de soluciones. Los principios básicos del algoritmo fueron presentados por Holland (1975) y están explicados con suficiente detalle en Goldberg y

Holland (1988). A continuación, se hace una breve descripción de los operadores que participan en la ejecución del algoritmo (Figura 3.1).



**Figura 3.1:** Diagrama de un GA Canónico.

En analogía directa con el fenómeno de evolución en la naturaleza, se seleccionan individuos favoreciendo a los de mayor calidad, evaluados a través de una función de aptitud o fitness. Luego, por lo general estos individuos se cruzarán y sus descendientes formarán parte de la nueva población. Estos descendientes a veces mutan. La versión canónica repite este proceso en cada generación hasta que se alcanza el punto de parada (Whitley, 1994).

El algoritmo comienza con una población inicial, que puede ser elaborada con anterioridad o generada aleatoriamente. Una población de soluciones aleatorias suele usarse cuando no se posee conocimiento del problema o no se desea influenciar la búsqueda en ningún sentido. La opción de comenzar el algoritmo con soluciones que satisfacen ciertos requerimientos como para formar parte de la población inicial, provoca una convergencia hacia una solución óptima en menor tiempo.

Para que en las próximas etapas sea posible evaluar las soluciones propuestas, se debe contemplar la existencia de un proceso que asocie a cada individuo un valor proporcional de su aptitud respecto de los demás individuos de la población.

El proceso de selección se encarga de transmitir y mantener aquellas soluciones que se consideran importantes. Se considera primordial conservar los individuos de mejor aptitud para que estos luego puedan reproducirse. Aunque también es importante mantener y reproducir en menor medida, algunos individuos puedan contener información útil para futuras generaciones.

Un algoritmo genético puede utilizar distintas técnicas para seleccionar a los individuos que deben copiarse hacia la siguiente generación. Algunos de estos métodos son mutuamente exclusivos, y otros se pueden utilizar de manera combinada, algo que se hace a menudo. Algunas de las formas de selección más comunes son: selección elitista, por rueda de ruleta, por torneo, entre otras.

El cruzamiento es el operador encargado de intercambiar información genética entre un par de individuos. El objetivo consiste en obtener descendientes que mejoren la aptitud de los padres, se logra reemplazando algunos genes de un padre por los genes correspondientes del otro padre. Existen numerosas maneras de implementar este intercambio, los más empleados son los de cruce en un punto, cruce en dos puntos y cruce uniforme (Goldberg y Linge, R, 1985).

La mutación es la operación de introducir pequeñas variaciones en un individuo, consiste en reemplazar algunos de sus genes con cierta probabilidad. Este proceso, además de promover la búsqueda en nuevos sectores, participa activamente en el escape de óptimos locales. Especialmente en situaciones donde una buena solución encontrada rápidamente acaba con la diversidad genética, haciendo que la población se domine de individuos similares. El proceso de mutación más usual consiste en intercambiar un componente de la solución. Cabe

remarcar que la probabilidad de mutación considerada debe ser baja, generalmente menor al 1%, esto se debe a que los individuos mutados suelen ser de una aptitud menor a la que tenían anteriormente. Sin embargo, se considera un proceso necesario porque nos asegura que toda solución no tenga una probabilidad nula de ser examinada.

---

**Algoritmo GA:**


---

1.  $t:=0;$  // La primer generación.
  2. inicializar  $P(t);$  // Se genera una población aleatoria.
  3. **Mientras** no finalice **hacer** // No se alcancen todas las generaciones.
  4.  $t:=t+1;$  // Próxima generación.
  5.  $P(t):=selección P(t-1);$  // La mejor solución es elegida.
  6. cruzar  $P(t);$  // Las soluciones son cruzadas.
  7. mutar  $P(t);$  // Los descendientes son mutados
  8. evaluar  $P(t)$  // Se evalúa la aptitud de cada solución.
  9. **Fin Mientras**
- 

*Algoritmo 3.1: Pseudocódigo GA.*

Los operadores descritos son los responsables de obtener el equilibrio entre la exploración y la explotación. El balance dinámico del proceso de búsqueda se logra haciendo que al principio exista una alta diversificación y baja intensificación, esta situación debe ir cambiando paulatinamente para que al final del proceso se invierta esta relación. Las estrategias que se usen para los operadores afectan directamente al rendimiento del algoritmo.

### Recocido Simulado

El Recocido Simulado o Simulated Annealing es un método de búsqueda que se fundamenta en el proceso físico de calentamiento de un sólido, seguido por un enfriamiento

paulatino, hasta lograr un estado cristalino con una estructura casi perfecta (Kirkpatrick et al., 1983). Existen múltiples investigaciones empleando la versión canónica del método (Aarts y Korst ,2002), así como también versiones híbridas del mismo (Hedar y Fukushima, 2002).

El algoritmo emplea una variable llamada temperatura ( $T$ ) como parámetro de control, este valor indica en qué medida podremos aceptar una solución peor que la actual. Dicha variable inicia en un valor alto y se reduce a medida que avanza las iteraciones, en función de una variable de enfriamiento  $\alpha$ . En cada iteración, se genera un vecino de la solución actual, se evalúa su función de aptitud y se aplica el criterio de aceptación para ver si se sustituye o no a dicha solución.

El criterio de aceptación consiste en aceptar la solución vecina en caso de que sea mejor que la actual (como en un método de búsqueda clásica). Sin embargo, cuando la solución generada es peor que la mejor solución actual, es posible aceptarla como nueva solución actual dependiendo de la diferencia que exista entre la calidad de las dos soluciones ( $\Delta$ ) y de la temperatura actual  $T$ :  $P_{\text{aceptar}} = e^{(-\Delta/T)}$ . Esta última característica es la que posibilita el escape de óptimos locales.

Se puede observar que a mayor temperatura mayor es la posibilidad de aceptar peores soluciones, esto ocurre en las primeras iteraciones favoreciendo el proceso de exploración del espacio de búsqueda, y como la temperatura va disminuyendo progresivamente a medida que avanzan las mismas se intensifica el proceso de explotación. Además, a menor diferencia de calidad, mayor es la probabilidad de aceptar una solución peor.

Así se procede con la forma más habitual que consiste en diversificar al principio e intensificar al final.

**Algoritmo SA:**

- 
1.  $S_{act} \leftarrow \text{Inicialización } ();$  // Se inicializa con una solución aleatoria.
  2.  $T \leftarrow T_0;$  // Se inicializa la temperatura.
  3. **Mientras** no finalice **hacer** // No se alcancen todas las generaciones.
  4. **Desde**  $Cont=1$  hasta  $L(t)$  **hacer**; // Velocidad de enfriamiento.
  5.  $S_{cand} \leftarrow \text{SelectSolutionN}(S_{act});$  // Se elige un vecino (Nueva Solución).
  6.  $\Delta \leftarrow \text{cost}(S_{cand}) - \text{cost}(S_{act});$  // Diferencia entre la solución actual y vecina.
  7. **Si**  $(U(0,1) < e^{(-\Delta/T)})$  o  $(\Delta < 0)$  **entonces** // Aplica el criterio de aceptabilidad.
  8.  $S_{act} \leftarrow S_{cand};$  // Se reemplaza la solución.
  9. **fin Si**
  10.  $T \leftarrow \alpha(t)$  // Mecanismo de enfriamiento
  11. **Fin desde**
  12. **Fin Mientras**
- 

*Algoritmo 3.2: Pseudocódigo SA.*

### Optimización por Colonia de Hormigas

La Optimización por Colonia de Hormigas se inspira en la inteligencia colectiva de las comunidades de hormigas naturales para resolver problemas de optimización. Consiste en un procedimiento estocástico que construye soluciones simulando el mecanismo utilizado por las hormigas en su búsqueda de alimentos. Está inspirado en el depósito de feromona realizado en el camino hacia la fuente de alimentos, que luego es utilizado para reflejar la experiencia adquirida en la búsqueda de la solución (Dorigo et al., 2006).

El conjunto de hormigas coopera en la búsqueda de la solución. Esta cooperación es la base para la obtención de soluciones de alta calidad, ya que una hormiga por sí sola obtiene

soluciones pobres, producto de su visión casi nula. Una colonia de hormigas se mueve de manera estocástica en función de dos componentes: i. la información específica del problema y ii. los rastros de feromona existentes en el recorrido en ese momento.

Uno de los primeros problemas resueltos con esta metodología fue el Problema del Agente Viajero (TSP: Travelling Salesman Problem). Aquí, la elección de la hormiga consiste en seleccionar la próxima ciudad a visitar en función de la feromona depositada en los arcos que la unen (Solomon, 1987).

En los últimos años se ha observado un creciente interés en la resolución de problemas de optimización combinatoria mediante algoritmos ACO (Dorigo y Stützle, 2010). Especialmente en problemas asociados al ruteo de vehículos (Bell y McMullen, 2004). También se ha comprobado que la técnica complementada con búsqueda locales obtiene rendimientos favorables cuando el problema es muy complejo (Maniezzo et al., 2004).

La estructura de ACO se basa en cuatro componentes principales.

- La inicialización de los siguientes parámetros: La cantidad de iteraciones para el proceso de búsqueda, la cantidad de hormigas utilizadas en cada iteración, el factor de incidencia de la información determinística disponible, el factor de incidencia de la cantidad de feromona acumulada y el porcentaje de feromona perdida durante el proceso de evaporación.
- La construcción de soluciones: Cada hormiga construye de forma iterativa una solución partiendo desde un sitio inicial, en su trayecto el algoritmo utilizará los dos tipos de información antes mencionada.
- La incorporación de feromona: Esto implica modificar los valores de las variables (rastro de feromonas) que serán transmitidas a las demás hormigas para perfeccionar



sus búsquedas. Esta acción se realiza luego de que una hormiga haya concluido su recorrido, es decir, habiendo obtenido una solución al problema.

- La evaporación de feromona: Luego de que todas las hormigas hayan construido las soluciones, al final de una iteración, se realiza el proceso de evaporación de la feromona, con la intención de “olvidar” aquellos caminos que fueron obtenidos en etapas anteriores a la ejecución del algoritmo, esto es para favorecer la exploración del espacio de búsqueda y evitar la convergencia prematura que derive en un óptimo local.

---

#### Algoritmo ACO:

---

1. *InicializarParametros();* // *CantIteraciones, CantHormigas, Feromona.*
  2. **Desde** *i=1* hasta *CantIteraciones*  
**hacer**
  3. **Desde** *j=1* hasta *CantHormigas*  
**hacer**
  4. **Repetir**
  5. *Hormiga[j] ← PróximoDestino();* // *Elige el próximo destino para la hormiga j.*
  6. hasta **CompletarRecorrido**
  7. *Costo[j] ← Hormiga[j];* // *El costo asociado a la hormiga j.*
  8. *Feromona[j] ← Hormiga[j];* // *Se actualiza la feromona según el recorrido.*
  9. **Si** (*SoluciónActual < MejorSolucion*) **entonces** // *Se reemplaza la solución.*
  10. *MejorSolucion ← SolucionActual;*
  11. **Fin si**
  12. **Fin Desde**
  13. *EvaporarFeromona(Hormiga(j));*
  14. **Fin Desde**
- 

*Algoritmo 3.3: Pseudocódigo ACO.*

---

### 3.3 EVALUACION COMPARATIVA

Las metaheurísticas no aseguran la obtención de soluciones óptimas, ni desempeños similares para todos los problemas. De esta manera, al menos por ahora, no es posible establecer un ranking de las mejores técnicas. No existe una demostración de que una metaheurística sea superior a otra en todos los casos. Incluso, el teorema NFL (No Free Lunch) establece que si una metaheurística supera a otra en un conjunto de problemas, entonces es de esperar que exista una cantidad similar de problemas donde ocurra lo contrario (Wolpert y Macready, 1997).

Las principales causas por lo que no se pueden establecer comparaciones precisas tienen que ver con:

- La dependencia del problema: En la mayoría de los casos, cuando se implementa una metaheurística se la adapta al problema en cuestión. Así, cuando cambian las condiciones del problema, también se modifica su desempeño, y por ende los resultados obtenidos.
- Las múltiples variantes a las técnicas existentes: Incorporar una variante a una metaheurística también puede hacer que se obtengan resultados que la versión canónica no podía conseguir. La gran mayoría de estas alteraciones están direccionadas a modificar el proceso de búsqueda en cuestiones que tienen que ver con las tareas de intensificación y diversificación que trae consigo cada método. Algunas definiciones existentes en la literatura explican la importancia de estos procedimientos y la necesidad de obtener un equilibrio para guiar la búsqueda hacia una mejor solución (Rochat y Taillard, 1995).
- Encontrar los valores óptimos de los parámetros: Aunque existen muchas investigaciones en este sentido, teóricamente no existe la posibilidad de parametrizar de manera universal (Back y Schwefel, 1993). Generalmente, el desarrollo de un algoritmo incluye una etapa de ajuste de parámetros. Este proceso puede realizarse tras la evaluación de distintos

valores en múltiples ejecuciones o mediante el empleo de algún proceso de sintonización de parámetros (Coy et al., 2001). En la mayoría de los casos, estos valores también dependen del problema de optimización para el cual se está desarrollando el algoritmo (Hart y Bellew, 1991).

En estas condiciones, el rendimiento de una metaheurística es analizado en función de la calidad de la solución obtenida y el tiempo que le demandó al método obtenerla. También es importante considerar la simplicidad del método. Un algoritmo sencillo contribuye a su verificación y mantenimiento. Y por sobre todas las cosas hace más simple su adaptación a problemas distintos.

De la clasificación anterior, nuestra experiencia indica que las metaheurísticas basadas en trayectorias (SA) responden mejor al proceso de intensificación, y las técnicas que utilizan poblaciones de soluciones (GA y ACO) realizan una mejor diversificación de la búsqueda. Esto implica mejores tiempos computacionales para la primera, pero a cambio de una mayor probabilidad de quedar atrapado en un óptimo local.

Respecto a los tiempos de cómputo, también notamos ventajas de SA respecto a GA y ACO, remarcando nuevamente que los valores encontrados por SA son óptimos locales que muchas veces se ven superados por las soluciones de las técnicas poblacionales. Esta característica se ve acentuada cuando se configuran los algoritmos para que dispongan de un mayor tiempo de búsqueda, como por ejemplo incrementar el número de generaciones.

En lo que se refiere a la simplicidad del método, existe concordancia de que SA es uno de los métodos más sencillos y fáciles de implementar. Sin embargo, ninguno de los métodos enunciados debería representar mayores inconvenientes al momento de su desarrollo.

Con el objeto de lograr avances en estas direcciones, la comunidad científica está haciendo hincapié en aspectos tales como, la hibridación para utilizar componentes de distintas técnicas en un solo método. Para mejorar los tiempos computacionales se recomienda generar

técnicas que realizan procesamientos en paralelo. En cuanto a la obtención de métodos simples y que se adapten fácilmente a distintos problemas se sugieren las técnicas hiperheurísticas. Para profundizar los primeros dos aspectos se redactaron los próximos dos apartados, para incursionar en el tema de las hiperheurísticas se presenta una revisión de la estrategia en el capítulo 6 y se propone una hiperheurística basada en metaheurísticas para resolver una instancia del LRP en el capítulo 7.

### **3.4 HIBRIDACIÓN**

La aplicación de metaheurísticas en instancias realmente grandes, hace que sus fortalezas y debilidades se hagan más notorias. Muchas veces la hibridación, entendida como la combinación del potencial de distintas técnicas deriva en mejoras considerables (Raidl, 2006).

En los últimos años, se incrementó el uso de estos algoritmos. El éxito de la idea se ve reflejado en la gran cantidad de aplicaciones de estas técnicas a problemas reales y el creciente número de investigaciones sobre esta temática (Talbi, 2009).

Respecto al GA, existe una tendencia a hibridar dicha técnica con métodos de búsqueda local. La decisión se fundamenta en el hecho de contar con una población finita, lo que podría implicar una interacción pobre entre los genes (El-Mihoub, 2006).

Una práctica común en las metaheurística ACO, consiste en hibridar la técnica introduciendo nuevos operadores que impidan el estancamiento en óptimos locales, producto de la acumulación de feromona. En Balseiro et al. (2011), se propone un algoritmo ACO hibridado con heurísticas de inserción para una instancia del VRP. En este caso, la inclusión de nuevos operadores facilita el proceso de inclusión de nuevos clientes en las últimas etapas del algoritmo. Por otro lado, Zhao et al. (2010) propone el algoritmo MACO (Mutated Ant Colony Optimization), el cual aplica el operador de mutación en la mejor solución después de

cada iteración. El proceso consiste en modificar uno o más elementos de la solución, para que en el caso de que la solución modificada sea mejor, quedarse con ella.

La hibridación puede consistir en combinar partes de técnicas metaheurísticas claramente distinguibles, la colaboración de un método exacto o simplemente combinar operadores o representaciones de alguna técnica. Con frecuencia se suele utilizar una técnica que encuentre zonas favorables rápidamente y a partir de ese momento utilizar un componente con la velocidad de convergencia como los basados en búsqueda local. El principal objetivo de la hibridación se corresponde con la necesidad de expandir la diversidad de soluciones y evitar la convergencia prematura.

### **3.5 ESTRATEGIAS PARALELAS**

Si bien el uso de metaheurísticas contribuye a disminuir el tiempo de cómputo para la resolución de un problema, este tiempo muchas veces continua siendo más elevado de lo necesario.

En estos casos, utilizar plataformas paralelas para la ejecución de metaheurísticas es una alternativa que puede acelerar la ejecución de los algoritmos. Así, es posible obtener soluciones en un lapso menor de tiempo e incluso muchas veces se mejora la calidad de las soluciones encontradas. Existen múltiples propuestas que se enfocan en la paralelización de metaheurísticas (Cung et al., 2002; Alba, 2005). Actualmente, surge de manera natural incluir el paralelismo en la generación de un método de búsqueda.

La estrategia de paralelización puede realizarse de distintas maneras que pueden o no alterar la trayectoria de la búsqueda. En Crainic y Toulouse (2003), se describen tres formas de paralelismo muy utilizadas.

- i. Paralelizar instrucciones de un procedimiento incluido en una iteración del método. Es muy simple de aplicar, obtiene mejoras en el tiempo de cálculo,

aunque no modifica la solución final, puesto que la trayectoria de la búsqueda es la misma.

- ii. Paralelizar el dominio del problema. Implica dividir el espacio de búsqueda para que cada proceso busque en los sub-espacios definidos. Generalmente se implementa con una topología Maestro-Esclavo. Aquí el camino hacia la solución final no es el mismo que su par secuencial.
- iii. Definir varios procesos que se sincronizan y cooperan para buscar en un mismo espacio de búsqueda. El tiempo de procesamiento disminuye significativamente, aunque se incrementa la complejidad para controlar la sincronización y evitar la superposición de las trayectorias de búsqueda.

También suelen utilizarse alguna otra clase de estructuración para el procesamiento paralelo. Entre los esquemas clásicos para estructurar el conjunto de soluciones están el modelo distribuido o de grano grueso (Alba, 1999) y el modelo celular o de grano fino (Alba y Tomassini, 2002). En el primer caso las soluciones se dividen entre un conjunto de islas que ejecutan la metaheurística de manera secuencial. Las islas cooperan entre sí intercambiando información para diversificar la búsqueda. En el segundo caso, las metaheurísticas celulares se basan en el concepto de vecindario (Alba y Dorronsoro, 2008). Aquí, cada solución tiene a su alrededor un conjunto de soluciones vecinas donde se lleva a cabo la explotación.

El desempeño de los métodos que emplean estrategias paralelas suele evaluarse con métricas especiales como el Speedup y la Eficiencia. El Speedup es la razón entre el tiempo medio de ejecución de la versión secuencial y paralela del mismo algoritmo. La eficiencia computacional es el cociente entre el Speedup y la cantidad de procesadores utilizados. Un valor de eficiencia mayor a 1 significa que los elementos de procesamiento en conjunto logran mejores resultados que por separado.

# CAPÍTULO 4

## METAHEURÍSTICAS APLICADAS AL IPTP

---

### *Contenidos de este capítulo*

En este capítulo se plantea el problema denominado Intercity Public Transport Problem (IPTP) (Rodriguez et al., 2011), se establecen la representación de una solución al problema y la función que se utiliza para evaluar las potenciales soluciones. A los fines de valorar los algoritmos propuestos, se utilizó un caso de estudio real, la línea de transporte interurbana que une las ciudades de Bahía Blanca y Punta Alta, Argentina.

En primera instancia se compararon los resultados obtenidos al resolver el problema mediante GA y SA. Debido al mejor rendimiento de SA, se decidió perfeccionarlo. Se incorporó una memoria auxiliar para evitar realizar evaluaciones no necesarias y se aplicaron técnicas de programación paralela. Como resultado se logró mejorar la calidad de las soluciones obtenidas y disminuir el tiempo computacional.

## 4.1 INTRODUCCIÓN

La logística del transporte es un aspecto vital para el éxito de una región. Cada día miles de pasajeros viajan de una ciudad a otra por diferentes motivos, se proyecta que el número de personas que realizan estos viajes continúe aumentando. En muchos casos, los usuarios comparten el mismo recorrido por un período de tiempo significativo, por lo que la planificación del sistema de transporte cobra cada vez más importancia.

Una opción válida, ante el incremento de la demanda, es aumentar los recursos y reorganizar la asignación de flota y personal. Sin embargo, si no es posible realizar dicha inversión para introducir en la línea nuevas unidades y más choferes, entonces una alternativa razonable es optimizar la ruta y las frecuencias de los autobuses (Baaj y Mahmassani, 1991; Desaulniers y Hickman, 2003), con el fin de mejorar el sistema de transporte interurbano. Esta posibilidad generalmente está regulada por el estado y afecta en el proceso de asignación de recursos (Ceder y Wilson, 1986).

Nuestro trabajo presenta un estudio que permitirá a los empresarios tomar decisiones que proporcionen beneficios en los servicios de transporte. El procedimiento se centra en la optimización de las rutas, la ubicación de las paradas y el establecimiento de las frecuencias que deben seguir los autobuses interurbanos de transporte público. En Guihaire y Hao (2008) se abordan objetivos similares en materia de transporte urbano. Sin embargo, en nuestro modelo empleamos un software de simulación de tráfico para incluir las particularidades que diferencian el transporte interurbano del urbano. El transporte interurbano se identifica principalmente por las siguientes características:

- Por lo general, existen menos paradas y se encuentran más alejadas entre sí.
- Hay menos interacción entre los vehículos lo que provoca un menor número de detenciones no programadas.



- Hay segmentos de rutas con características especiales (por ejemplo, límites de velocidad más altos y mayor número de carriles).

La Planificación de operaciones de vehículos implica tomar un gran número de decisiones. Además, como el número de opciones posibles crece exponencialmente con el tamaño del problema, no es viable la resolución de problemas complejos con métodos exactos. En consecuencia, es necesario utilizar metaheurísticas, las cuales son procedimientos alternativos que no garantizan la solución óptima, pero proporcionan soluciones de calidad dentro de un tiempo de cálculo razonable (Colorni et al. 1996).

Para las pruebas del modelo, se han implementado algoritmos en la arquitectura MALLBA (Alba et al., 2007). MALLBA es una librería de software implementada en C++ que permite implementar fácilmente modelos de algoritmos metaheurísticos híbridos y paralelos.

Las simulaciones que realiza el método se llevaron a cabo mediante el software SUMO (Simulator of Urban MObility) (Behrisch et al., 2011). Este software nos permite realizar simulaciones que emulan la movilidad vehicular a nivel microscópico, es de código abierto, altamente portable y capaz de soportar mapas completos de ciudades reales, como los obtenidos a través de GoogleMaps o OpenStreetMap (ver Figura 4.1).



*Figura 4.1: Mapa en OpenStreetMap y el resultado obtenido en SUMO.*

En SUMO se pueden definir propiedades como la aceleración y desaceleración de vehículos, la capacidad del conductor, la velocidad máxima de los vehículos, las direcciones de las calles y los tiempos de espera. De esta manera, SUMO permite la simulación precisa del comportamiento del tráfico, proporcionando todos los datos necesarios para calcular la aptitud de cada solución (Behrisch et al., 2011).

## 4.2 ITP: PLANTEAMIENTO DEL PROBLEMA

La problemática descrita puede ser modelada como una variante del VRP a la que denominamos ITP (Toth y Vigo, 2002). El modelo generado puede entenderse como un módulo que pertenece a un sistema mucho más complejo de modelos de decisión. Consideramos que la resolución del ITP se centra en el diseño correcto de las rutas y una frecuencia adecuada de salidas para los autobuses que deben atravesar un conjunto de paradas establecidas previamente.

Se describe una línea interurbana como una secuencia de paradas, que deberán ser visitadas por los autobuses con una frecuencia determinada. El objetivo del modelo es minimizar el costo total de brindar un servicio que satisfaga las condiciones mencionadas. Así, para cada configuración se tendrá en cuenta el número total de kilómetros recorridos por cada uno de los autobuses, los tiempos demandados y la cantidad de detenciones no planificadas que ocurrieron, penalizando aquellos autobuses que no completaron el recorrido.

### 4.2.1 ITP: REPRESENTACION DE LA SOLUCIÓN

Una solución al ITP es una secuencia de paradas, la cual está representada por un vector de  $n$  elementos, donde  $n$  es el número de paradas que debe visitar el autobús y un número entero  $F$  que indica la frecuencia de salida (ver Figura 4.2). Además se conserva un vector

adicional de la misma dimensión para representar la ciudad a la que pertenece cada una de las paradas (Rodríguez et al., 2011).

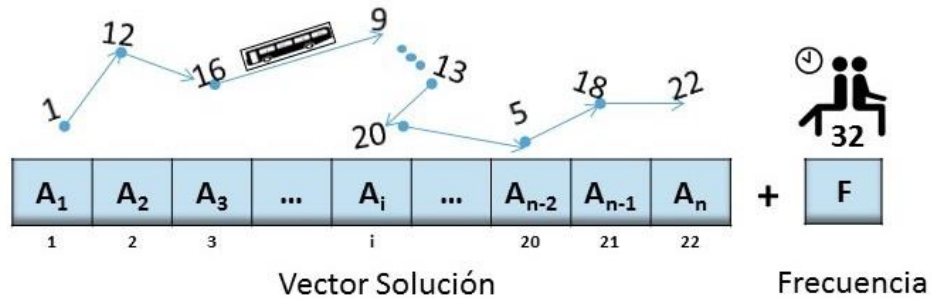


Figura 4.2: Representación de la solución para el ITP.

#### 4.2.2 ITP: FUNCIÓN DE APTITUD

En la función de aptitud o fitness se representan los intereses de los usuarios y los operadores. Es una medida de la calidad de la solución expresada mediante una estimación del costo total. Encontrar las rutas y frecuencias adecuadas para los autobuses se formula como un problema de optimización que consiste en minimizar el tiempo de viaje del autobús, el costo de la ruta para cada vehículo y el número de vehículos que cumple satisfactoriamente con las frecuencias impuestas (ver Ecuación 14).

$$\text{Min } Z = \frac{(T + R + ST)}{V} \quad (14)$$

##### Símbolos

$t_i$  El tiempo total del viaje para el autobús  $i$  en segundos

$T = \sum_{i=0}^{NV+V} t_i$

$v_i$  El autobús  $i$ .

$V$  El número de autobuses que llegan a destino.

$NV$  El número de autobuses que no llegan a destino.

$r_i$  El costo de la ruta para el autobús  $i$ .

$$R \quad \sum_{i=0}^{NV+V} r_i$$

$s_i$  El número de veces que el autobús  $i$  debe detenerse en un lugar no planificado.

$$ST \quad \sum_{i=0}^{NV+V} s_i$$

$S$  El tiempo total de simulación.

Para obtener los valores de aptitud asociados a las distintas soluciones se utilizó un software de simulación SUMO (Krajzewicz et al., 2002), el cual permite simular un escenario similar al real.

### 4.3 IPTP: LÍNEA INTERURBANA BAHÍA BLANCA – PUNTA ALTA

El caso de estudio analizado corresponde a una línea de pasajeros interurbanos que conecta las ciudades de Bahía Blanca y Punta Alta, ambas ciudades ubicadas en Argentina (ver <http://www.gpsurbana.com/>, elija la línea 319). El servicio actual tiene defectos importantes que inducen a las personas a vivir en la ciudad más cercana - por lo general en Bahía Blanca - para poder trabajar o estudiar. Nuestro enfoque ayuda a lograr un sistema de transporte flexible (animar a la gente a vivir en Punta Alta, dándoles la posibilidad de trabajar o estudiar en Bahía Blanca sin mayores problemas de transporte). Por lo tanto, este estudio es una contribución a la descentralización.

Las 22 paradas existentes están distribuidas en las ciudades de Bahía Blanca y Punta Alta. Se conoce el punto inicial y final del recorrido que tiene una longitud aproximada de 30 km. La frecuencia de partida de los autobuses, por disposiciones gubernamentales, debe estar entre los 20 y 50 minutos y el tiempo de simulación utilizado fue de 8 horas (una jornada completa).

Bahía Blanca es una ciudad argentina, cabecera del partido homónimo, con 450.000 habitantes, dos universidades nacionales, edificios gubernamentales, embajadas internacionales y una superficie poblada de 2.247 km<sup>2</sup>. Punta Alta es una ciudad con 60.000

habitantes, posee una de las bases navales más importantes del país llamada Puerto General Belgrano y una superficie poblada de 1.312 km<sup>2</sup>. Ambas están situadas al sur de la provincia de Buenos Aires y se comunican a través de una línea interurbana que tiene una demanda aproximada de dos mil pasajeros por día.

El IPTP es una variante del VRP y como este pertenece a la clase NP-Completo, sus variantes también lo hacen (Lenstra y Kan, 1981). Entonces, la solución de este problema por medio de un método de búsqueda exhaustiva requeriría un excesivo tiempo de procesamiento, además merece recordarse que su complejidad computacional aumenta exponencialmente al incrementarse el número de paradas. Por lo tanto, es deseable emplear metaheurísticas con el fin de obtener respuestas satisfactorias en un tiempo razonable.

En las restantes secciones del presente capítulo presentamos la evolución de nuestras investigaciones respecto al IPTP. Desarrollamos dos procedimientos basados en GA y SA. Luego presentamos una versión adaptada de la técnica SA que incorpora una memoria auxiliar y un nuevo método de selección de vecinos. Finalmente, adecuamos el algoritmo para que sea capaz de ejecutarse en múltiples procesadores. Los detalles sobre estos trabajos son presentados en los próximos tres apartados.

#### **4.4 EVALUACIÓN COMPARATIVA DE GA Y SA**

Procedimos a resolver el IPTP mediante algoritmos no deterministas, como lo son las metaheurísticas GA y SA. Sus principales beneficios se centran en su aptitud para escapar de óptimos locales, su capacidad de adaptación a los cambios y su independencia al problema.

Además del prestigio actual obtenido tal como se fundamentó en el capítulo 3, la elección de las dos metaheurísticas GA y SA está fuertemente ligada a sus marcadas diferencias. Estas técnicas difieren fundamentalmente en el número de soluciones que procesan al mismo tiempo y en la cantidad de memoria que utilizan. Por un lado, GA es un enfoque

basado en poblaciones mostrando la evolución del algoritmo en distintos puntos del espacio de búsqueda, utilizando información histórica de búsquedas anteriores. Por otro lado, el SA es un método basado en trayectorias, utiliza solamente una solución a la vez, marcando una trayectoria en el espacio de búsqueda; su acción se determina exclusivamente por el estado de la información actual.

Con vistas a una comparación algorítmica válida entre las dos técnicas, se empleó la misma representación para codificar todas las soluciones posibles y también la misma función de aptitud para evaluar la calidad de las soluciones.

#### 4.4.1 RESULTADOS Y CONCLUSIONES DE GA vs. SA

Previo al análisis de ambos algoritmos se realizaron una serie de pruebas empíricas con el fin de ajustar los parámetros de cada algoritmo. Finalmente, se llevaron a cabo 30 ejecuciones independientes con la configuración que se muestran en las Tablas 4.1 y 4.2. Las implementaciones se llevaron a cabo usando la librería MALLBA bajo el lenguaje C++ y se ejecutaron en una PC Intel Core 2 Duo, con un procesador de 2,53 GHz y 1 GB de RAM.

Generaciones	Individuos	Descendientes	Cruzamiento	Mutación
100	50	100	0.7	0.01

**Tabla 4.1:** Configuración de parámetros para el Algoritmo Genético.

Evaluaciones	Longitud Cadena de Markov	Temperatura inicial
300	10	0.99

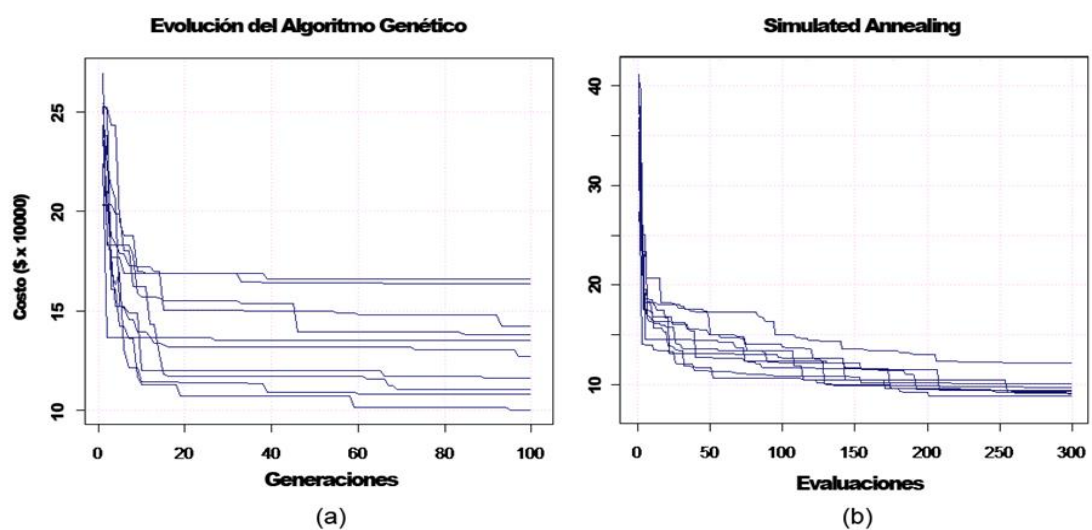
**Tabla 4.2:** Configuración de parámetros para Simulated Annealing.

Un primer análisis con respecto al uso del simulador en el cálculo de la función de aptitud para cada solución indica que el simulador consume el 95% del tiempo total de ejecución del algoritmo, el proceso de cargar el mapa de la región en estudio es la tarea más costosa

computacionalmente. Como esta actividad se lleva a cabo siempre que se necesite evaluar una solución, resulta natural establecer directrices para reducir el número de simulaciones en cada generación.

En este estudio computacional, el GA proporcionó resultados satisfactorios para la resolución del IPTP, incluso antes de llegar a las 100 generaciones (ver Figura 4.3a). Sin embargo, la situación detectada sobre el excesivo tiempo de cálculo del simulador, implica que al calcular la aptitud de la solución, el tiempo de cálculo del algoritmo se vea directamente afectado por la ejecución del simulador.

Debido a que GA es un enfoque basado en poblaciones, este cálculo se hace en muchas ocasiones, lo que implica un tiempo de procesamiento más alto. En cambio, SA un método basado en trayectorias, entonces necesita menos evaluaciones por iteración, reduciendo así el tiempo de cálculo. Además, nos permite ampliar el número de iteraciones a 300, obteniendo mejores resultados a los obtenidos por el GA (ver Figura 4.3b). En definitiva, 300 evaluaciones realizadas con SA son suficientes para obtener mejores soluciones de las que puede encontrar GA en 100 generaciones.



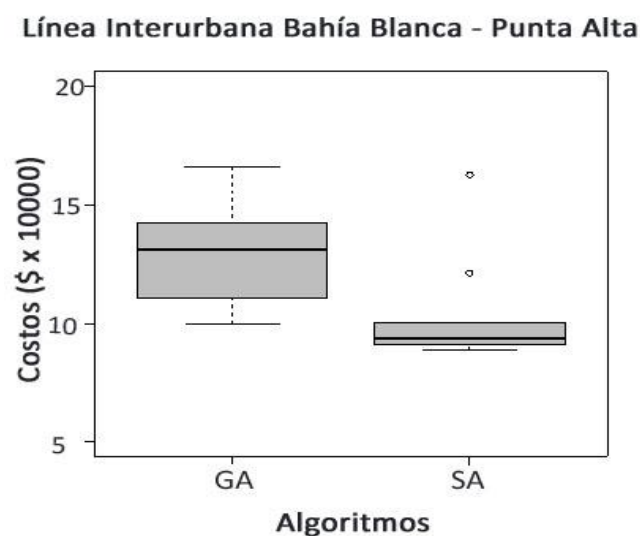
**Figura 4.3:** Evolución del Algoritmo Genético (a) y Simulated Annealing (b).

Con el fin de proporcionar comparaciones estadísticamente significativas, hemos aplicado el test de rangos signados de Wilcoxon (Wilcox, 1987) a las distribuciones numéricas de los resultados, en lugar de concentrarse en otras medidas populares, como la media y la desviación estándar (Sheskin, 2003). La Tabla 4.3 muestra los valores correspondientes. El nivel de confianza establecido en 95%, permite asegurar que todos estos resultados son estadísticamente diferentes si se traducen en un valor de  $p < 0,05$ .

	Media	Mediana	Desviación estándar
GA	1.35E+05	1.32E+05	2.82E+04
SA	1.14E+05	1.06E+05	3.93E+04

**Tabla 4.3:** Media, mediana y desviación estándar para los valores de aptitud obtenidos.

Para la línea interurbana Bahía Blanca-Punta Alta, las diferencias entre las distribuciones del GA y el SA en las 30 ejecuciones independientes mostraron valores de  $p$  muy inferiores a 0,05. Por lo tanto, se puede inferir que el SA obtuvo estadísticamente mejores resultados que el GA. Un resumen de estos resultados se puede ver en el diagrama de caja que se muestra en la Figura 4.4. El mismo confirma que SA muestra mejores cuartiles superiores e inferiores que GA.



**Figura 4.4:** Diagrama de caja con los resultados obtenidos por GA y SA.



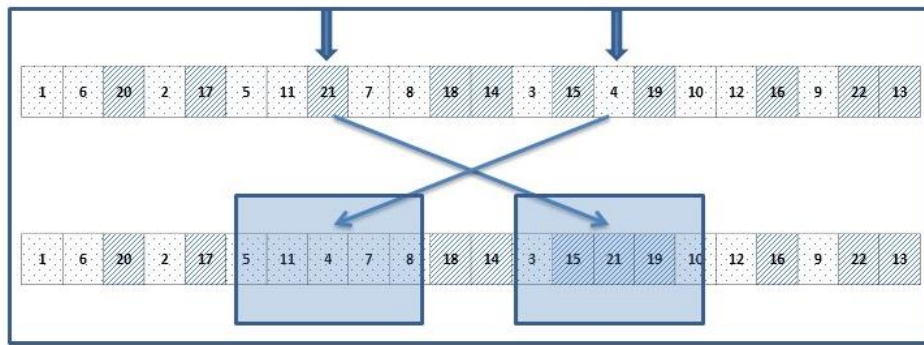
En definitiva, para el problema del transporte público interurbano tanto GA como SA se comportan satisfactoriamente. Sobre la base del análisis anterior, se puede recomendar el uso de un procedimiento SA para resolver versiones simples de un IPTP. El SA presenta mejores valores en la media y la mediana respecto al GA. Sin embargo, la aplicación de GA no debe descartarse, especialmente en los casos más complejos del IPTP.

#### **4.5 ADAPTACIÓN DE UN MODELO SA**

La evolución de las primeras investigaciones la presentamos como un algoritmo Simulated Annealing Adaptado (ASA, por sus siglas en inglés). Continuamos con el estudio que resuelve la instancia del problema de transporte público interurbano (Rodríguez et al., 2014a). Este problema de optimización combinatoria se resuelve con eficacia por medio de un método robusto. Su buen desempeño se logra mediante la incorporación de una memoria auxiliar y un novedoso método de selección de vecinos a la metaheurística SA. A continuación, se describen los detalles de la implementación.

La técnica SA intenta evitar ser atrapada en un óptimo local aceptando soluciones de menor calidad con cierta frecuencia, aumentando el valor de la función objetivo. En general, el enfoque más común es considerar que el vecino de una solución difiere en sólo un intercambio de elementos, la mayoría de las veces los dos elementos son elegidos al azar.

Por el contrario, en nuestro trabajo hemos diseñado un método de selección de vecino ad-hoc, donde se guía el intercambio. Hemos dado una mayor probabilidad a un intercambio entre dos paradas que pertenecen a ciudades distintas (ver Figura 4.5).



**Figura 4.5:** Un ejemplo del método de selección de vecinos.

Además, detectamos el beneficio de hacer que esta probabilidad aumente a medida que las iteraciones avanzan. De esta manera, al final del proceso, el algoritmo tiende a terminar con una solución representada por un vector cuyos elementos correspondientes a una misma ciudad, se encuentran contiguos.

La Figura 4.6, es una representación de la estructura básica de nuestro algoritmo ASA (Adapted Simulated Annealing). En primer lugar, una solución potencial (PS: Potential Solution) se evalúa y, seguidamente, se selecciona una solución vecina (NS: Neighbor Solution). Entonces, se procede a evaluar NS. Si NS es mejor que PS, entonces NS reemplaza PS, de lo contrario la peor solución igualmente se acepta con una probabilidad controlada por la Ecuación 15.

$$U(0,1) < e^{-\delta T} \tag{15}$$

A pesar de haber hecho una buena elección de los parámetros, los tiempos de cálculo no fueron tan satisfactorios como lo esperado. El motivo es el mismo al del trabajo inicial, el mayor tiempo de cálculo, aproximadamente el 90 % del tiempo global en todos los casos, se ocupa durante la simulación. Entonces, el tiempo de ejecución crece significativamente siempre que el algoritmo tiene que llamar al simulador.

Además, se detecta que durante las últimas iteraciones del algoritmo, las soluciones que ya habían sido evaluadas son enviadas al simulador nuevamente, lo que implica un consumo de recursos innecesarios. Para superar este inconveniente, el ASA incorpora una memoria auxiliar. De este modo, las soluciones con mayor probabilidad de ser re-evaluadas se mantienen en memoria. Entonces, antes de llamar al simulador, el algoritmo comprueba si la solución está incluida en la memoria auxiliar. Si es así, el valor de aptitud se recupera rápidamente. De esta manera, se logra un comportamiento satisfactorio en el rendimiento del algoritmo.

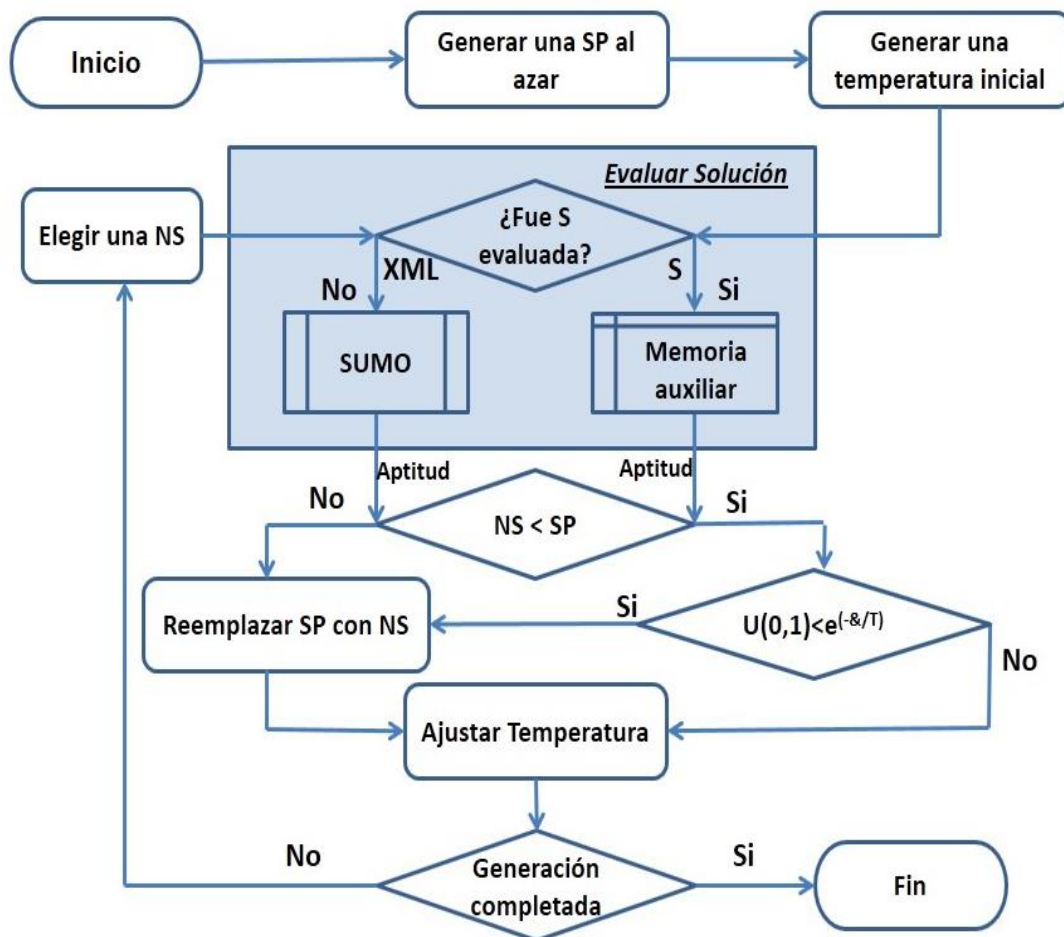


Figura 4.6: Diagrama de flujo de nuestro proceso ASA.

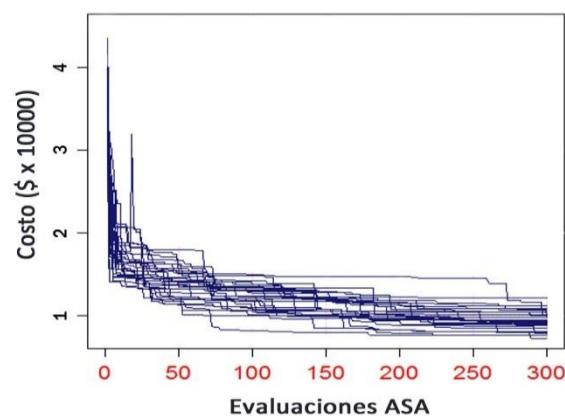
El proceso de selección de soluciones a ser mantenidas en la memoria se centra en aquellas soluciones con mayor probabilidad de ser elegidas como solución vecina, teniendo en cuenta su valor de aptitud como un atributo secundario. Para esto, es necesario mantener un

valor igual al número de coincidencias entre PS y NS, valor que hemos denominado Vecino Probable (LN: Likely Neighbour). Este índice muestra la cantidad de elementos coincidentes entre un vector NS en comparación con el vector PS. La memoria se ordena en función de LN, se actualiza en cada generación de vecino, y se ordena solo cuando aparece una mejor solución.

El tamaño de la memoria auxiliar se debe establecer de acuerdo a las dimensiones del problema. Después de varias pruebas (Rodríguez et al., 2011), una alternativa válida consiste en almacenar en la memoria el doble de la cantidad de paradas ( $\text{DimM} = 2p$ ). En este trabajo, se mantienen 44 soluciones vecinas, cada parada es representada por un entero, al igual que el índice LN. Como la aplicación hace uso de 4 bytes para almacenar un número entero. Entonces, el consumo de memoria ( $\text{ConM} = 4(p+1)(2p)$ ) no resulta significativo.

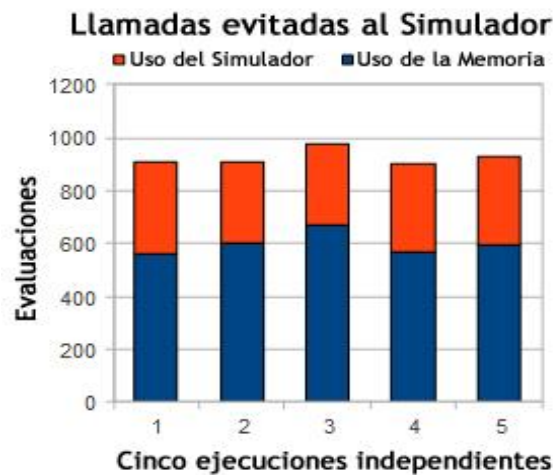
#### 4.5.1 RESULTADOS Y CONCLUSIONES DE ASA

Las ejecuciones se llevaron a cabo en una PC estándar: Intel Core 2 Duo de 2.53 GHz y 1 GB de RAM. Presentamos 30 ejecuciones independientes donde se obtuvieron los resultados mostrados en la Figura 4.7. Se observa que los valores de aptitud disminuyen a medida que progresan las evaluaciones, exhibiendo una robustez satisfactoria de ASA en la resolución del IPTP.



*Figura 4.7: Evolución del algoritmo ASA.*

La ventaja del método ASA al trabajar con una memoria auxiliar se refleja en una reducción media del 20 % en el tiempo de ejecución. La implementación que prioriza la calidad del vecino para mantener una solución en la memoria induce una mejora adicional del 12 % en el desempeño computacional. La Figura 4.8 refleja las llamadas evitadas al simulador.



*Figura 4.8: Llamadas evitadas al simulador.*

Esta propuesta se ha traducido en una opción efectiva y eficiente. La puesta en práctica del algoritmo ASA demostró ser capaz de producir soluciones de alta calidad en los tiempos apropiados.

Hemos construido un modelo eficiente que intenta minimizar el número de llamadas al simulador gracias a la adición de una memoria auxiliar y un novedoso método de selección de vecinos que aprovecha la información disponible del problema para acelerar la búsqueda. También hemos demostrado que nuestra técnica se puede aplicar a un escenario real.

Los resultados computacionales indican claramente que el enfoque propuesto constituye una mejora en la capacidad de búsqueda de soluciones de alta calidad y que facilita la convergencia.

Hasta aquí trabajamos sobre una programación secuencial, a continuación efectuamos una implementación de una estrategia paralela con el objeto de obtener mejores soluciones, reducir los tiempos de cómputos y aprovechar los recursos disponibles.

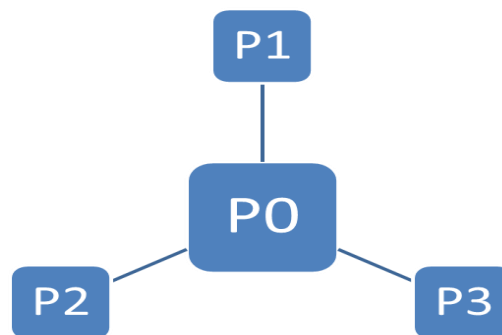
#### **4.6 ESTRATEGIA PARALELA DE SA**

Los algoritmos secuenciales suelen tornarse ineficientes al abordar problemas complejos, sobre todo cuando la evaluación de la función de aptitud es muy costosa en cuanto a tiempo de cómputo. Además, la plataforma de desarrollo para un algoritmo secuencial no favorece la búsqueda y por lo general ocurre que convergen prematuramente hacia valores sub-óptimos. Entonces, para contrarrestar estos problemas surge la necesidad de usar técnicas que paralelicen la ejecución de los algoritmos. En este caso, se optó por desarrollar un algoritmo que utiliza múltiples procesadores. Esta característica permite ampliar la cantidad de sectores de búsqueda dentro del espacio de soluciones aumentando la diversidad y por ende minimizando la probabilidad de converger prematuramente.

La arquitectura de computación paralela multiprocesador que empleamos sigue el diseño basado en esqueletos de la biblioteca MALLBA. Esta biblioteca soporta la ejecución de algoritmos concurrentes en procesadores multi-núcleo, permitiendo la ejecución de varias tareas en forma simultánea. De esta manera, disponemos de una mayor ventaja, el hecho de que los núcleos estén integrados a un mismo circuito, hace que se obtenga una reducción de los tiempos de comunicación entre los procesos.

Utilizamos una topología que distingue dos tipos de procesos, un proceso maestro ( $P_0$ ) que mantiene un estado global del problema y que es el encargado de recolectar los resultados obtenidos en cada generación por cada uno de los restantes procesos ( $P_i$ ). En este trabajo se operaron tres de estos procesos esclavos, cada uno de ellos emplea un procesador desarrollando la búsqueda en direcciones distintas (ver Figura 4.9).

El algoritmo sigue un modelo síncrono, lo que significa que el proceso maestro espera a que todos los procesos finalicen su tarea antes de iniciar la siguiente generación (Rodríguez et al., 2012). La comunicación entre los procesos se realiza mediante el software Netstream (Alba, 2001), que nos ofrece una interfaz orientada a objetos sobre la biblioteca de pasos de mensajes MPI (Gropp et. al, 1996).



**Figura 4.9:** Topología Maestro-Esclavo

#### 4.6.1 RESULTADOS Y CONCLUSIONES DE SA PARALELO

La ejecución de nuestro algoritmo SAP (Simulated Annealing Paralelo) se llevó a cabo utilizando parámetros con los mismos valores que para el SAS (Simulated Annealing Secuencial) como se presenta en la Tabla 4.4. El equipo de cómputo utilizado fue una computadora IBM con procesador AMD Phenom QUAD CORE SERIE 9000 y 8GB de memoria RAM.

Evaluaciones	Longitud de la cadena de Markov	Temperatura Inicial
300	10	0.99

**Tabla 4.4:** Valores de los parámetros del algoritmo.

Los resultados del algoritmo paralelo propuesto y su par secuencial se resumen y se comparan en la Tabla 4.5, luego de 30 ejecuciones independientes. Para cada algoritmo se presentan los promedios y mejores valores de la función objetivo, el número de iteración en la

cual se obtuvo ese mejor valor, el tiempo promedio de ejecución y las medidas para evaluar el rendimiento como lo son el Speedup y la Eficiencia.

Algoritmo	# Procesa_ dores	Aptitud Promedio	Mejor aptitud	Iteración	Tiempo	Speedup	Eficiencia
SAS	1	92468.2	86770.7	289	12.27E+5	1.28	32%
SAP	4	91741.9	88842.6	263	95.53E+4		

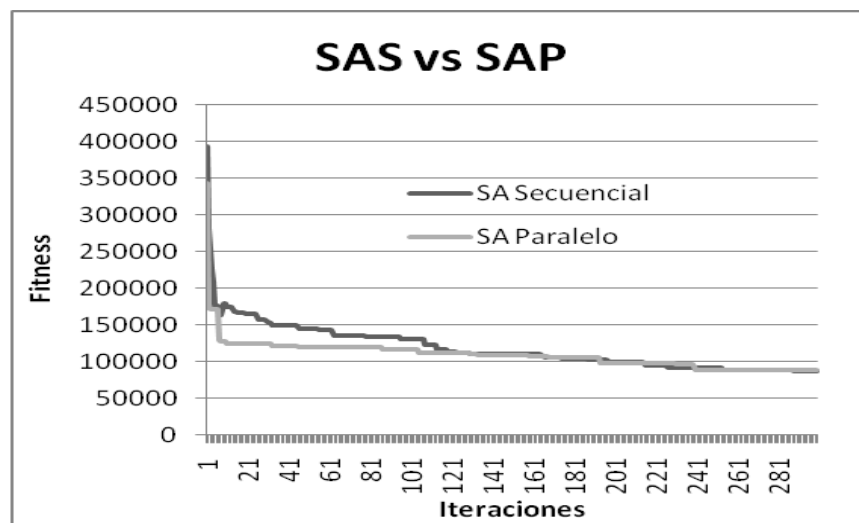
**Tabla 4.5:** Comparación del algoritmo secuencial con el paralelo.

Recordemos que el Speedup es el factor de mejora del rendimiento por haber procesado en paralelo y la eficiencia es una comparación entre el grado de Speedup y el mejor valor obtenido.

De los resultados obtenidos podemos afirmar que no existen diferencias significativas entre los valores promedios de aptitud. Sin embargo, al analizar los tiempos de cálculo, podemos observar una marcada diferencia en favor del algoritmo paralelo. La utilización de cuatro procesadores otorga una ganancia en la eficiencia del procesamiento cercano al 32%. Con respecto al Speedup, este escenario muestra un comportamiento sub-lineal. A pesar de que, naturalmente, la mejora no es proporcional al número de procesadores, este comportamiento lo consideramos satisfactorio y prometedor.

También podemos observar en la Figura 4.10 que el modelo paralelo encuentra mejores valores en las primeras iteraciones y no necesita tantas iteraciones para encontrar una buena solución final. Motivo por el cual permite concluir que la ganancia de tiempo puede ser aún mayor comparado con el algoritmo secuencial.





**Figura 4.10:** Evolución del algoritmo Paralelo y Secuencial.

Mediante la paralelización del algoritmo SA, se consigue reducir el tiempo de ejecución sin afectar la calidad de las soluciones obtenidas, esto dicho en comparación con nuestro antecesor: el algoritmo SAS.

La estructura paralela que sigue un modelo maestro-esclavo también contribuye a la diversidad para cada proceso en ejecución. Debemos recalcar que en el problema del transporte interurbano se trabaja con instancias más grandes a la presentada en este trabajo, lo que implica un elevado grado de procesamiento y una necesidad de paralelizar cuando es necesario reducir los tiempos de cómputos.



# CAPÍTULO 5

## METAHEURÍSTICAS PARA PROBLEMAS DE LOCALIZACIÓN Y RUTEO

---

### *Contenidos de este capítulo*

Presentamos el problema de localización y ruteo. Establecemos un modelo para la representación de los problemas estudiados.

Se presenta su aplicación en dos casos de estudios diferentes:

LRP aplicado al transporte público: El objetivo es optimizar el recorrido de los autobuses y la localización de las paradas a utilizar, para dar cobertura a una determinada región, minimizando los costos por la prestación del servicio.

LRP aplicado a la distribución de fluidos energéticos: El objetivo es establecer una red de cañerías para distribuir el producto desde su lugar de origen hacia plantas de procesamiento. Se contempla la posibilidad de instalar localizaciones intermedias que permitan almacenar temporalmente el producto para su posterior traslado.

## 5.1 INTRODUCCIÓN

Los problemas de localización también tienen numerosas aplicaciones reales (Campbell, 1994). Se caracterizan por la necesidad de instalar centros de servicios que faciliten el transporte de un determinado bien o producto. La incorporación de esta problemática al problema de ruteo descrito anteriormente da origen a lo que denominamos LRP.

El LRP es un problema que se encuadra en un área de investigación relativamente nueva con distintas aplicaciones y variados enfoques (Chandra Mohan y Baskaran, 2012). El estudio de Nagy y Salhi (2006) describe técnicas de solución y aplicaciones para este tipo de problema. Está demostrado que este es un problema NP-Completo (Wolsey, 2000), ya que abarca dos problemas NP- Completos: la ubicación de las instalaciones y la asignación de rutas para los vehículos (Lenstra y Kan, 1981). En este caso, la dificultad de resolver el problema crece exponencialmente con respecto al tamaño del problema, siendo recomendable la aplicación de metaheurísticas sobre todo en problemas de grandes dimensiones (Sörensen, 2013).

En la mayoría de los trabajos publicados sobre LRP se resuelven problemas asociados al transporte de un objeto: desde un depósito hacia un conjunto de clientes (Belenguer et. al, 2011). Muy pocos autores han estudiado este problema relacionándolo al transporte de personas. Una gran parte de ellos tratan el caso del transporte privado, mayormente del transporte escolar (Park y Kim, 2010). Nuestro objetivo es modelar aspectos del sistema de transporte público de pasajeros para una determinada región (Rodriguez et al., 2013b). Berger et al. (2007) plantea un modelo que integra tanto la localización de los depósitos como el ruteo de vehículos de manera simultánea.

A continuación, se presentan dos casos de estudio asociados al LRP. Una instancia asociada al transporte público y otra asociada a una red de cañerías, las cuales fueron resuelta mediante técnicas basadas en las metaheurísticas ACO y SA respectivamente. Los algoritmo

fueron implementado en el lenguaje Java y los experimentos computacionales fueron realizados en una PC Intel Core 2 duo, con un procesador de 2.53 Ghz. y 1 GB de RAM.

## 5.2 CASO DE ESTUDIO: TRANSPORTE PÚBLICO

En esta sección continuamos en la misma línea de trabajo, abordar un problema que contribuye a la construcción de un sistema eficiente en el traslado de las personas que utilizan el servicio del transporte público ofrecido por una empresa, que cuenta con una capacidad limitada de vehículos y que pretende maximizar sus beneficios por los servicios prestados. La principal diferencia con la problemática presentada en el capítulo anterior está dada en que en el IPTP las paradas se encuentran en lugares ya establecidos, en cambio ahora se debe establecer el lugar donde se instalaran las paradas de los autobuses.

El estudio implica decisiones importantes en dos aspectos fundamentales: la localización de las paradas y las rutas de los colectivos. Es importante remarcar que la primera es una resolución estratégica que se mantiene por un determinado periodo, en cambio las rutas establecidas suelen modificarse con mayor asiduidad. Esta particularidad permite la posibilidad de tratar el problema en dos fases, primero establecer las mejores localizaciones y luego diseñar los recorridos.

Las características de los pasajeros, las paradas y los vehículos dan lugar a diferentes variantes del problema. En nuestro trabajo, asumimos que la región en donde se aplica el modelo está dividida en zonas que agrupan a personas que comparten alguna característica relacionada a los viajes, fundamentalmente aquellas que pertenecen a un mismo vecindario. Esta zonificación es concebida a corto plazo y puede ser revisada para horizontes más alejados. Se considera la ubicación de las potenciales paradas en múltiples lugares de la región, en especial, en aquellos sitios en donde existen centros de atracción de viajes tales como escuelas, hospitales, centros comerciales, etc. Cada pasajero es asociado a una potencial parada; sin

embargo, si esta no es seleccionada, el pasajero será asignado a la parada más cercana de la misma región, satisfaciéndose siempre la demanda de los usuarios. Los autobuses utilizados son considerados con una capacidad homogénea determinada. Esto implica que si durante el recorrido de una ruta esta capacidad se ve sobrepasada, resulta necesario el empleo de otra unidad.

Hemos planteado un caso de estudio práctico con el objeto de comprender el modelo descrito y poder evaluar el comportamiento del algoritmo que proponemos. En este caso se considera una región asociada a la ciudad de Bahía Blanca, Argentina. El requerimiento empresarial consiste en determinar las paradas a utilizar, la asignación de los pasajeros a estas paradas y establecer las rutas de los autobuses para satisfacer la demanda de los pasajeros minimizando el costo global y optimizando el servicio. El costo global incluye los costos fijos de apertura de las paradas, los costos fijos asociados a los autobuses utilizados y los costos por recorrer los trayectos de las rutas establecidas. Los colectivos deberán atravesar la ciudad, partiendo y finalizando sus recorridos en la estación central, cuya ubicación ya está establecida.

La versión simplificada del caso de estudio consiste en una región dividida en 3 zonas en las que se distribuyen 8 paradas, además de la estación central identificada como parada 0. Cada una de las paradas tiene asociada una demanda determinada que deberá ser satisfecha utilizando colectivos de igual capacidad. El algoritmo determinará las paradas y los recorridos para satisfacer la demanda con mínimo costo.



Figura 5.1: Representación de una instancia de este modelo.

La instancia descrita está representada en la Figura 5.1. Los datos de entrada al modelo son:

- a) La cantidad de potenciales paradas. ( $k=8$ ).
- b) Un costo de instalación para cada parada.  $H \in \mathbb{R}^{1 \times 8}$ ; por ejemplo,  $h_2=32$ .
- c) Una matriz de costo de transporte para un autobús al trasladarse de una parada a otra. ( $C \in \mathbb{R}^{8 \times 8}$ ; por ejemplo,  $c_{23}=45$ ).
- d) Una matriz de costo por el traslado de un pasajero desde su parada (no utilizada), hacia una parada cercana que se utiliza en su misma zona. ( $M \in \mathbb{R}^{8 \times 8}$  por ejemplo,  $m_{45}=16$ ).

- e) Un vector de demandas para cada una de las paradas.  $D \in \mathbb{R}^{1 \times 8}$ ; por ejemplo,  $d_3=11$ ).
- f) Una matriz binaria de asignación de paradas a zonas.  $(A \in \mathbb{R}^{3 \times 8}$ ; por ejemplo,  $a_{27}=1$ ).
- g) La capacidad de los colectivos. ( $Q=32$ ).

Entre paréntesis arriba se muestran algunos valores asociados a este caso de estudio.

### 5.2.1 IMPLEMENTACIÓN

Nuestra propuesta consiste en evaluar paradas y recorridos a través de un modelo lineal de programación entera diseñado para minimizar los costos totales por la prestación del servicio. El modelo corresponde a un problema de localización y ruteo con restricciones de capacidad en los vehículos. Podemos representarlo como un problema de grafos de la siguiente manera:

Sea  $G = (V, A)$  un grafo completo no dirigido, en el cual  $V = \{1, \dots, n\}$  es el conjunto de vértices y  $A$  es el conjunto de arcos. El conjunto de vértices corresponde a las paradas potenciales. Un valor no negativo  $c_{ij}$  es asociado con cada arco  $(i,j)$  perteneciente a  $A$ , representa el costo de realizar el recorrido desde la parada  $i$  hasta la parada  $j$ . Está disponible una flota homogénea  $(F)$  de autobuses  $v$ , cada uno con capacidad  $Q$ . Existe un costo de instalación  $h_k$  para cada una de las paradas, y se contempla la aplicación de un costo  $m_{ij}$  siempre que por la no utilización de la parada  $i$ , el pasajero deba trasladarse a una parada cercana  $j$ .

Así, el modelo puede formularse como:

$$\text{Min}Z = \sum_{i \in V} \sum_{j \in V} \sum_{v \in F} c_{ij} x_{ijv} + \sum_k h_k y_k + \sum_{i \in V} \sum_{j \in V} \sum_{k \in V} m_{ij} x_{ijk} \quad (16)$$



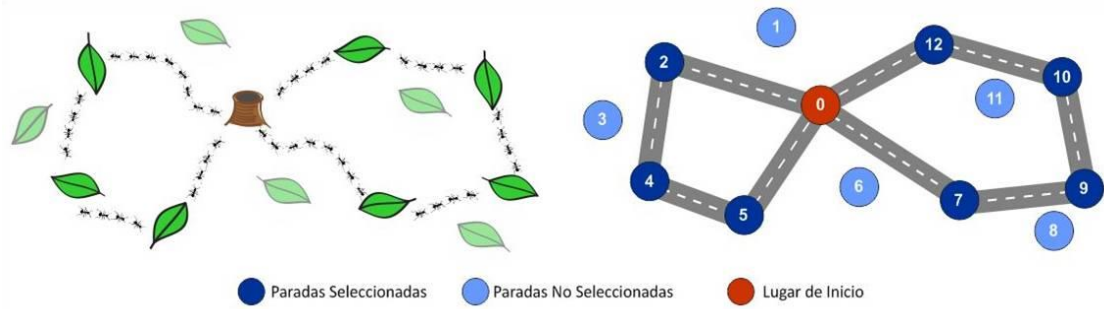
Sujeto a las restricciones indicadas en la Tabla 5.1.

Restricción	Propósito
$W_{0v} = 1$	Todos los vehículos utilizan la estación central (parada 0).
$U_{0v} = 1$	Todos los vehículos parten desde la estación central.
$\sum_1^n a_{pk} y_k \geq 1$	Cada pasajero dispondrá de al menos una parada en su región.
$\sum_1^n d_k w_{kv} \leq Q$	No se debe exceder la capacidad de los colectivos.
$2 \leq u_{kv} \leq \sum_1^n w_{k'v}$	Las paradas se ordenan entre dos y el número de paradas utilizadas.
$\sum_{i=0}^n x_{ijk} = \sum_{j=1}^{n+1} x_{jik}$	Si un autobús llega a una parada también sale de ella, salvo la estación central.
$\sum_{i \notin S} \sum_{j \in S} x_{jik} \geq r(S), \forall S \subset V$	Impide la existencia de sub-circuitos (sub-tours).

**Tabla 5.1:** Restricciones del modelo.

La formulación empleada permite resolver el problema de manera exacta en instancias no demasiado grandes. Las metaheurísticas han demostrado ser una alternativa interesante para resolver problemas de grandes dimensiones, determinando buenas soluciones con razonables tiempos de cómputo (Gendreau y Potvin, 2005). Basados en el excelente desempeño de la metaheurística ACO en problemas similares, elegimos utilizarla como base para solucionar el problema planteado anteriormente.

La aplicación de ACO se lleva a cabo representando la solución como un vector de enteros, donde cada elemento representa una de las potenciales localizaciones. Cada una de las hormigas construirá una solución completa, en cada iteración se evalúa la calidad las soluciones encontradas, y se actualiza la cantidad de feromona en función de las paradas utilizadas y los resultados obtenidos. Al finalizar cada iteración se procede con la actividad de evaporación de feromona. El resultado del algoritmo es un conjunto de paradas a utilizar y las rutas generadas (Figura 5.2).



**Figura 5.2:** Inspiración y comportamiento de la metaheurística ACO.

Los parámetros empleados (Tabla 5.2) fueron elegidos a través de múltiples simulaciones.

<i>CantHormigas</i>	<i>Iteraciones</i>	<i>a</i>	<i>b</i>	<i>e</i>
100	30	2	1	0.8

**Tabla 5.2:** Parámetros utilizados durante las simulaciones.

El termino PróximoDestino (línea 5, Algoritmo 3.3), es una función probabilística utilizada para influenciar la elección de la próxima parada. La misma tiene la expresión presentada en la Ecuación 17.

$$P_{ij}^k = \frac{(\tau_{ij}^a)(\eta_{ij}^b)}{\sum(\tau_{ij}^a)(\eta_{ij}^b)} \tag{17}$$

Donde  $P_{ij}^k$  es la probabilidad de que la hormiga  $k$  ubicada en la parada  $i$  elija la parada  $j$ . El símbolo  $\tau$  corresponde al componente asociado a la información determinística que se dispone, el símbolo  $\eta$  corresponde a la feromona depositada. Con los parámetros  $a$  y  $b$  se establece el nivel de importancia para cada uno de estos componentes. En este trabajo establecimos que los aspectos determinísticos tienen el doble de importancia que los aspectos aleatorios. Por último, el parámetro  $e$  (Tabla 5.2), indica el porcentaje de información que se eliminará al momento de realizar el proceso de evaporación de la feromona.

ACO necesita de un procedimiento para evaluar cada una de las soluciones propuestas. Para ello, introducimos dentro del procedimiento ACO, el paquete GAMS con el resolvidor CPLEX 11.2, de esta manera individualizamos las paradas convenientes de usar y el recorrido asociado.

### 5.2.2 RESULTADOS Y CONCLUSIONES

El caso de estudio planteado en la sección anterior puede derivar en distintas instancias más complejas y cercanas a la realidad. Trabajando en la misma región, hemos considerado casos de 25 y 50 paradas, dividiendo la región en 3, 6 y 12 zonas. Esto dio lugar a las 6 instancias presentadas en la Tabla 5.3.

Instancia	Paradas	Zonas	Mejor Solución (rutas)	Costo (\$)	Promedio	Tiempo CPU (s)
1	8	3	0 2 6 4 0	479	479	7
2	25	3	0 20 4 9 0	8864	8926	23
3	25	6	0 20 15 0 0 22 5 0 0 11 2 0	9684	10272	32
4	25	12	0 11 15 0 0 2 20 0 0 22 5 0	11523	12358	68
5	50	6	0 6 29 0 0 33 27 0 0 25 37 0	45552	46752	89
6	50	12	0 9 27 0 0 29 2 20 15 0 0 33 35 19 0 0 21 22 46 0	89936	96.890	116

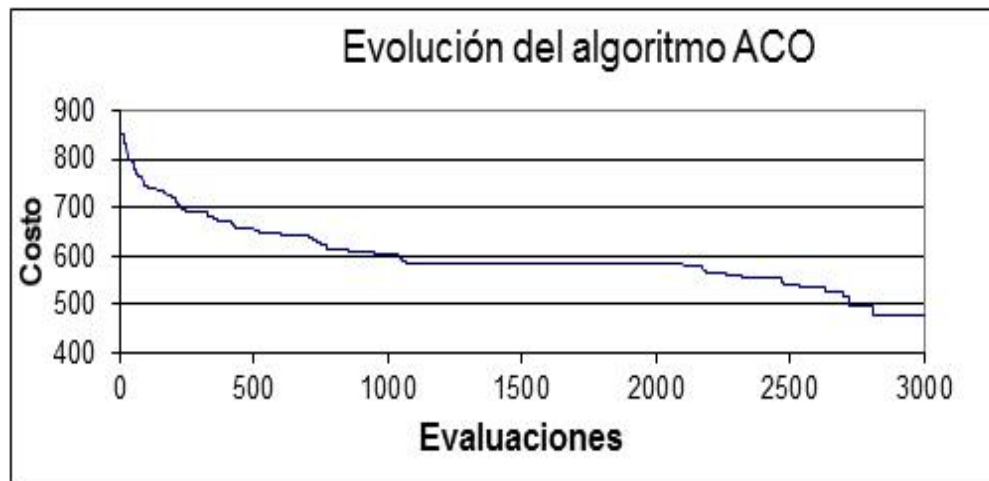
**Tabla 5.3:** Resultados computacionales para las instancias consideradas.

La cuarta columna de la Tabla 5.3 reporta las rutas que se necesitan generar para satisfacer la demanda. Por ejemplo, para la instancia 3, se necesitan 3 rutas que visitan 2 paradas cada una, partiendo y finalizando en la estación central 0. La quinta columna indica el costo asociado a la mejor solución obtenida, mientras la sexta muestra el promedio de los costos de las mejores soluciones logradas en todas las ejecuciones. Comparando estas dos columnas se desprende que en una instancia pequeña (con 8 paradas y 3 zonas) el algoritmo encuentra la

mejor solución en todas las ejecuciones. Al incrementar las paradas a 25, el promedio de las soluciones continúa siendo cercano a la mejor solución encontrada, aunque el tiempo de cómputo del algoritmo crece considerablemente (de 7 seg. a 23 seg.).

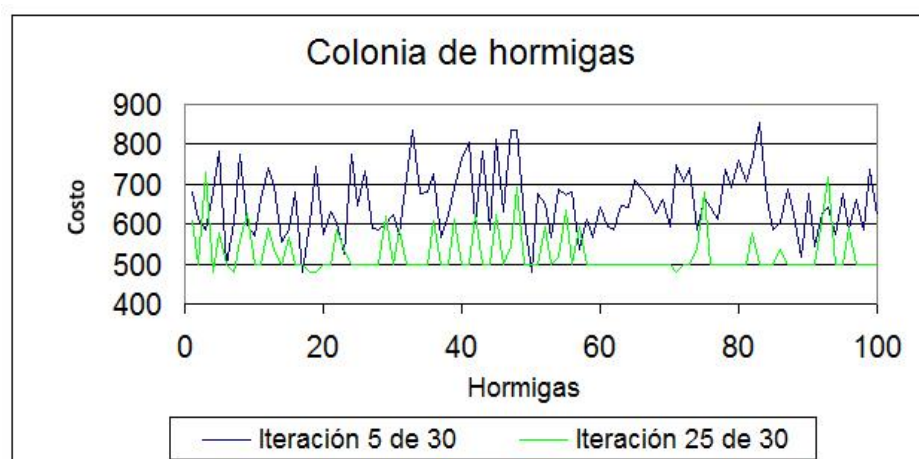
Esta tendencia se acentúa cuando se incrementa el tamaño del problema; sin embargo, considerando que la cantidad de combinaciones posibles aumenta de manera exponencial, el tiempo de cómputo mantiene un aumento paulatino que satisface nuestras pretensiones, porque nunca supera los 2 minutos. También es destacable la robustez del algoritmo en todas las instancias. Este rendimiento puede observarse mediante la cercanía entre el mejor resultado encontrado y el promedio de todas las ejecuciones, en el peor de los casos para las instancias más grandes la diferencia no superó el 8%.

La capacidad algorítmica de mejorar las soluciones obtenidas a medida que avanzan las iteraciones puede verse reflejada en la Figura 5.3, obtenida en la ejecución de la instancia 1, utilizando 100 hormigas y 30 iteraciones. Se muestran todos los costos (aptitud) obtenidos en una ejecución completa. El total de 3000 evaluaciones realizadas demuestra que el algoritmo recorre eficientemente el espacio de búsqueda. Un gran porcentaje de veces (aproximadamente el 40% de las veces) se obtiene una misma solución con aptitud = 583; sin embargo, continuando con el proceso de búsqueda se logra obtener mejores resultados hasta encontrar el menor costo (479).



**Figura 5.3:** Evolución de ACO en una ejecución completa.

La Figura 5.4 se refiere a la misma instancia y también refleja el buen comportamiento del algoritmo ACO. Muestra las diferencias entre una iteración al inicio de la ejecución (#5) y una iteración al final de la misma (#25), para las 100 hormigas artificiales utilizadas. Se puede observar que, a medida que avanzan las iteraciones, aumenta la cantidad de veces que se obtienen las mejores soluciones. La existencia de algunos picos está asociada a soluciones no satisfactorias. Esto se debe al componente estocástico del algoritmo durante la acción de evaporación de feromona con el fin de evitar óptimos locales.



**Figura 5.4:** Progreso de las iteraciones para la instancia 1.

De las pruebas realizadas en todas las instancias, observamos la particular influencia de la demanda de pasajeros y de la capacidad del colectivo. Los resultados anteriores se corresponden con una capacidad para 32 pasajeros, y se observa que resultaron rutas con escasas paradas. Como consecuencia, se decidió probar la instancia 3 con colectivos para 38 y 45 pasajeros. La Tabla 5.4 muestra que aumentando la capacidad del autobús es posible lograr el objetivo con 2 recorridos en lugar de 3 (ver Tabla 5.4). Esto implica la utilización de menos unidades, derivando en la reducción de un porcentaje aproximado del 14% y 17% de los costos totales para las instancias 3a y 3b, respectivamente.

<b>Instancia 3</b>	<b>Capacidad</b>	<b>Mejor Solución (rutas)</b>	<b>Costo (\$)</b>	<b>Promedio (\$)</b>	<b>Tiempo CPU (s)</b>
A	38	0 2 20 15 0 0 9 5 22 0	8328	8962	36
B	45	0 2 20 15 0 0 11 22 5 0	7953	8196	41

**Tabla 5.4:** Resultados computacionales para la instancia 3, con autobuses de 38 y 45 pasajeros.

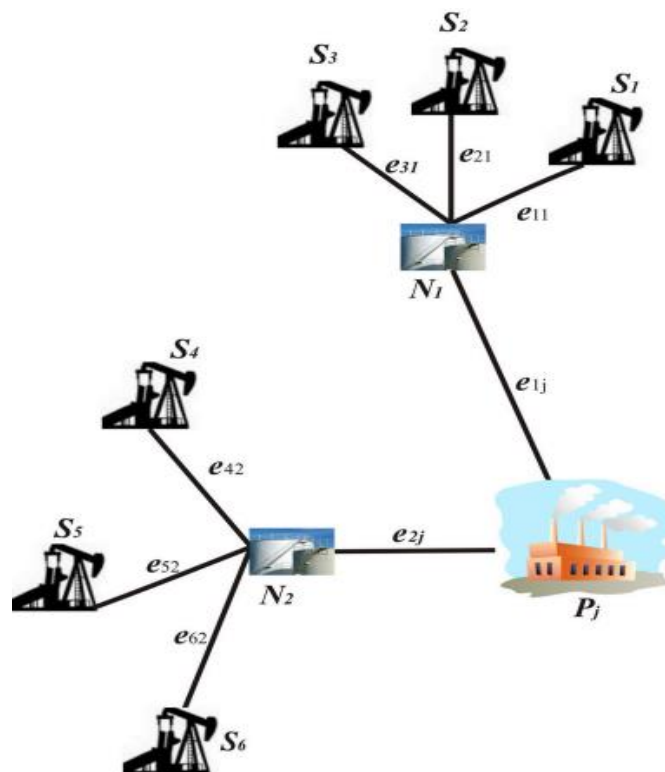
La propuesta basada en optimización por colonia de hormigas y la utilización del software GAMS resuelve el problema de localización y ruteo con restricciones de capacidad. La utilización de una metaheurística permite abordar problemas de grandes dimensiones y obtener resultados satisfactorios. El algoritmo resuelve la localización de paradas en función de una zonificación previa en la región de interés. Esta condición nos permite satisfacer la demanda de los usuarios en cada sector, minimizando los costos globales por la prestación del servicio. Finalmente, es importante destacar que el algoritmo diseñado es general, permitiendo su aplicación a distintas versiones de un LRP.

### 5.3 CASO DE ESTUDIO: RED DE CAÑERIAS

Una aplicación distinta del problema de localización y ruteo se presenta como el diseño de una red de cañerías para la distribución de hidrocarburos, junto a la elección de un conjunto

de ubicaciones intermedias que concentren la producción a los fines de optimizar el transporte desde distintos yacimientos hacia plantas de procesamiento (Figura 5.5).

El diseño de una red de cañerías para la distribución es un problema clásico de optimización. En los últimos años aumentó el interés y la necesidad de contar con un software de optimización como herramienta que ayude a encontrar una configuración exitosa. Cuando la planificación se realiza correctamente, el uso de cañerías resulta ser más económico, seguro y confiable que otros medios de transporte. Además, permite lograr mayor competitividad en un mercado de creciente demanda.



*Figura 5.5: Esquema del problema.*

Además del diseño de la red y la selección de las rutas, en estos tipos de problemas se suelen tener en cuenta aspectos tales como: las necesidades del mercado, el cumplimiento de las regulaciones existentes, los riesgos ambientales, el análisis de inventario, las

especificaciones de las cañerías y sus componentes. Muchos de los trabajos han sido resueltos mediante el uso de metaheurísticas (Costa et al., 2000; Rui et al., 2010 y Yaghini et al., 2012).

Aquí se presenta un modelo como un problema de localización combinado con elementos de transporte. Se desea minimizar los costos de instalación y transporte ( $Z$ ) con el objeto de enviar de manera eficiente la producción de los campos petrolíferos ( $S_i$ ) a las plantas de procesamiento ( $P_j$ ). El modelo evalúa la localización de nodos concentradores intermedios ( $N_i$ ) que recolectan el flujo enviado desde varios yacimientos para su posterior traslado en una única cañería. Este problema combinatorio consiste en determinar un subconjunto de múltiples opciones para los posibles lugares cuyos costos varían de acuerdo a su ubicación geográfica. Además, para cada solución potencial al problema de localización, es decir, para cada subconjunto de nodos concentradores, se necesita generar un plan de transporte. En realidad, la instalación de nodos concentradores intermedios se asocia a un problema de transbordo. Sin embargo, cualquier problema de transbordo se puede convertir fácilmente en un problema de transporte equivalente (Winston, 2004). Cabe remarcar que un problema equivalente tiene la misma solución que el problema original.

El diseño ingenieril detallado de este problema en particular fue efectuado como un prototipo descrito en (Oteiza et al., 2015). En base a esta experimentación, se decidió llevar a cabo su resolución por medio de la técnica metaheurística SA y el software GAMS (Brooke et al., 1998). Con SA se logra obtener las ubicaciones para las localizaciones intermedias y luego, con GAMS se resuelve el problema de transporte. Esta formulación se clasifica como un problema NP-Completo (Kariv y Hakim, 1979). Entonces, conseguir soluciones factibles a través de métodos heurísticos, en lugar de métodos exactos, se justifica plenamente.



### 5.3.1 METODOLOGÍAS DE RESOLUCIÓN

El espacio de soluciones está representado por medio de vectores binarios  $v$ . Cada vector  $v$  de  $n$  bits se corresponde con las  $n$  ubicaciones posibles para los nodos concentradores, asignándoles el valor 1 a cada nodo cuya instalación ha sido propuesta y 0 en caso contrario. El procedimiento se puede resumir en la Figura 5.6, donde se describe el proceso de inicio, en el cual se carga la matriz de costos y los parámetros a utilizar, para luego generar una solución aleatoria. Después, para cada una de las iteraciones establecidas se mantiene la temperatura según la longitud de la cadena de Markov elegida y se genera una nueva solución con la matriz de costos modificada, que será evaluada mediante GAMS para determinar su valor de aptitud final.

En base a tres aspectos: la diferencia entre las dos soluciones, una función probabilística basada en esta diferencia y la temperatura actual, se resuelve cuál de las dos soluciones mantener. De esta manera, es posible aceptar soluciones peores a la mejor encontrada y así lograr escapar de los óptimos locales.

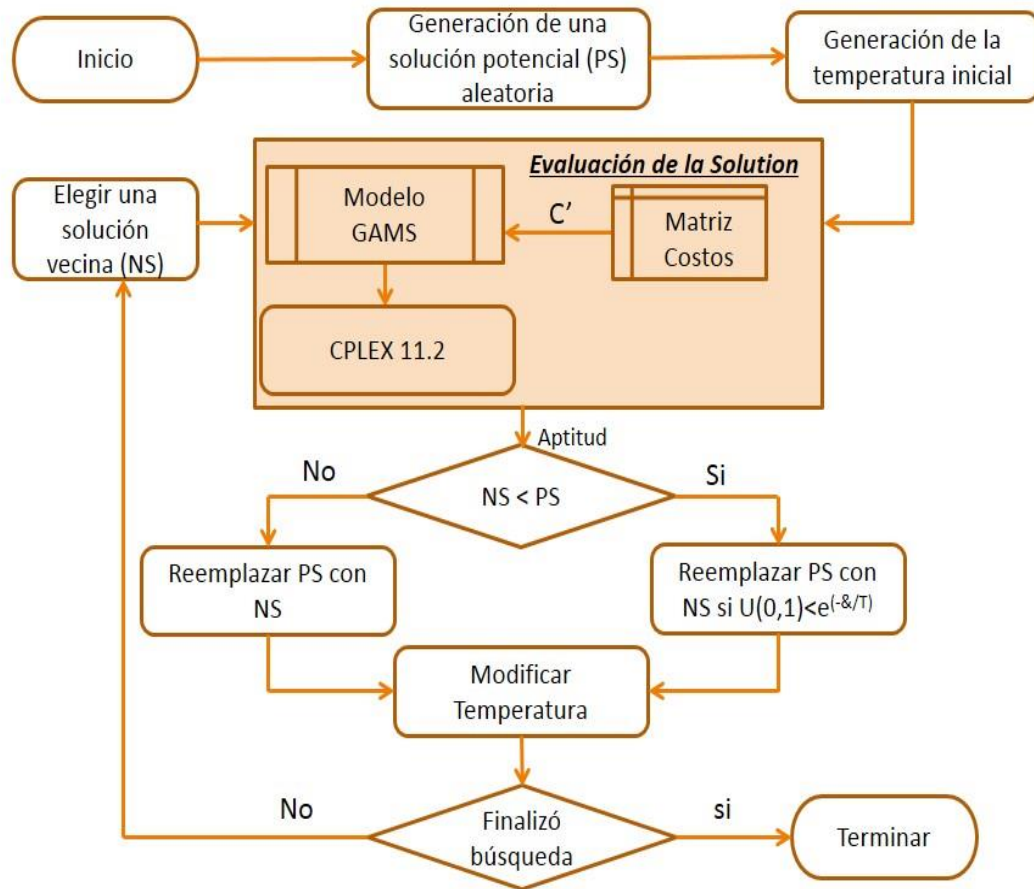


Figura 5.6: Diagrama SA - GAMS

GAMS permite resolver problemas de manera rápida y eficiente. Su principal ventaja es que proporciona la posibilidad de aplicar diferentes resolvers (lineal, no lineal, modelos enteros, etc.) a los modelos generados: Sólo se deben hacer pequeños cambios en el archivo de entrada. Para modelar los casos de estudio en GAMS, se simplificó el análisis considerando que la producción de todos los campos es transportada hacia alguna planta destino. El costo de construcción de cada nodo concentrador se calculó en función de la altura sobre el nivel del mar donde está localizado. Finalmente, se tuvo en cuenta las distancias y diferencias de altura con el fin de calcular el costo de llevar la producción de un lugar a otro.

Para cada solución potencial, el algoritmo genera una modificación de la matriz de costos  $C$ . Posteriormente, GAMS emplea esa matriz  $C$  modificada con el fin de evaluar el valor de aptitud correspondiente a  $Z$ . Cuando no se elige el  $k$ -ésimo nodo concentrador ( $N_k = 0$ ), todas las entradas de  $C_{ij}$  para  $i=1, \dots, m$ , y  $j=F+k$  son penalizadas con un valor extremadamente alto para que  $N_k$  nunca se tenga en cuenta.

La Figura 5.7 representa un pequeño ejemplo en el que se ilustra la acción algorítmica en la configuración de los nodos en  $C$ . Cuando un nodo no es considerado en la configuración, por ejemplo,  $N_2$ , la columna que corresponde a ese nodo asume valores muy altos, es decir, las entradas  $C_{i7}$  ( $i = 1, 9$ ) tienden a infinito. De esta manera, cuando GAMS emplea la matriz  $C$  modificada,  $N_2$  no será elegido por su alto costo. La Figura 5.7 también muestra a la derecha la red resultante para  $N_2 = 0$ .

### Una pequeña red de cañerías.

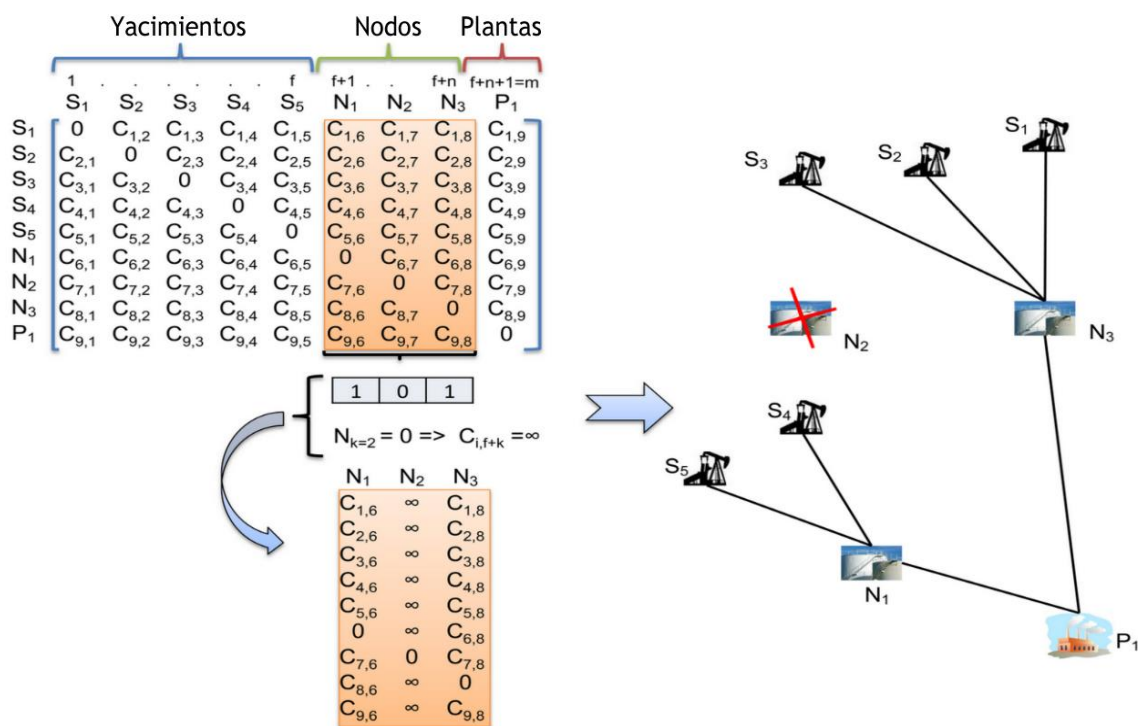


Figura 5.7: Modificaciones de la matriz de costos cuando se descarta el nodo  $N_2$ .

### 5.3.2 DISCUSIÓN DE RESULTADOS

El costo asociado de la matriz  $C$  se genera sólo una vez, al comienzo del procedimiento. Entonces, para cada etapa, sus elementos se modifican de acuerdo a los nodos activos. La técnica ahorra almacenamiento y evita estructuras de datos adicionales que afectarían el rendimiento computacional. El algoritmo es capaz de encontrar la cantidad adecuada de nodos concentradores y sus localizaciones más convenientes.

El desempeño computacional del enfoque propuesto fue evaluado mediante la resolución de distintos casos de estudio reales existentes en la provincia de Santa Cruz, Argentina, en los que se evalúan distintas combinaciones de yacimientos y nodos concentradores (Rodríguez et al., 2013a). La Tabla 5.5 muestra 2 ejemplos representativos junto a la solución óptima, el valor medio obtenido y el tiempo computacional requerido.

<b>Ejemplo</b>	<b>Yacimientos</b>	<b>Nodos Concentradores</b>	<b>Mejor Solución [ <math>10^6</math> \$]</b>	<b>Media [ <math>10^6</math> \$]</b>	<b>Tiempo CPU [seg]</b>
1	5	3	360	360	1.27
2	21	28	623	634	73.28

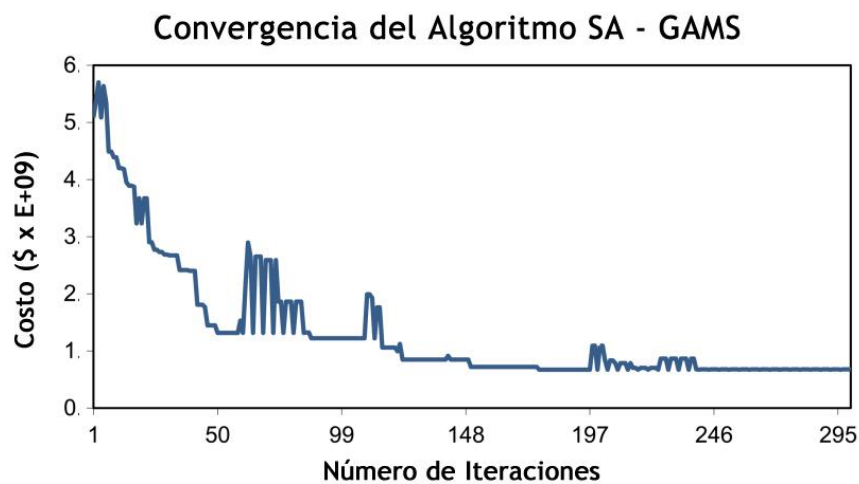
*Tabla 5.5: Resultados obtenidos para los dos ejemplos presentados.*

Los resultados obtenidos por el algoritmo son apropiados y se obtienen en tiempos relativamente cortos. La Tabla 5.5 proporciona los valores finales. A los fines comparativos, la mejor solución obtenida se muestra en la cuarta columna. Los valores reportados en la quinta columna representan el valor promedio de aptitud de 30 ejecuciones. El algoritmo ha demostrado ser robusto porque, en todas las pruebas, la mejor solución se obtuvo en más del 60 % de las ejecuciones, alcanzando el resto de las veces valores que están muy cerca de la misma. Además, luego de ejecutar reiteradas veces el algoritmo observamos que la solución que aparece con más frecuencia puede ser adoptada como la mejor.

El rendimiento de la metaheurística se ve directamente afectado por los valores elegidos para los parámetros. Estos valores se fijaron después de una fase de prueba preliminar, donde se estudió la calidad de las soluciones obtenidas y el tiempo de cálculo que se requería para obtenerlas. De acuerdo con esta búsqueda, los mejores valores adoptados para los ejemplos son los siguientes:

- Descenso de temperatura = 0,99
- Longitud de la cadena de Markov = 3, (rango de pruebas: [1, 10])
- Número de iteraciones = 100, (rango de pruebas: [50, 300])

La Figura 5.8 muestra el comportamiento del algoritmo en una sola ejecución. Se destaca la forma en que disminuye el valor de aptitud durante la primera mitad de las iteraciones. Cuando se registra un pico sucede que se ha aceptado una solución peor en pos de una nueva dirección en la búsqueda. Además, la explotación en las áreas más prometedoras se realiza en la segunda mitad de las iteraciones. Por lo tanto, la probabilidad de aceptar una solución peor en ese momento es menor. De esta manera, se consigue flexibilidad en el algoritmo y se logra escapar de óptimos locales, mejorando así considerablemente su rendimiento.



*Figura 5.8: Convergencia del Algoritmo SA - GAMS.*

Las bondades del algoritmo fueron demostradas en el análisis de los resultados. Fue programado cuidadosamente para asegurar la eficiencia y la flexibilidad. Además, es eficiente porque requiere tiempos de cálculo relativamente cortos, mientras que es flexible porque logra escapar de óptimos locales hasta encontrar una buena solución.

# CAPÍTULO 6

## HIPERHEURÍSTICAS

---

### *Contenidos de este capítulo*

Se presenta el concepto de hiperheurística. Sus principales características y los beneficios que se podrían llegar a obtener por su utilización. Establecemos una moderna clasificación de las hiperheurísticas y exponemos las nuevas tendencias asociadas al uso de las mismas.

Este capítulo es importante porque muestra una visión general de las hiperheurísticas, facilitándose así la comprensión de la técnica que se propone en el capítulo siguiente.

## 6.1 INTRODUCCIÓN

Las heurísticas y especialmente las metaheurísticas demostraron ser capaces de obtener excelentes resultados en problemas de optimización combinatorios complejos. Sin embargo, no son capaces de asegurar que una técnica funcione de manera satisfactoria en múltiples instancias (Wolpert y Macready, 1997). Esta es la principal razón por la que las hiperheurísticas surgen como nuevas estrategias para resolver esta clase de problemas.

Una hiperheurística es un método de búsqueda general que funciona a un nivel más alto de abstracción que el enfoque metaheurístico. Es aplicable a múltiples instancias de un problema e incluso puede utilizarse para resolver problemas distintos. El método dispone de un conjunto de técnicas de resolución de problemas y gestiona la ejecución de la técnica apropiada en un determinado momento del proceso de búsqueda.

El término hiperheurística empezó a ser utilizado en el año 2000. Aquí se presenta la idea de la metodología como una heurística para elegir heurísticas (Cowling et al., 2001). En Burke et al. (2003) se define hiperheurística como un método de búsqueda o mecanismo de aprendizaje para seleccionar o generar heurísticas que resuelvan problemas de búsqueda computacionales. Estamos en presencia de un enfoque moderno, donde se prioriza la capacidad de detectar cual es la mejor manera de resolver la búsqueda cuando se presenta una situación particular, sin prestar demasiada atención al problema en sí mismo.

Una hiperheurística puede ayudarse de cualquier tipo de heurística de bajo nivel, las cuales son vistas como un sistema de caja negra que colabora ofreciendo una nueva solución potencial al problema. Este proceso contribuye con aspectos muy deseados por los investigadores en la resolución de problemas, esto es, un enfoque general, simple y fácil de implementar (Ross, 2005).



Con una hiperheurística se pretende automatizar el diseño, generando un marco que controle la elección de la heurística apropiada considerando las características de su desempeño y las condiciones actuales del problema. Esta situación alcanza mayor consideración si se tiene en cuenta que la mayor parte del tiempo que se utiliza para resolver un problema, se lo ocupa en adaptar una técnica para que funcione adecuadamente.

Las hiperheurísticas tienen como objetivo principal el aporte de soluciones generalizadas a los problemas, en lugar de apuntar a los mejores resultados (Ross, 2005). Entonces, lo que se persigue es incrementar la generalidad de los métodos (Cowling et al., 2001). En cambio, las metaheurísticas más optimizadas requieren de cierta experiencia en el dominio del problema y la realización de múltiples experimentos, lo cual trae asociado un costo significativo para su implementación. En definitiva, se pretende obtener una técnica lo suficientemente robusta como para operar con eficacia en un conjunto de problemas distintos.

La manera en que una hiperheurística guía la búsqueda de una solución consiste en emplear indicadores de desempeños que están relacionados al funcionamiento de las heurísticas de bajo nivel. Estos indicadores suelen relacionarse a la calidad de la solución obtenida y el tiempo de cómputo empleado. La hiperheurística no manipula directamente las soluciones encontradas, ni tiene información respecto a cómo está compuesta la función objetivo. Solo requiere conocer si es una función de maximización o minimización.

Estos lineamientos están expresados en la definición de hiperheurística presentada en Chakhlevitch y Cowling (2008):

1. Una hiperheurística es una heurística de más alto nivel que administra un conjunto de heurísticas de bajo nivel.
2. Una hiperheurística busca un buen método para resolver el problema en lugar de una buena solución.

3. Una hiperheurística no precisa información específica del problema. El proceso general solo necesita conocer las técnicas de bajo nivel que puede utilizar y la función objetivo a ser minimizada o maximizada.

Una revisión detallada sobre hiperheurísticas se puede encontrar en Burke et al. (2009). Al no trabajar con información directa del dominio del problema, puede utilizar el mismo método en muchos problemas sin emplear demasiado tiempo ni recursos en adaptarlo para cada uno de ellos (Ozcan et al., 2008). Sin embargo, la generalidad conseguida por las hiperheurísticas puede derivar en una pérdida del rendimiento en comparación con aquellos métodos que han sido diseñados a medida para un problema. En muchos casos esta situación no representa un problema, debido a que el esfuerzo para diseñar una técnica a medida es muy superior y generalmente no es posible su reutilización.

## 6.2 CLASIFICACIÓN

De manera análoga a la clasificación de metaheurísticas. Las hiperheurísticas se pueden dividir en constructivas e iterativas. Las constructivas utilizan una técnica de bajo nivel para generar cada parte de una solución en cada iteración y las iterativas inician con una solución completa y de forma iterativa van modificando la solución siguiendo una dirección de búsqueda en función de estructuras vecinas (Bai, 2005).

Otra forma de clasificación diferencia las hiperheurísticas con aprendizaje y sin aprendizaje. En las primeras, las llamadas a las técnicas de bajo nivel se realizan acorde a una secuencia predeterminada. En cambio, las hiperheurísticas con aprendizaje van modificando las preferencias de elección en función del rendimiento histórico de las técnicas (Soubeiga, 2003).

Otros autores (Chakhlevitch y Cowling, 2008) clasifican las hiperheurísticas en cuatro categorías:

1. Aleatorias: Se elige la técnica a utilizar al azar. Existen algunas variantes en esta categoría. Por ejemplo luego de elegir una técnica, se la aplica hasta que no ofrezca mejoras, en ese caso se vuelve a elegir otra técnica.
2. Codiciosas: En cada iteración se aplican todas las técnicas, seleccionando la que mejores resultados ofrezca, luego se cambia de técnica.
3. Basadas en metaheurísticas: La solución se representa como un conjunto de metaheurísticas que se deben aplicar siguiendo un cierto orden. Dicho orden se va modificando según las particularidades de cada metaheurísticas.
4. Con aprendizaje: La elección se basa en la eficacia acumulada de cada técnica desde el inicio del programa. En cada iteración se recompensa y castiga a las técnicas para ir estableciendo un orden de prioridades.

### **6.3 HIPERHEURÍSTICA PARALELA**

Mediante el empleo de algoritmos paralelos, se pretende la ejecución de sentencias en varias unidades de procesamiento al mismo tiempo. Estas instrucciones pueden ser complementarias para la realización de una determinada tarea. Los beneficios de una ejecución paralela se traducen en disminuciones de los tiempos de cómputos y un mejor aprovechamiento de recursos.

La implementación de un sistema paralelo implica establecer la arquitectura y el modelo de programación a utilizar. Respecto a la arquitectura paralela, una categorización muy representativa es la taxonomía de Flynn (1996). Consiste en definir cuatro categorías de paralelismo en función del número de instrucciones concurrentes y el flujo de datos disponibles (ver Tabla 6.1).

	<i>Instrucción simple</i>	<i>Instrucción Múltiple</i>
<i>Datos Simple</i>	SISD	MISD
<i>Datos Múltiples</i>	SIMD	MIMD

**Tabla 6.1:** Taxonomía de Flynn.

*Una instrucción, un dato (SISD):* Un sistema con un único procesador que ejecuta las instrucciones secuencialmente. No hace uso del beneficio del paralelismo.

*Múltiples instrucciones, un dato (MISD):* No suele utilizarse debido a que disponer de múltiples instrucciones solo es efectivo cuando existen múltiples flujos de datos.

*Una instrucción, múltiples datos (SIMD):* Un equipo donde una única instrucción actúa sobre un conjunto de datos.

*Múltiples instrucciones, múltiples datos (MIMD):* Un sistema con varios procesadores autónomos que ejecutan simultáneamente instrucciones diferentes sobre datos diferentes.

Una vez establecida la arquitectura del hardware, se debe diseñar el software de manera que sea posible el uso concurrente de múltiples procesadores. Tradicionalmente, los programas fueron desarrollados para ejecutarse de manera secuencial. El aprovechamiento de los beneficios que ofrece el paralelismo tiene un costo importante al momento de desarrollar el software. Implica contemplar cuestiones como la comunicación, la sincronización, formas distintas de manipular y distribuir los datos, balancear la carga de los procesadores, mayor tasa de errores durante la programación, entre otras.

Los modelos de programación paralela pueden ser clasificados en función de:

- I. *La interacción de los procesos:* Se establece la manera en que los procesos se comunican entre sí. Los modelos más difundidos son:
  - a) *Memoria Compartida*, donde todos los procesadores son capaces de acceder a un mismo espacio de direcciones global. Aquí, se debe controlar los accesos concurrentes

para evitar incoherencias y además disponer de un ancho de banda suficiente como para que todos los procesos accedan sin inconvenientes.

b) *Memoria Distribuida*, donde cada proceso tienen asignada una memoria local, obteniendo un funcionamiento independiente. Como los procesadores están conectados, cuando se necesita acceder a datos de otro procesador estos deben enviarse por esa misma red. Aquí, se debe implementar mecanismos de comunicación entre las tareas paralelas.

II. *La descomposición de problemas*: Existen distintas clasificaciones basadas en la descomposición (Hansen, 1993). Las estrategias más representativas son:

a) *Maestro-Esclavo*, donde el problema se divide en tareas que son ejecutadas por los esclavos, los cuales envían los resultados al maestro para que pueda constituir la solución final.

b) *Modelo de Islas*, permite distribuir la carga de procesamiento (Biazzini et al., 2009). Se dividen los datos entre todos los procesadores, los cuales ejecutan el mismo código. En algunos casos, un procesador puede requerir un dato calculado por otro, por lo que pueden definirse mecanismos de comunicación entre los mismos.

Las características descritas de una hiperheurística son propicias para el máximo aprovechamiento de una plataforma paralela. La posibilidad de direccionar distintas opciones de búsqueda permite la obtención de mejores resultados. De ese modo se justifica el desarrollo de una hiperheurística paralela.

Una hiperheurística paralela está claramente influenciada por dos aspectos principales.

- i. El tiempo de ejecución de cada uno de los algoritmos de bajo nivel.
- ii. El tiempo demandado y la cantidad de veces que se realice el proceso de sincronización.

Es necesario encontrar un equilibrio en estos procesos. Si las sincronizaciones no son suficientes deriva en el mal aprovechamiento de los descubrimientos realizados por los otros algoritmos. En caso de sincronizar con demasiada frecuencia provoca que los algoritmos no tengan el tiempo necesario para evolucionar correctamente. En Rattadilok et al (2005), se propone un modelo para el empleo general de heurísticas de bajo nivel aprovechando los beneficios que proporciona la computación paralela.

En el siguiente capítulo nos enfocaremos en la hiperheurística novedosa desarrollada en esta tesis doctoral para la resolución del LRP.

# CAPÍTULO 7

## DISEÑO DE UNA HIPERHEURÍSTICA PARA EL LRP

---

### *Contenidos de este capítulo*

Aquí, se propone una técnica hiperheurística basada en las metaheurísticas GA, SA y ACO, para luego aplicarla a un problema de localización y transporte.

Mostramos detalles del modelo generado, la forma en que se controla y selecciona la técnica a utilizar y las particularidades que contemplan la ejecución del algoritmo en paralelo. Esta propuesta posibilita la obtención de mejores resultados.

Luego de la exposición de los resultados obtenidos, se realiza un análisis de la contribución de cada técnica en la búsqueda de la solución deseada.

## 7.1 INTRODUCCIÓN

A partir de las consideraciones establecidas en el capítulo anterior, en este capítulo introducimos una nueva hiperheurística paralela basada en metaheurísticas (HHP). Deseamos obtener un método que sea capaz de resolver diferentes instancias asociadas a problemas de localización y transporte. Esta hiperheurística opera a un nivel superior al de las técnicas metaheurísticas, de las cuales se asiste para guiar la búsqueda.

En general, cuando se hace referencia a una hiperheurística basada en metaheurísticas, se relaciona a una técnica que tiene el comportamiento de una metaheurística y que funciona en un espacio de búsqueda de distintas heurísticas (Chakhlevitch y Cowling, 2008), como por ejemplo Algoritmos Genéticos (Cowling et al., 2002) y Búsqueda Tabú (Burke et al., 2007). De esta manera, el algoritmo metaheurístico buscará cual es la mejor técnica a utilizar, sin conocimientos del problema en cuestión.

Sin embargo, nuestra propuesta consiste en utilizar un conjunto de metaheurísticas como técnicas de bajo nivel de la hiperheurística. Una versión preliminar del diseño de esta propuesta fue presentada en Rodríguez et al. (2014b), allí se proponía el uso de SA y ACO como técnicas de resolución de bajo nivel de una hiperheurística. Durante su ejecución, la elección de la técnica a emplear se elegía al azar hasta tanto se cumpla con la condición de parada. En la versión que presentamos en esta tesis varias mejoras han sido incorporadas con el objeto de perfeccionar el proceso de búsqueda:

- Se incorporó a la metaheurística GA como tercer método de resolución, con el objeto de obtener un mejor recorrido en el espacio de búsqueda.
- Se implementó un mecanismo que guía la elección en función del estado de la búsqueda y el rendimiento observado en las últimas iteraciones. La hiperheurística lleva un control del resultado encontrado por cada algoritmo, de manera que cuando hay un



estancamiento de la búsqueda en alguna de las técnicas se favorezca el uso de alguna otra. Además, cuando una técnica encuentra una mejor solución se activa un componente que privilegia la elección de la misma técnica para la próxima elección.

- Se desarrolló un proceso que realiza un intercambio de datos permitiendo que una técnica disponga de información descubierta por otra. Esta funcionalidad, logra por un lado acelerar el proceso de búsqueda y también funciona como herramienta para escapar de óptimos locales.

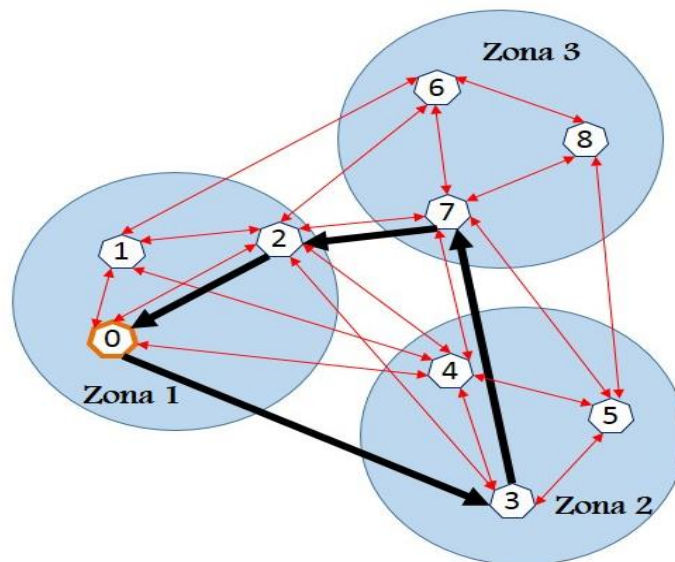
- Por último, se incorpora una estrategia paralela basada en la ejecución de hilos. Así, cuando se cuenta con varios procesadores se puede aprovechar las características de una hiperheurística haciendo que las técnicas de bajo nivel se ejecuten en simultáneo. De esta manera, se logra disminuir los tiempos computacionales considerablemente.

La versión del LRP que se pretende resolver, busca optimizar el servicio de transporte público de pasajeros de una determinada ciudad, en la que existe la posibilidad de instalar múltiples paradas. La región está dividida en zonas homogéneas en función de los viajes demandados. Para la zonificación se tienen en cuenta factores como cantidad de viviendas, población total, entre otros. Así se establece una asociación entre los pasajeros a las paradas y las paradas a las zonas.

El objetivo es minimizar el costo total por la instalación de una línea que satisfaga la demanda de todas las zonas. Para ello, se debe encontrar las mejores ubicaciones para las paradas a instalar y establecer los recorridos de los autobuses que partiendo de una estación central predeterminada, atravesarán dichas paradas. Se asume que los pasajeros están asociados a una parada potencial y cuando una parada no es seleccionada, sus pasajeros son asignados a la parada válida más cercana, de esa manera se satisface a toda la demanda.

## 7.2 MODELO PROPUESTO DE HIPERHEURISTICA PARA LRP

El modelo propuesto puede ser representado mediante un grafo, donde los vértices representan las paradas y las aristas representan las rutas disponibles para la circulación de los autobuses, que tendrán una capacidad homogénea  $q$ . El costo total que se pretende minimizar está compuesto por el costo de realizar los recorridos seleccionados, el costo de instalación de las paradas elegidas y el costo asociado al traslado de un pasajero de una parada no seleccionada a la parada más cercana. La formulación del problema es la misma que se presenta en la Sección 5.2.



*Figura 7.1: Representación de un prototipo.*

Un caso de estudio académico, con el que se realizaron las primeras pruebas consiste en una región dividida en tres zonas que contienen un total de ocho paradas potenciales (ver Figura 7.1).

Las metaheurísticas que optamos por utilizar dentro de la hiperheurística propuesta son GA, SA y ACO. La elección de estas metaheurísticas se fundamenta en la decisión de utilizar técnicas que emplean diferentes estrategias para encontrar una solución. GA y SA gestionan la

búsqueda manipulando soluciones completas, en cambio ACO incorpora iterativamente un componente hasta completar una estructura que representa la solución. Otra diferencia fundamental es que GA y ACO son técnicas basadas en poblaciones, en cambio SA realiza la búsqueda modificando una única solución, generando así una trayectoria entre la solución inicial y la final.

La conveniencia de utilizar técnicas conceptualmente distintas radica en la posibilidad de disponer de una estrategia adecuada en los distintos estados por los que puede atravesar la búsqueda. En general, al comienzo del proceso conviene abarcar la mayor parte del espacio de búsqueda posible, en esa situación conviene priorizar el uso de técnicas poblacionales, fundamentalmente GA por su capacidad de búsqueda en múltiples vecindarios. En otras etapas del proceso conviene explotar los sectores favorables, este escenario es propicio para técnicas como SA.

La evaluación del estado del problema se hace en función del resultado encontrado por cada una de las técnicas en las últimas  $j$  ejecuciones. El valor  $j$  funciona como la dimensión de una memoria en la que se recopila información relacionada a:

- La cantidad de veces que finalizó con éxito la ejecución ( $E_{sol}$ ), esto implica haber encontrado una mejor solución.
- La cantidad de veces que no pudo mejorar la solución actual ( $R_{sol}$ ).
- El tiempo que demandó obtener las últimas soluciones ( $T_{sol}$ ).

Con estas medidas de desempeño se asigna un valor de fuerza (peso) a cada metaheurística. De esta manera, es posible hacer que la técnica con mayor peso disponga de más chances de ser seleccionada para su ejecución. El pseudocódigo 7.1 presenta los procesos más importantes de la hiperheurística propuesta.

**Algoritmo HHP:**

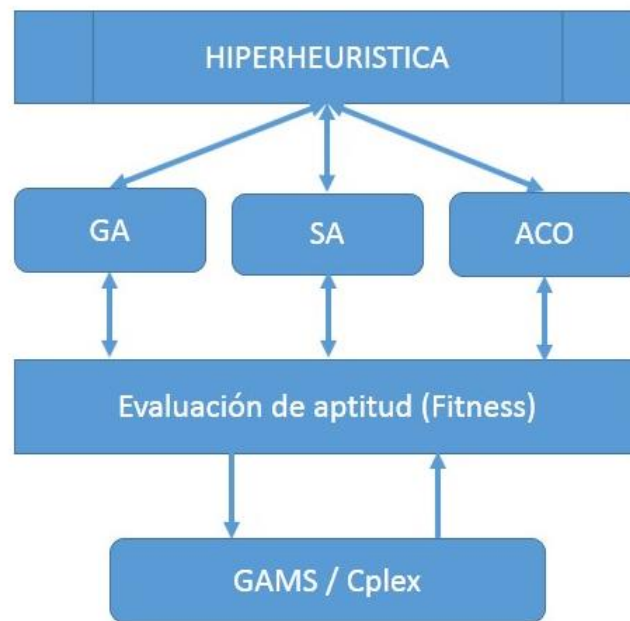
- 
1.  $S_{act} \leftarrow$  Inicialización (); // Inicializar Solución.
  2. **Mientras** no finalice **hacer** // No se alcancen todas las generaciones.
  3. Estado  $\leftarrow$  EvaluarEstado( $E_{sol}$ ,  $R_{sol}$ ,  $T_{sol}$ ); //Análisis de la evolución del método
  4.  $H \leftarrow$  SeleccionarTécnica(Estado); // Elección de la técnica Metaheurística a emplear.
  5. AplicarTécnica( $H$ ); // Se aplica la Metaheurística seleccionada.
  6. CompartirInformacion(); // Intercambio de información entre las técnicas.
  7. **Fin Mientras**
- 

**Algoritmo 7.1: Pseudocódigo HHP.**

Por último, cuando una técnica encuentra una mejor solución, se procede a compartir el resultado encontrado con las demás, de la siguiente manera:

- GA incorpora a la solución como un individuo más reemplazándolo por algún otro seleccionado al azar.
- ACO aumenta la cantidad de feromona en el recorrido asociado a la solución encontrada.
- SA adopta la nueva solución como solución inicial.

Las tres metaheurísticas funcionan utilizando la misma función de aptitud, que es evaluada en este caso mediante GAMS (Figura 7.2). El flujo de información entre la hiperheurística y las metaheurísticas intervinientes no contiene información referida al problema. Este vínculo es para intercambiar información respecto a las soluciones encontradas y a los recursos utilizados para obtenerla. Se pretende evaluar y comparar el rendimiento de la hiperheurística con cada una de las metaheurísticas ejecutadas de forma individual.



*Figura 7.2: Modelo de la hiperheurística.*

### 7.3 IMPLEMENTACIÓN PARALELA

Cuando existe la necesidad de disminuir los tiempos de cómputos, desarrollar algoritmos que se ejecutan en paralelo resulta una opción adecuada. La posibilidad de contar con múltiples procesadores nos permite implementar una programación basada en hilos de manera que dos o más procesos se ejecuten de manera simultánea.

Empleamos una arquitectura multicore, una tecnología donde los procesadores están incorporados al mismo chip. Esta condición permite una comunicación más rápida entre los procesadores. Además, un sistema de memoria compartida para que todos los procesos tengan acceso a los datos almacenados.

Nuestra hiperheurística está preparada para funcionar en una plataforma que soporte múltiples hilos, a los fines de ejecutar cada uno de ellos en un procesador disponible. De esta manera, para cada hilo se selecciona una metaheurística y se procede con su ejecución hasta su finalización, momento en que se reportan los resultados obtenidos. En ese momento, se procede

con el intercambio de información hacia las demás técnicas que podrían estar en plena ejecución, razón por la cual se necesita un proceso de sincronización para coordinar el acceso a los datos compartidos y así evitar inconsistencias en el sistema.

Nuestro algoritmo implementado en Java soluciona ese inconveniente haciendo uso de propiedades intrínsecas del lenguaje, que consisten en definir los métodos o bloques de código a los que se accede en exclusión mutua. Así, es posible tener en ejecución más de una técnica a la vez, permitiéndonos obtener un mejor proceso de búsqueda que podría llevarnos hacia soluciones más precisas.

## **7.4 EVALUACIÓN DE RESULTADOS**

Los algoritmos fueron implementados en el lenguaje Java y los experimentos computacionales fueron realizados en una PC Intel Core i5-3330 con 8 GB de RAM. En cuanto a los recursos utilizados para la ejecución del algoritmo, cabe mencionar que el equipo empleado no posee limitaciones de procesamiento ni tampoco de memoria, para justificar un análisis en particular en ese sentido.

El algoritmo HHP se utilizó para resolver el prototipo planteado (Caso I) en la Sección 7.2, con la intención de estudiar su comportamiento. Finalmente, también fue empleado para resolver una instancia real (Caso II) que cuenta con 50 localizaciones de potenciales paradas, distribuidas en una región compuesta por 12 zonas (Ver Sección 5.2). La discusión de resultados que se efectúa a continuación, se realiza en base a los valores obtenidos a partir de 30 ejecuciones independientes.

La Tabla 7.1 muestra las configuraciones utilizadas para los dos casos. Se presenta también la mejor solución encontrada, el costo asociado a dicha solución, el promedio de las soluciones obtenidas y el tiempo computacional demandado. En el Caso I, como era de esperar por su reducido tamaño, el algoritmo encuentra siempre la mejor solución posible. Para el Caso

II, la cercanía entre la mejor solución encontrada y el promedio de todas las soluciones deja en evidencia la robustez del algoritmo.

Caso	Paradas	Zonas	Mejor Solución (rutas)	Costo (\$)	Promedio (\$)	Tiempo CPU (s)
I	8	3	0 3 7 2 0	244	244	11
II	50	12	0 29 4 27 5 26 22 40 2 39 19 25 33 0	4452	5274	29

**Tabla 7.1:** Resultados computacionales para los casos estudiados.

A partir de aquí, presentamos los resultados obtenidos por la técnica propuesta durante su aplicación al caso de estudio con 50 paradas y 12 zonas (Caso II). En la Tabla 7.2, presentamos detalles de los resultados encontrados por las metaheurísticas GA, SA y ACO ejecutadas de manera individual y los comparamos con los resultados de HHP. Se ha realizado esta comparación en función de la mejor solución encontrada y de los tiempos computacionales incurridos.

Algoritmo	HHP	GA	SA	ACO
Mejor Tiempo CPU (seg.)	19,28	96,58	15,47	102,63
Tiempo Promedio CPU (seg.)	30,67	143,43	31,26	109,52
Mejor Solución (\$)	4452	5332	4772	5493
Solución Promedio (\$)	5691,22	7680,3	5703,6	7783,55

**Tabla 7.2:** Comparación de Algoritmos.

HHP encuentra la mejor solución y tiene el mejor promedio de las soluciones obtenidas. Respecto al tiempo de cómputo, solo es superado en algunas ejecuciones por SA, aunque la supera ligeramente en el promedio de los tiempos utilizados.

También es posible valorar el rendimiento de los algoritmos mediante el número de evaluaciones que realizan. Se desea conseguir la mayor cantidad de óptimos parciales durante una ejecución con un mínimo número de evaluaciones. Por lo tanto, un porcentaje de eficacia puede calcularse como el cociente entre cantidad de óptimos y cantidad de evaluaciones. La

Tabla 7.3 presenta la cantidad de evaluaciones realizadas por cada uno de los algoritmos, así como también la cantidad de veces que en una evaluación se alcanzó el óptimo. Vistos los porcentajes de eficacia, HHP supera a las metaheurísticas comparadas.

La diferencia entre las evaluaciones que realizan las técnicas poblacionales (GA y ACO) y la técnica basada en trayectorias (SA) es significativa. Sin embargo, el esfuerzo computacional realizado por GA y ACO no se ve reflejado en el porcentaje de casos exitosos. En ese aspecto, se observa un mejor rendimiento de SA y otro muy superior de HHP. Merece destacarse que HHP necesita un número considerablemente menor de evaluaciones que las metaheurísticas estudiadas y además obtiene un gran porcentaje de nuevas mejores soluciones en dichas evaluaciones.

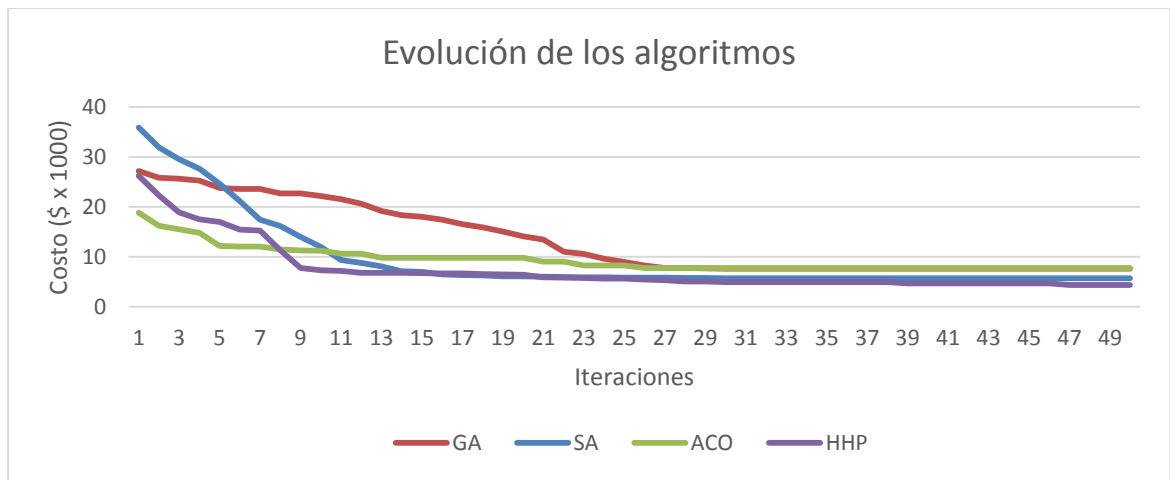
Algoritmo	Evaluaciones	Óptimos	Eficacia (%)
<b>HHP</b>	2174	37	1,70
<b>SA</b>	12626	58	0,46
<b>ACO</b>	30755	59	0,19
<b>GA</b>	73908	67	0,09

*Tabla 7.3: Comparación de Algoritmos.*

La Figura 7.3 muestra el desempeño temporal de los algoritmos, a medida que transcurren las iteraciones. El gráfico se basa en los valores promedios sobre treinta ejecuciones diferentes para cada uno de los algoritmos. El rendimiento superior de HHP se fundamenta en la colaboración entre las técnicas participantes y el control que se ejerce para influenciar la técnica a utilizar en determinados estados del problema.

Notamos que tanto HHP como las técnicas poblacionales GA y ACO, encuentran rápidamente mejores soluciones, mientras que SA tiene dificultades para encontrar soluciones satisfactorias rápidamente. Sin embargo, a partir de la segunda mitad de las iteraciones se observa que todas las soluciones son de la misma magnitud.



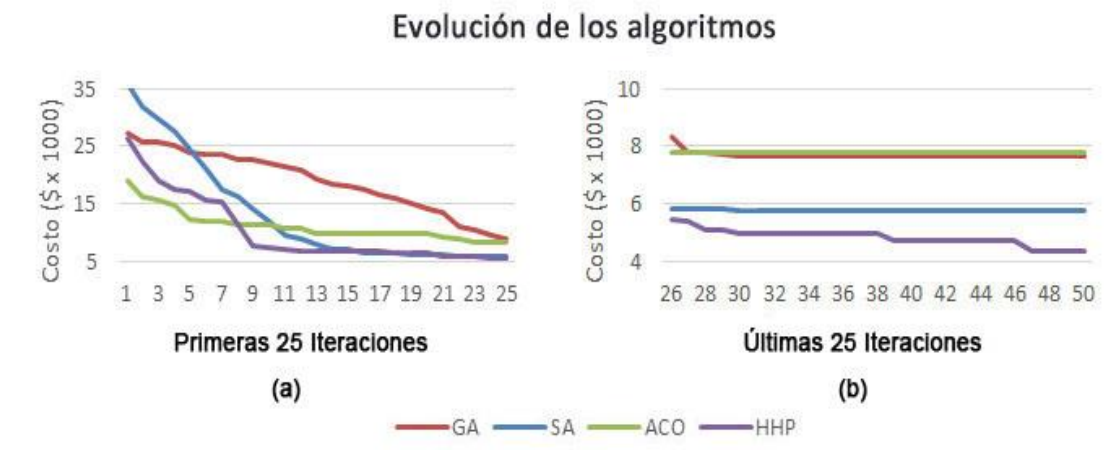


**Figura 7.3:** Evolución de los algoritmos.

Con el objeto de visualizar de mejor manera las bondades del algoritmo producto de los mecanismos de intercambio de información y de control de selección se presenta la Figura 7.4.

La información que se intercambia está asociada a las soluciones de buena calidad que fueron encontradas por cada una de las metaheurísticas. Esta es utilizada para acelerar la convergencia en las primeras iteraciones del algoritmo. Se puede observar que HHP es capaz de encontrar buenas soluciones rápidamente. En esta etapa se destaca el rendimiento de las técnicas poblacionales, por su capacidad de direccionar la búsqueda en distintos sectores del espacio de soluciones (Figura 7.4a).

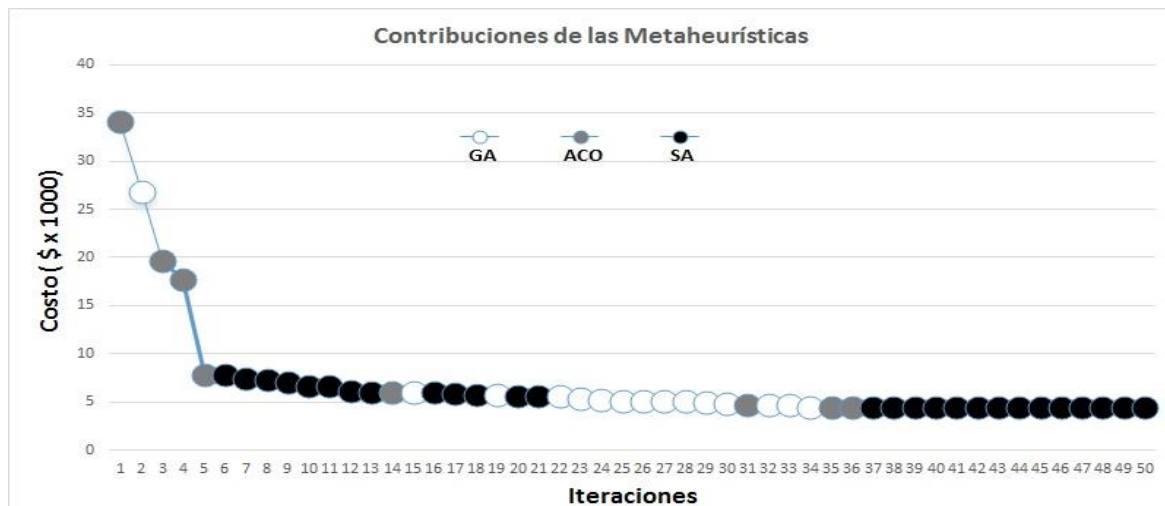
El control de selección de la técnica apropiada sirve para que cuando una metaheurística queda estancada en un óptimo local, se fomente la utilización de una técnica distinta para avanzar en la búsqueda de una mejor solución. Este proceso cobra mayor importancia a partir de la segunda mitad de las iteraciones, cuando las soluciones de las metaheurísticas son de la misma magnitud y empiezan a verse afectadas por óptimos locales. (Figura 7.4b).



**Figura 7.4:** Aceleración de la convergencia (a) – Escape de óptimos locales (b).

Generalmente, ocurre que una técnica poblacional contribuye a la mejora de una técnica basada en trayectorias y viceversa. Es decir que cuando SA queda atrapado en un óptimo local, GA y ACO pueden encontrar la manera de salir de él y en ese momento SA probablemente pueda llegar a mejorar esa solución. También, cuando SA y ACO se encuentran en un óptimo local la información ofrecida por SA ayuda a evitarlo, encontrando una mejor solución, y como además el nuevo individuo mejora la población, permite nuevas oportunidades de continuar la búsqueda.

No existe una marcada diferencia sobre el grado de colaboración de alguna de las técnicas, es decir que todas se ayudan de manera proporcional. Una característica de funcionamiento interesante consiste en la observación de que en la mayoría de las veces la mejor solución obtenida por la HHP fue encontrada por SA, producto de su mejor rendimiento en el proceso de explotación durante la última etapa del proceso. También se puede concluir que ACO es la técnica que mejores resultados ofrece en las primeras iteraciones, mientras que al finalizar encuentra dificultades para escapar de un óptimo local. La Figura 7.5 muestra la contribución que tuvo cada técnica para obtener mejores soluciones a lo largo de una ejecución con 50 iteraciones.



**Figura 7.5:** Soluciones encontradas por las distintas técnicas.

Respecto al tiempo de cómputo demandado, debemos mencionar que existen estrategias que podrían presentar mejoras en este sentido. Dependiendo de las características del problema y la necesidad de contar con soluciones rápidamente, es posible configurar parámetros para mejorar aún más el tiempo computacional. Naturalmente, si se dispone de información externa para usarse como solución inicial, el algoritmo también podría emplear menos tiempo de cómputo y ahorrar cálculos innecesarios como los que se hacen para la técnica SA al principio del proceso. De todas maneras, esa posibilidad atenta contra la generalidad de la técnica propuesta y su posibilidad de aplicar nuestro enfoque a múltiples problemas de optimización combinatoria.

En conclusión, los resultados obtenidos en el problema abordado en esta tesis demuestran la efectividad de nuestra propuesta. La hiperheurística logra un alto rendimiento en la búsqueda de soluciones, superando ampliamente a los obtenidos con las metaheurísticas GA, SA y ACO. En primera instancia se debe destacar que HHP es capaz de obtener soluciones de mejor calidad empleando menos tiempo de cómputo. Los valores medios exhibidos indican que estas condiciones se presentan en la gran mayoría de las ejecuciones realizadas, demostrando la robustez del algoritmo.

La implementación se llevó a cabo haciendo hincapié en hacer el mejor uso de las metaheurísticas participantes, así como también de los recursos de hardware disponibles. La programación de procesos que se ejecutan de manera sincronizada bajo una plataforma basada en hilos, permiten explotar el paralelismo en un entorno multiprocesador. Así, se optimiza el funcionamiento, incrementando la velocidad de ejecución del algoritmo.

Se destacan las contribuciones de los mecanismos de *intercambio de información entre las técnicas* y el *de control y selección de la técnica*. El primero permite acelerar la convergencia en el proceso de búsqueda y el segundo procura mejorar el escape de óptimos locales. Además, merece remarcarse que con este enfoque se puede disminuir la cantidad de soluciones que se necesitan evaluar, a la vez que se aumenta el porcentaje de veces en que se encuentra una mejor solución durante la ejecución de la hiperheurística.

# CAPÍTULO 8

## CONCLUSIONES

---

### *Contenidos de este capítulo*

Se establecen las conclusiones derivadas del trabajo realizado. Un resumen de las publicaciones generadas a partir de nuestras investigaciones. Además, un listado de actividades que podrían ser realizadas en una etapa posterior a la presentación de la tesis.

## 8.1 CONCLUSIONES

En esta tesis doctoral se han propuesto distintas técnicas basadas en metaheurísticas para la resolución de problemas asociados al transporte. En primera instancia se planteó el problema denominado IPTP, considerando las paradas de pasajeros como preestablecidas. Posteriormente se incorporó la dificultad de tener que elegir las paradas a utilizar entre múltiples posibilidades distribuidas en toda la región de estudio.

Consideramos que la importancia que tienen estos problemas para la sociedad en su conjunto justifica las investigaciones realizadas y el esfuerzo dedicado. Además de los problemas de transporte público, también se evaluaron los algoritmos desarrollados en problemas distintos al transporte de personas. Se realizó un estudio para optimizar la instalación de una red de cañerías para transportar un producto (gas, petróleo, derivados del petróleo, etc.) desde su lugar de origen hacia plantas de procesamiento, con la opción de instalar centros intermedios que almacenen temporalmente la producción antes de ser enviada al destino.

Los problemas de optimización combinatoria mencionados se categorizan como NP-Completos, debido a su dificultad para encontrar una solución óptima. Por esta razón se proponen técnicas basadas en metaheurísticas para la resolución de los problemas, las cuales son capaces de encontrar soluciones aproximadas en tiempos razonables.

En primer lugar realizamos una revisión de la literatura referente a las principales técnicas metaheurísticas y también sobre la problemática relacionada al transporte, especialmente al transporte público de pasajeros. En estos problemas existen intereses encontrados, por un lado se pretende mejorar el servicio pero también disminuir los costos. Tras reconocer que existen múltiples factores que influyen directamente en la solución al

problema, decidimos enfocarnos en algunos de los temas fundamentales como lo son los recorridos, las frecuencias y los autobuses a utilizar.

Se implementaron algoritmos que han tenido un desempeño destacable. Para resolver el IPTP se desarrollaron dos técnicas, una basada en GA y otra basada en SA. Merece destacarse que se incorporaron a los algoritmos características especiales que inciden directamente en el éxito alcanzado. Por ejemplo, la definición de un método de selección de vecinos nos sirvió para acelerar el proceso de búsqueda y la implementación de una memoria auxiliar se utilizó para evitar hacer evaluaciones innecesarias.

Además, nos orientamos a utilizar un software externo para realizar las evaluaciones de las potenciales soluciones. El motivo de esta decisión se circunscribe a la existencia en el mercado de distintas alternativas que son capaces de representar lo suficientemente bien el modelo planteado. Por ejemplo, para este primer problema utilizamos el programa SUMO, que nos permite simular un escenario de transporte con un gran nivel de detalle, el cual habría sido muy difícil de conseguir si hubiéramos empleado un modelo matemático.

De una comparación entre los dos algoritmos desarrollados, se concluye que si bien ambos tienen un buen comportamiento, SA es capaz de obtener mejores soluciones producto de la menor cantidad de evaluaciones que necesita realizar hasta que el algoritmo converge. Aunque también se reconoce que el GA realiza una búsqueda más general, logrando explorar gran parte del espacio de búsqueda.

A partir de esta conclusión y en el afán de obtener mejores tiempos computacionales se implementó una estrategia paralela para el algoritmo SA. Utilizamos la librería MALLBA, la cual facilita el desarrollo del algoritmo que se escribió con el lenguaje C++. Esta implementación paralela es capaz de disminuir considerablemente el tiempo de cómputo sin afectar la calidad de las soluciones, en comparación con su par secuencial. También merece destacarse que esta ganancia de tiempo aumenta cuando se incrementa el tamaño del problema.

A continuación se resolvió incorporar al estudio, la problemática asociada a la elección y localización de las paradas de pasajeros. Para este problema optamos por resolverlo basándonos en la técnica ACO. Esta elección se basa en la intención de valorar una técnica con la particularidad de que construye las soluciones de manera incremental.

Generamos un algoritmo ACO que incorpora variantes para aprovechar el conocimiento disponible del problema y así acelerar la convergencia. Debido al éxito de los resultados obtenidos, decidimos evaluar su rendimiento en un problema distinto al del transporte público de pasajeros. Así surgió la posibilidad de modelar un sistema de transporte de fluidos mediante cañerías. El modelo considera un conjunto de orígenes, destinos y la posibilidad de instalar nodos concentradores intermedios que almacenen temporalmente el producto, antes de ser enviado a un destino. Tras adaptar nuestro algoritmo a ese problema, demostramos que la técnica es robusta, flexible y eficiente.

En esta etapa de nuestra investigación se cuenta con algoritmos capaces de resolver de manera eficiente distintos problemas. Y además se conoce cuáles son las debilidades y fortalezas de cada una de las técnicas estudiadas. Teniendo en cuenta la dificultad que nos representó tener que adaptar cada método a un problema en particular, se consideró apropiado empezar a trabajar sobre la generalidad de los métodos. Así, surgieron nuestras investigaciones sobre hiperheurísticas.

Se generó una técnica hiperheurística que funciona a un nivel superior, la cual va eligiendo de manera automática una de las metaheurísticas disponibles (GA, SA y ACO) para su aplicación. Esta elección es controlada en función de los distintos estados por los que va atravesando el problema, a la vez que se va compartiendo información relevante encontrada a lo largo de la búsqueda. El método se inicia ejecutando cualquiera de las técnicas disponibles sin prioridad alguna. Luego, la participación de los dos mecanismos incluidos en el algoritmo



(*intercambio de información y selección y control*) guiarán la búsqueda hacia una solución satisfactoria.

Para compartir información implementamos un proceso que informa el descubrimiento de una mejor solución presentándoselas a las demás técnicas. Entonces, se propuso que esta solución pase a ser parte de la población en GA, se incremente el nivel de feromona en la configuración que identifica a esa solución en la técnica ACO y se reemplace a la solución existente en SA. Así, logramos que un descubrimiento encontrado por alguno de los métodos pueda ser aprovechado por todos los demás.

El mecanismo de selección y control administra la conveniencia de la aplicabilidad de uno u otro método. Se basa en el seguimiento de la solución obtenida en cada iteración haciendo que si esta solución fue mejorada se priorice el uso de la misma técnica y en caso de que a medida que avanza el proceso no se logren mejoras se tienda a utilizar alguna otra técnica que permita evitar un posible óptimo local.

La hiperheurística propuesta logra un alto rendimiento, mejora los resultados encontrados por las técnicas GA, SA y ACO ejecutadas de manera individual y además, gracias a la implementación bajo una plataforma basada en hilos, logra disminuir los tiempos de computo, a la vez que aprovecha los recursos de hardware multiprocesador disponibles.

En definitiva, una administración eficiente de la información adquirida, un cuidadoso mecanismo de selección y control de técnicas, y el hecho de tener algoritmos desarrollados apropiadamente nos permitió obtener una técnica hiperheurística que no solo demostró un mejor rendimiento en comparación con las metaheurísticas que intervienen, sino que también nos ofrece cierta generalidad en el método que se traduce en una considerable disminución en la dificultad para aplicar una técnica aproximada para la resolución de un problema de optimización combinatoria.

## 8.2 PRINCIPALES CONTRIBUCIONES CIENTIFICAS

Se ha efectuado amplia difusión del presente trabajo de investigación, como se refleja en las siguientes publicaciones en revistas internacionales indexadas en ISI y congresos del área:

### Trabajos Publicados en Revistas Internacionales:

1. Rodriguez, D.A., Oteiza, P.P., Brignole, N.B., “Simulated-Annealing Optimization for Hydrocarbon Pipeline Networks” *Industrial & Engineering Chemistry Research* (ISSN (printed): 0888-5885, ISSN (electronic): 1520-5045, DOI: 10.1021/ie400022g, 52, 25, 8579-8588, **2013**.
2. Rodriguez D.A., Olivera A.C., Brignole N.B., “Vehicle Routing for Public Transport with Adapted Simulated Annealing, *Latin American Applied Research*. (ISSN: 0327-0793), 44(3):247-252, **2014**.

### Trabajos presentados en Congresos

1. Oteiza P.P., De Meio M., Rodriguez D.A., Viego V. y Brignole N.B. “Metaheuristic Techniques for the Optimal Design of NGL Pipelining”, 25th European Symposium on Computer Aided Process Engineering – ESCAPE 25. Copenhagen, (Resumen aceptado), **2015**.
2. Oteiza P.P., Rodriguez D.A., Brignole N.B. “Diseño De Un Sistema De Control Multivariable Mediante Algoritmos Genéticos” RITEQ 2014: III Reunión Interdisciplinaria de Tecnología y Procesos Químicos, Los Cocos, Córdoba, Argentina, 13-16/4/**2014**.

### Los siguientes trabajos fueron Publicados en Revistas Nacionales con copyright (disponibles on-line) que contienen Trabajos difundidos en Congresos:

3. Rodriguez D., Olivera A.C., Brignole N.B. “Metaheuristics Applied To the Intercity Public Transport Problem”, *Mec. Comp.* (ISSN: 1666-6070), Vol. XXX, 2169-2179,

2011. - ENIEF 2011: 19° Congreso sobre Métodos Numéricos y sus Aplicaciones, AMCA, Rosario, Argentina, 1-4/11/2011.

4. Rodríguez D.A., Olivera A.C., Brignole N.B, "Una Estrategia Paralela Con Simulated Annealing Para El Problema Del Transporte Público Interurbano" *Mec. Comp.* (ISSN: 1666-6070), Vol. XXXI, 3523-3530, 2012. MECOM 2012: X Congreso sobre Mecánica Computacional, AMCA, Publ. en CD (ISSN: 1666-6070), Salta, Argentina, 13-16/11/2012.

5. Rodríguez D.A., Olivera A.C., Brignole N.B, "Metaheurística ACO Para Resolver Un Problema De Transporte" *Mec. Comp.* (ISSN: 1666-6070), Vol. XXXII, 2665-2675, 2013. ENIEF 2013: XX Congreso sobre Métodos Numéricos y sus Aplicaciones, Mendoza, Argentina, 18-22/11/2013.

6. Oteiza P.P., Rodríguez D.A., Brignole N.B, "Sistema De Control Para El Almacenamiento Intermedio De Gasolina Natural" *Mec. Comp.* (ISSN: 1666-6070), Vol. XXXII, 2503-2516, 2013. ENIEF 2013: XX Congreso sobre Métodos Numéricos y sus Aplicaciones, Mendoza, Argentina, 18-22/11/2013.

7. Rodríguez D.A., Olivera A.C., Brignole N.B, "Hiperheurística Diseñada para un Problema de Localización y Transporte Público" *Mec. Comp.* (ISSN: 1666-6070), Vol. XXXIII, 2513-2521, 2014. ENIEF 2014: XXI Congreso sobre Métodos Numéricos y sus Aplicaciones, Bariloche, Argentina, 23-26/9/2014.

### 8.3 TRABAJOS FUTUROS

A futuro se vislumbran interesantes perspectivas para continuar las investigaciones realizadas en esta tesis. Entre los temas que quedan abiertos a partir de este trabajo podemos mencionar:

- **Mejora de la hiperheurística:** En la tesis se logró complementar varias técnicas metaheurísticas en una única hiperheurística, sin embargo creemos que es posible mejorar los resultados de la búsqueda enriqueciendo el conocimiento de una técnica con información descubierta por otra. El intercambio de información mejoraría la exploración en el espacio de búsqueda y ayudaría a escapar de óptimos locales.
- **Estudio y utilización de otras metaheurísticas:** Sería interesante incorporar a la hiperheurística desarrollada algunas técnicas que no fueron tenidas en cuenta para la generación de nuestros algoritmos. Por ejemplo, existen métodos de reconocida trayectoria y potencialidades, tal como Optimización por Cúmulos de Partículas (PSO: Particle Swarm Optimization) (Eberhart y Shi, 2001). También vale la pena considerar otras técnicas recientes que prometen ser exitosas en múltiples problemas; por ejemplo, el Algoritmo de Búsqueda Gravitacional (GSA: Gravitational Search Algorithm) (Rashedi, 2009).
- **Nuevos Paradigmas de Programación Paralela:** En vistas a optimizar el rendimiento de los algoritmos, se podrían desarrollar nuevas formas de paralelismo que permita aprovechar los recursos de hardware y obtener mejores tiempos computacionales.
- **Mejorar el modelo:** Reconociendo la dificultad de contemplar todas las características presentes en un modelo de transporte público, convendría enriquecer el modelo propuesto con aspectos tales como una demanda dinámica de pasajeros, congestión vehicular en ciertos horarios o rutas con características diferentes. Resolver una instancia real como lo es el Área Metropolitana de Salta podría resultar una excelente prueba para nuestro modelo.
- **Un software alternativo para evaluar soluciones:** En un principio las evaluaciones de las soluciones potenciales fueron llevadas a cabo con el simulador SUMO. En trabajos posteriores utilizamos el paquete GAMS. Las características de ambos software presentan limitaciones si lo que se pretende es simular el comportamiento vehicular en un escenario complejo. Sería interesante trabajar con un software de simulación a gran escala,

donde se pueda evaluar instancias grandes de un problema y permita incorporar los algoritmos desarrollados.

- Solución a otros problemas: Respecto al problema estudiado, un modelo de Localización-Asignación intenta buscar soluciones a múltiples situaciones de la vida real. Resultaría muy útil resolver problemas que, al igual que el transporte público de pasajeros, contribuyen al bienestar de la sociedad en su conjunto. En el campo de la optimización existen problemas que son muy estudiados, como recolección de residuos, el transporte escolar, la cobertura de antenas, entre otros, y que pueden ser esquematizados y resueltos con nuestro modelo.

- Un proyecto más ambicioso podría ser desarrollar una aplicación completa que utilice nuestros algoritmos y que sirva de soporte para los encargados de la administración del transporte público. Un sistema de gestión que permita modelar fácilmente una región y genere alternativas que mejoren el servicio.



# REFERENCIAS

## A

Aarts, E., & Korst, J. (2002). Selected topics in simulated annealing. In *Essays and surveys in metaheuristics* (pp. 1-37). Springer US.

Alba, E. (1999). Análisis y diseño de algoritmos genéticos paralelos distribuidos. PhD Tesis, Universidad de Málaga.

Alba, E. (2001). NetStream: a Flexible and Simple OOP Message Passing Service for LAN/WAN Utilization.

Alba, E., & Dorronsoro, B. (2009). *Cellular genetic algorithms* (Vol. 42). Springer.

Alba, E., Luque, G., Garcia-Nieto, J., & Ordonez, G. (2007). MALLBA: a software library to design efficient optimisation algorithms. *International Journal of Innovative Computing and Applications*, 1(1), 74-85.

Alba, E., & Tomassini, M. (2002). Parallelism and evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on*, 6(5), 443-462.

Alvarenga, G. B., Mateus, G. R., & De Tomi, G. (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34(6), 1561-1584.

## B

Baaj, M. H., & Mahmassani, H. S. (1991). An AI-based approach for transit route system planning and design. *Journal of Advanced Transportation*, 25(2), 187-209.

Bäck, T., & Schwefel, H. P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, 1(1), 1-23.

Balseiro, S. R., Loiseau, I., & Ramonet, J. (2011). An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. *Computers & Operations Research*, 38(6), 954-966.

Behrisch, M., Bieker, L., Erdmann, J., & Krajzewicz, D. (2011). Sumo-simulation of urban mobility-an overview. In *SIMUL 2011. The Third International Conference on Advances in System Simulation* (pp. 55-60).

Belenguer, J. M., Benavent, E., Prins, C., Prodhon, C., & Wolfler Calvo, R. (2011). A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6), 931-941.

Bell, J. E., & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1), 41-48.

Berger, R. T., Coullard, C. R., & Daskin, M. S. (2007). Location-routing problems with distance constraints. *Transportation Science*, 41(1), 29-43.

Biazzini, M., Bánhelyi, B., Montesor, A., & Jelasity, M. (2009). Distributed hyper-heuristics for real parameter optimization. In Proceedings of the 11th Annual conference on Genetic and evolutionary computation (pp. 1339-1346). ACM.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3), 268-308.

Blum, C., Puchinger, J., Raidl, G. R., & Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, 11(6), 4135-4151.

Brooke, A., Kendrick, D., Meeraus, A., Raman, R., & America, U. (1998). The General Algebraic Modeling System. *GAMS Development Corporation*.

Burke, E. K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., & Qu, R. (2009). A survey of hyper-heuristics. *Computer Science Technical Report No. NOTTCS-TR-SUB-0906241418-2747, School of Computer Science and Information Technology, University of Nottingham*.

Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., & Schulenburg, S. (2003). Hyper-heuristics: An emerging direction in modern search technology. In *Handbook of metaheuristics* (pp. 457-474). Springer US.

Burke, E., McCollum, B., Meisels, A., Petrovic, S., & Qu, R. (2007). A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, 176(1), 177-192.

## C

Ceder, A., & Wilson, N. H. (1986). Bus network design. *Transportation Research Part B: Methodological*, 20(4), 331-344.

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1), 41-51.

Chakhlevitch, K., & Cowling, P. (2008). Hyperheuristics: recent developments. In *Adaptive and multilevel metaheuristics* (pp. 3-29). Springer Berlin Heidelberg.

Chandra Mohan, B., & Baskaran, R. (2012). A survey: Ant Colony Optimization based recent research and implementation on several engineering domain. *Expert Systems with Applications*, 39(4), 4618-4627.

Cipriani, E., Gori, S., & Petrelli, M. (2012). Transit network design: A procedure and an application to a large urban area. *Transportation Research Part C: Emerging Technologies*, 20(1), 3-14.



Colorni, A., Dorigo, M., Maffioli, F., Maniezzo, V., Righini, G. I. O. V. A. N. N. I., & Trubian, M. (1996). Heuristics from nature for hard combinatorial optimization problems. *International Transactions in Operational Research*, 3(1), 1-21.

Costa, A. L. H., De Medeiros, J. L., & Pessoa, F. L. P. (2000). Optimization of pipe networks including pumps by simulated annealing. *Brazilian Journal of Chemical Engineering*, 17(4-7), 887-896.

Cook, S. A. (1971, May). The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing* (pp. 151-158). ACM.

Coy, S. P., Golden, B. L., Runger, G. C., & Wasil, E. A. (2001). Using experimental design to find effective parameter settings for heuristics. *Journal of Heuristics*, 7(1), 77-97.

Crainic, T. G., & Toulouse, M. (2003). *Parallel strategies for meta-heuristics* (pp. 475-513). Springer US.

Cung, V. D., Martins, S. L., Ribeiro, C. C., & Roucairol, C. (2002). Strategies for the parallel implementation of metaheuristics. In *Essays and surveys in metaheuristics* (pp. 263-308). Springer US.

Cowling, P., Kendall, G., & Han, L. (2002). An investigation of a hyperheuristic genetic algorithm applied to a trainer scheduling problem. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on* (Vol. 2, pp. 1185-1190). IEEE.

Cowling, P., Kendall, G., & Soubeiga, E. (2001). A hyperheuristic approach to scheduling a sales summit. In *Practice and Theory of Automated Timetabling III* (pp. 176-190). Springer Berlin Heidelberg.

## D

Darwin, C. (1859). On the origins of species by means of natural selection. *London: Murray*.

Desaulniers, G., & Hickman, M. (2003). Public transit. *Transportation, handbooks in operations research and management science*, 69-127.

Desrochers, M., Lenstra, J. K., & Savelsbergh, M. W. (1990). A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research*, 46(3), 322-332.

Dooley, A. E., Parker, W. J., & Blair, H. T. (2005). Modelling of transport costs and logistics for on-farm milk segregation in New Zealand dairying. *Computers and electronics in agriculture*, 48(2), 75-91.

Dondo, R., Méndez, C. A., & Cerdá, J. (2003). An optimal approach to the multiple-depot heterogeneous vehicle routing problem with time window and capacity constraints. *Latin American applied research*, 33(2), 129-134.

Dorigo, M. (Ed.). (2006). *Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006, Brussels, Belgium, September 4-7, 2006, Proceedings* (Vol. 4150). Springer.

Dorigo, M., & Stützle, T. (2010). Ant colony optimization: overview and recent advances. In *Handbook of metaheuristics* (pp. 227-263). Springer US.

Drexl, M., Schneider, M., & Professorship, D. S. E. A. (2013). A survey of Location-Routing Problems. Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL).

## E

Eberhart, R. C., & Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on* (Vol. 1, pp. 81-86). IEEE.

El-Mihoub, T. A., Hopgood, A. A., Nolle, L., & Battersby, A. (2006). Hybrid Genetic Algorithms: A Review. *Engineering Letters*, 13(2), 124-137.

## F

Feo, T. A., & Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2), 109-133.

Flynn, M. J., & Rudd, K. W. (1996). Parallel architectures. *ACM Computing Surveys (CSUR)*, 28(1), 67-70.

## G

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management science*, 40(10), 1276-1290.

Gendreau, M., & Potvin, J. Y. (2005). Metaheuristics in combinatorial optimization. *Annals of Operations Research*, 140(1), 189-213.

Giaglis, G. M., Minis, I., Tatarakis, A., & Zeimpekis, V. (2004). Minimizing logistics risk through real-time vehicle routing and mobile technologies: Research to date and future trends. *International Journal of Physical Distribution & Logistics Management*, 34(9), 749-764.

Glover, F. (1990). Tabu search—part II. *ORSA Journal on computing*, 2(1), 4-32.

Glover, F., & Kochenberger, G. A. (Eds.). (2003). *Handbook of metaheuristics*. Springer.

Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, 3(2), 95-99.

Goldberg, D. E., & Lingle, R. (1985, July). Alleles, loci, and the traveling salesman problem. In *Proceedings of an International Conference on Genetic Algorithms and Their Applications* (Vol. 154). Lawrence Erlbaum, Hillsdale, NJ.

Gronau, R. (2000). Optimum diversity in the public transport market. *Journal of transport economics and policy*, 21-41.

Gropp, W., Lusk, E., Doss, N., & Skjellum, A. (1996). A high-performance, portable implementation of the MPI message passing interface standard. *Parallel computing*, 22(6), 789-828.

Grotschel, M., & Lovász, L. (1995). Combinatorial optimization. *Handbook of combinatorics*, 2, 1541-1597.

Guihaire, V., & Hao, J. K. (2008). Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice*, 42(10), 1251-1273.

## H

Hansen, P. B. (1993). Model programs for computational science: A programming methodology for multicomputers. *Concurrency: practice and experience*, 5(5), 407-423.

Hart, W. E., & Belew, R. K. (1991, June). Optimising an Arbitrary Function is Hard for the Genetic Algorithm. In *ICGA* (pp. 190-195).

Hedar, A. R., & Fukushima, M. (2002). Hybrid simulated annealing and direct search method for nonlinear unconstrained global optimization. *Optimization Methods and Software*, 17(5), 891-912.

## I

## J

Jansson, J. O. (1980). A simple bus line model for optimisation of service frequency and bus size. *Journal of Transport Economics and Policy*, 53-80.

## K

Kariv, O., & Hakimi, S. L. (1979). An algorithmic approach to network location problems. II: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3), 539-560.

Kennedy, J. (2010). Particle swarm optimization. In *Encyclopedia of Machine Learning* (pp. 760-766). Springer US.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.

Krajzewicz, D., Hertkorn, G., Rössel, C., & Wagner, P. (2002). Sumo (simulation of urban mobility). In *Proc. of the 4th middle east symposium on simulation and modelling* (pp. 183-187).

## L

---

Laporte, G. (1987). Location routing problems. In: Golden, B.L., Assad, A.A. (Eds.), *Vehicle Routing: Methods and Studies*. North-Holland, Amsterdam, 163–198

Lenstra, J. K., & Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2), 221-227.

## M

Maniezzo, V., Gambardella, L. M., De Luigi, F. (2004) Ant Colony Optimization. *New Optimization Techniques in Engineering*, by Onwubolu, G. C., and B. V. Babu, Springer-Verlag Berlin Heidelberg, pp. 101-117.

Maffioli, F. (2003). The vehicle routing problem: A book review. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2), 149-153.

Maldonado, C. E., & Gómez Cruz, N. A. (2010). *El mundo de las ciencias de la complejidad*. Universidad del Rosario.

Mauttone, A. (2005). *Optimización de recorridos y frecuencias en sistemas de transporte público urbano colectivo* (Doctoral dissertation, Master thesis in computer science, Universidad de la República, Uruguay).

Mauttone, A., & Urquhart, M. E. (2009). A route set construction algorithm for the transit network design problem. *Computers & Operations Research*, 36(8), 2440-2449.

Murray, A. T. (2003). A coverage model for improving public transit system accessibility and expanding access. *Annals of Operations Research*, 123(1-4), 143-156.

## N

Nagy, G., & Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2), 649-672.

## O

Oaks, S., & Wong, H. (1999). Java threads. " O'Reilly Media, Inc."

Olivera, A. C., Frutos, M., Carballido, J. A., Ponzoni, I., & Brignole, N. B. (2009). Bus Network Scheduling Problem: Grasp+ eas With Pisa\* simulation. In *Bio-Inspired Systems: Computational and Ambient Intelligence* (pp. 1272-1279). Springer Berlin Heidelberg.

Ortuzar, J de D., & Willumsen, L. G. (1994). *Modelling transport*. Wiley.

Oteiza P.P., De Meio M., Rodriguez D.A., Viego V. & Brignole N.B. (2015). "Metaheuristic Techniques for the Optimal Design of NGL Pipelining", 25th European Symposium on Computer Aided Process Engineering – ESCAPE 25. Copenhagen, (Resumen enviado).

Özcan, E., Bilgin, B., & Korkmaz, E. E. (2008). A comprehensive analysis of hyper-heuristics. *Intelligent Data Analysis*, 12(1), 3-23.

## P

Park, J., & Kim, B. I. (2010). The school bus routing problem: A review. *European Journal of operational research*, 202(2), 311-319.

## Q

## R

Raidl, G. R. (2006). A unified view on hybrid metaheuristics. In *Hybrid Metaheuristics* (pp. 1-12). Springer Berlin Heidelberg.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*, 179(13), 2232-2248.

Rattadilok, P., Gaw, A., & Kwan, R. S. (2005). Distributed choice function hyper-heuristics for timetabling and scheduling. In *Practice and Theory of Automated Timetabling V* (pp. 51-67). Springer Berlin Heidelberg.

Reeves, C. R. (1993). *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc.

Rodríguez D. A., Olivera, & Brignole N. B. (2011). Meta-heuristics applied to the intercity transport problem. ENIEF 2011 XIX Congreso sobre Métodos Numéricos y sus Aplicaciones. *Mecánica Computacional* ISSN 1666-6070 Vol XXX, pp. 2169-2179.

Rodríguez D. A., Olivera, & Brignole N. B. (2012). Una Estrategia Paralela con Simulated Annealing para el Problema del Transporte Público Interurbano. MECOM 2012 X Congreso Argentino de Mecánica Computacional. Vol XXXI, pp. 3523-3530.

Rodríguez, D. A., Oteiza, P. P., & Brignole, N. B. (2013a). Simulated Annealing Optimization for Hydrocarbon Pipeline Networks. *Industrial & Engineering Chemistry Research*, 52(25), 8579-8588.

Rodríguez D. A., Olivera, & Brignole N. B. (2013b). Metaheurística ACO para Resolver un Problema de Transporte. ENIEF 2013. XX Congreso sobre Métodos Numéricos y sus Aplicaciones. *Mecánica Computacional*. Vol XXXII, pp. 2665-2675.

Rodriguez, D. A., Olivera, A. C. & Brignole N. B. (2014a). Vehicle Routing for Public Transport with Adapted Simulated Annealing, *Latin American Applied Research (LAAR)*. (ISSN: 0327-0793), 44(3):247-252.

Rodriguez D.A., Olivera A.C. & Brignole N.B. (2014b) Hiperheurística Diseñada para un Problema de Localización y Transporte Público. ENIEF 2014: XXI Congreso sobre Métodos Numéricos y sus Aplicaciones. *Mecánica Computacional*. Vol XXXIII, pp. 2513-2521.

Rochat, Y., & Taillard, É. D. (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics*, 1(1), 147-167.

Ross, P. (2005). Hyper-heuristics. In *Search methodologies* (pp. 529-556). Springer US.

Rui, X., Gongyuan, L., & Dong, H. (2010, June). Simulated annealing algorithm for solving a bi-level optimization model on high-speed railway station location. In *Information and Computing (ICIC), 2010 Third International Conference on* (Vol. 2, pp. 159-162). IEEE.

## S

Sa'adah, S., Ross, P., & Paechter, B. (2004). Improving vehicle routing using a customer waiting time colony. In *Evolutionary Computation in Combinatorial Optimization* (pp. 188-198). Springer Berlin Heidelberg.

Sheskin, D. J. (2003). *Handbook of parametric and nonparametric statistical procedures*. Press.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2), 254-265.

Sörensen, K. (2013). Metaheuristics - the metaphor exposed. *International Transactions in Operational Research*.

Soubeiga, E. (2003). *Development and application of hyperheuristics to personnel scheduling* (Doctoral dissertation, University of Nottingham).

Steiglitz, K., & Papadimitriou, C. H. (1982). *Combinatorial optimization: Algorithms and complexity*. Printice-Hall, New Jersey.

## T

Talbi, E. G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.

Toth, P., & Vigo, D. (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123(1), 487-512.

## U

## V

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2014). A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234(3), 658-673.

## W

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and computing*, 4(2), 65-85.

Wilcoxon, R. R. (1987). *New statistical procedures for the social sciences: Modern solutions to basic problems*. Psychology Press.

Winston, W. L. (2004) *Operations Research: Applications and Algorithms*; 4th Ed, Brooks/Cole Duxbury Press, Chapter 7.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *Evolutionary Computation*, IEEE Transactions on, 1(1), 67-82.

Wolsey, L. A. (2000). Integer programming. *IIE Transactions*, 32(273-285), 2-58.

## X

Xinchao, Z. (2011). Simulated annealing algorithm with adaptive neighborhood. *Applied Soft Computing*, 11(2), 1827-1836.

## Y

Yaghini, M., Momeni, M., & Sarmadi, M. (2012). A Simplex-based simulated annealing algorithm for node-arc capacitated multicommodity network design. *Applied Soft Computing*, 12(9), 2997-3003.

Yan, S., & Chen, H. L. (2002). A scheduling model and a solution algorithm for inter-city bus carriers. *Transportation Research Part A: Policy and Practice*, 36(9), 805-825.

## Z

Zhao, N., Wu, Z., Zhao, Y., & Quan, T. (2010). Ant colony optimization algorithm with mutation mechanism and its applications. *Expert Systems with Applications*, 37(7), 4805-4810.