

Universidad de las Ciencias Informáticas

Facultad 6



Análisis y Diseño de Algoritmos Genéticos para Problemas de Optimización de Rutas en GeoQ

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autora: Ailema Vázquez García

Tutor: MSc. Yesnier Bravo García

Junio de 2012

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos que la Universidad de las Ciencias Informáticas tiene los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los ____ días del mes de _____ del año _____.

Ailema Vázquez García

MSc. Yesnier Bravo García

Firma del Autor

Firma del Tutor

Agradecimientos

En primer lugar quisiera agradecer a mis padres Lázaro y Teresita, por ser los mejores padres del mundo y motores impulsores de mi carrera. El amor que me demuestran día tras día, fue y es primordial para alcanzar mis metas.

Quisiera agradecer a Yesnier, por brindarme su paciencia y ayuda incondicional en estos cinco años. Fuiste indispensable para este trabajo, como lo eres en tantos aspectos de mi vida.

No puedo dejar de mencionar a mi familia: mi hermana Eilen, mi abuela Amelia, tíos, primos, gracias a todos por estar a mi lado en los momentos más tristes y no olvidarse de mí en los alegres.

A todos mis amigos con los que he compartido muy buenos momentos y que durante estos cinco años han sido mi otra familia.

En fin, a todas las personas que aportaron su granito de arena para que este sueño se hiciera realidad.

Dedicatoria

A mis padres, por ser seres extraordinarios.

A mi novio, por estar a mi lado en todo momento.

A mi familia, por el amor y la comprensión a lo largo de mi vida.

A todos gracias, los amo.

Resumen

Actualmente existe un dominio de problemas denominados Problemas de Optimización de Rutas, que tienen como objetivo determinar las rutas que mejor satisfacen un número de restricciones. Las estrategias utilizadas para solucionar Problemas de Optimización de Rutas van desde las técnicas exactas hasta las probabilísticas. Los algoritmos exactos requieren de tiempos de ejecución que se elevan en la medida que aumenta el tamaño de las instancias de problemas, por lo que resultan de escasa utilidad práctica. Entre las técnicas probabilísticas, las metaheurísticas, y en específico los Algoritmos Genéticos, son de las más usadas con resultados exitosos. Básicamente, porque resultan menos afectados por los óptimos locales que otras técnicas y son intrínsecamente paralelos ya que operan de forma simultánea con varias soluciones. Por otro lado, el desarrollo tecnológico de los Sistemas de Información Geográfica ha dado potencialidad adicional a las herramientas que resuelven Problemas de Optimización de Rutas. Estos sistemas pueden ser utilizados para realizar el pre-procesamiento de los datos necesarios para el funcionamiento del Algoritmo Genético, así como para la posterior visualización de las rutas. Debido a la dificultad de los Problemas de Optimización de Rutas durante la investigación se identificaron tres componentes de los Algoritmos Genéticos que pudieran modificarse en busca de un mejor desempeño, la inicialización de la población, el cruzamiento y la mutación. Las estrategias propuestas aprovechan las facilidades que ofrecen los Sistemas de Información Geográfica como GeoQ para analizar la visibilidad y proximidad entre puntos y consultar distancias e intersecciones. Los resultados experimentales corroboran la efectividad de los operadores propuestos.

Palabras clave: Algoritmos Genéticos, Problemas de Optimización de Rutas, Sistemas de Información Geográfica

Índice de Contenido

Introducción	1
CAPÍTULO 1 Problemas de Optimización de Rutas en GeoQ	4
1.1 <i>Introducción</i>	4
1.2 <i>Problemas de Optimización de Rutas</i>	4
1.2.1 Problema del Camino Mínimo	6
1.2.2 Problema del Recurso más Cercano	7
1.2.3 Problema del Agente Viajero	8
1.2.4 Problema del Ruteo de Vehículos	10
1.3 <i>Sistemas de Información Geográfica</i>	11
1.3.1 SIG que solucionan POR	13
1.3.2 GeoQ	13
1.4 <i>Conclusiones</i>	17
CAPÍTULO 2 Algoritmos Genéticos para Problemas de Optimización de Rutas	18
2.1 <i>Introducción</i>	18
2.2 <i>Funcionamiento de los AG</i>	18
2.2.1 Población inicial	19
2.2.2 Operador de cruzamiento	20
2.2.3 Operador de mutación	21
2.3 <i>Estrategias actuales enfocadas a POR</i>	21
2.4 <i>Complejidad computacional</i>	26
2.5 <i>Conclusiones</i>	26
CAPÍTULO 3 Diseño de Nuevos Componentes de Algoritmos Genéticos	27
3.1 <i>Introducción</i>	27
3.2 <i>Propuesta de inicialización de la población</i>	27
3.2.1 Inicialización Proporcional a la Proximidad (IPP)	28
3.2.2 Inicialización en Doble Sentido (IDS)	29
3.3 <i>Propuesta de operadores de cruzamiento</i>	31
3.3.1 Cruzamiento Basado en Costo (CBC)	31
3.3.2 Variación Cruzamiento Uniforme (VCU)	32
3.4 <i>Propuesta de operadores de mutación</i>	33
3.4.1 Variación Mutación Inversión (VMI)	33
3.4.2 Mutación por Reagrupamiento de Bloques (MRB)	34
3.5 <i>Conclusiones</i>	35
CAPÍTULO 4 Análisis de los Resultados	36

4.1	<i>Introducción</i>	36
4.2	<i>Variantes de inicialización aplicadas</i>	37
4.2.1	<i>Impacto en la Diversidad de la Población</i>	38
4.3	<i>Estudio de los operadores CBC y VCU</i>	39
4.4	<i>Estudio de los operadores VMI y MRB</i>	42
4.5	<i>Conclusiones</i>	44
	Conclusiones Generales	46
	Recomendaciones	47
	Bibliografía	48
	Glosario de Términos	51

Índice de Tablas

<i>Tabla 1 Parámetros del algoritmo</i>	<i>36</i>
<i>Tabla 2 Valor de aptitud medio para las estrategias de inicialización.....</i>	<i>37</i>
<i>Tabla 3 Valor de desviación estándar luego la inicialización</i>	<i>38</i>
<i>Tabla 4 Valor de aptitud medio para las estrategias de cruce</i>	<i>40</i>
<i>Tabla 5 Valor de aptitud medio para las estrategias de mutación</i>	<i>42</i>
<i>Tabla 6 Valor de desviación estándar media de las estrategias de mutación</i>	<i>43</i>
<i>Tabla 7 Valor de aptitud medio para las propuestas finales de AG</i>	<i>44</i>

Índice de Figuras

<i>Figura 1.2-1 Posible solución al PCM</i>	6
<i>Figura 1.2-2 Posible solución al PRC</i>	7
<i>Figura 1.2-3 Posible solución al TSP</i>	8
<i>Figura 1.2-4 Taxonomía de heurísticas específicas para el TSP</i>	9
<i>Figura 1.2-5 Posible solución al VRP</i>	10
<i>Figura 1.3-1 Componentes de un SIG</i>	12
<i>Figura 1.3-2 Funcionalidad de análisis</i>	14
<i>Figura 2.2-1 Esquema del funcionamiento de un AG</i>	19
<i>Figura 2.3-1 Soluciones para el PCM</i>	22
<i>Figura 2.3-2 Soluciones para el PRC</i>	22
<i>Figura 2.3-3 Soluciones para el TSP</i>	23
<i>Figura 2.3-4 Soluciones para el VRP</i>	23
<i>Figura 2.3-5 Valor de aptitud para las soluciones del PCM y PRC</i>	24
<i>Figura 2.3-6 Valor de aptitud para las soluciones del TSP</i>	24
<i>Figura 2.3-7 Valor de aptitud para las soluciones del VRP</i>	25
<i>Figura 3.2-1 Selección proporcional a la proximidad</i>	28
<i>Figura 3.2-2 Espacio de soluciones</i>	30
<i>Figura 3.3-1 Realizando el CBC</i>	31
<i>Figura 3.3-2 Copiando el 50% de los genes</i>	32
<i>Figura 3.3-3 Completando con los genes de los padres contrarios</i>	32
<i>Figura 3.4-1 Buscar dos posiciones aleatorias cuyo intercambio permita una disminución del costo</i>	33
<i>Figura 3.4-2 Aplicando la inversión</i>	33
<i>Figura 3.4-3 Buscar cuatro posiciones aleatorias que permitan disminución del costo</i>	34
<i>Figura 3.4-4 Reagrupando los bloques</i>	34
<i>Figura 4.2-1 Por ciento de éxito para las estrategias de inicialización</i>	39
<i>Figura 4.3-1 Valor de aptitud medio para las estrategias de cruce para el TSP</i>	41
<i>Figura 4.3-2 Por ciento de éxito para las estrategias de cruce</i>	41
<i>Figura 4.4-1 Valor de aptitud medio para las estrategias de cruce para el TSP</i>	43

Introducción

Actualmente existe un dominio de problemas denominados Problemas de Optimización de Rutas (POR), que tienen como objetivo determinar las rutas que mejor satisfagan un número de restricciones. Dado un conjunto de locaciones geográficas, una ruta se define como un camino que parte de un punto (origen) y llega a otro punto (destino), pasando por puntos consecutivamente vecinos (puntos conectados directamente). La conexión entre dos puntos tiene asociado un costo y la sumatoria de los costos de cada una de las conexiones de la ruta determinan su costo general. Partiendo de que entre dos puntos pueden existir varias rutas, el objetivo es hallar la ruta de menor costo (o mayor beneficio). Determinar el camino de costo mínimo entre dos puntos o identificar el recurso más cercano a una emergencia, son algunos ejemplos de POR. El número de localizaciones geográficas y la cantidad de restricciones involucradas, influyen directamente en la dificultad de estos problemas. Dicho esto, solucionar las instancias más grandes tiene mayor trascendencia económica y social.

Varias estrategias han sido utilizadas para solucionar POR a lo largo de los años. Ellas van desde las técnicas exactas a las probabilísticas. Los algoritmos exactos garantizan encontrar la mejor solución, pero requieren de tiempos de ejecución que se elevan exponencialmente con el tamaño de las instancias de problemas. De este modo, resultan de escasa utilidad práctica, ya que solo son efectivos para solucionar las instancias más pequeñas. Por su lado, las metaheurísticas son técnicas que exploran el espacio de soluciones aplicando métodos probabilísticos y se concentran en la explotación de las zonas más prometedoras (1). Las investigaciones referentes a la aplicación de estas técnicas han demostrado que permiten encontrar buenas soluciones en tiempos razonables. Muchas metaheurísticas han sido utilizadas para resolver POR; entre ellas, los Algoritmos Genéticos (AG) (2) son una de las más usadas (3), (4), (5) y (6). Estos son métodos que crean una población inicial que evoluciona aplicando operadores de selección, cruzamiento y mutación. El paralelismo con que se opera la población y el hecho de que resultan menos afectados por los óptimos locales que otras técnicas (7), los convierten en una opción viable para el tratamiento de POR. Sin embargo, hasta donde se conoce, no existe un estudio estructurado, unificado y comprensivo de las distintas variantes de AG aplicadas a POR. Como se verá en este trabajo, estos problemas presentan características comunes que permiten el análisis y diseño de representaciones y operadores genéticos generales.

Por otro lado, el surgimiento y desarrollo de los Sistemas de Información Geográfica (SIG) ha dado una potencialidad adicional a las herramientas que resuelven POR (8). Estos sistemas permiten la extracción de los datos necesarios para el funcionamiento de los algoritmos, además de la visualización de las rutas intuitivamente en un mapa. En el Proyecto SIG-Desktop del Centro de Desarrollo GEySED, perteneciente a la Facultad 6 de la Universidad de las Ciencias Informáticas, se desarrolla el SIG GeoQ. En este sistema, las funciones de análisis y visualización no son explotadas al máximo en la solución de POR. GeoQ provee funcionalidades de intersección, cálculo de distancias e identificación y creación dinámica de objetos espaciales, que permiten interactuar con los datos de los POR. Estos sistemas presentan los datos espaciales enriquecidos, lo que permite modelar más restricciones que extrayendo los datos de forma tradicional y así, resolver problemas más reales. Este sistema tiene tres componentes que permiten solucionar algunos POR. En primer lugar, la componente pgRouting (9), que tiene la restricción de que los datos tienen que almacenarse en una base de datos PostgreSQL. También tiene un complemento que resuelve únicamente el Problema del Camino Mínimo utilizando el algoritmo de Dijkstra sobre un grafo reducido. En este caso, la desventaja radica en que cada vez que se quiera reducir el grafo se realiza nuevamente todo el procesamiento, afectándose el tiempo de respuesta de la aplicación. Por último, provee el complemento Road Graph, que resuelve el Problema del Camino Mínimo utilizando Dijkstra sobre un grafo que guarda las intersecciones entre líneas. Este complemento requiere que se esté utilizando como base de GeoQ la plataforma Quantum Gis versión 1.6 o mayor.

En este contexto queda enmarcado el presente trabajo que tiene como **problema científico** ¿cómo aumentar la eficacia de los AG básicos para resolver POR en GeoQ? Se centra el **objeto de estudio** en el proceso de resolución de POR en SIG y el **campo de acción** en el diseño de AG para solucionar POR en GeoQ. Para darle solución al problema, se plantea como **objetivo** diseñar nuevos operadores de AG usando funciones de análisis de GeoQ.

Esta investigación **defiende la idea** de que si se diseñan nuevos operadores de AG usando funciones de análisis de GeoQ entonces se podrá aumentar la eficacia de los AG básicos para POR en GeoQ.

Para alcanzar el objetivo propuesto se planificaron las siguientes tareas de investigación:

1. Determinar los POR que servirán para armar un marco de representación uniforme para los AG, así como probar el desempeño de los algoritmos con distintos operadores.
2. Caracterizar los AG aplicados a POR.

3. Identificar características de los SIG como GeoQ para facilitar la implementación de AG eficaces para resolver POR, basados en información geo-referenciada del problema.
4. Analizar de forma estructurada, unificada y exhaustiva las estrategias de representación y operadores genéticos actuales orientados a POR en base a resultados cuantitativos y a un análisis de significación estadística.
5. Elaborar una propuesta de AG para la solución eficiente de POR en la aplicación GeoQ.
6. Validar la propuesta mediante el diseño e implementación de un componente de análisis de rutas para GeoQ.

Durante el desarrollo de la investigación se utilizan varios métodos científicos. Desde el punto de vista teórico, el análisis histórico lógico permitirá estudiar las especificaciones de los AG y los POR. Además, el análisis y síntesis facilitará obtener y asimilar los contenidos relacionados con las estrategias de representación, selección, cruzamiento y mutación de AG para POR. Finalmente, la experimentación se utilizará para validar empíricamente la idea a defender de la investigación. El análisis de los resultados se realiza por medio de métodos estadísticos.

Estructura del contenido

- El capítulo 1, “Problemas de Optimización de Rutas en GeoQ”, define formalmente los problemas de optimización seleccionados para realizar la investigación. Además describe los beneficios de los SIG como complemento para la resolución y representación de estos problemas.
- El capítulo 2, “Algoritmos Genéticos para Problemas de Optimización de Rutas”, introduce los AG y sus componentes fundamentales, así como las distintas variantes de codificación, mutación y cruzamiento de las soluciones que existen en la literatura. Además realiza un estudio estructurado y unificado de los AG propuestos para POR.
- El capítulo 3, “Diseño de Nuevos Componentes de Algoritmos Genéticos”, propone nuevas estrategias de inicialización, realiza mejoras a operadores ya existentes e introduce nuevos operadores de cruzamiento y mutación.
- El capítulo 4, “Análisis de los resultados”, muestra y discute los resultados de los experimentos que permiten validar la efectividad de las mejoras y nuevos operadores propuestos. Además, valida el beneficio de aprovechar las capacidades de cómputo de los SIG para mejorar el desempeño de AG para POR.

CAPÍTULO 1 Problemas de Optimización de Rutas en GeoQ

1.1 Introducción

En este capítulo se definen los POR estudiados durante esta investigación. Primeramente se definen de forma unificada sus características generales y luego se plantean las especificidades para cada problema. Luego, se introducen aspectos fundamentales de los SIG que servirán para entender las propuestas de solución.

1.2 Problemas de Optimización de Rutas

Los POR estudiados en esta investigación presentan similitudes que permiten caracterizarlos de forma unificada. En primer lugar, todos son problemas de optimización cuya función objetivo es encontrar K rutas en una red de N puntos tal que el costo del recorrido total sea mínimo, como se plantea a continuación:

$$\min \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^N C_{n_i, n_j} x_{n_i, n_j}^k$$

Sujeto a:

$$\sum_{k=1}^K \sum_{i=1}^N x_{n_i, n_j}^k = 1, \quad j = 1, \dots, N \quad (I)$$

$$\sum_{i=1}^N x_{n_i, n_j}^k - \sum_{i=1}^N x_{n_j, n_i}^k = 0 \quad (II)$$

En esta definición general, denotamos $C_{i,j}$ como el costo de ir desde el punto i al punto j , y $x_{n_i,n_j}^k \in \{0,1\}$ como una variable de decisión que representa la presencia o no de la conexión (i,j) en la ruta k . Sujeto a las restricciones de que (I) cada punto debe ser visitado una sola vez y que (II) cada recorrido llega y sale de cada punto visitado. Además, cada problema puede incorporar otras restricciones.

Una ruta consiste en una secuencia de m ($1 < m \leq N$) puntos consecutivamente vecinos $\vec{r} = n_{l_1}, n_{l_2}, \dots, n_{l_m}$, donde l_1, l_2, \dots, l_m denota una selección del conjunto de índices $L = \{1,2, \dots, N\}$. A continuación se ilustra una ruta, indicando todas las variables de decisión que toman valor 1:

$$n_{l_1} \xrightarrow{x_{l_1,l_2}} n_{l_2}, \dots, \xrightarrow{x_{l_{m-1},l_m}} n_{l_m}$$

Otro de los aspectos que se pueden definir de forma unificada es la representación de la red de puntos. La misma se puede modelar mediante un grafo $G = (V, E, C)$, donde:

- $V \rightarrow$ Representa el conjunto de todos los puntos i que conforman la red; o sea, todos los puntos disponibles para formar parte de una ruta.
- $E \rightarrow$ Representa el conjunto de las posibles conexiones (i,j) entre los puntos.
- $C \rightarrow$ Representa la matriz de costo.

En esta definición la información de la red se consulta de una matriz de costo. Esta es una de las estructuras de datos más usadas por su simplicidad para modelar grafos. Es empleada en este trabajo porque permite un acceso en orden constante $O(1)$ a cada elemento. La misma consiste en una matriz de orden $O(N^2)$. Cada intersección (i,j) guarda el costo de ir del punto i al punto j , que se toma como Inf en caso de que no exista una conexión (i,j) en E . A continuación se define formalmente esta función utilizando distancia euclidiana:

$$c_{i,j} = \begin{cases} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \rightarrow Si (i,j) \in E \\ Inf \rightarrow Si (i,j) \notin E \end{cases}$$

En problemas reales, la distancia euclidiana no es correcta porque no considera las conexiones curvas. Precisamente, este trabajo propone utilizar las funcionalidades de edición de los SIG para modelar estas conexiones como segmentos de líneas y así poder usar la distancia euclidiana.

A continuación, se definen cuatro POR seleccionados para realizar el estudio. Los problemas son planteados en orden ascendente de dificultad.

1.2.1 Problema del Camino Mínimo

El Problema del Camino Mínimo (PCM) constituye uno de los POR más básico. Consiste en hallar el camino de costo mínimo entre dos puntos cualesquiera. El costo del camino puede estar en función de diferentes variables, o una combinación de ellas, como son:

1. Distancia entre dos puntos de la ruta, usualmente la más utilizada.
2. Velocidad máxima de ir de un punto a otro de la ruta.

La Figura 1.2-1 muestra una posible solución al problema.

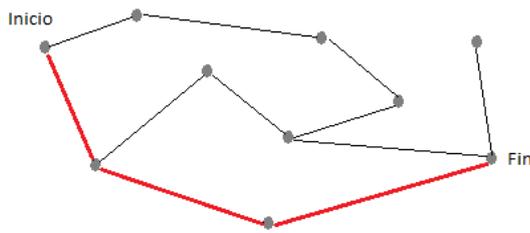


Figura 1.2-1 Posible solución al PCM

En este problema se mantiene la función objetivo general con la especificidad de que se busca una única ruta ($K = 1$). A las restricciones generales definidas se agregan:

$$\sum_{j=1}^N x_{n_a, n_j} - \sum_{i=1}^N x_{n_i, n_a} = 1 \quad (IIIa)$$

$$\sum_{i=1}^N x_{n_i, n_b} - \sum_{j=1}^N x_{n_b, n_j} = 1 \quad (IVa)$$

Donde n_a denota el punto origen, desde donde debe partir cada ruta (restricción (IIIa)), y n_b el punto destino a donde debe arribar cada ruta (restricción (IVa)).

La aplicación de este problema aparece usualmente en las industrias de telecomunicaciones y transporte, buscando reducir tiempo y dinero invertido. También, se utilizan en el desarrollo de circuitos

integrados, sistemas mecánicos e hidráulicos, la administración de proyectos y horarios de trabajo y la planificación del tráfico urbano.

Entre los algoritmos revisados para resolver el PCM de forma determinista, se encuentra Dijkstra (10), primer método conocido que lo soluciona. Luego, aparecen versiones del mismo que incorporan nuevas estructuras de datos (11) y logran implementaciones más eficientes. Los algoritmos mencionados hasta el momento, tienen la limitación de que comienzan a ser ineficientes cuando el número de puntos aumenta. Sin embargo, los avances en el campo de la optimización, han motivado la aparición de nuevos algoritmos utilizando técnicas metaheurísticas como en (12), (3) y (13). Estas técnicas aumentan la exploración del espacio de soluciones, lo que permite resolver instancias del PCM más grandes.

1.2.2 Problema del Recurso más Cercano

El Problema del Recurso más Cercano (PRC) considera a un subconjunto de los puntos como recursos o facilidades. Un recurso representa un punto de interés en determinado negocio, como hospitales, estaciones de bomberos y policías, etc. Este problema consiste en hallar el recurso más cercano a un punto dado. La Figura 1.2-2 muestra una posible solución al problema.

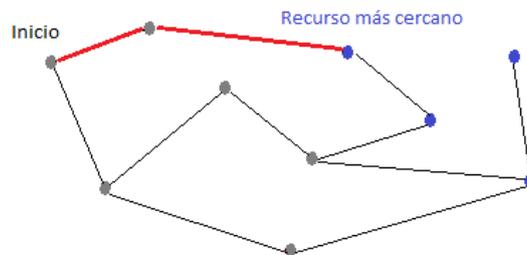


Figura 1.2-2 Posible solución al PRC

Sin pérdida de generalidad, se considera que el conjunto de recursos R es un subconjunto propio del conjunto de puntos ($R \subset V$). En este caso igualmente se busca una única ruta ($K = 1$), por lo que la función objetivo se mantiene igual que para el problema anterior. Se hereda la restricción (IIIa) del PCM y se define la siguiente condición adicional:

$$\sum_{i=1}^N x_{n_i, n_f} - \sum_{j=1}^N x_{n_f, n_j} = 1 \quad (IVb)$$

La restricción (IVb) plantea que toda ruta termine en alguno de los recursos, denotados como n_f .

Una de las aplicaciones más importantes del PRC es en los sistemas de gestión de incidencias. En estos sistemas, los recursos o facilidades lo constituyen las ambulancias, carros de bomberos, etc., en dependencia del tipo de incidente. Una vez que se produce una emergencia, se selecciona del conjunto de recursos el más cercano al punto donde se originó el incidente.

Para el tratamiento del PRC resalta el algoritmo Dijkstra (14), con la limitante de que aumenta considerablemente el tiempo de ejecución cuando aumenta la dificultad del problema. Algunas metaheurísticas como las empleadas en (15), (16), (4) y (17), muestran buenos resultados en la localización de facilidades. Específicamente los AG desarrollados en estos trabajos, son aplicables con éxito a problemas de gran escala. Estas investigaciones destacan que los AG tardan más en llegar a la solución que otras técnicas probabilísticas, pero la calidad de la misma es en algunos casos mejor que la registrada en la bibliografía.

1.2.3 Problema del Agente Viajero

El Problema del Agente Viajero o Problema del Viajante de Comercio (TSP, por sus siglas en inglés) es uno de los problemas de optimización combinatoria más conocido. El mismo consiste en:

“Encontrar entre un conjunto de ciudades, una ruta que partiendo de una ciudad y regresando a ella, recorra todas las ciudades por el camino más corto.” (18) (ver Figura 1.2-3)

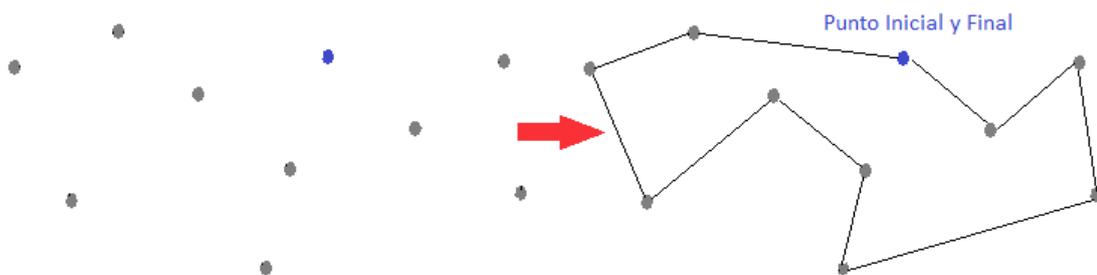


Figura 1.2-3 Posible solución al TSP

Es decir, que consiste en encontrar una ruta que permita recorrer todos los puntos, sin que estos sean repetidos y que termine en el punto inicial.

De forma similar que en los problemas anteriores, se busca una única ruta ($K = 1$). Así, la función objetivo se mantiene constante y se agrega al conjunto de restricciones básicas:

$$|\vec{r}| = N \quad (IIIc)$$

Esta restricción garantiza que sean recorridos todos los puntos.

La aplicación de este problema se ve en las empresas que tienen redes de distribución y requieren recorrer todos los puntos que la componen minimizando costos. Además, los métodos de búsqueda exitosos basados en el TSP tienen impacto en la biología molecular.

Para el TSP, en (19), aparece como posible solución la técnica de propósito general Ramificación y Corte (Branch and Cut). Esta técnica comprueba exhaustivamente las soluciones posibles, descartando en forma inteligente ciertas “direcciones de búsqueda”. Por su lado, las heurísticas de mejoramiento iterativo, como 2-Opt (20) y su generalización k-Opt, mejoran una solución establecida inicialmente. Consisten en borrar k arcos existentes, e insertar k arcos nuevos. Tienen la desventaja de que la solución inicial influye la posibilidad de caer en óptimos locales, de ahí que suelen ser utilizadas como complemento de otras técnicas. Otras heurísticas de propósito especial como el vecino más cercano y la inserción más cercana (21), garantizan encontrar buenos resultados. Estas técnicas se limitan a ser aplicadas en instancias que cumplan con “la desigualdad del triángulo”. La Figura 1.2-4 muestra una taxonomía de las principales heurísticas que se han empleado para solucionar este problema (22).

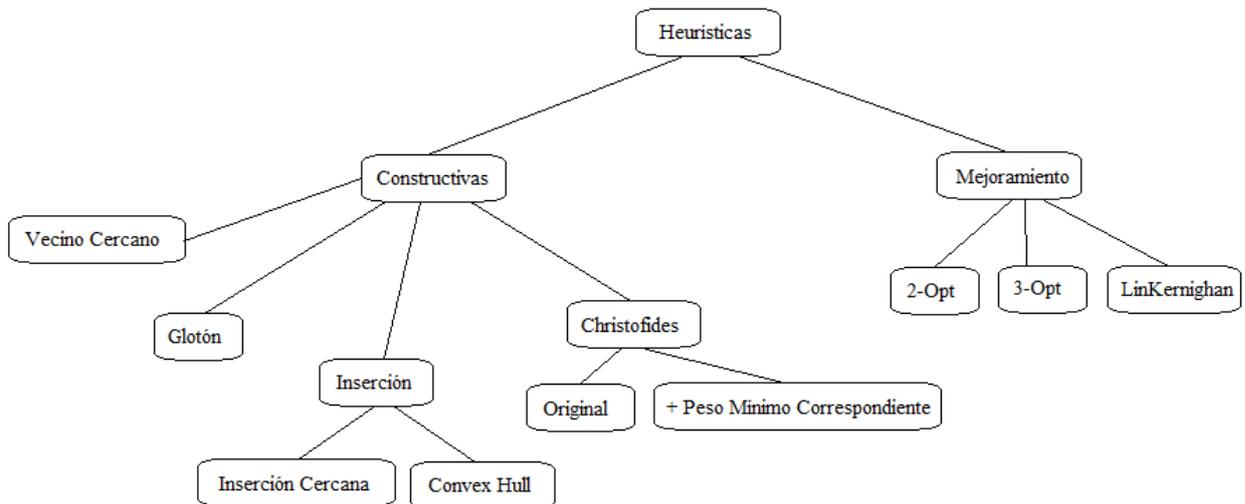


Figura 1.2-4 Taxonomía de heurísticas específicas para el TSP

De modo general, estas heurísticas están divididas en dos grandes grupos, aquellas para la creación de rutas y otras que permiten partiendo de una ruta inicial mejorarla iterativamente. Por su lado, los AG

expuestos en (18), (20), (23) y (5), destacan que los buenos resultados de estas técnicas dependen en gran medida de la selección de una correcta representación, cruzamiento y mutación.

1.2.4 Problema del Ruteo de Vehículos

El Problema del Ruteo de Vehículos (VRP, por sus siglas en inglés) es una versión extendida del TSP. En este caso, el número de rutas está determinado por la cantidad de vehículos disponibles y las ciudades o puntos que distribuyen los vehículos. Asociado a los vehículos se encuentra el concepto de capacidad, que puede variar o no para cada vehículo. El problema consiste en hallar una distribución de rutas para cada uno de los vehículos, de modo que el conjunto de todas las rutas recorra cada uno de los puntos. La Figura 1.2-5 muestra una red de puntos y una posible solución para recorrer la misma con tres vehículos.

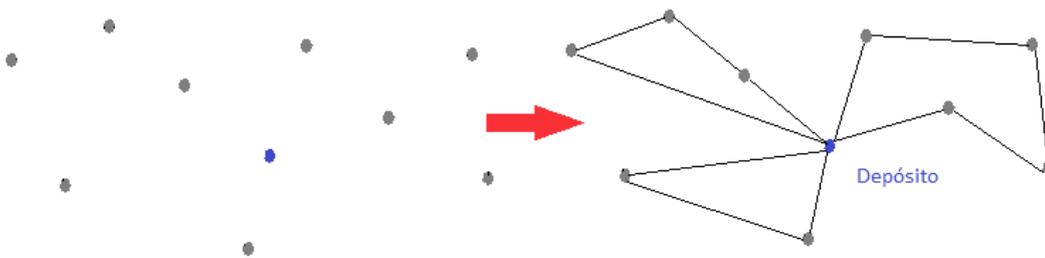


Figura 1.2-5 Posible solución al VRP

En (24), se citan las variantes más importantes de este problema, que incorporan restricciones de tiempo de los recorridos, capacidad de los vehículos, recolección y entrega, e incluso combinaciones de ellas. En esta investigación, la variante utilizada es la que especifica que las capacidades de los vehículos de la flota sean homogéneas, conocida como CVRP por sus siglas en inglés.

Para el caso de CVRP, se define la capacidad asociada a cada vehículo C y la demanda de cada punto o ciudad D . El depósito es considerado como el punto n_0 . En este problema intervienen un conjunto de vehículos, que determinan la cantidad de rutas que se van a optimizar en la función objetivo general. Además de las restricciones básicas se plantean:

$$\sum_{j=1}^N x_{n_i, n_j}^k - \sum_{j=1}^N x_{n_j, n_i}^k = y_i^k \quad (II d)$$

$$\sum_{k=1}^K \sum_{i=1}^N D_{n_i} y_{n_i}^k \leq C_k \quad (IV d)$$

$$\sum_{i=1}^N x_{n_i, n_0} = \sum_{j=1}^N x_{n_0, n_j} = K \quad (Vd)$$

$$\sum_{k=1}^K y_{n_i}^k = 1 \quad (VIId)$$

Donde la $y_{n_i}^k \in \{0,1\}$ es una variable de decisión que indica si el punto i será atendido por el vehículo k . Primeramente, se redefine la restricción (II) como la (IIId), para indicar que si un vehículo llega o sale del punto i es porque lo va a atender. Se hereda la restricción (IIIc) del problema anterior. Por su lado, la restricción (IVd) restringe la capacidad máxima de cada vehículo y la (Vd) establece que cada sub-ruta empiece y termine en el punto depósito. Por último, se garantiza que un cliente solo pueda ser atendido por un vehículo (VIId).

Las aplicaciones de este problema son muy similares a las del TSP. Igualmente se ven beneficiadas las empresas con cadenas de suministros y en este caso particular con flotas de más de un vehículo. Entre las técnicas exactas que resuelven el VRP se encuentra nuevamente el algoritmo de propósito general Ramificación y Corte, en tres versiones propuestas por Augerat en 1994 (25), Baldacci (26) y Lysgaard (27) en 2004. En (28), se reporta que los algoritmos anteriores son capaces de resolver problemas de hasta 135 puntos. Con respecto a las técnicas probabilísticas, en (6), (29), (30) y (31), se muestran evidencias de la calidad de las soluciones ofrecidas por los AG. Estos trabajos describen potentes algoritmos para resolver instancias grandes del VRP, utilizando en unos casos AG paralelos.

El desarrollo de aplicaciones en función de resolver problemas orientados a la toma de decisiones, como es el caso de los POR, requiere de un módulo que permita la adquisición de datos. En este punto, los SIG facilitan la manipulación y visualización de las redes que son utilizadas como base de los POR.

1.3 Sistemas de Información Geográfica

Los Sistemas de Información Geográfica (SIG) surgen en la década de los 60s (32), con el objetivo de gestionar grandes volúmenes de datos que guardan relación con localizaciones geográficas. Actualmente son muy utilizados para representar y trabajar sobre las redes de distribución utilizadas en los POR.

Los SIG involucran una serie de componentes (ver Figura 1.3-1), que garantizan su funcionamiento y actualización constante. Todos estos componentes armonizados correctamente, permiten el tratamiento y visualización de información geográfica.



Figura 1.3-1 Componentes de un SIG

Los SIG tienen una forma peculiar de gestionar la información espacial. El sistema separa la información en diferentes capas temáticas y las almacena de forma independiente. De esta forma, el trabajo con cada capa se agiliza y simplifica. Además, se posibilita relacionar la información a través de la topología de los objetos, a fin de crear nueva información.

En los SIG es posible representar el mundo real mediante dos modelos: *ráster* y *vectorial*. El modelo vectorial almacena la información en forma de objetos de tipo punto, línea y polígono mediante sus coordenadas. Este formato resulta adecuado para la representación de objetos geométricos reales, como las redes de carreteras utilizadas en los POR. El acceso y cálculo de distancias se simplifica por el hecho de que las entidades geográficas se representen como objetos. Por su lado, el modelo ráster no es apto para la manipulación de los datos, pues pierden definición con los cambios de escala. Además, en este modelo la identificación de objetos y cálculo de distancia es muy costoso. Por este motivo, el modelo empleado es el vectorial.

Aunque existen otras herramientas que permiten la gestión de datos espaciales (p.ej. trabajo directo con las bases de datos), los SIG marcan una diferencia en cuanto al resto de las variantes. Ellos permiten la visualización de datos geográficos en un mapa, con la facilidad de aplicar a los objetos funcionalidades de identificación, medición y edición. Las componentes de los SIG permiten la explotación total de las bondades de los datos geográficos. Así, son las herramientas seleccionadas en esta investigación para obtener los datos que permitan solucionar los POR y visualizar las rutas.

1.3.1 SIG que solucionan POR

El conjunto de herramientas que provee ArcGIS (de ESRI, por sus siglas en inglés) tiene un potente módulo de análisis de redes (33). Esta herramienta es una de las más abarcadoras en cuanto a POR refiere, ya que permite solucionar los cuatro problemas que se están estudiando. Su principal desventaja es que es una herramienta propietaria, lo que impide el análisis de las funcionalidades existentes y la incorporación de nuevas. Dentro de las herramientas libres, destaca el sistema gvSIG (34), que permite solucionar el PCM y el PRC. Otras herramientas libres analizadas en (35) son los SIG Open JUMP y KOSMO. El primero resuelve el PRC y el segundo soluciona el PCM. Todos estos sistemas son similares y no ofrecen mejoras con respecto a la elección del SIG GeoQ, ya previamente realizada en el proyecto. Este SIG está basado en la plataforma QuantumGis. Provee además de las funcionalidades básicas, un módulo de herramientas de análisis avanzadas, enfocadas a facilitar el manejo de la información vectorial. Por tanto, el SIG GeoQ es el sistema que se va a utilizar durante la investigación.

1.3.2 GeoQ

El SIG GeoQ (36) es desarrollado por el proyecto SIG-Desktop de la Facultad 6 de la Universidad de las Ciencias Informáticas. Admite el trabajo con datos vectoriales, lo que posibilita la explotación de las facilidades de este modelo en el tratamiento de POR.

Actualmente GeoQ cuenta con tres complementos que permiten solucionar algunos de los POR tratados en esta investigación. Primeramente el complemento pgRouting (9), que soluciona el PCM y el TSP. Para el cálculo del camino mínimo emplea tres algoritmos: Dijkstra, A-Star, Shooting-Star. El primero es la clásica versión de Dijkstra que partiendo de un punto inicial busca el camino más cercano a un objetivo. Por su lado, el algoritmo A-Star, adiciona información geográfica del punto inicial y del objetivo a cada línea en la red. De esta forma, permite que la búsqueda del camino priorice las líneas más cercanas al objetivo. Finalmente, el algoritmo Shooting-Star, se especializa en las rutas de líneas a líneas y no de puntos a puntos como los algoritmos anteriores. Define un conjunto de relaciones entre las líneas e igual que el A-Star prioriza aquellas más cercanas al objetivo de la búsqueda. Este complemento requiere para su utilización que los datos se encuentren almacenados en un SGBD PostgreSQL. En segundo lugar, se encuentra un complemento basado en la reducción de grafos que resuelve únicamente el PCM. El grafo se reduce a partir de una relación de equivalencia, que se emplea también de forma inversa para expandir dinámicamente las porciones. Una relación de equivalencia agrupa elementos entre los que se establece un vínculo. Luego de que el grafo se encuentra reducido se aplica el algoritmo de Dijkstra. La desventaja de este método radica en que

siempre que cambie la relación de equivalencia, ya sea porque cambie el mapa o se desee una nueva reducción, es necesario realizar nuevamente todo el procesamiento. Este proceso afecta el tiempo de respuesta de la aplicación. Por último, provee el complemento Road Graph, que resuelve el PCM utilizando Dijkstra sobre un grafo que guarda las intersecciones entre líneas. Permite realizar la optimización atendiendo a criterios de longitud y tiempo. En este caso, requiere que se esté utilizando como base de GeoQ la plataforma Quantum Gis versión 1.6 o mayor.

Los SIG como GeoQ, proveen un conjunto de funcionalidades de análisis de datos espaciales. Estas funciones permiten integrar la información distribuida en capas y transformarla en datos de entrada para procedimientos más avanzados, como muestra la Figura 1.3-2.

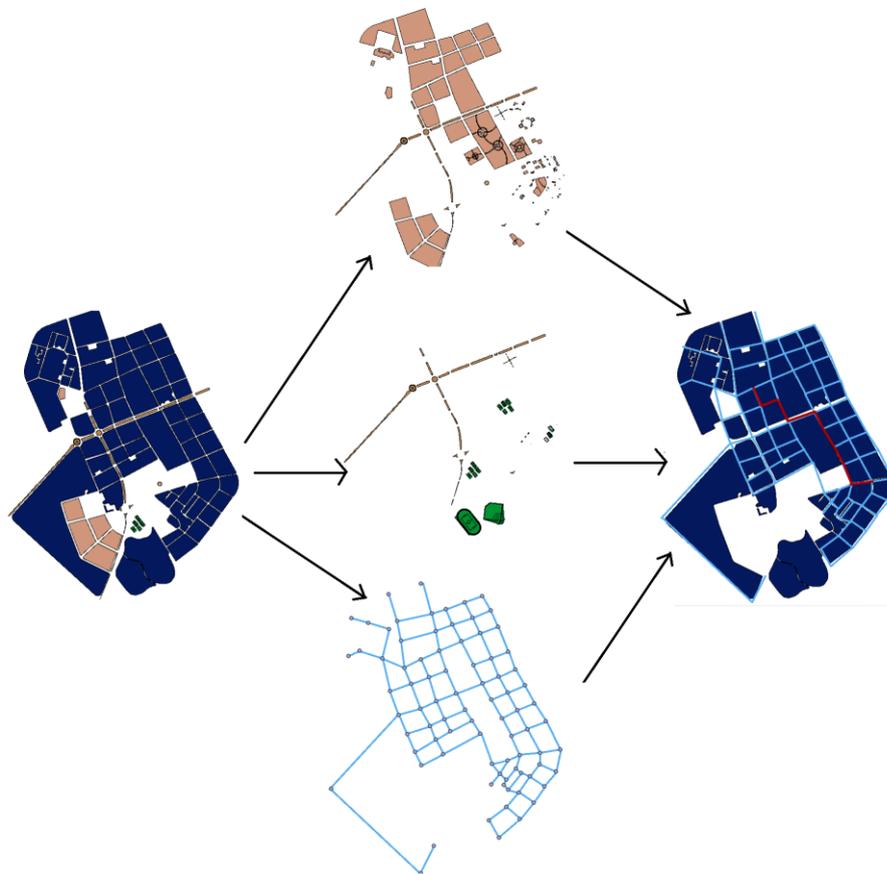


Figura 1.3-2 Funcionalidad de análisis

Dentro de las funciones básicas que provee GeoQ son utilizadas la intersección de objetos, el cálculo de distancias, la identificación de objetos y la creación dinámica de objetos.

Además de estas funcionalidades básicas, GeoQ cuenta con un conjunto de herramientas avanzadas llamado *ftools*. Este complemento permite el manejo de datos espaciales y ofrece funcionalidades de análisis, investigación y geometría.

En particular, destacan las dos funciones que se describen a continuación:

- **Matriz de distancia:** Permite obtener una matriz de distancia entre dos capas de puntos. Esta funcionalidad no integra la información existente en todas las capas, por lo que no tiene en cuenta la visibilidad real entre los puntos. Así, la matriz guarda distancias entre puntos entre los que se encuentran objetos que impiden su comunicación.
- **Análisis de vecinos más próximos:** Calcula estadísticas de los vecinos más próximos. En este caso, no ofrece puntos vecinos a otro, por lo que no resulta útil.

Las funcionalidades anteriores no permiten obtener los datos necesarios para resolver POR. Los siguientes sub-acápites detallan los problemas encontrados en estas dos funciones y ofrecen soluciones.

1.3.2.1 Matriz de Costo

GeoQ permite hallar la matriz de costo basándose en la existencia de una capa de puntos. Esta funcionalidad se puede combinar con la de Extraer Nodos de una Capa, que devuelve los puntos que se encuentran en los vértices de los polígonos en una nueva capa. No obstante, la matriz resultante no es factible para utilizar como entrada de datos para los algoritmos que resuelven POR, porque no tiene en cuenta la visibilidad entre cada uno de los puntos; o sea, no considera la posible intersección con otros objetos de la cartografía.

Esta investigación propone un procedimiento para obtener una matriz de costo eficiente, partiendo de la existencia de una capa de puntos. La propuesta guarda la distancia entre dos puntos siempre y cuando estos sean visibles entre sí; o sea, que el segmento de línea que definen no se intersecte con ninguna otra geometría del resto de las capas.

Sin embargo, en los problemas reales la matriz de costo guarda muy pocas intersecciones. Esto se debe a que en las carreteras reales, en una esquina suelen intersectarse solamente cuatro calles. Estas matrices son conocidas como matriz esparcida. En este punto, existen varias formas de optimizar el almacenamiento de estas estructuras. La forma más trivial es almacenando la matriz utilizando listas enlazadas y no arreglos bidimensionales. Se crea una lista ordenada que almacena los

índices de las filas de la matriz original y para cada fila una lista de pares formados por el índice de la columna y el costo asociado. La principal ventaja que trae esta representación es que se eliminan los ceros innecesarios y se gestiona de forma más eficiente la memoria. Por otro lado, las estrategias de reducción sobre grafos permiten analizar en detalle las propiedades del grafo, considerando únicamente los vértices significativos para analizar. Se basan en descomponer el grafo general en subgrafos, que a su vez serán considerados como vértices del grafo reducido. Estos nuevos vértices serán conectados atendiendo a reglas prefijadas. Las reducciones son realizadas por medio de eliminación o contracción de aristas, fusión de nodos y reducción de caminos.

1.3.2.2 Análisis de Visibilidad

La visibilidad es uno de los conceptos más estudiados en la Geometría Computacional. En el contexto de un SIG, dos puntos a y b son visibles si el segmento (imaginario) \overline{ab} no intersecta a ninguno de los objetos espaciales que se encuentran en el resto de las capas. GeoQ tiene la funcionalidad de Análisis de vecinos próximos, que ofrece datos estadísticos como la distancia media entre puntos, pero que no son de utilidad para el trabajo de los algoritmos.

Otras técnicas para el análisis de la visibilidad implican un pre-procesamiento de los datos. Es el caso de los grafos de visibilidad (37), donde cada vértice está enlazado con todos sus vértices visibles. Los mejores algoritmos para la construcción de estos grafos tienen complejidad $O(n^2)$, pero permiten reducir la complejidad de búsqueda de vecinos a $O(v)$, siendo v el número total de vecinos. Aunque en el peor de los casos v es igual a n , en problemas reales el número de vecinos suele no pasar de cuatro, de modo que disminuye considerablemente la complejidad.

Otra de las estructuras más utilizadas para el tratamiento de la visibilidad son los Diagramas de Voronoi (38). Un diagrama de Voronoi codifica la información de proximidad entre elementos. Partiendo de la existencia de un conjunto de puntos P , se define el Diagrama de Voronoi como la subdivisión del plano en n celdas, una para cada punto de P . Los métodos Incremental y Divide y Vencerás permiten construir estos diagramas, con complejidades $O(n \log n)$ y $O(n)$ respectivamente. Localizar el vecino más cercano a un punto dentro de una región de Voronoi se reduce a tiempo $O(\log n)$. Sin embargo, esta estructura se define en base a una clase de objetivos determinados en el mapa, y no tiene en cuenta la capa de rutas. Así, un punto puede ser el más cercano según las regiones definidas dada la clase seleccionada, pero considerando el recorrido por la capa de rutas puede resultar más cerca un punto de otra región de Voronoi.

El análisis anterior indica que aún hay trabajo que hacer en el tratamiento de la visibilidad, de modo que los métodos planteados puedan utilizarse en la resolución de POR con AG. En esta investigación, el método para hallar vecindad usa una matriz de costo creada como se propone en el acápite anterior. La búsqueda de vecinos para este caso tiene complejidad $O(n)$, pero con la variante de representación como matriz esparcida usando listas enlazadas se reduce a $O(v)$, siendo v el número total de vecinos.

1.4 Conclusiones

Todos los problemas objeto de estudio en esta investigación tienen características comunes, que permiten sean agrupados como POR. Estos son problemas de optimización que basan su funcionamiento en una red de puntos localizados geográficamente. La existencia de POR y su dificultad, ha devenido en un alto interés científico por encontrar algoritmos que los solucionen de forma eficiente. El análisis realizado en este capítulo ha demostrado que una gran variedad de métodos, tanto exactos como probabilísticos, han sido empleados con este objetivo. La utilización de un algoritmo exacto restringe la dificultad de los problemas que pueden solucionarse en tiempos aceptables, puesto que las instancias con mayor impacto social tardan mucho en ejecutarse. Sin embargo, las metaheurísticas son técnicas probabilísticas que ofrecen soluciones de calidad en tiempos razonables. Existen muchas adaptaciones de metaheurísticas que solucionan POR. El estudio del arte arrojó, que los AG se han utilizado en la totalidad de los problemas planteados con buenos resultados, de ahí que sean la estrategia elegida por esta investigación.

Al mismo tiempo, a medida que las redes de puntos se hacen más grandes se requieren sistemas que permitan manejar los datos de entrada y visualizar los resultados de forma sencilla, como es el caso de los SIG. En específico GeoQ, el SIG seleccionado, provee además de las funcionalidades básicas de análisis vectorial, un módulo de herramientas llamado *ftools*. Este complemento presenta problemas en la funcionalidad que obtiene la matriz de costo y que realiza el análisis de vecindad. De modo general, la matriz de costo no tiene en cuenta la interacción con objetos geográficos de otras capas y sólo se ofrecen datos estadísticos de los vecinos.

CAPÍTULO 2 Algoritmos Genéticos para Problemas de Optimización de Rutas

2.1 Introducción

John Holland es considerado el creador de los Algoritmos Genéticos, aunque el término fue mencionado inicialmente por Bagley en 1967 (39). Según Goldberg en (2) los AG son:

“Algoritmos de búsqueda basados en los mecanismos de selección natural y genética natural. Combinan la supervivencia de los más compatibles entre las estructuras de cadenas, con una estructura de información ya aleatorizada, intercambiada para construir un algoritmo de búsqueda con algunas de las capacidades de innovación de la búsqueda humana.”

La efectividad de los AG se basa en el paralelismo implícito que presentan. Este paralelismo se basa en el modelo de esquemas y la teoría de los bloques constructivos. Según Goldberg en (2), un esquema es un modelo de similitud que comprende un conjunto de cadenas con símbolos iguales en determinadas posiciones. En (40), Holland plantea el teorema fundamental de los AG, que expresa que los bloques constructivos (esquemas cortos, con aptitud superior a la media) tienen mayor probabilidad de propagarse a lo largo del proceso evolutivo. Así, ambos llegan a la conclusión de que al analizar una determinada generación de una población de tamaño n , se están analizando simultáneamente del orden de n^3 esquemas diferentes. De esta forma, en cada generación el espacio de búsqueda se explora en muchas más direcciones de las que se ven a simple vista.

2.2 Funcionamiento de los AG

La Figura 2.2-1 muestra cómo funcionan los AG.

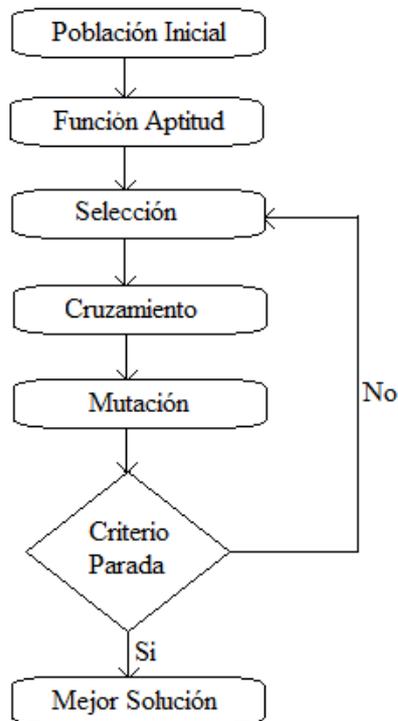


Figura 2.2-1 Esquema del funcionamiento de un AG

Los AG son métodos poblacionales que se basan en la creación de una población inicial con individuos codificados. Esta codificación consiste en una forma de modelar las características de una posible solución al problema que se va a resolver. Luego, se procede a evaluar la población para determinar el valor de aptitud de cada uno de los cromosomas. Este valor es determinado por la función de aptitud, que guía el proceso de búsqueda por medio del operador de selección. Luego, se realiza un ciclo evolutivo compuesto por tres procesos fundamentales, selección, cruzamiento y mutación. En cada evolución se seleccionan individuos para ser cruzados y los descendientes quedan pendientes a mutaciones. Cada evolución tiene la finalidad de obtener nuevos cromosomas que hereden las mejores características de la población actual. Este proceso se repite hasta que se cumple un determinado criterio de parada, que usualmente está determinado por un número de generaciones.

Los siguientes sub-acápites describen en detalle los componentes que van a ser modificados durante la investigación.

2.2.1 Población inicial

Existen dos elementos fundamentales a considerar respecto a la población inicial en los AG: el tamaño de la población inicial y el procedimiento empleado para inicializarla.

Se estimaba que el tamaño de la población debía aumentar de forma exponencial con la dificultad de los problemas para poder generar buenas soluciones. No obstante, recientes estudios demuestran que con poblaciones pequeñas se obtienen buenos resultados. Así, no se ven afectados los costos computacionales de memoria y tiempo como sucede con poblaciones excesivas.

Respecto a los procedimientos para generar la población inicial, son utilizadas tanto la inicialización aleatoria como la heurística, aunque más frecuentemente la primera. La inicialización aleatoria genera genes aleatorios hasta que se completa el cromosoma. El uso extendido de esta inicialización se debe a que los individuos creados aleatoriamente pueden ser muy malos, pero tener buenos bloques constructivos para el avance del algoritmo. Por su lado, la inicialización heurística crea individuos atendiendo a un determinado criterio e intenta que sean de calidad. Esta calidad en la población inicial pudiera terminar en una convergencia prematura hacia un óptimo local. En los AG, la calidad de las soluciones finales debe ser producto del avance de las generaciones y no de la calidad de la población inicial. Sin embargo, cuando la dificultad de los problemas aumenta, como cuando se intenta resolver un POR en una red grande, es necesario utilizar heurísticas constructivas para lograr la convergencia del algoritmo. En el desarrollo de estas heurísticas se pudieran aprovechar las funcionalidades de vecindad de los SIG. Así, incorporando conocimiento vinculado al problema se puede mejorar el proceso de evolución.

2.2.2 Operador de cruzamiento

El operador de cruzamiento es el encargado del intercambio de material genético entre dos individuos de la población (padres). Cuando se produce el cruce, se crea una descendencia que combina los genes de ambos padres, posibilitando la explotación del espacio de soluciones. Estos operadores están sujetos a una probabilidad de cruce (valor entre 0 y 1) que suele tomar valor 0.6. Entre los operadores más conocidos se encuentran:

- Único punto: Consiste en la selección aleatoria de una posición dentro del cromosoma. Luego, se intercambian los genes desde la posición seleccionada hasta el final del cromosoma.
- N Puntos: Generalización del cruzamiento de un punto, suele utilizar un valor de $n = 2$ (siendo n el número de puntos), debido a que minimiza los efectos destructivos. Consiste en la selección aleatoria de n posiciones en los cromosomas y el intercambio de los genes que se encuentran entre las posiciones.

- Uniforme: Consiste en formar los hijos con genes que se escogen de forma aleatoria de uno u otro padre. Tiene un efecto destructivo mucho más elevado que las técnicas anteriormente expuestas.

2.2.3 Operador de mutación

El operador de mutación afecta a cada hijo de forma independiente. Consiste en la modificación de forma aleatoria de los genes del cromosoma. La mutación garantiza que ninguna solución dentro del espacio de búsqueda tenga probabilidad 0 de ser examinada. Permite la exploración del espacio de soluciones y la introducción de diversidad en la población. Estos operadores están sujetos a una probabilidad de mutación (valor entre 0 y 1) que suele tomar valor 0.001. Esta probabilidad tan baja es porque al igual que ocurre en la naturaleza, las mutaciones no son muy habituales. Entre los operadores más conocidos se encuentran:

- Simple: Consiste en seleccionar un único gen de forma aleatoria y variarlo.
- Intercambio: Consiste en seleccionar dos posiciones de forma aleatoria e intercambiar los genes que se encuentran en ellas.
- Inversión: Consiste en seleccionar dos posiciones de forma aleatoria e invertir los genes que se encuentran entre ellas.
- Inserción: Consiste en seleccionar un gen de forma aleatoria e insertarlo en una nueva posición seleccionada de la misma forma.
- Dispersión: Funciona de forma similar al operador de inserción solo que aplica para una sub-cadena de genes. Consiste en seleccionar dos posiciones de forma aleatoria. La sub-cadena de genes que se encuentra comprendida entre estas posiciones se inserta en una nueva posición seleccionada también de forma aleatoria.

2.3 Estrategias actuales enfocadas a POR

En primer lugar, se muestran las estrategias actuales enfocadas a la codificación de las soluciones. De modo general, en todos los POR se considera un mapa representando una red de calles de una ciudad. En esta red cada punto (intersección entre calle) tiene un identificador desde 1 hasta N , que es la cantidad total de puntos en la red.

En el PCM, la secuencia de puntos es delimitada por un punto origen y uno destino. En los trabajos (3) y (13), cada solución se codifica como una secuencia de números enteros (identificadores de los puntos en el mapa). Estos trabajos destacan que esta representación entera es la más apropiada para

modelar estos problemas que establecen secuencias. La Figura 2.3-1 muestra que en este problema los cromosomas no tienen que tener las mismas longitudes. Esto sucede porque el objetivo está centrado en llegar a un destino, de modo que cuando este punto aparezca en la secuencia, no es necesario continuar almacenando más puntos.

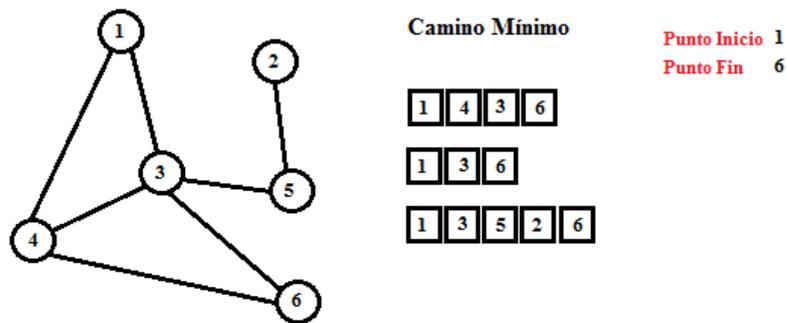


Figura 2.3-1 Soluciones para el PCM

Para el caso del PRC, igualmente se parte de un punto inicial y se finaliza cuando se ha llegado a un recurso. La representación en este caso no almacena los identificadores de los puntos, sino las direcciones en las que se encuentran los mismos. Esta dirección igualmente es un número entero, que suele no exceder el valor de 4 en escenarios reales, ya que las intersecciones en las vías suelen realizarse entre dos calles. En los trabajos consultados se han utilizado codificaciones binarias (17), pero más se destaca el uso de la entera (41). La Figura 2.3-2 muestra que al igual que en el problema anterior los cromosomas son de longitud variable. En este caso específico, esto se debe a que los recursos están en diferentes regiones y por ende no se requieren la misma cantidad de pasos para llegar a ellos. Como el objetivo se centra en localizar el recurso más próximo, los puntos finales de las rutas (posibles direcciones a los recursos) son variables.

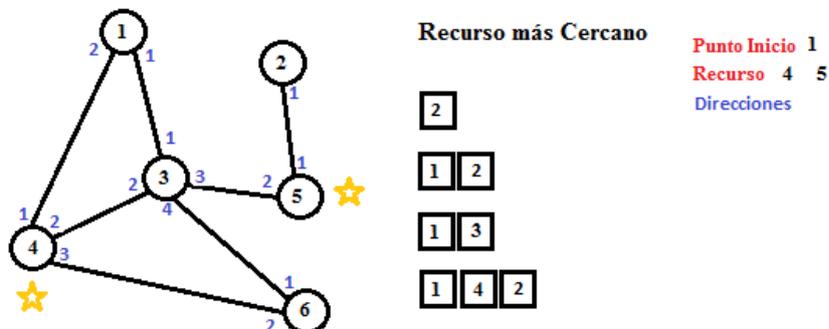


Figura 2.3-2 Soluciones para el PRC

En el caso del TSP, el cromosoma parte de un punto inicial y recorre el resto de los puntos para luego regresar al inicio. El estado del arte sugiere que se ha incursionado en las representaciones matriciales (42), no obstante predomina el uso de la entera (18). La Figura 2.3-3 muestra que en este caso los cromosomas van a ser de longitud constante, ya que todos los puntos tienen que recorrerse en la ruta.

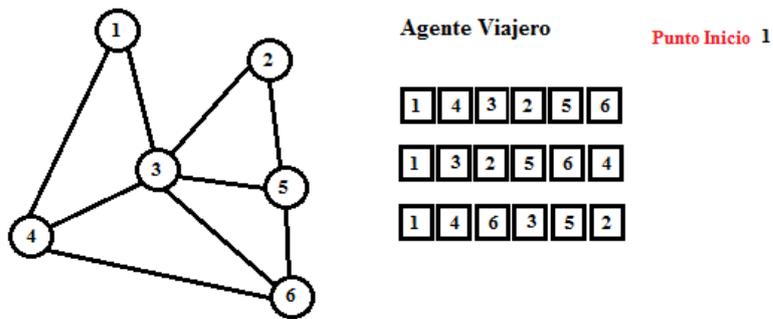


Figura 2.3-3 Soluciones para el TSP

Para el VRP, la secuencia está dada por un conjunto de sub-secuencias de enteros dependiendo del número de vehículos. Entre las sub-secuencias no debe haber puntos en común y unidas deben contener todos los puntos. La bibliografía consultada utiliza en todos los casos la codificación entera (30), (43). La Figura 2.3-4 muestra que al igual que el problema anterior la longitud del cromosoma permanece constante, ya que las sub-secuencias se representan de forma continua. Estas sub-secuencias se delimitan por las v posiciones que se adicionan al final del vector de enteros y que dependen del número de vehículos. En cada una de estas posiciones se almacena el número de puntos que contiene la sub-secuencia del vehículo v_i .

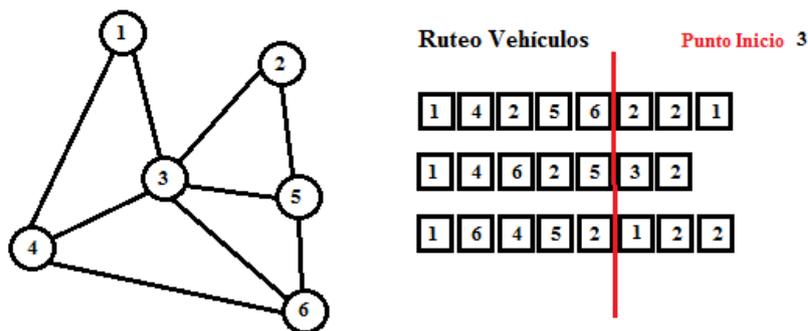


Figura 2.3-4 Soluciones para el VRP

Finalmente, se identifica a la codificación entera como una alternativa viable para generalizar una estrategia de representación que permita modelar POR en los AG. Esta codificación permite

representar a las rutas en su forma natural, estableciendo un orden entre el conjunto de puntos que permita fácilmente distinguir un recorrido.

A continuación se describen las estrategias enfocadas a determinar el valor de aptitud de las soluciones. Para el PCM y el PRC, el valor de aptitud se determina como la sumatoria de los costos de ir desde el punto inicial hasta el final o recurso (ver Figura 2.3-5).

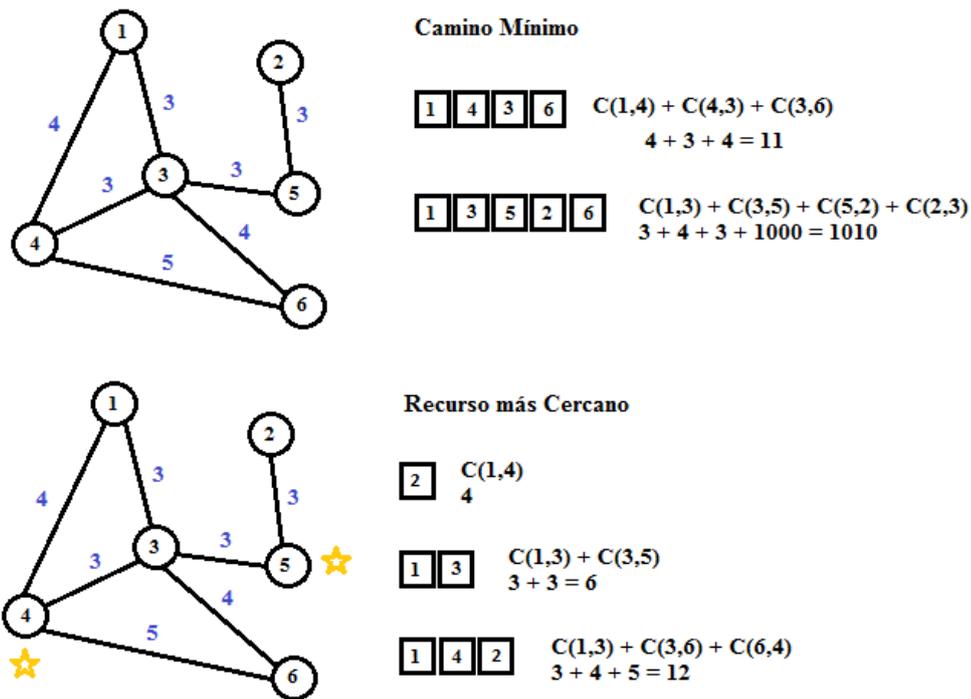


Figura 2.3-5 Valor de aptitud para las soluciones del PCM y PRC

Para el caso particular del TSP, la Figura 2.3-6 muestra que es necesario adicionarle a la sumatoria de recorrer la secuencia, el costo de ir desde el último punto de la misma hasta el inicial.

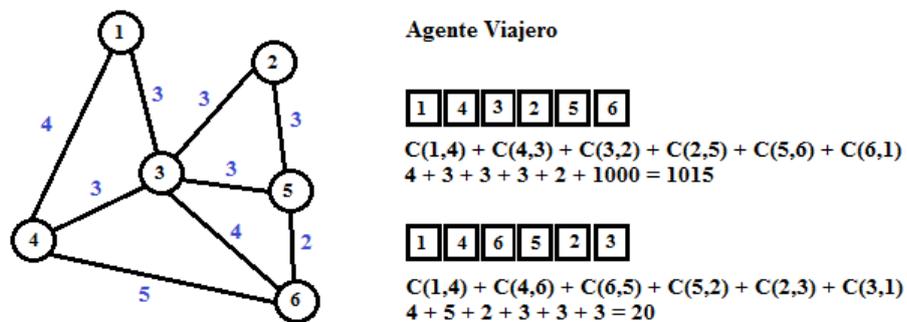


Figura 2.3-6 Valor de aptitud para las soluciones del TSP

Por su lado, para el VRP, la Figura 2.3-7 muestra que es necesario delimitar las sub-secuencias para cada vehículo, de modo que el valor de aptitud para la solución sea la sumatoria de transitar cada una de las sub-secuencias. Además, hay que sumar al valor general el costo de ir desde el punto inicial al inicio de cada sub-secuencia y desde el final de estas nuevamente al punto inicial.

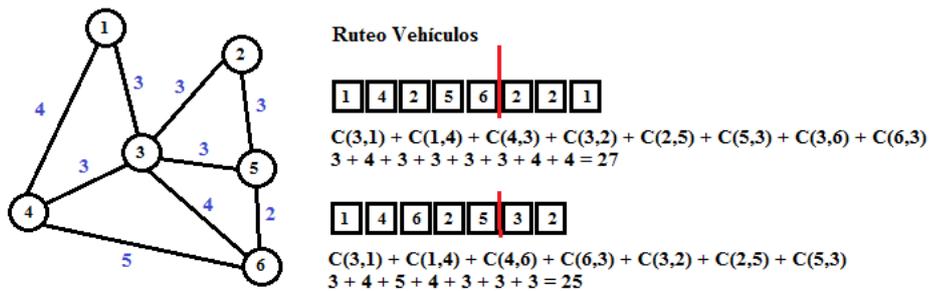


Figura 2.3-7 Valor de aptitud para las soluciones del VRP

El operador de selección que se va a emplear es el Torneo Binario, ya que es el empleado en la mayor parte de la bibliografía consultada. El tamaño del torneo permite mantener una presión de selección baja, que posibilite la exploración del espacio de soluciones. El algoritmo va a ejecutarse bajo un esquema de selección generacional. Este esquema plantea que en cada paso de la evolución se genera una población completa.

Los AG diseñados para el PCM utilizan cruzamientos de uno y dos puntos en (12) y (44). Por otra parte, para solucionar el PRC se emplean variaciones del cruzamiento de un punto en (4). A medida que los problemas se tornan más difíciles, se requieren nuevos operadores para alcanzar el óptimo, tal es el caso de los operadores diseñados para el TSP y el VRP. Para el primer problema, en (18), se utilizan los cruzamientos PMX, OX, AP y POS, que varían tanto las posiciones como el orden de los genes en la solución. Además, en (23), se emplea un cruzamiento cíclico, que consiste en seleccionar cada elemento de alguno de los padres y colocarlos en la posición en que estaba previamente. En (5), se emplean los cruzamientos CX, MOX y se reiteran PMX y OX. En (45) y (46), se emplea el operador DPX, que intenta crear un descendiente que preserve la misma distancia entre ambos padres. Para el segundo problema, resaltan los cruces BOC y HLC en (30), que integran el concepto de cuadro delimitador. Además, en (6), se emplea un operador que consiste en donar un número aleatorio de destinos de un determinado vehículo a otro. Por último, en (31), se emplea el cruzamiento genérico, que selecciona una sub-ruta aleatoria de un padre y la inserta después del punto más cercano geográficamente al punto inicial de la sub-ruta en el otro padre.

Por su lado, las mutaciones para el PCM en (12) y (44), consisten en seleccionar una posición aleatoria y regenerar el cromosoma a partir de ella. Además, son utilizadas variaciones de la mutación de un único punto en (47). Para el caso del TSP, en (18) y (23), se emplea una variación simple; o sea, se selecciona una ciudad para variar y se intercambia con la nueva, para evitar que se repitan genes y se introduzca una falsa solución. Además, en (20), se emplea la mutación Cadena de 2 Cambios, que consiste en reemplazar cuatro arcos de la ruta. Por último, en (5) y (48), se emplean los intercambios 2-Opt, 3-Opt y OR-Opt, que realizan cambios en las soluciones en busca de mejoras en el valor de aptitud. En (46), se emplea el operador 4-Cambio, que transforma la solución de modo que la mutación se distancie solamente en 4 de la no mutada. Finalmente, para el VRP, en (6), se emplea el operador de inversión para un único vehículo seleccionado aleatoriamente. En (29), se realiza un intercambio OR-Opt y en (30) se emplea una mutación simple.

2.4 Complejidad computacional

Fijamos n como el número de cromosomas o tamaño de la población y l como la longitud del cromosoma. Para los AG generacionales, se establece que el torneo binario tiene complejidad $O(n * (n/2))$. El cruzamiento, mutación, selección y almacenamiento temporal de la población tiene complejidad $O(2 * l + l + C_{ev} + l)$. Por su lado, el reemplazo es orden $O(n * l)$ y el cálculo de las estadísticas $O(n)$. Finalmente, la complejidad del algoritmo es $O(n^2 + n * l)$ según (49).

2.5 Conclusiones

Los AG son una de las metaheurísticas más utilizadas para solucionar POR. Aunque no garantizan encontrar siempre la solución óptima, empíricamente han demostrado ofrecer soluciones aceptables en un tiempo razonable (7). Este comportamiento se justifica porque resultan menos afectados por los óptimos locales y porque operan de forma simultánea con varias soluciones. Los POR son problemas que tienen un espacio de búsqueda limitado, es posible definirles una función de aptitud y existe una codificación fácil de implementar, lo que facilita el diseño de los AG. El estudio realizado permitió identificar una representación sencilla que permite modelar todos los problemas, la codificación entera. La creación de la población inicial debe tener un alto grado de aleatoriedad, pero pudiera optimizarse con alguna heurística para lograr la convergencia de las instancias más grandes. De modo general, el cruzamiento y la mutación son variables y suelen complementarse con técnicas de mejoramiento de rutas. Por último, la función de aptitud se mantiene similar para todos los problemas, basándose en la sumatoria de los costos de transitar las rutas, sujeta a las restricciones particulares.

CAPÍTULO 3 Diseño de Nuevos Componentes de Algoritmos Genéticos

3.1 Introducción

Durante la revisión bibliográfica realizada, se identificaron tres posibles elementos que pudieran ser modificados para mejorar el desempeño de los AG para POR. Primeramente, se describen dos nuevas estrategias para la inicialización de la población: IPP e IPP en Doble Sentido. La primera se basa en la visibilidad entre puntos, hallada por medio del SIG GeoQ. Por su lado, la segunda utiliza la IPP para crear soluciones partiendo de punto inicial hasta el final y viceversa. Esta estrategia se basa en la existencia de buenos bloques constructivos en el inicio y fin del espacio de búsqueda, cuando el mismo se encuentra reducido. Existen dos procesos fundamentales dentro del ciclo evolutivo que permiten la explotación y exploración del espacio de soluciones, el cruce y mutación respectivamente. Esta investigación propone dos operadores de cruzamiento y dos de mutación.

3.2 Propuesta de inicialización de la población

Esta investigación propone dos nuevas formas de inicialización. La primera aprovecha las capacidades del SIG GeoQ y propone construir las rutas basándose en los conceptos de visibilidad y proximidad entre puntos. Partiendo del punto inicial, adiciona puntos a la ruta dando mayor probabilidad a los más próximos al punto precedente, hasta que llega al punto final. La segunda propuesta, utiliza como base la IPP para crear las soluciones desde el punto inicial hasta el final y en sentido contrario. Esta variación, permite aprovechar los buenos bloques constructivos que existen tanto al inicio como al fin del espacio de soluciones, cuando el mismo se encuentra reducido.

3.2.1 Inicialización Proporcional a la Proximidad (IPP)

Dos conceptos son utilizados en esta primera propuesta para la creación de la población inicial en los AG, llamada Inicialización Proporcional a la Proximidad.

- Visibilidad: Actualmente esta información se extrae de la matriz de costo. No obstante existen métodos más eficientes analizados en la sección Análisis de Visibilidad
- Proximidad: Mientras la distancia entre dos puntos visibles sea menor, estos son más próximos.

Selección Proporcional a la Proximidad: Cada punto tiene asociada una lista con sus puntos visibles. Se organiza una ruleta según su proximidad (ver Figura 3.2-1), de modo que aquellos puntos más próximos ocupen mayor espacio y tengan mayor probabilidad de ser seleccionados para formar la ruta.

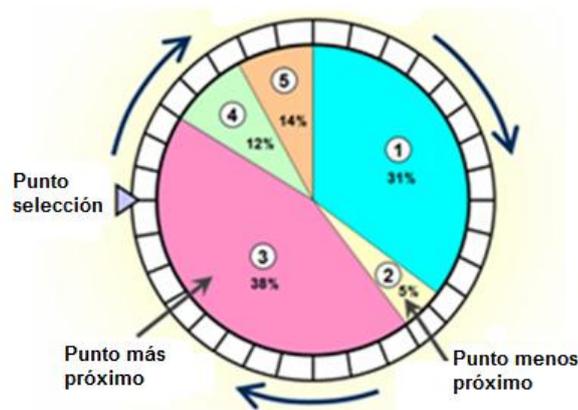


Figura 3.2-1 Selección proporcional a la proximidad

La construcción basada en la visibilidad y proximidad persigue crear menos rutas falsas y con mejores bloques constructivos que las producidas con una estrategia aleatoria.

En los siguientes algoritmos se parte de un punto inicial y se adicionan a la ruta nuevos puntos siempre que sean visibles a su punto precedente. Estos puntos son seleccionados proporcionalmente a su proximidad. Esta selección depende de un valor aleatorio, lo que permite que el proceso no sea determinista. Si la ruta no se ha completado y todos los puntos visibles al último están incluidos, se le da un tratamiento especial a la solución dependiendo del problema. Los siguientes algoritmos detallan el tratamiento específico para cada problema. El *Algoritmo 1* es específico para el PCM y el PRC. Por su lado, el *Algoritmo 2* detalla el método para el TSP y el VRP.

Algoritmo 1. Init_Prop_Prox[Xinicial,Xfinal]

```

i := 0;
Xi := Xinicial;
Mientras Xi <> Xfinal
  Ruta[i] := Xi;
  Vi := visibles_no_visitados[Xi];
  Si |Vi| > 0 entonces
    Xi := selección_proporcional[Vi];
  Sino
    Xi := Xfinal;
  Fin
  i := i + 1;
Fin

```

Para el caso del PCM, se coloca el punto final y como esta es la condición de parada se termina el procedimiento. Por su lado, para el PRC se buscan los recursos que se comuniquen directamente con el punto actual y se adiciona a la solución el que se encuentre a menor distancia. De esta forma, termina el procedimiento porque se ha llegado a un recurso.

Algoritmo 2. Init_Prop_Prox[Xinicial]

```

i := 0;
Xi := Xinicial;
Mientras i < Npuntos
  Ruta[i] := Xi;
  Vi := visibles_no_visitados[Xi];
  Si |Vi| > 0 entonces
    Xi := selección_proporcional[Vi];
  Sino
    Xi := buscar_punto_no_recorrido[Ruta];
  Fin
  i := i + 1;
Fin

```

Para el caso del VRP el proceso que describe el *Algoritmo 2* se realiza para cada una de las sub-rutas, determinadas por el número de vehículos. En caso de que no haya puntos visibles no visitados, para ambos problemas se coloca un punto que no esté en la ruta y se continúa el procedimiento.

3.2.2 Inicialización en Doble Sentido (IDS)

El siguiente procedimiento es una extensión del anterior. Propone crear un 50% de las soluciones desde el punto inicial hasta el final y el resto desde el punto final hasta el inicial, utilizando en ambos casos la IPP. La Figura 3.2-2 muestra que a medida que se avanza en la creación de una solución, aumenta considerablemente el espacio de posibles soluciones. Sin embargo, al aproximarse al punto final el espacio se encuentra reducido. Al inicializar una solución comenzando por el punto inicial en busca del fin, está contendrá buenos bloques constructivos preferentemente al inicio, ya que se

encuentra contraído el espacio de soluciones. Este procedimiento intenta recrear este comportamiento inicializando las soluciones partiendo del punto final en busca del inicial. De esta forma, se crearán unas soluciones con buenos bloques constructivos en el inicio y otras en el fin. Este comportamiento es aprovechado por las estrategias de cruzamiento al mezclar soluciones creadas de ambas formas.

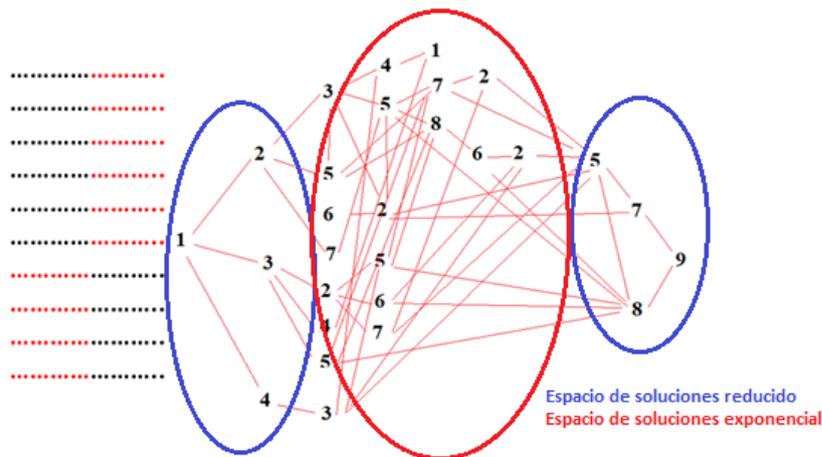


Figura 3.2-2 Espacio de soluciones

Los siguientes algoritmos coinciden en la descripción general del proceso de inicialización. Un 50% de las soluciones se crea partiendo del punto inicial, hasta llegar al final y el resto de forma contraria. En este caso se han realizado tres pseudocódigos. El *Algoritmo 3* es específico para el PCM y el *Algoritmo 4* para el PRC. Mientras que el *Algoritmo 5* se aplica para el TSP y el VRP.

Algoritmo 3. Init_Dbl_Sen[Xinicial,Xfinal,N]

```

Si random[0,1] > 0.5 entonces
    Ruta := Init_Prop_Prox[Xinicial, Xfinal];
Sino
    Ruta := Init_Prop_Prox_I[Xfinal, Xinicial];
Fin
    
```

Para el PCM, la función *Init_Prop_Prox* crea las soluciones partiendo del inicio y hasta el fin. Por su lado, la función *Init_Prop_Prox_I* crea las soluciones partiendo del punto final hasta el inicial y luego invierte la solución para que se mantenga válida.

Algoritmo 4. Init_Dbl_Sen[Xinicial, N].

```

R := selección_recursos[];
Para i:=1 Hasta N Paso N/k
    Ruta := Init_Dbl_Sen[Xinicial, R(k), N/k];
    k := k+1;
Fin
    
```

Para el PRC, se requieren rutas que lleguen hasta todos los recursos. La función *selección_recursos* realiza una previa selección de un recurso de forma aleatoria. Luego, se utiliza como base el *Algoritmo 3* para crear una ruta partiendo desde el punto inicial hasta llegar al recurso seleccionado o de forma inversa.

Algoritmo 5. Init_Dbl_Sen[Xinicial]

```
Ruta := Init_Prop_Prox[Xinicial];
Si random[0,1] > 0.5 entonces
    Ruta := Invertir[];
Fin
```

Para el TSP y el VRP, siempre se crea una solución partiendo del punto inicial hasta recorrer el resto de los puntos. Luego, la función *Invertir* invierte la ruta con excepción del punto inicial, que se mantiene en la misma posición. Las inversiones para el caso del VRP, se realizan de forma independiente en cada uno de los recorridos parciales.

3.3 Propuesta de operadores de cruzamiento

La mayoría de los operadores de cruzamiento utilizados para tratar POR, han sido diseñados de forma específica aprovechando las particularidades de cada problema. Sin embargo, el diseño de cruzamientos generales permitirá validar si estos ofrecen mejores resultados para varios problemas que los tradicionales. Las siguientes sub-secciones describen dos nuevos operadores de cruzamiento. Los ejemplos de las figuras son para el caso del PCM, pero en el resto de los problemas es similar.

3.3.1 Cruzamiento Basado en Costo (CBC)

Este operador de cruce utiliza los costos ($C(i,j)$) entre cada par de puntos. Consiste en realizar el intercambio que se muestra en la Figura 3.3-1 en un por ciento de genes de cada padre, siempre que repercuta en una disminución de la aptitud de la solución.

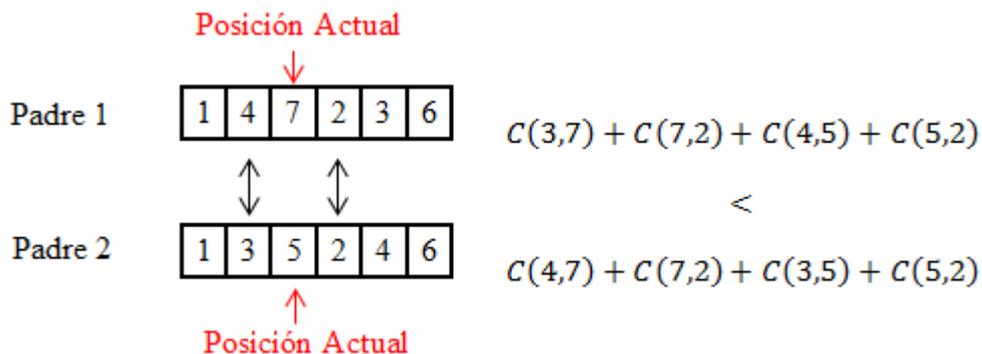


Figura 3.3-1 Realizando el CBC

Los padres intercambian el mismo número de genes y éstos se encuentran situados en las mismas posiciones.

Dependiendo de la naturaleza del problema, es necesario arreglar la solución para que se mantengan las restricciones del problema. Por ejemplo, para el TSP, el intercambio requiere que se acomoden los genes para que no se repitan. En este caso, se busca el gen que viene del padre uno en el padre dos y se inserta en su posición el gen que se encuentra en la posición del intercambio. Este operador va a permitir explotar el espacio de soluciones vecinas a la inicial con mejoras en la aptitud.

3.3.2 Variación Cruzamiento Uniforme (VCU)

El siguiente operador de cruzamiento que se propone se basa en las posiciones que ocupan los genes en los cromosomas padres.

Inicialmente, se copian el 50% de los genes de un padre en las mismas posiciones en que se encuentran a un hijo. Como se observa en la Figura 3.3-2, las posiciones no tienen que coincidir en ambos padres.

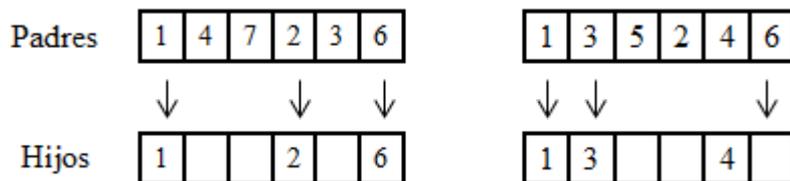


Figura 3.3-2 Copiando el 50% de los genes

Luego, se completan los hijos con los elementos del padre contrario en el orden en que se encuentran en el mismo (ver Figura 3.3-3).

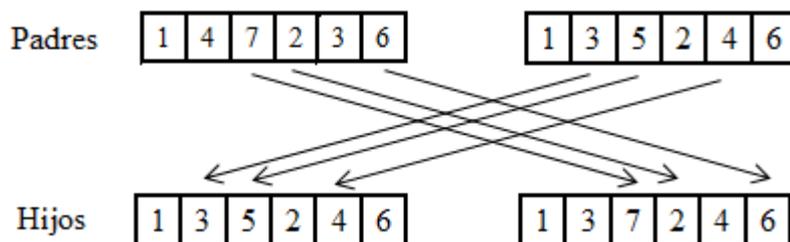


Figura 3.3-3 Completando con los genes de los padres contrarios

Igualmente, dependiendo de las particularidades de cada problema se realizan los movimientos necesarios para corregir los cromosomas. La inserción de los genes manteniendo el orden permite que

los cromosomas hijos hereden buenas características de los padres, a pesar de las variaciones aleatorias que se realizan en un primer momento.

3.4 Propuesta de operadores de mutación

Los operadores de mutación deben realizar cambios más o menos bruscos, que permitan explorar nuevas regiones del espacio de soluciones para que el algoritmo no se estanque. Las siguientes subsecciones describen dos nuevos operadores de mutación, el primero realiza un cambio pequeño y el segundo provoca mayor dispersión en la solución. Las figuras reflejan ejemplos para el caso del TSP, pero para el resto de los problemas ocurre de forma similar.

3.4.1 Variación Mutación Inversión (VMI)

Este operador consiste en una variación del operador de Inversión. Parte de seleccionar dos posiciones aleatorias, tal que un intercambio en sus genes adyacentes genere una disminución del costo total de la ruta, como se ilustra en la Figura 3.4-1.

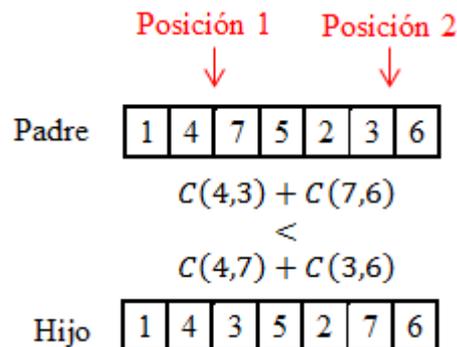


Figura 3.4-1 Buscar dos posiciones aleatorias cuyo intercambio permita una disminución del costo

Luego, se aplica la inversión al resto de los elementos que quedan comprendidos entre las posiciones seleccionadas (ver Figura 3.4-2).

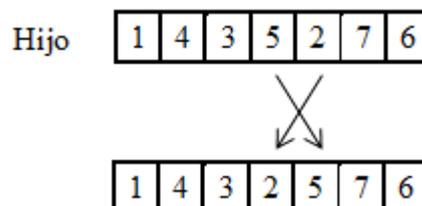


Figura 3.4-2 Aplicando la inversión

Este operador va a permitir explorar nuevas y mejores regiones del espacio de búsqueda, ya que la inversión solo se realizará en aquellos casos en que aumente la calidad de la solución inicial.

3.4.2 Mutación por Reagrupamiento de Bloques (MRB)

Este último operador de mutación consiste en seleccionar cuatro posiciones aleatorias, de modo que un intercambio entre ellas permita una disminución del costo de la ruta, como se observa en la Figura 3.4-3. Las posiciones dos y cuatro vienen a continuación de la uno y tres respectivamente.

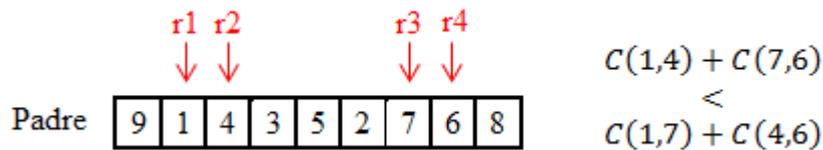


Figura 3.4-3 Buscar cuatro posiciones aleatorias que permitan disminución del costo

Si es factible reagrupar los genes que se encuentran en estas posiciones, se procede a identificar tres bloques comprendidos entre las 4 posiciones, como se ilustra en la Figura 3.4-4.

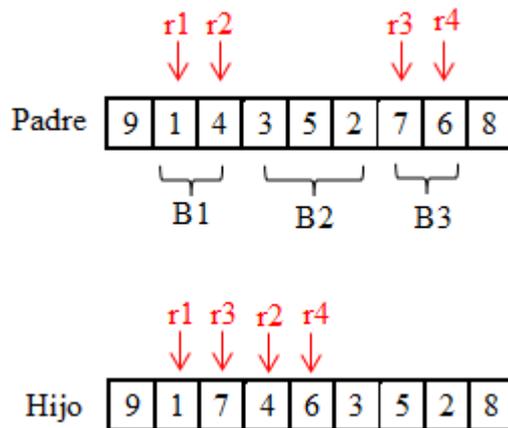


Figura 3.4-4 Reagrupando los bloques

Luego del gen de la primera posición, se inserta el gen de la tercera posición. Seguido, se coloca el gen que se encontraba en la segunda posición. Luego, se inserta el gen de la cuarta posición y se completa con los genes entre las posiciones dos y tres sin incluirlas.

Este operador tiene la intención de agrupar los genes que se encuentran en las cuatro posiciones seleccionadas de forma aleatoria. De esta forma, se crea un nuevo bloque constructivo que mejora la aptitud de la solución. Además, a lo largo de las generaciones este bloque puede ser heredado por otras soluciones por medio de las estrategias de cruce.

3.5 Conclusiones

Ambas inicializaciones propuestas aprovechan las facilidades que ofrecen los SIG para determinar la visibilidad y proximidad entre puntos. La idea es crear una estructura que permita que los puntos más próximos a otro tengan mayor probabilidad de ser elegidos en la formación de la ruta. Estas construcciones propician una disminución considerable de las falsas soluciones. Por su lado, los operadores de cruzamiento propuestos, se basan tanto en los costos entre puntos (genes), como en las posiciones entre estos. El operador CBC intenta intercambiar un por ciento de genes entre los padres, siempre que dicho intercambio repercuta en una mejora en la aptitud de las soluciones. No sucede lo mismo con el operador VCU, que mantiene los genes que se encuentran en un 50% de las posiciones y el resto los incorpora del padre contrario en el mismo orden en que aparecen. Mantener el orden de los genes en el padre posibilita que perduren buenas características. Los operadores de mutación propuestos, intentan realizar cambios siempre en función de una mejora en la aptitud de las soluciones. La VMI, es una mejora a la MI, que propone que se realice la inversión siempre que esta mejore la aptitud. Por otro lado, la MRB, reorganiza los bloques constructivos que se encuentran entre cuatro posiciones seleccionadas de forma aleatoria. Todas las estrategias de inicialización, cruzamiento y mutación diseñadas en este capítulo, serán contrastadas con las estrategias más utilizadas para cada uno de los problemas. Los resultados permitirán identificar si las propuestas realizadas ofrecen mejores resultados para todos los problemas que aquellas que han sido diseñadas de forma específica.

CAPÍTULO 4 Análisis de los Resultados

4.1 Introducción

Este capítulo presenta el diseño de las pruebas y los resultados obtenidos al contrastar las estrategias básicas de AG con las propuestas realizadas.

En estos experimentos las instancias utilizadas del PCM y del PRC fueron generadas de forma aleatoria, con 10, 30, 50 y 200 puntos. Para el TSP, se utilizaron las instancias publicadas en TSPLIB (50): ulysses16, eil51 y st70, de 16, 51 y 70 puntos respectivamente. Por último, para el VRP, igualmente se utilizan las instancias publicadas en TSPLIB: eil22, eil30 y eil51, de 22, 30 y 51 puntos, respectivamente.

Los algoritmos genéticos utilizados son generacionales, con un total de 512 individuos en la población. La selección de los padres que van a reproducirse en el ciclo evolutivo, se realiza por medio del torneo binario. Esta misma estrategia es empleada para seleccionar los descendientes que van a formar parte de la nueva población. Siempre se realiza cruzamiento y mutación solamente un 5% de las ocasiones. El algoritmo se termina cuando se realizan 500 generaciones. Estos parámetros quedan resumidos en la Tabla 1.

Tabla 1 Parámetros del algoritmo

<i>Tamaño de la Población</i>	512
<i>Selección de Padres</i>	Torneo (<i>tamaño</i> = 2)
<i>Probabilidad de Cruzamiento</i>	$p_c = 1$
<i>Probabilidad de Mutación</i>	$p_m = 0.05$
<i>Selección de Hijos</i>	Torneo (<i>tamaño</i> = 2)
<i>Condición de Parada</i>	500 Generaciones

Como es normal, para analizar el desempeño de algoritmos de naturaleza estocástica, se realizaron 30 corridas independientes y se computó la media de los valores obtenidos. Se comparó el rendimiento de

los algoritmos atendiendo a los valores de aptitud medio y diversidad. Esta diversidad es computada como la desviación estándar de la población con respecto a la media, que expresa la diferencia fenotípica que existe entre todos los individuos. Primeramente, se realiza la prueba Kolmogorov-Smirnov, que permite chequear si los datos siguen una distribución normal. En caso de que así sea, se realiza la prueba de ANOVA para comparar las medias. En otro caso, se usa la prueba de Kruskal-Wallis para comparar las medianas. El nivel de confianza que se utiliza es 95%.

El lenguaje de programación que se utilizó en la confección de los algoritmos fue C++. Todos los experimentos fueron realizados en una PC con un procesador Intel Core 2 Duo T7100, con velocidad 1.8 GHz y 2 GB de RAM. El sistema operativo utilizado fue GNU/Linux Ubuntu versión 11.04.

4.2 Variantes de inicialización aplicadas

En este primer acápite se comparan las variantes de inicialización propuestas en el capítulo anterior: Inicialización Proporcional a la Proximidad (IPP) e Inicialización en Doble Sentido (IDS), con respecto a la inicialización aleatoria del AG tradicional.

La Tabla 2 resume los valores de aptitud medio para cada una de las instancias de POR estudiados.

Tabla 2 Valor de aptitud medio para las estrategias de inicialización

<i>Problema</i>	<i>Instancia</i>	<i>Óptimo</i>	<i>AG</i>	<i>AG+IPP</i>	<i>AG+IDS</i>
PCM	r10	23	23.3	*23.0	23.0
	r30	14	21.1	*14.6	14.6
	r50	7	24.7	*7.1	7.1
	r200	3	36.7	3.6	*3.5
PRC	r10	23	*23.0	23.0	23.0
	r30	9	*9.0	9.0	9.0
	r50	7	9.0	*7.0	7.0
	r200	1	41.0	*1.0	1.0
TSP	ulysses16	6859	6946.6	*6933.5	7072.7
	eil51	426	595.7	*591.9	599.7
	st70	675	1171.5	1187.5	*1164.5
VRP	eil22	380	386.7	*355.0	368.4
	eil30	554	614.9	611.3	*606.2
	eil51	624	*629.0	635.6	630.7

De modo general, los mejores valores son registrados por las estrategias propuestas, marcando diferencias significativas sobre los valores obtenidos por la inicialización aleatoria. El valor medio de aptitud permite concluir, que ambas propuestas crean una población con más posibilidades de evolucionar hacia el óptimo que la estrategia aleatoria. Esta última crea en muchos casos soluciones

no válidas, que afectan el comportamiento del algoritmo y requieren de un tratamiento especial en la función de aptitud. La principal ventaja de las inicializaciones propuestas es que reducen considerablemente la creación de falsas soluciones. Por último, las construcciones basadas en la proximidad y visibilidad entre puntos utilizada en ambas propuestas, crean de forma inherente buenos bloques constructivos.

4.2.1 Impacto en la Diversidad de la Población

Normalmente, la población inicial se crea de forma aleatoria. Esto permite crear soluciones que en principio tienen mala calidad, pero pudieran contener buenos bloques constructivos para el avance del algoritmo. La aleatoriedad de estas estrategias posibilita que la población sea muy diversa; o sea, que las soluciones se encuentren más dispersas por el espacio de búsqueda. Por su lado, las técnicas heurísticas crean las soluciones con cierta aleatoriedad atendiendo a criterios específicos. La incorporación de estas técnicas aumenta la eficacia de los algoritmos para tratar los problemas más difíciles. Sin embargo, su uso produce una disminución de la diversidad de la población y la convergencia temprana del algoritmo hacia un óptimo local.

La Tabla 3 muestra para cada una de las instancias de los problemas estudiados, los valores de diversidad luego de que se inicializa la población.

Tabla 3 Valor de desviación estándar luego la inicialización

<i>Problema</i>	<i>Instancia</i>	<i>AG</i>	<i>AG+IPP</i>	<i>AG+IDS</i>
PCM	r10	189.9	61.7	60.0
	r30	659.7	124.2	130.9
	r50	1086.1	191.6	211.0
	r200	4431.7	707.0	718.0
PRC	r10	46.6	24.6	942.7
	r30	126.1	60.3	631.2
	r50	96.9	77.0	722.4
	r200	652.4	133.5	215.6
TSP	ulysses16	1126.9	1167.3	1789.2
	eil51	88.2	96.6	98.1
	st70	180.7	197.4	195.3
VRP	eil22	65.4	69.7	69.4
	eil30	160.8	143.9	149.4
	eil51	84.9	99.7	99.5

En los dos primeros problemas que se tratan, la inicialización aleatoria como norma obtiene los valores más altos de diversidad. Sin embargo, los valores exorbitantes están provocados por las falsas

soluciones que esta estrategia introduce. Estos valores, lejos de mejorar el comportamiento del algoritmo son contraproducentes para el mismo, como se muestra en la Tabla 2. Para el resto de los problemas, los valores de diversidad se muestran más uniformes. A medida que aumenta la dificultad de los problemas, las heurísticas propuestas crean poblaciones con diversidad similar a las obtenidas por las poblaciones creadas de forma aleatoria. Estos valores de diversidad evidencian que las dos propuestas crean una población variada y sin la presencia de muchas falsas soluciones, lo que repercute positivamente en la efectividad del algoritmo.

Finalmente, la Figura 4.2-1 muestra la influencia de las estrategias de inicialización en el por ciento de éxito de los algoritmos.

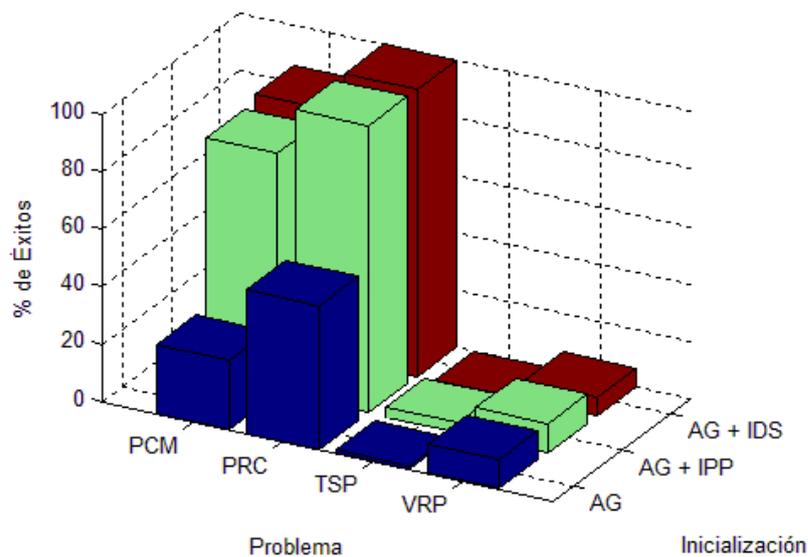


Figura 4.2-1 Por ciento de éxito para las estrategias de inicialización

A medida que aumenta la dificultad de los problemas, disminuye la eficacia de los algoritmos. Para el PCM y el PRC, las estrategias de IDS e IPP obtienen los por cientos más altos. En el resto de los problemas, la IPP es la estrategia de mayor eficacia. Así, la mejora introducida por las estrategias de inicialización propuestas en este trabajo, repercute de forma positiva en la evolución de las poblaciones de los algoritmos y su eficacia.

4.3 Estudio de los operadores CBC y VCU

Esta sección está encaminada al estudio de las estrategias de cruzamiento propuestas, en contraste con las más utilizadas para cada uno de los problemas. Para el PCM y el PRC, el cruzamiento de un

punto se considera como el básico. Por su lado, para el TSP y el VRP, se consideraron los operadores PMX (18) y la propuesta de (6) respectivamente.

La Tabla 4 resume los valores de aptitud medio para las instancias de problema que se estudian.

Tabla 4 Valor de aptitud medio para las estrategias de cruce

<i>Problema</i>	<i>Instancia</i>	<i>Óptimo</i>	<i>AG</i>	<i>AG+CBC</i>	<i>AG+VCU</i>
PCM	r10	23	23.3	23.3	*23.0
	r30	14	21.1	22.5	*16.2
	r50	7	24.7	33.5	*13.3
	r200	3	36.7	58.4	*18.1
PRC	r10	23	*23.0	23.0	23.0
	r30	9	*9.0	9.0	9.0
	r50	7	*9.0	9.0	9.0
	r200	1	*41.0	41.0	41.0
TSP	ulysses16	6859	6946.6	6952.7	*6899.9
	eil51	426	595.7	605.8	*558.8
	st70	675	*1171.5	1182.9	1263.1
VRP	eil22	380	*386.7	455.0	460.1
	eil30	554	*614.9	707.7	711.3
	eil51	624	*629.0	661.8	759.4

De forma general, se destacan los resultados obtenidos por el operador VCU con todos los problemas, exceptuando únicamente al VRP. Inicialmente, este operador traspasa el 50% de los genes de forma aleatoria directamente de padre a hijo, lo que produce una variación significativa en la solución. No obstante, el resto de los genes son transportados en el mismo orden en que aparecen en los padres contrarios, lo que posibilita la herencia de buenas características. El análisis del valor de aptitud medio destaca como la mejor estrategia a la VCU, porque a pesar de no resultar mejor para el último problema, en el resto marca diferencias significativas con las otras técnicas.

Un elemento fundamental que deben propiciar las estrategias de cruzamiento, es la explotación del espacio de búsqueda. El cruzamiento es un proceso esencial, ya que posibilita que los individuos exploten el espacio de búsqueda en sus inmediaciones, teniendo en cuenta las potencialidades que el resto de los individuos puedan haber descubierto. Posibilita que cada solución no avance por su propia cuenta, sino por la interacción con el resto. El resultado esperado es que exista una transferencia de información entre los candidatos prósperos, que potencie producir buenos descendientes conservando las virtudes de ambos padres.

La Figura 4.3-1 muestra el avance del valor de aptitud medio para el TSP.

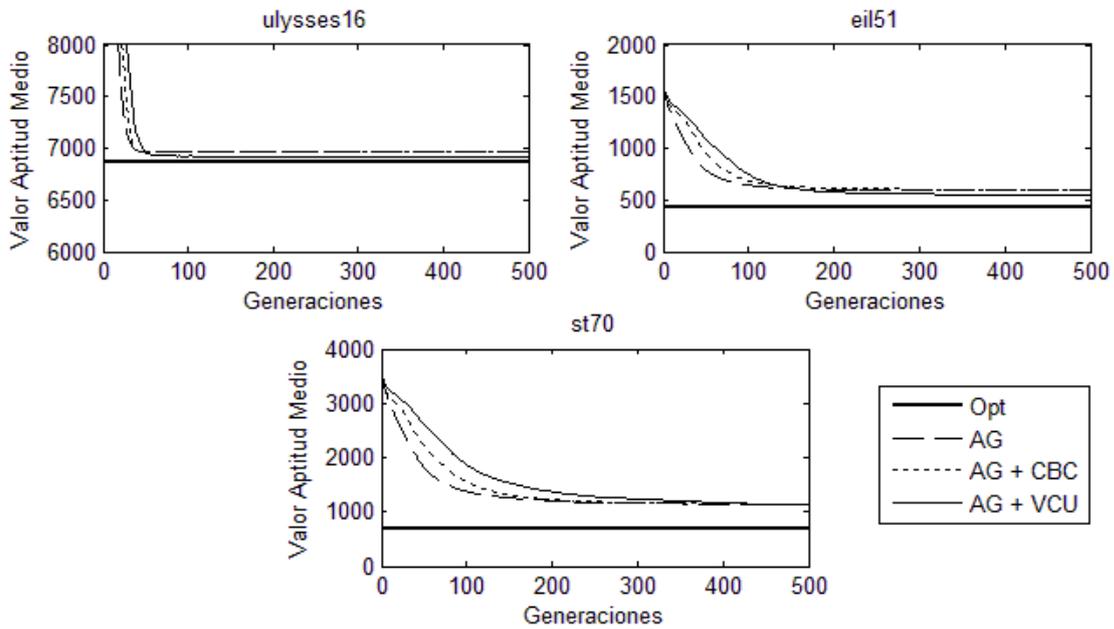


Figura 4.3-1 Valor de aptitud medio para las estrategias de cruce para el TSP

El operador VCU comienza siempre con valores de aptitud medio superiores al resto de los operadores. Sin embargo, en un promedio de 150 generaciones mejora los valores obtenidos por los operadores básicos y CBC. Este avance más lento sugiere que este operador explota mejor las soluciones vecinas a las más prometedoras, lo que repercute positivamente en la efectividad (ver Figura 4.3-2).

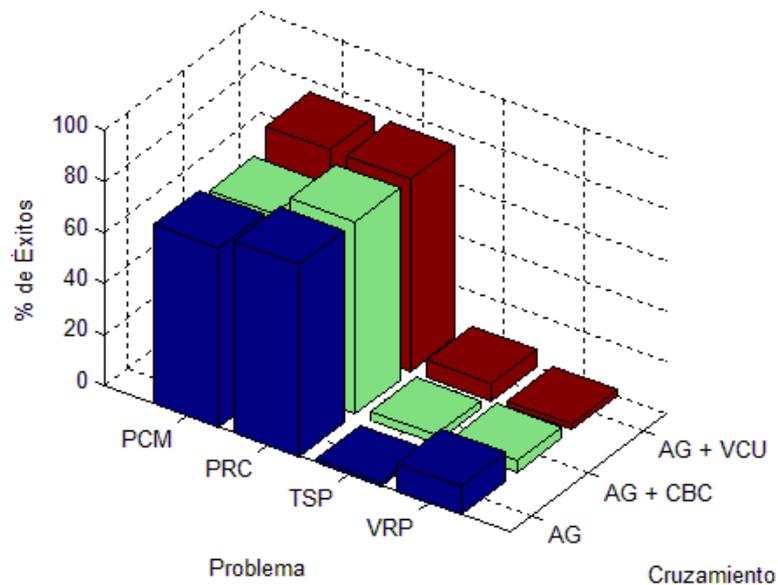


Figura 4.3-2 Por ciento de éxito para las estrategias de cruce

Esta gráfica muestra que la estrategia VCU, obtiene los mejores por cientos para los tres primeros problemas, exceptuando únicamente el VRP. Este comportamiento se justifica porque realiza una mejor revisión de las soluciones vecinas.

4.4 Estudio de los operadores VMI y MRB.

En esta última sección se realiza un análisis de las estrategias de mutación propuestas, en contraste con la mutación Inversión.

La Tabla 5 resume los valores de aptitud medio para los problemas estudiados.

Tabla 5 Valor de aptitud medio para las estrategias de mutación

<i>Problema</i>	<i>Instancia</i>	<i>Óptimo</i>	<i>AG</i>	<i>AG+VMI</i>	<i>AG+MRB</i>
PCM	r10	23	23.3	*23.0	23.0
	r30	14	21.1	*17.7	19.5
	r50	7	24.7	*24.6	25.3
	r200	3	36.7	47.5	*31.5
PRC	r10	23	*23.0	23.0	23.0
	r30	9	*9.0	9.0	9.0
	r50	7	*9.0	9.0	9.0
	r200	1	*41.0	41.0	41.0
TSP	ulysses16	6859	6946.6	*6882.3	6962.8
	eil51	426	595.7	*471.7	568.9
	st70	675	1171.5	*770.8	1034.0
VRP	eil22	380	*386.7	410.7	438.5
	eil30	554	*614.9	663.7	709.6
	eil51	624	*629.0	638.8	739.1

Los mejores valores son registrados por el operador de mutación es VMI, exceptuando igualmente al VRP. Este comportamiento está provocado porque este operador realiza una inversión que posibilite además de alterar la solución, una mejora en la aptitud. La Figura 4.4-1, es una muestra del comportamiento del valor de aptitud medio para el TSP. Se observa que a medida que aumenta el tamaño de las instancias de problema, los operadores MI y MRB, se distancian más del óptimo. No sucede lo mismo con el operador VMI, que alcanza el óptimo en la primera instancia y para el resto registra valores muy cercanos al óptimo.

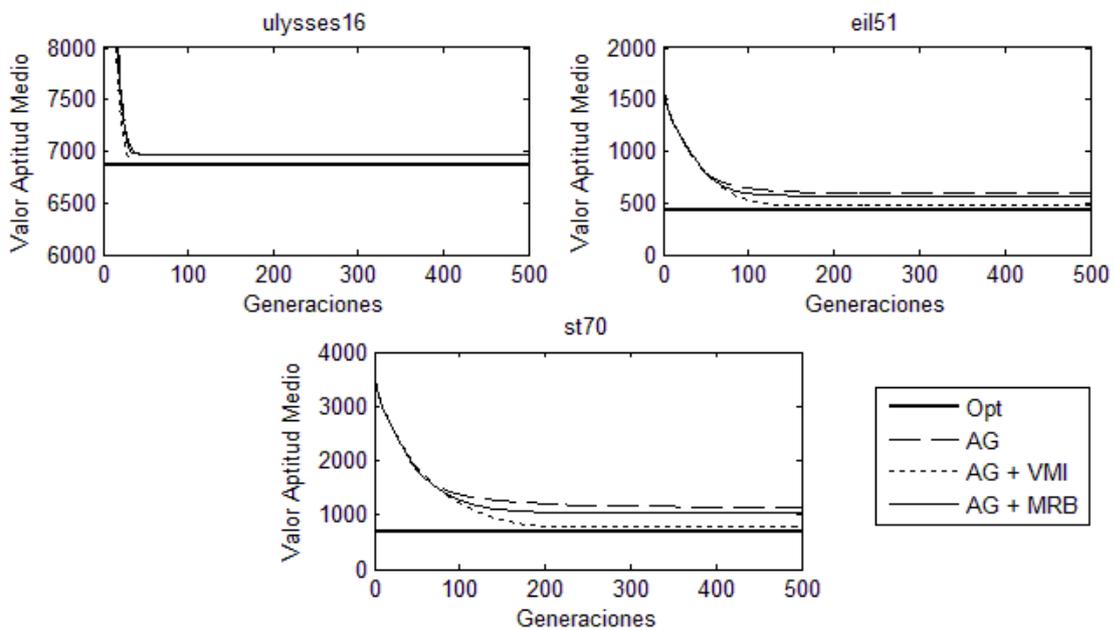


Figura 4.4-1 Valor de aptitud medio para las estrategias de cruce para el TSP

Uno de los elementos fundamentales que debe propiciar un operador de mutación, es la exploración del espacio de soluciones. Estos operadores deben posibilitar que la mayor parte de las áreas del espacio de búsqueda sean exploradas en algún momento. Esta exploración es determinada por medio de la diversidad. A medida que la población de soluciones sea más semejante la diversidad es menor.

La Tabla 6 muestra un análisis de la diversidad media de los algoritmos.

Tabla 6 Valor de desviación estándar media de las estrategias de mutación

<i>Problema</i>	<i>Instancia</i>	<i>AG</i>	<i>AG+VMI</i>	<i>AG+MRB</i>
PCM	r10	1.0	1.0	1.0
	r30	3.2	3.1	3.1
	r50	6.0	5.2	5.0
	r200	22.0	20.6	21.0
PRC	r10	0.1	0.1	0.1
	r30	0.4	0.4	0.4
	r50	0.3	0.3	0.3
	r200	2.4	2.3	2.4
TSP	ulysses16	142.2	27.7	32.1
	eil51	9.9	7.0	6.1
	st70	19.0	17.9	14.6
VRP	eil22	11.9	8.9	10.9
	eil30	21.8	17.2	20.9
	eil51	14.9	12.2	15.3

Los valores de diversidad son uniformes para cada uno de los problemas. Esto sugiere, que los operadores propuestos propician en buena medida (teniendo como referencia al operador MI) la exploración de nuevas áreas de soluciones.

Finalmente, la propuesta de AG utiliza la IPP, el operador de cruzamiento VCU y de mutación VMI. La Tabla 7, resume los valores de aptitud medio para cada una de las instancias de POR, contrastando el AG básico con la propuesta final de AG.

Tabla 7 Valor de aptitud medio para las propuestas finales de AG

<i>Problema</i>	<i>Instancia</i>	<i>Óptimo</i>	<i>AG</i>	<i>AG+IPP+VCU+VMI</i>
PCM	r10	23	23.3	*23.0
	r30	14	21.1	*14.5
	r50	7	24.7	*7.1
	r200	3	36.7	*3.0
PRC	r10	23	*23.0	23.0
	r30	9	*9.0	9.0
	r50	7	9.0	*7.0
	r200	1	41.0	*1.0
TSP	ulysses16	6859	6946.6	*6925.5
	eil51	426	595.7	*535.7
	st70	675	*1171.5	1274.9
VRP	eil22	380	*386.7	446.1
	eil30	554	*614.9	689.2
	eil51	624	*629.0	778.2

El AG propuesto obtiene los mejores valores para casi todas las instancias, exceptuando casos en que tiene resultados sin diferencias significativas del AG básico. Un caso especial ocurre con el VRP, siendo el AG básico el que registra los mejores valores, con diferencias significativas del AG propuesto. Sin embargo, la propuesta de AG realizada tiene como norma un mejor comportamiento. Lo que permite concluir que las estrategias propuestas de forma general para los POR, funcionan mejor que aquellas que han sido diseñadas específicamente para cada problema.

4.5 Conclusiones

Con respecto a las estrategias de inicialización propuestas, ambas obtienen mejores resultados que la inicialización aleatoria. Este comportamiento está justificado porque estas estrategias reducen considerablemente la creación de falsas soluciones. Además, la creación basada en la visibilidad y proximidad entre puntos permite crear buenos bloques constructivos, que luego serán aprovechados por las estrategias de cruzamiento. Por su lado, de los cruzamientos propuestos, CBC y VCU, el último

es el que obtiene los mejores resultados. Su funcionamiento permite que perduren buenas características de los padres, lo que garantiza la explotación del espacio de soluciones. Con respecto a los operadores de mutación, VMI y MRB, el primero de ellos destaca como la mejor estrategia. Básicamente, porque la inversión repercute en una mejoría de la aptitud de la solución y a largo plazo mejora la exploración del espacio de soluciones. Para las pruebas realizadas y dentro del rango de instancias elegidas, se ha comprobado que la IPP e IDS y los operadores VCU y VMI, obtienen mejores resultados que los operadores básicos verificados. De esta forma, los algoritmos propuestos son viables para su utilización en el análisis de rutas en GeoQ.

Conclusiones Generales

La resolución de POR tiene un alto impacto en la sociedad debido a la reducción de costos que ello implica. Los AG son una de las técnicas más utilizadas para solucionar de forma eficiente instancias reales de estos problemas. En esta investigación se propone la incorporación de métodos basados en AG para resolver POR en los SIG, específicamente GeoQ, haciendo uso de conceptos y capacidades propias de estos sistemas. Las aportaciones fundamentales se enuncian a continuación:

- Se caracterizan los POR de forma estructurada y unificada, proponiendo una definición general como un problema de optimización, lo que permite plantear cada problema atendiendo a sus restricciones específicas.
- Se caracterizan las funcionalidades de los SIG que permiten una fácil obtención y manipulación de los datos espaciales asociados a los POR.
- Se proponen nuevos operadores para la inicialización de la población, el cruzamiento y la mutación. Ellos explotan los conceptos de visibilidad y proximidad entre puntos y las funciones de análisis asociadas a los SIG, específicamente de GeoQ. Para casi la totalidad de instancias de POR resueltas, las propuestas mejoran la eficacia de los AG básicos.
- Con la selección de los mejores operadores propuestos, se implementa una componente para el análisis de rutas en GeoQ. La arquitectura base de este componente se fundamenta en la definición unificada de POR para lograr alto grado de reusabilidad.

Recomendaciones

Durante la investigación fueron identificadas posibles mejoras para obtener la matriz de costo y realizar el análisis de visibilidad, por lo que se recomienda:

- Optimizar la matriz de costo mediante el uso de listas enlazadas y de la reducción de grafos.
- Realizar el análisis de vecinos utilizando Grafos de Visibilidad.
- Propiciar la interacción del AG con el SIG, por medio de consultas dinámicas a funciones de identificación e intersección de objetos.

Bibliografía

1. **Valero, Francisco Luna.** *Metaheurísticas avanzadas para problemas reales en redes de telecomunicaciones.* Málaga : s.n., 2008.
2. **Goldberg, DE.** *Genetic Algorithm in search, optimization and machine learning.* s.l. : Addison-Wesley Publishing Co. Inc., 1989.
3. **Behzadi, S. and Alesheikh, Ali A.** *A PSEUDO GENETIC ALGORITHM FOR SOLVING BEST PATH PROBLEM.* Tehran : The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2008.
4. **Wadhwa, Vaishali and Garg, Deepak.** *Facility Location Problem Using Genetic Algorithm: A Review.* Patiala : Research Journal of Computer Systems Engineering- An International Journal, 2011. 2230-8563.
5. **Bryant, Kylie.** *Genetic Algorithms and the Traveling Salesman Problem.* s.l. : Harvey Mudd, 2000.
6. **Villarreal, Nydia Paola Rondón.** *Algoritmos genéticos en la solución de problemas de ruteo de vehículos con capacidad CVRP.* Colombia : s.n.
7. **González, Enrique López and Cuervo, Cristina Mendaña.** *NUEVAS TECNOLOGÍAS PARA EL MARKETING PÚBLICO: MINERÍA DE DATOS Y TÉCNICAS BIO-INSPIRADAS.* León : Universidad de León. E-24071.
8. **WU, Qishi and SHAN, Jeffrey J.** *THE APPLICATION OF GENETIC ALGORITHM IN GIS NETWORK ANALYSIS.* Amsterdam : International Archives of Photogrammetry and Remote Sensing, 2000. 47907-1284.
9. **pgRouting Project.** pgRouting Project. *Open Source Routing Library.* [Online] [Cited: 10 28, 2011.] <http://www.pgrouting.org/>.
10. **Na, Zhao.** *AN IMPROVED ALGORITHM OF THE SHORTEST PATH SEARCH PROBLEM IN GIS FIELD.* Taiyuan : American Journal of Engineering and Technology Research, 2011. 030012.
11. **Ahuja, Ravindra K., et al., et al.** *Faster Algorithms for the Shortest Path Problem.* s.l. : ACM, 1990. 0004-5411.
12. **Gonen, Bilal.** *Genetic Algorithm Finding the Shortest Path in Networks.* Reno : University of Nevada. 89502.
13. **Ahn, Chang Wook and Ramakrishna, R. S.** *A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations.* s.l. : IEEE, 2002. 1089-778X.
14. **Zhan, F. Benjamin.** *Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures.* Texas : Journal of Geographic Information and Decision Analysis, 1997.
15. **Ramachandran, Naveen.** *Accident Emergency Response And Routing Software (AERARS) using Genetic Algorithm.* Tamil Nadu : International Journal on Computer Science and Engineering (IJCSSE), 2011. 0975-3397.

16. **Boffey, B, Yates, D and Galvao, RD.** *An algorithm to locate perinatal facilities in the municipality of Rio de Janeiro.* Rio de Janeiro : Journal of the Operational Research Society, 2003. 0160-5682/03.
17. **Jaramillo, Jorge H., Bhadury, Joy and Batta, Rajan.** *On The Use of Genetic Algorithms to Solve Location Problems.* s.l. : Computers and Operations Research, 2002. 761-779.
18. **Acuña, Jorge.** *ESTUDIO EXPERIMENTAL CON DIFERENTES OPERADORES DE CRUCE PARA LA RESOLUCIÓN DEL PROBLEMA DEL VIAJANTE.* s.l. : ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA, 2010.
19. **Larsen, Jesper and Clausen, Jens.** *Branch and Cut for TSP.* Denmark : Informatics and Mathematical Modelling.
20. **Holstein, Diana.** *Una Metaheurística Co-evolutiva para el Problema del Viajante de Comercio.*
21. **Mercado, Roger Z. Ríos and Velarde, Jose Luis Gonzáles.** *Heurísticas para la solución del TSP.* Monterrey : Ingenierías, 2000.
22. *Heuristics for the Traveling Salesman Problem.* **Nilsson, Christian.** s.l. : Linkoping University.
23. **Karova, Milena, Smarkov, Vassil and Penev, Stoyan.** *Genetic operators crossover and mutation in solving the TSP problem.* s.l. : International Conference on Computer Systems and Technologies, 2005.
24. **Soto, Daniel, Soto, Wilson and Pinzón, Yoan.** *Una metaheurística híbrida aplicada a un problema de planificación de rutas.* Medellín : Revista Avances en Sistemas Informáticos, 2008. 1657-7663.
25. **Augerat, P., et al., et al.** *Computational results with a branch and cut code for the capacitated vehicle routing problem.* Grenoble : ARTEMIS-IMAG, 1995. RR949-M.
26. **Baldacci, R., Hadjiconstantinou, E.A. and Mingozi, A.** *An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation.* 2004.
27. **Lysgaard, J., Letchford, A.N. and Eglese, R.W.** *A new branch-and-cut algorithm for the capacitated vehicle routing problem.* s.l. : Math. Program, 2004.
28. **Baldacci, R., Christofides, N. and Mingozi, A.** *An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts.* s.l. : Mathematical Programming, 2007.
29. **Bräysy, Olli.** *Genetic Algorithms for the Vehicle Routing Problem with Time Windows.* Vaasa : Special issue on Bioinformatics and Genetic Algorithms, 2001. FIN-65101.
30. **Bjarnadóttir, Áslaug Sóley.** *Solving the Vehicle Routing Problem with Genetic Algorithms.* Denmark : Technical University of Denmark, 2004.
31. **Dorronsoro, Bernabé, et al., et al.** *Un Algoritmo Genético Híbrido Paralelo para Instancias Complejas del Problema VRP.* Málaga : Universidad de Málaga.
32. **Barroso, Alfonso Rubio and Puebla, Javier Gutierréz.** *LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICOS: ORIGEN Y PERSPECTIVAS.* Madrid : Revista General de Información y Documentación, Universidad Complutense, 1997.
33. ArcGIS Network Analyst | Overview. [Online] ESRI. [Cited: 11 15, 2011.] <http://www.esri.com/software/arcgis/extensions/networkanalyst/index.html>.
34. Portal gvSIG — gvSIG. [Online] [Cited: 09 15, 2011.] <http://www.gvsig.org/web/>.
35. **Margalef, Júlia Gilavert and Polo, Càrol Puig.** *Estudio comparativo de herramientas SIG Libres aplicadas a contextos de cooperación al desarrollo.* Girona : II JORNADAS DE SIG LIBRE.

36. **Equipo Desarrollo Quantum GIS.** *Quantum GIS- Guía de usuario.* 2009.
37. **Alvarado, Lidia Ortega.** Geometría Computacional- Visibilidad, Planificación de Trayectorias y Colisiones. [Online] Enero 1, 2010. [Cited: Diciembre 1, 2011.] <http://wwwdi.ujaen.es/asignaturas/gc/tema7.odp>.
38. —. Geometría Computacional - El Diagrama de Voronoi. [Online] Noviembre 14, 2011. [Cited: Diciembre 5, 2012.] <http://wwwdi.ujaen.es/asignaturas/gc/tema5.odp>.
39. **Rodríguez-Piñero, Piedad Tolmos.** *Introducción a los algoritmos genéticos y sus aplicaciones.* Madrid : s.n.
40. **Holland, John.** *Adaptation in Natural and Artificial Systems.* Michigan : University of Michigan Press, 1975.
41. **Ramachandran, Naveen and Devi, G.** *Accident Emergency Response And Routing Software (AERARS) using Genetic Algorithm.* Tamil Nadu : Naveen Ramachandran et al. / International Journal on Computer Science and Engineering, 2011. 0975-3397.
42. **KHAN, FOZIA HANIF, et al., et al.** *SOLVING TSP PROBLEM BY USING GENETIC ALGORITHM.* s.l. : International Journal of Basic & Applied Sciences. 96110-1414.
43. **Pereira, Francisco B., et al., et al.** *GVR: a New Genetic Representation for the Vehicle Routing Problem.* Portugal : Centro de Informática e Sistemas da Universidade de Coimbra.
44. **Kusetogullari, H, Leeson, M S and Hines, E L.** *Genetic Algorithm Approach to Solve the Shortest Disjoint Path Problem for Optimized Rainbow Network Flow of Multiple Description Codes.* Warwick : London Communications Symposium, 2009.
45. *New Genetic Local Search Operators for the Traveling Salesman.* **Freisleben, Bernd and Merz, Peter.** Proceedings of the 4th Conference on Parallel Problem Solving from Nature - PPSN IV : Springer, 1996, Vol. 1141. 890-899.
46. *A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems.* **Freisleben, Bernd and Merz, Peter.** Nagoya : Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, 1996. 616-621.
47. **May, Ming-Der.** *APPLYING GENETIC ALGORITHM ON SELECTING EMERGENT MEDICAL STATION BEFORE DISASTERS.* Taiwan : LungHwa Institute of Technology.
48. *Blending Heuristics with a Population-Based Approach: A "Memetic" Algorithm for the Traveling Salesman Problem.* **Moscato, Pablo and Tinetti, Fernando.** 1992.
49. **Alba, Enrique and Troya, José M.** *Improving flexibility and efficiency by adding parallelism to genetic algorithms.* Málaga : Kluwer Academic Publishers, 2002. 0960-3174.
50. **Reinelt, Gerhard.** TSPLIB. [Online] Universität Heidelberg, Agosto 6, 2008. [Cited: Febrero 13, 2012.] <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.

Glosario de Términos

A-Star: También llamado “A Estrella”, es un algoritmo de búsqueda en grafos. Representado por Peter E. Hart, Nils J. Nilsson y Bertram Rápale en 1968. Este algoritmo es el único que garantiza, sea cual sea la función heurística, que se tenga en cuenta el camino recorrido.

Algoritmo Genético Paralelo: Algoritmo genético donde el paralelismo reside en que la ejecución de algunas operaciones se solapa en el tiempo. Existen diversos modelos de paralelismo: grano grueso, grano fino, paralelismo global, mixto.

AP (Alternating Position Crossover): Operador genético de cruce que crea un descendiente alternado los genes de los padres. Para ello se introduce alternativamente un gen de cada progenitor hasta haber completado el descendiente. Si el gen a introducir ya se encuentra en el descendiente, se pasa al siguiente del mismo padre y así se evita tener genes repetidos. Por ejemplo, a partir de (2 6 4 1 3 5) y (2 1 6 5 4 3) se puede generar (2 6 1 4 5 3).

BOC (Biggest Overlap Crossover): Operador genético de cruce que utiliza la geografía de la solución. Consiste en crear una copia del primer padre en el descendiente y seleccionar una sub-ruta del segundo padre. Luego, se calcula el cuadro delimitador de la selección y sus miembros son eliminados del primer padre para prevenir duplicados. Después, comparando el cuadro delimitador de la sub-ruta con el de cada ruta en el primer padre se determinan las tres rutas más cercanas. Por último, la sub-ruta es insertada en la ruta con menor demanda de la forma menos costosa posible.

Convergencia prematura: Problema que presentan ciertos métodos de búsqueda consistente en que el algoritmo tiende hacia una solución única que no es la óptima.

Cuadro delimitador: Cuadrado más pequeño que contiene a todos los elementos de una ruta.

CX (Cycle Crossover): Operador genético de cruce que crea dos descendientes seleccionando cada elemento de uno de los padres y colocándolo en la posición en que estaba anteriormente. Por ejemplo, a partir de (1 2 3 4 5 6 7 8) y (8 5 2 1 3 6 4 7) se selecciona la primera posición del primer padre se generan (1 5 2 4 3 6 7 8) y (8 2 3 1 5 6 4 7).

Desigualdad del triángulo: Especifica que para viajar de una ciudad a otra es más corto hacerlo directamente que pasando por una ciudad intermedia. Formalmente, tiene que cumplirse que la distancia de i a j para cualquier par de puntos debe ser menor o igual que la distancia de i a k más la distancia de k a j , para cualquier otro punto k .

Dijkstra: También llamado algoritmo de caminos mínimos, es un algoritmo para determinar el camino más corto dado un vértice origen al resto de vértices en un grafo con pesos en cada arista. Su nombre se refiere a Edsger Dijkstra, quien lo describió por primera vez en 1959.

DPX (Distance Preserving Crossover): Operador genético de cruce que consiste en incorporar en los descendientes los arcos comunes a ambos padres y luego, unir los fragmentos utilizando arcos no pertenecientes a ninguno de los padres. Por ejemplo, a partir de (1 2 3 4 5) y (1 2 5 4 3) se genera (1 2 5 3 4).

HLC (Horizontal Line Crossover): Operador genético de cruce que combina a la par genes de ambos padres. Consiste en trazar una línea horizontal a través del depósito que divide a los progenitores en dos partes. Los cuadros delimitadores son utilizados para identificar las rutas. Las rutas del primer padre que tienen cuadros delimitadores por encima de la línea trazada son movidas para la primera parte del descendiente. Luego, el mismo se completa con las rutas del segundo padre que tienen cuadros delimitadores por debajo de la línea.

k-Opt: Heurística de mejoramiento iterativo. Dada una ruta, se tratará de mejorarla eliminando k arcos y reemplazándolos por un nuevo conjunto de k arcos que reconectan la ruta. A medida que se usa un mayor k , el mínimo local es mejor pero el costo computacional aumenta. Por esta razón, usualmente se utilizan las movidas 2-Opt y 3-Opt.

MOX (Modified Order Crossover): Operador genético de cruce que funciona de forma similar al OX. Consiste en seleccionar un punto de cruce en los padres y los genes que se encuentran antes de este punto se quedan iguales. Luego, se reordenan los genes que se encuentran después del punto de cruce en el orden en que aparecen en el segundo padre. Por ejemplo, a partir de (1 2 3 | 4 5 6) y (3 6 4 | 2 1 5) elegiría un punto de cruce (marcado con barras) y generaría los hijos (1 2 3 | 6 4 5) y (3 6 4 | 1 2 5).

OR-Opt: Operador que selecciona un conjunto de genes y chequea si pueden ser insertados entre otros genes conectados, en busca de una reducción del costo de la ruta. En caso de que se produzca una disminución del costo, se ejecuta el cambio.

OX (Order Crossover): Operador genético de cruce que genera dos descendientes eligiendo para cada uno de ellos un sub-recorrido de uno de los progenitores y preservando el orden relativo de los restantes valores presentes en el otro progenitor. Explota la propiedad de que el orden de los elementos (no su posición) es importante. Por ejemplo, a partir de (1 2 3 | 4 5 6 7 | 8 9) y (4 5 2 | 1 8 7 6 | 9 3) elegiría dos puntos de cruce (marcados con barras) y generaría los hijos (2 1 8 4 5 6 7 9 3) y (3 4 5 1 8 7 6 9 2).

PMX (Partially Mapped Crossover): Operador genético de cruce que produce dos descendientes eligiendo un patrón de uno de los dos progenitores y preservando el orden y la posición de tantos valores como sea posible del otro progenitor (para cada uno de ellos). Explota las similitudes en el valor y el orden. Por ejemplo, a partir de (1 2 3 | 4 5 6 | 7 8) y (3 7 5 | 1 6 8 | 2 4) elegiría dos puntos de cruce (marcados con barras) y generaría los descendientes (4 2 3 1 6 8 7 5) y (3 7 8 4 5 6 2 1).

POS (Position Based Crossover): Operador genético de cruce que crea un descendiente copiando inicialmente los genes que se encuentran en un número de posiciones seleccionadas aleatoriamente. Después, se introducen el resto de los genes del otro padre, empezado desde el principio e introduciendo sólo los genes que no se encuentran en el descendiente. Por ejemplo, a partir de (2 5 1 3 4 6 8 7) y (2 3 1 5 8 7 4 6) se introducen las posiciones 2 5 y 7, para generar (2 3 5 1 8 6 4 7).

Ramificación y Corte (Branch and Cut): Técnica de propósito general que se basa en reglas de ramificación. Comprueba exhaustivamente las soluciones posibles, descartando en forma inteligente ciertas “direcciones de búsqueda”.

Torneo Binario: Estrategia de selección que consiste en seleccionar dos soluciones de la población de forma aleatoria. Por último, entre ellas se elige aquella de mayor aptitud.