No hay posibilidad de que un descubrimiento se deba tan solo a la casualidad. Necesariamente, la invención y el descubrimiento, ya sea en las matemáticas o en cualquier otro campo, tienen lugar a través de la combinación de las ideas.

J. Hadamard

5. Fundamentos del algoritmo genético como meta-heurística propuesta.

5.1 Introducción: breve historia del antecedente de los algoritmos genéticos.

La naturaleza ha sido siempre fuente de inspiración para el hombre. Comprender los complejos mecanismos por los cuales los seres vivos hemos alcanzado nuestra forma actual ha sido una de las tareas más ambiciosas del hombre, y sin lugar a dudas el trabajo realizado hace 140 años por el naturalista inglés Charles Darwin constituye una de las contribuciones más importantes en esta dirección.

El 1 de julio de 1858, Darwin presentó su controversial "Teoría de la Evolución Natural" ante la Sociedad Linnean de Londres, revolucionando las ciencias biológicas y cambiando incluso la filosofía del pensamiento de los seres humanos. Actualmente el paradigma neo-darwinista de la evolución sostiene que el mecanismo evolutivo de las especies e individuos está sustentado en cuatro procesos principales: reproducción, mutación, competencia y selección, todos frecuentemente resumidos en la frase "sobrevivencia del más apto y fuerte".

Alan Turing y Stanislaw Ulam se cuentan entre los científicos más célebres que pensaron en la evolución natural como el mecanismo que hizo posible el desarrollo de cualidades altamente complejas de las especies. Turing[59], argumentó que "una conexión obvia entre aprendizaje y evolución" debe de existir en los mecanismos de la cognición humana. En el Laboratorio de Los Alamos, Ulam y Von Neumann, modelaron en una computadora la velocidad con la cual las mutaciones favorables se esparcen entre los individuos de una población sujeta a los mecanismos de "sobre vivencia del más fuerte", concluyendo que "la reproducción sexual acelera tremendamente el ritmo de crecimiento de las mutaciones favorables, comparada con la reproducción asexual que progresa de manera lineal". [60]

[59] Turing, A. (1950), *Computer Machinery and Intelligence*, MIND, EUA, pag 433-462.

[60] Bednarek, A. (1984), *Analogies between Analogies: The mathematical reports of S.M. Ulam and his Los Alamos collaborators*, University of California Press, EUA.

En el ámbito Europeo (Alemania), probablemente uno de los pioneros más importantes de la computación evolutiva sea Hans Bremmerman. Su trabajo estuvo dirigido al estudio y uso de los principios de la evolución natural como mecanismos de optimización. Bremmerman no sólo utilizó los conceptos de "aptitud", "selección", "mutación", "población", y "genotipo", que muy pocos asociaban en aquella época con la resolución de problemas de optimización, sino que además afirmó que "la evolución biológica es un proceso de optimización"[61] .

Este enunciado que parece tan simple, es la filosofía que da soporte a los métodos que en conjunto se conocen ahora con el nombre de Algoritmos Genéticos. Así pues, resulta fácil comprender que si acaso hemos de considerar a la evolución natural como un proceso de optimización, dado que la evolución ha sido capaz de optimizar organismos hasta hacerlos aptos para sobrevivir, entonces, de modelarse este principio adecuadamente, podemos emplear el mismo principio para encontrar la mejor solución para un problema de optimización matemático.

[61] Bremermann, H. (1962), *Optimization through Evolution and Recombination*, Spartan Books, Alemania, pag 35-80.

5.2 La versión americana y la versión alemana del algoritmo genético.

Así pues, el concepto del Algoritmo Genético, se desarrolla a ambos lados del océano Atlántico a partir de la década de los 60's. El algoritmo correspondiente a la variante Alemana, con la intención de diferenciarlo hay quienes lo conocen como "Algoritmo basado en Estrategias Evolutivas" mientras que a la versión Americana simplemente se le denomina "Algoritmo Genético". A continuación en la Figura 5.1, se muestra un breve resumen de acontecimientos que tanto en Norte América así como en Alemania se suscitaron a lo largo de estos últimos 40 años de progreso en el desarrollo de los Algoritmos Genéticos:

	ALEMANIA	NORTE AMÉRICA
1962		John Holland publica un artículo sobre Sistemas Adaptativos en el Journal del ACM
1963	Los estudiantes Ingo Rechenberg y Hans-Paul Schwefel comienzan a trabajar juntos en el túnel de viento de la Universidad Tecnológica de Berlín.	
1973	Rechenberg publica su disertación que aborda la teoría (1+1) de las EEs y la mayor parte de la nomenclatura adoptada posteriormente por los investigadores en esta área; se describe la "regla de 1/5".	
1975	La disertación de Schwefel incluye modelos y experimentos para poblaciones multi-miembro (con más de dos individuos).	Holland publica libro sobre Sistemas Adaptivos que marca el principio del actual auge de los Algoritmos Genéticos
1977	Schwefel introduce los operadores de recombinación; aparece la primera edición de su libro sobre Estrategias Evolutivas	
1985		La Primera Conferencia Internacional sobre Algoritmos Genéticos se realiza en la Universidad Carnegie-Mellon
1989		David Goldberg publica su libro sobre Algoritmos Genéticos propiciando mayor interés en el tema
1995	Schwefel publica una revisión de su libro sobre EEs . Thomas Bäck introduce las EEs a una audiencia aún mayor en 1996 con un nuevo libro de computación evolutiva	

Figura 5.1 Acontecimientos históricos en el desarrollo de los algoritmos genéticos [62].

[62] Coello, Carlos. (1995), *Introducción a los Algoritmos Genéticos*, "Tecnologías de Información y Estrategias de Negocios, Año 3, No. 17", México, pag 4.

A continuación en la Figura 5.2, se muestran ambas variantes del Algoritmo Genético, tanto la versión Alemana a la izquierda como versión desarrollada en los Estados Unidos a la derecha. Como puede observarse en el diagrama lógico los operadores involucrados en el algoritmo genético son básicamente los mismos, lo único que cambia es el orden en que se llevan a cabo.

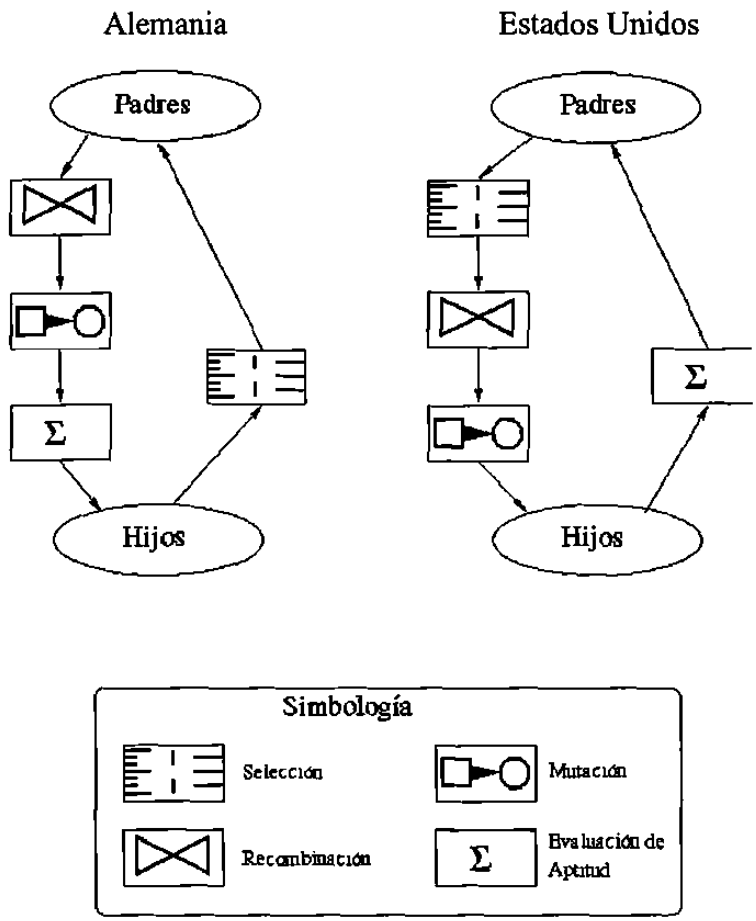


Figura 5.2 Variantes del algoritmo genético [63].

Aunque existen similitudes entre ambas versiones, vale la pena diferenciar que el Algoritmo Alemán basado en las Estrategias Evolutivas se especializa en *resolver problemas de optimización* de funciones reales (variables continuas), mientras que los Algoritmos Genéticos se concentran mucho más en la solución de problemas de optimización discreta. Otra diferencia notable es que para la implementación computacional de los operadores de selección, cruce y mutación en los Algoritmos Genéticos, el componente estocástico es mucho más intenso que en la versión alemana debido al grado de estructuración determinístico que suelen tener estos últimos.

[63] Hoffmeister, F. (1992), *Genetic Algorithms and Evolution Strategies: Similarities and Differences,* "Technical Report No.1", University of Dortmund, Alemania, pag 70-120.

Aunque originalmente la versión Alemana se diferenciaba de la versión Americana en que en la primera no solo se evoluciona la solución sino también los parámetros del algoritmo que propiamente llevan a converger en la solución; esta característica ha sido en los últimos años retomada por el Algoritmo Genético Americano con la finalidad de incorporarlo como parte de su estrategia en su implementación computacional.

Otra diferencia entre ambas versiones radica en que la versión Alemana, por enfocarse en la solución de problemas de optimización para variables continuas, ha dedicado mucho mas esfuerzo en lo relacionado a depurar la estrategia de guía (brújula) para mejorar su eficiencia durante el proceso de búsqueda del espacio solución en la fase de escalamiento. Para entender lo anterior, basta preguntarse: ¿qué debemos seleccionar, un organismo cerca de la solución pero mal orientado, o uno alejado de la solución pero correctamente orientado?

En la opinión del postulante, la principal diferencia entre la versión Alemana y la versión Americana estriba en que en la primera los organismos no solo están constituidos por su propio material genético (genotipo), sino que también poseen su propio control (fenotipo). Sería algo así como permitirle a cada organismo tener su propia personalidad y voluntad. De esto último entonces, resulta razonable preguntarse si esta doble característica no resulta excesiva para tan solo implementar el mecanismo de selección natural en los organismos de la población. Es precisamente este problema de doble constitución de la versión alemana lo que representa su talón de Aquiles en la práctica. Debemos puntualizar que en este asunto en la versión Americana, el ajuste de los parámetros se caracteriza no al nivel de cada organismo sino a nivel global de la población.

De cualquier manera es claro comprender que la versión Americana nos es más útil por enfocarse hacia problemas de optimización discreta que es exactamente hacia donde nos estamos dirigiendo, ya que nuestro problema de investigación está referido a un caso de ruteo de distribución en el área de logística.

5.3 Introducción a los parámetros de operación del algoritmo genético.

El Dr. John Holland, investigador de la Universidad de Michigan, consciente de la importancia que la selección natural ha tenido en el proceso de la evolución natural, a fines de la década de los 60s desarrolló una técnica que permitió simular dicho proceso biológico a través de un algoritmo computacional. Su objetivo original era lograr que las computadoras aprendieran por sí mismas [64]. Eventualmente la técnica que inventó Holland se llamó "Algoritmo Genético" tras la publicación de su libro en 1975. Los Algoritmos Genéticos son técnicas de búsqueda que permiten explorar de forma eficiente el espacio solución de un problema. El proceso de búsqueda de la solución óptima esta basado en la teoría de la evolución de Darwin mediante el cual, los mecanismos de genética y de selección natural aplicados en el proceso evolutivo estimulan la formación de nuevos organismos mejor adaptados al medio ambiente.

Otra definición de lo que es un Algoritmo Genético se ofrece por John Koza (traducción libre):

> *"Se trata de un algoritmo matemático paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud."* [65]

En los últimos años, la comunidad científica internacional ha mostrado un creciente interés en esta nueva técnica la cual se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven al adaptarse más fácilmente a los cambios que se producen en su entorno. Los Algoritmos Genéticos gozan hoy en día de enorme popularidad en todo el mundo, debido a su generalidad y sencillez conceptual.

La aplicación más común de los algoritmos genéticos ha sido para la solución de problemas de

---

[64] Holland, John. (1975), *Adaptation in Natural and Artificial Systems*; University of Michigan Press, EUA, pag 211.
[65] Koza, John. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, EUA, pag 319.

optimización matemática, en donde han mostrado ser eficientes. Sin embargo, no todos los problemas pueden ser apropiados para esta técnica. A continuación se enumeran algunas características que el problema debe tener para aplicar un algoritmo genético de acuerdo a la empírica del postulante:

1. El espacio de búsqueda para el problema debe estar delimitado dentro de ciertos límites: Lo recomendable es aplicar el algoritmo para resolver problemas que tengan espacios de búsqueda discretos. Esta característica le ofrece naturalidad al algoritmo y le permite atender inclusive espacios de solución inclusive grandes. Su aplicación al tratarse de espacios de búsqueda continuos resulta menos eficiente por lo cual preferentemente se limita a casos donde exista un rango de soluciones relativamente pequeño.

2. Es necesario definir una función de aptitud que nos indique qué tan buena o mala es cada característica revisada en el organismo. La función de aptitud no es otra cosa que la función objetivo del problema de optimización matemático. La función de aptitud debe ser capaz de "castigar" a las malas soluciones (organismos), y de "premiar" a las buenas, de forma que sean estas últimas las que se propaguen con mayor rapidez en la población.

3. Las soluciones para el problema deben ser codificadas de una forma que resulte fácil su implementación algorítmica. La codificación más común es a través de cadenas binarias, aunque también se han empleado otras codificaciones basadas en números reales y letras. La codificación binaria ha gozado de mucha popularidad debido a que es la técnica que propuso originalmente Holland, y además porque resulta muy sencilla de implementar.

Existen diversos operadores que se utilizan en los algoritmos genéticos para guiar el proceso evolutivo y lograr la generación de nuevas poblaciones de individuos a partir de los existentes. A continuación se describen los operadores que el postulante implementará en el algoritmo genético que será aplicado como propuesta de solución para el problema de investigación:

1. Población y su función de desempeño: se parte de la existencia de una población inicial, que dará paso a nuevas generaciones a través de la ejecución de un proceso similar a lo que ocurre en la naturaleza.

2. Cromosomas y genes: son soluciones potenciales, que van formando la población de organismos. Los cromosomas se constituyen a su vez por genes los cuales contienen la información genética codificada del organismo.
3. Selección: competencia para realizar el cruzamiento de organismos.
4. Cruzamiento o Reproducción: generación de nuevos organismos a partir de la combinación del contenido genético de dos organismos antecesores que ya existan en la población.
5. Mutación: alteración del contenido genético de nuevos organismos sin que intervengan los padres, a partir de pequeños cambios en uno o más genes de un determinado cromosoma.

Aunque todos los parámetros antes expuestos son importantes, en la Figura 5.3, se resaltan dos de ellos de manera particular: la selección natural y el cruce o reproducción.

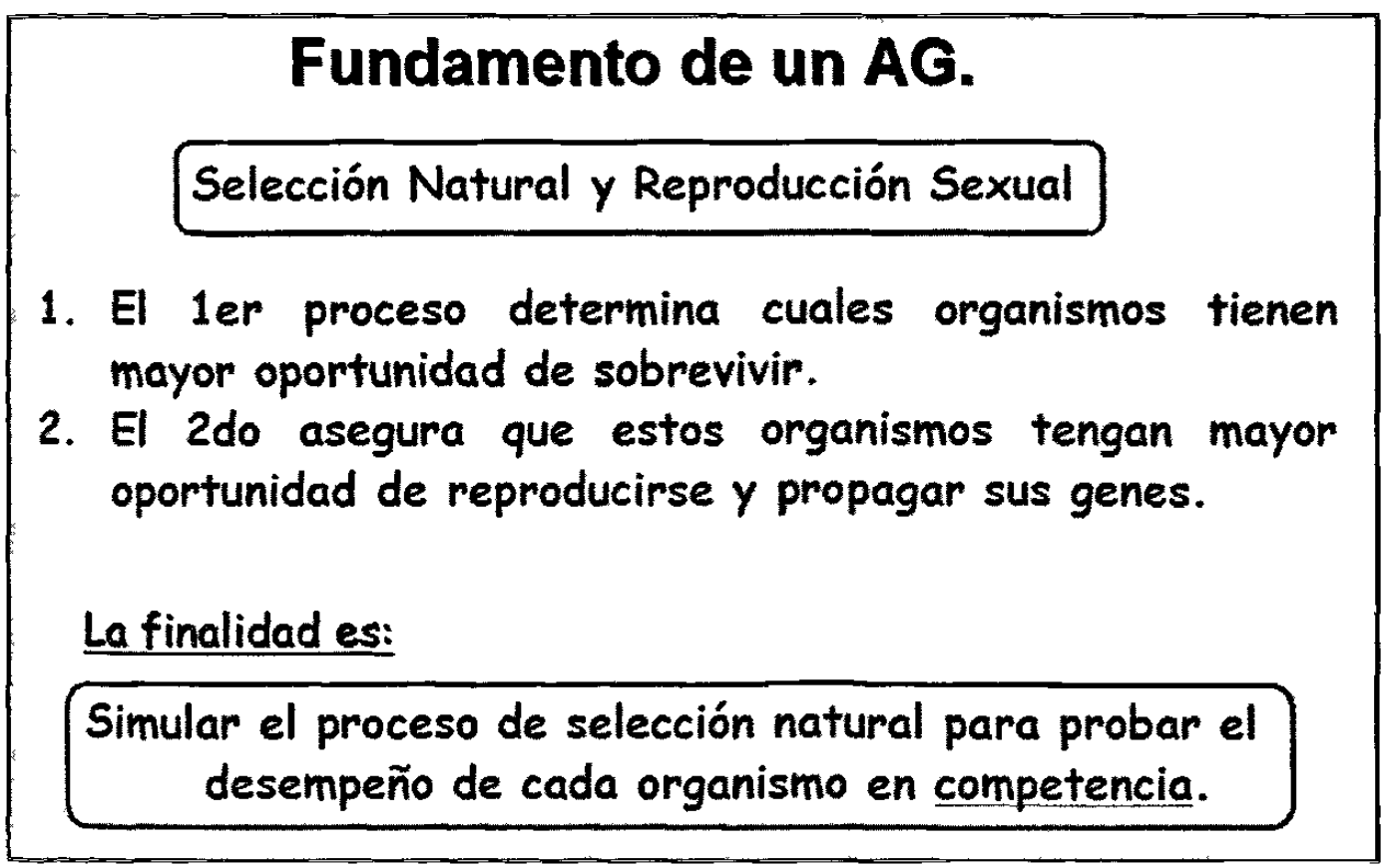


Figura 5.3 Fundamento del algoritmo genético.

En los siguientes puntos trataremos de manera específica cada uno de estos parámetros para su mejor comprensión.

5.4 La población de organismos y la función de desempeño.

Los componentes de la población son denominados organismos o individuos. La estructura de los organismos es determinada a priori y es la misma para toda la población. La estructura precisa de los organismos es dependiente del dominio del problema, lo cual implica que para cada problema tiene que idearse una representación adecuada del organismo (genoma). No obstante lo anterior, la operación del Algoritmo Genético es independiente del dominio del problema.

El tamaño de la Población "P", representa ser uno de los parámetros más importantes del Algoritmo Genético, esto es la representación del conjunto de soluciones potenciales para el problema. El tamaño de la población puede variar a lo largo de las generaciones del Algoritmo, aunque usualmente dicho parámetro permanece estático.

Una característica que los problemas deben de tener, es la definición de una medida comparativa de las soluciones (organismos) que compiten. Debe de existir un mecanismo derivado del dominio del problema que nos permita asignar una medida de desempeño (o fittness) a cada uno de los organismos de la población y que éste realmente sea representativo de su calidad como solución. A este parámetro se le denomina medida o desempeño (o también aptitud). Un organismo con mayor aptitud representa una mejor solución para el problema, lo cual en las condiciones específicas de éste puede representar una solución correcta o inclusive la misma solución óptima para el problema.

Es evidente que lograr una alta correlación entre el valor de aptitud y la cercanía a la solución que cada individuo representa resulta finalmente crítico. Esto se propicia mediante la representación o codificación apropiada del problema, y por supuesto, mediante una función de aptitud que correctamente logre hacer este mapeo asignando un valor de calidad adecuado a cada organismo de acuerdo a su desempeño. Precisamente esto es lo que permite indicar qué tan buena es una solución con respecto al resto de la población.

5.5 La definición del cromosoma del organismo y sus genes.

Las unidades básicas de codificación que permiten distinguir a cada uno de los atributos de un ser vivo son lo que conforman los genes de un individuo. Son estos mismos genes los que permiten que los atributos más deseables se transmitan a sus descendientes al momento del acto de reproducción.

La principal característica de un Algoritmo Genético es que éste no requiere de información de dominio o conocimiento respecto al procedimiento para solucionar un problema. La justificación para el uso de representaciones binarias como esquema principal de codificación de los organismos en los Algoritmos Genéticos se remonta al trabajo desarrollado por Holland en 1974. La codificación binaria permite un mayor grado de paralelismo implícito en el algoritmo debido simplemente a la cantidad combinatoria de bloques esquemáticos que una representación binaria puede llegar a cubrir respecto a una representación por ejemplo del tipo decimal.

Un bloque esquemático o simplemente esquema, no es otra cosa que un patrón que se visualiza al revisar en varios organismos de la población, un subconjunto de genes que comparten similitudes a lo largo de la longitud cromosómica del genoma [66]. Es luego fácil comprender que al haber una mayor diversidad de esquemas a poder estar siendo permutados, entonces esto incrementa la diversidad y por consiguiente la probabilidad de que un buen bloque pueda llegar a generarse que devenga a su vez en la solución óptima para el problema.

Lo anterior, no significa que no puedan utilizarse codificaciones de mayor cardinalidad, es decir que sean diferentes a la codificación binaria. Lo único es estar consciente de que la codificación binaria finalmente será aquella que podrá ofrecer una mayor cantidad de esquemas posibles que cualquier otro mecanismo de codificación. Por tanto, entre mayor cantidad de genes esté representando a un genoma, mucho más sensibilidad tendrá el algoritmo para la formación de bloques constructores. Más adelante se dedicará un apartado para revisar más a detalle el tema de la codificación de los organismos (revisar el apartado 5.8).

---

[66] Goldberg, D. (1995), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Pub Co, University of Massachusetts, EUA, pag 45-87.

En la Figura 5.4, se ejemplifica aún más el concepto de codificación en el desarrollo de una representación binaria para el manejo de las soluciones en el Algoritmo Genético. En la figura se menciona como es que dentro del algoritmo genético es posible incorporar en el genoma de los organismos, ciertas propiedades y características que a priori se "estima", por parte del modelador, puedan ser favorables para la evolución de la población. A este componente del algoritmo genético, el postulante le nombra como el efecto de "La mano de Dios".

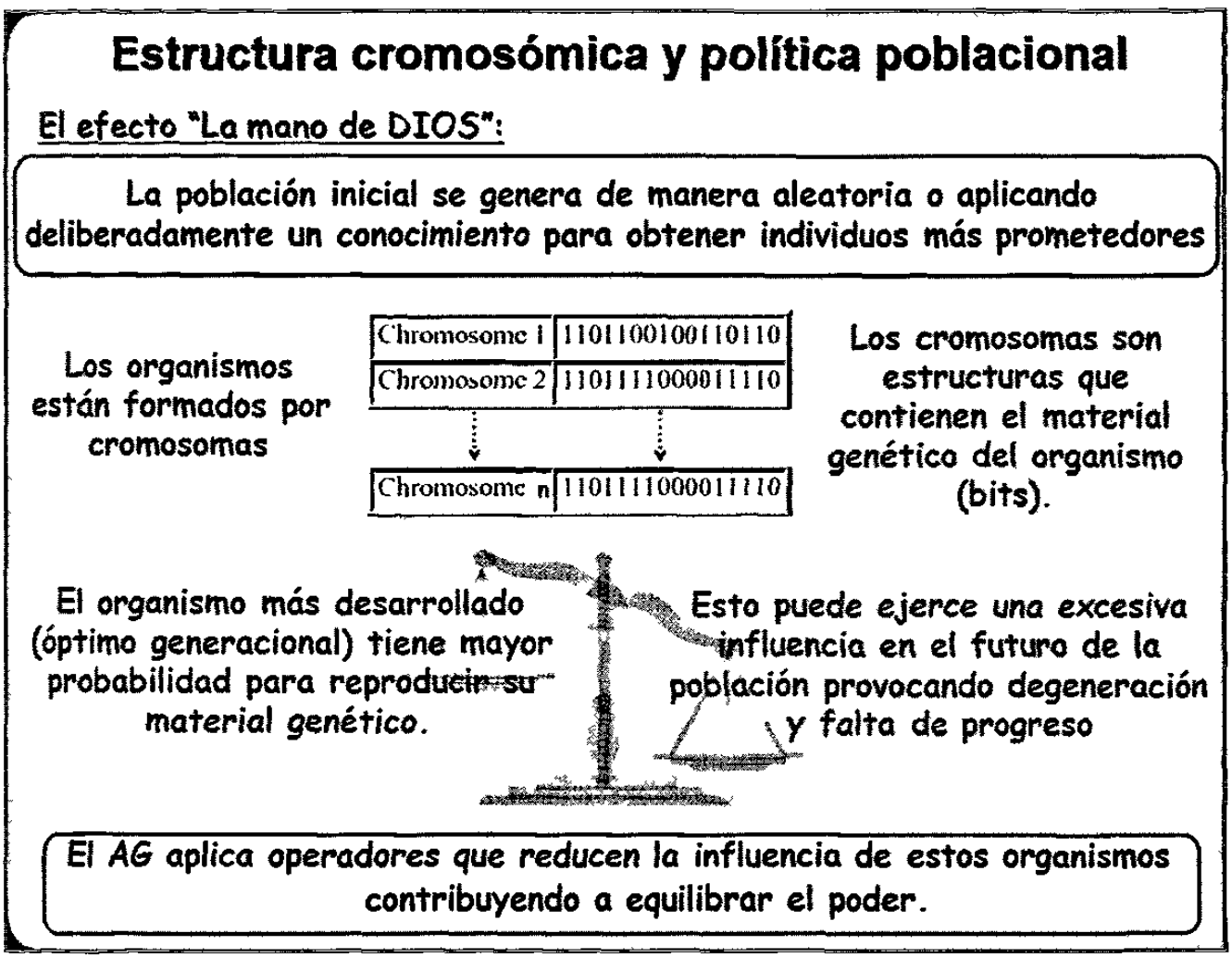


Figura 5.4 Desarrollo de la estructura cromosómica y política poblacional.

5.6 El operador de selección.

La selección es el operador que escoge preferentemente a los organismos con mayor aptitud de una población. El grado en el cual mejores valores de aptitud son preferidos sobre los más pobres se denomina presión de la selección. Existen dos métodos de selección que fueron aplicados en la implementación del Algoritmo Genético propuesto para determinar cuales de los organismos de la población son los que tendrán mayor probabilidad de descendencia: el de Ruleta y el de Torneo.

1. La Ruleta: fue desarrollado por Goldberg[67] , el método consiste en crear una ruleta en la que cada organismo tiene asignada una fracción del área de la ruleta en proporción a su grado de aptitud. Sin que nos refiramos a una *función de aptitud en particular, supongamos que se tiene una población de* 5 cromosomas cuyas aptitudes están dadas por los valores mostrados en la siguiente Tabla 5.1:

Tabla 5.1 Ejemplificación del método de selección de la ruleta

Cromosoma No.	Cadena	Aptitud	% del Total
1	11010110	254	24.5
2	10100111	47	4.5
3	00110110	457	44.1
4	01110010	194	18.7
5	11110010	85	8.2
Total		1037	100.0

Con los porcentajes mostrados en la cuarta columna de la Tabla 5.1, podemos pasar ahora a construir la ruleta de la Figura 5.5. Esta ruleta se gira una cantidad "n" de veces para determinar qué organismos se seleccionarán. La cantidad "n" esta determinada al menos por la cantidad de organismos que se desea mantener en la población. Debido a que a los organismos más aptos se les asignó un área mayor de la ruleta, se espera que sean seleccionados más veces que los menos aptos.

[67] Goldberg, David. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, EUA, pag 412.

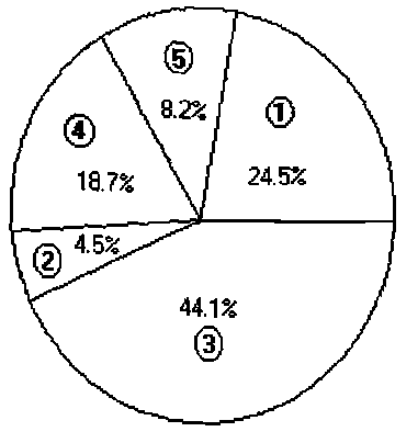


Figura 5.5 Ruleta que representa los valores de aptitud de la tabla 5.1

2. El Torneo: este método se basa en el sorteo de la población para luego hacer competir a los organismos que la integran en grupos de tamaño predefinido (normalmente en parejas). La finalidad de éste método es simular un torneo en el cual resultan ganadores aquéllos organismos que tengan valores de aptitud más altos. Esta técnica garantiza la obtención de múltiples copias de los mejores individuos de entre los progenitores de cada siguiente generación.

A continuación en la Figura 5.6, se explica de manera breve la forma como está íntimamente relacionada la función de desempeño de un organismo con respecto a la forma como actúa el operador de selección en el algoritmo genético. Es decir, entre mayor sea la función de desempeño de un organismo mayor probabilidad tendrá en ser seleccionado para el proceso de cruzamiento sexual que se explica en el siguiente apartado.

Operador de Selección

El material genético en los organismos determina la función de desempeño.

La función de desempeño:

Simula la presión adaptativa que ejerce el medio ambiente sobre los organismos determinando que tan bien resuelve un organismo un problema.

Org.	f(x) Desempeño	P(x) Prob.
01001	5	19%
10000	12	46%
01110	9	35%

Los organismos son ordenados y seleccionados de acuerdo a su función de desempeño f(x).

Entre mejor sea el organismo, mayor probabilidad de descendencia tendrá para trasmitir su material genético.

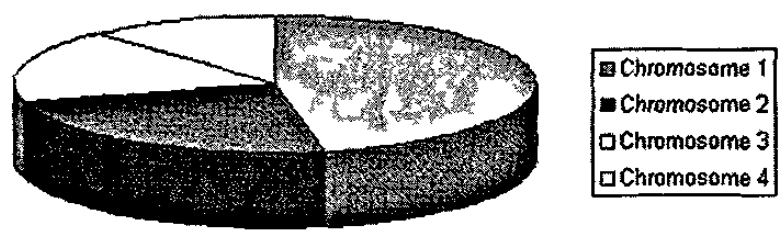


Figura 5.6 Funcionamiento del operador de selección genético.

5.7 El operador de cruce o reproducción.

Una vez realizada la selección, se procede a la reproducción sexual o cruza de los organismos seleccionados. En esta etapa, los sobrevivientes intercambian material genético y sus descendientes forman la población de la siguiente generación. El operador de cruce (o reproducción) es aquel por el cual los organismos de una población intercambian información genética. La implementación computacional del algoritmo genético propuesto, considera dos formas de reproducción sexual:

1. Uso de un punto único de cruza: éste se escoge de forma aleatoria sobre la longitud de la cadena que representa el cromosoma, y a partir de él se realiza el intercambio de material de los 2 individuos, tal y como se muestra en la Figura 5.7.

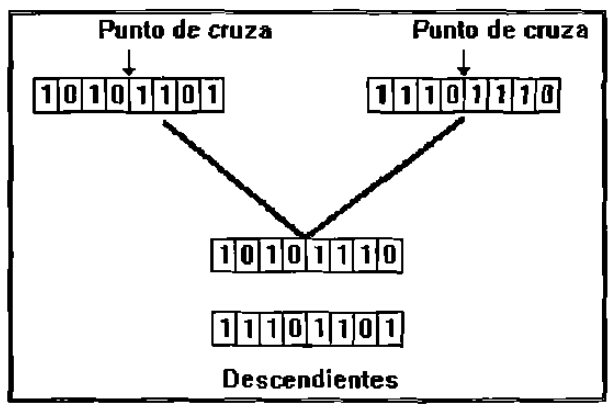


Figura 5.7 Un solo punto de cruza. Cada pareja de organismos da origen a 2 descendientes.

2. Uso de dos puntos de cruza: cuando se usan 2 puntos de cruza, se procede de manera similar, pero en este caso el intercambio se realiza en la forma mostrada en la Figura 5.8. Con este método de cruza entre 2 organismos se mantienen los genes de los extremos, y se intercambian los del centro.

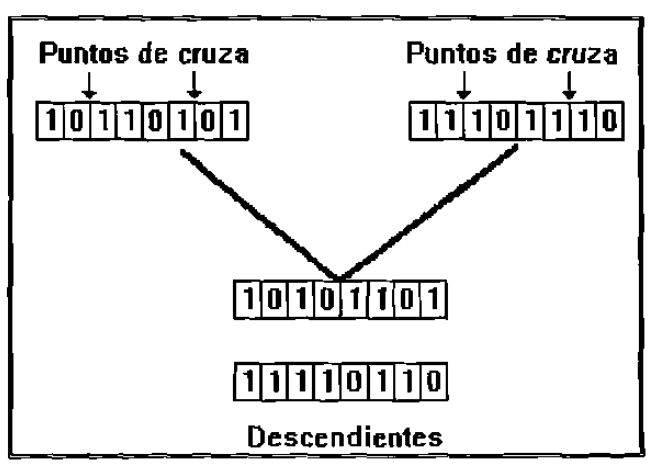


Figura 5.8 Dos puntos de cruza.

Normalmente la cruza se maneja dentro de la implementación del algoritmo genético como un parámetro expresado como un porcentaje que indica con qué frecuencia se efectúa dicho evento. Esto significa que no todas las parejas de organismos se cruzarán, sino que habrá algunas que pasarán intactas a la siguiente generación. Existe una variante del Algoritmo Genético en la que el organismo más apto a lo largo de las distintas generaciones no se cruza con nadie, y se mantiene intacto hasta que surge otro individuo mejor que él, que lo desplazará. A esta variante del algoritmo genético se le denomina "Elitismo" en virtud precisamente de lograr mantener el genoma "elite" de generación en generación hasta que aparezca un organismo con mejor desempeño que lo reemplace.

El operador de cruce se maneja como un parámetro en el Algoritmo Genético mediante un porcentaje que indica con qué frecuencia se efectuará dicho evento. El porcentaje de cruce normalmente es de alrededor del 60%, mientras que el porcentaje de mutación que a continuación se va a exponer en el siguiente apartado normalmente nunca supera el 5% [68]. En general, el porcentaje de mutación se fija mucho más bajo que el cruce.

A continuación en la figura 5.9, se explica de manera esquemática el mecanismo mediante el cual el proceso de cruzamiento ocurre. Es interesante verificar el comentario que se hace en función a como puede llegar a reducirse la diversidad genética del nuevo organismo al manipularse el material genético proveniente de un solo organismo padre y no de dos.

[68] Coello, Carlos. (1995), *Introducción a los Algoritmos Genéticos*, "Tecnologías de Información y Estrategias de Negocios, Año 3, No. 17", México, pag 5-11.

Cruce Genético y Reproducción

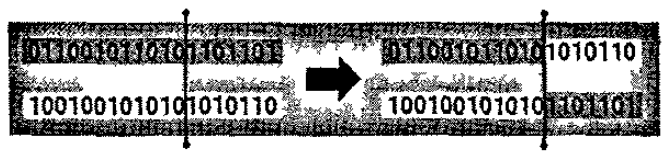
Este es el proceso más CRUCIAL de la evolución:

A diferencia de la clonación donde se reduce la diversidad genética ya que se manipula un solo material genético.

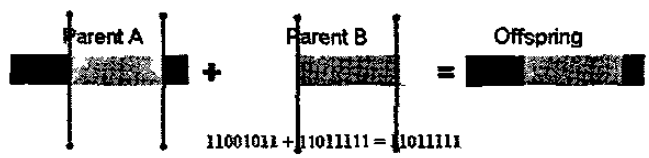
La idea básica es:

Intercambiar el contenido genético de dos individuos promedio con la expectativa de poder obtener un organismo mejor.

Operador de cruce simple



Operador de cruce doble



La probabilidad para la reproducción sexual de los organismos es proporcional a su nivel de desempeño.

Figura 5.9 Proceso de cruce genético y reproducción

5.8 El operador de mutación.

Además del operador de selección y de cruce, existe otro operador llamado mutación. La mutación es el operador que causa cambios aleatorios en los organismos de la población. El cambio se realiza a uno de los genes del cromosoma elegido aleatoriamente. Cuando se usa una representación binaria, el gene seleccionado se sustituye por su complemento, es decir un cero se cambia por uno y viceversa. Este operador permite la introducción de nuevo material cromosómico en la población, tal y como sucede en los equivalentes sistemas biológicos.

El operador de mutación aplicado en el Algoritmo Genético no es otra cosa más que un procedimiento de búsqueda escalando la colina (Hill climbing). Este proceso de escalamiento en el Algoritmo Genético es esencialmente aleatorio para producir las alteraciones esperadas en el contenido genético de los organismos. Las mutaciones que resulten en mejoras de desempeño en los organismos se conservan y el resto se desechan. No obstante, los desplazamientos a lo largo de los ejes del sistema de coordenadas durante el proceso de escalamiento, es poco probable que sean en línea recta sobre la dirección de búsqueda más corta hacia la solución óptima global. Esto significa que la trayectoria de búsqueda que sigue el proceso de mutación sigue un camino en forma de zigzag más o menos perpendicular al vector correspondiente a la dirección de la solución óptima. Este proceso de búsqueda en zigzag antes descrito se puede ejemplificar en la Figura 5.10.

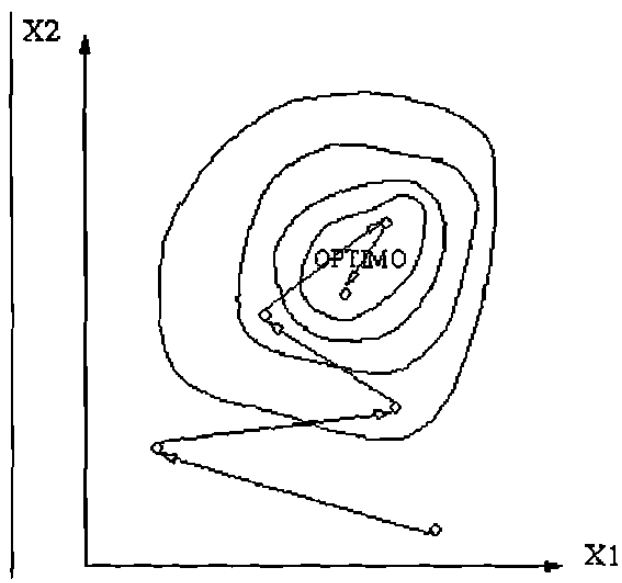


Figura 5.10 Proceso de escalamiento para la búsqueda de la solución óptima.

A continuación en la Figura 5.11, se desarrollan breves explicaciones respecto al factor de mutación y de elitismo ya antes mencionados. Es importante precisar que aunque los dos factores "parezcan" opuestos, en realidad lo que ocurre es que el primero se concentra en diversificar el contenido del genoma dentro de una población y el segundo busca perpetuar de generación en generación aquél genoma que resulte con mejor desempeño.

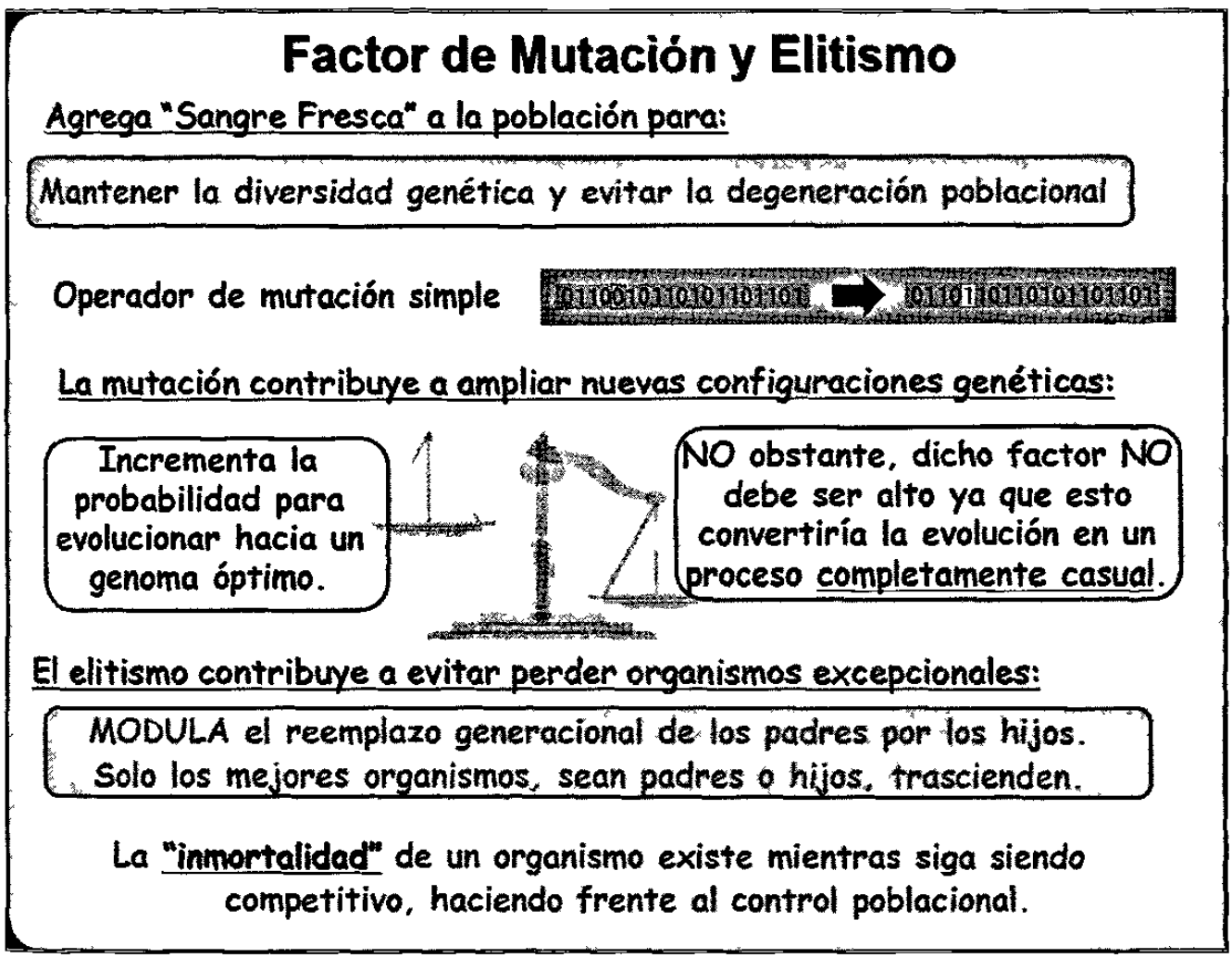


Figura 5.11 Afectación del factor de mutación y elitismo en el algoritmo genético.

5.9 Funcionamiento general del algoritmo genético simple.

La operación del algoritmo genético en su versión más simple puede ilustrarse con el siguiente pseudo-código presentado en la Figura 5.12:

```
generar población inicial, G(0);
evaluar G(0);
t:=0;
repetir
        t:=t+1;
        generar G(t) usando G(t-1);
        evaluar G(t);
hasta encontrar una solución;
```

Figura 5.12 Pseudo-código del algoritmo genético simple [69].

Con la finalidad de ofrecer una breve explicación al pseudo código anterior, iniciamos primero al generar aleatoriamente la población inicial, la cual estará constituida por un conjunto de organismos, o cadenas de caracteres que representan las soluciones posibles del problema. A cada uno de los organismos de esta población se le aplicará la función de aptitud a fin de saber qué tan buena es la solución que está codificando en el interior de su contenido genético. Habiendo evaluado la aptitud de cada organismo, se procede a la selección de los que se cruzarán en la siguiente generación. En este sentido es presumible pensar, que el Algoritmo Genético escogerá a los "mejores" organismos.

Un aspecto interesante en un Algoritmo Genético es determinar su punto de terminación. Si acaso supiéramos la respuesta óptima a un problema, entonces la respuesta a esto último es trivial ya que sólo es necesario incluir en el Algoritmo Genético una indicación que haga concluir el procedimiento de cómputo una vez que el algoritmo converge en la solución previamente definida como óptima. Sin embargo, eso casi nunca es posible, o por lo menos no es lo que se supone sea lo deseable. Debido a lo anterior, normalmente se usan 2 criterios principales de detención:

1. Iterar el Algoritmo Genético hasta un número máximo de generaciones.
2. Detenerlo una vez la población se haya estabilizado, es decir, cuando todos o la mayoría de los individuos tengan la misma aptitud. A esto se le llama homeóstasis.

[69] Buckles, Bill. (1992), *Genetic Algorithms*, IEEE Computer Society Press, EUA, pag 109.

En términos prácticos al aplicar el segundo método, una característica que se presenta cuando la población se estabiliza, es que esto guarda una relación estrecha respecto a la probabilidad de que exista la generación de mejores organismos. Dicho de otro modo, al haber organismos muy parecidos entre sí, la probabilidad de un enriquecimiento en la solución disminuye debido precisamente a esa falta de variedad genética de donde echar mano para mejorar la solución.

Con la finalidad de representar gráficamente lo antes expuesto, en la Figura 5.13, se ofrece una mayor explicación respecto al funcionamiento básico del algoritmo genético.

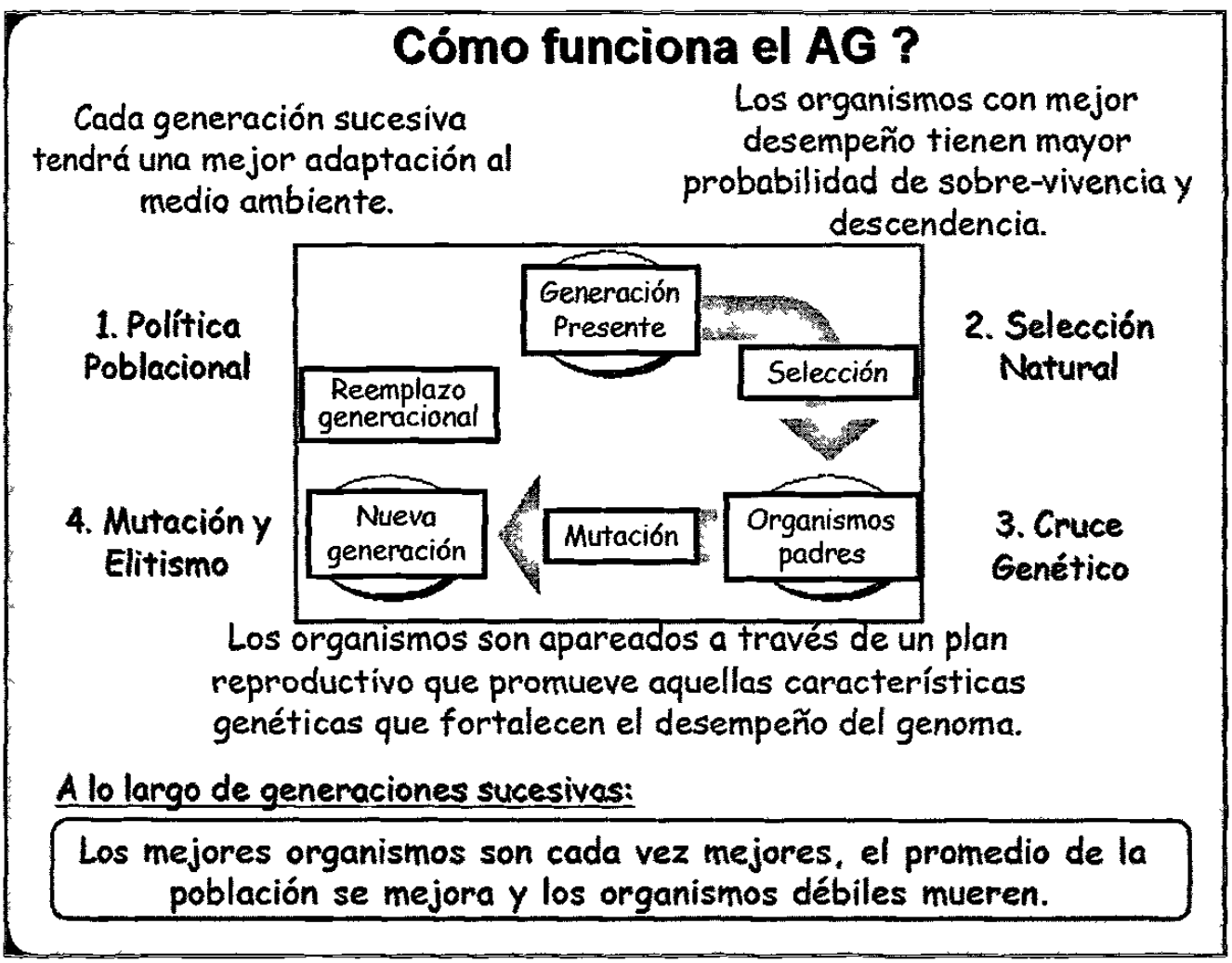


Figura 5.13 Funcionamiento del algoritmo genético.

5.10 El problema de la codificación de la solución (genoma) en los algoritmos genéticos.

En el asunto de la codificación en los Algoritmos Genéticos, es necesario diferenciar entre lo que es un genotipo y un fenotipo. El genotipo es precisamente la cadena cromosómica del organismo, la cual es utilizada para contener su información genética. A diferencia tenemos el fenotipo, el cual corresponde a los valores que se calculan una vez que se aplica el proceso de decodificación sobre el genotipo. Ambos elementos están íntimamente relacionados ya que un buen Algoritmo Genético requiere de un fenotipo apropiado que pueda contribuir a medir el nivel de desempeño de los organismos y a su vez un buen fenotipo requiere de un genotipo apropiado que logre "maximizar" la probabilidad de generar bloques esquemáticos que permitan evolucionar el desempeño del fenotipo[70].

Así pues, el paralelismo implícito al que se refería Holland en su Algoritmo Genético es precisamente el hecho de que mientras el algoritmo determina las mejores aptitudes de los organismos de una población, al mismo tiempo determina en forma implícita las características promedio de un número obviamente mucho mayor de cadenas cromosómicas. Lo anterior se logra a través precisamente de la explotación de los bloques constructores que se van verificando como patrones característicos en los organismos de la población.

No obstante la ventaja de la codificación binaria, es importante precisar que también existen dos desventajas principales:

1. En los problemas de optimización continua, es común que se lleguen a requerir cadenas cromosómicas demasiado largas lo cual propicia un desempeño extremadamente lento en la eficiencia del algoritmo al buscar la solución.
2. La aplicación de los operadores de cruce naturales pueden llegar a producir soluciones ilegales sobre la base de una codificación binaria.
3. La codificación binaria en forma natural no logra mapear adecuadamente el espacio de búsqueda con el espacio de representación. Dicho de una manera sencilla, quiere decir que aunque dos soluciones sean contiguas en el espacio solución, en la representación codificada de ambos, éstos pueden llegar a ser tan diferentes por varios órdenes de magnitud (muchos bits).

---

[70] Hollstein, R. (1991), *Artificial Genetic adaptation in computer control systems,* Phd Thesis, University of Michigan, EUA, pag 84.

En lo referente al último punto expuesto se han desarrollado investigaciones que han venido a resolver dicha dicotomía entre la adyacencia del espacio de búsqueda respecto a la adyacencia en el espacio de representación (o codificación). Uno de los métodos recientemente propuestos (1998), ha sido el código de Gray [71]. La virtud de la codificación en Gray es en principio que continua siendo binaria, solo que se le añade un procedimiento de transformación a través de la operación booleana "XOR" aplicada cada uno de los bits contenidos en el cromosoma de derecha a izquierda.

Investigaciones han comprobado que la aplicación de este tipo de codificaciones, han resultado en una mejora sustancial en el desempeño del Algoritmo Genético ya que se disminuye la cantidad de óptimos locales como consecuencia del uso de una representación mucho más apegada al propio espacio de búsqueda[72].

En lo referente a la codificación de números reales que requieren el manejo de decimales con alta precisión, el problema de la codificación binaria es inevitablemente crítico, con o sin el uso de la codificación de Gray mencionado antes. Para esto se han desarrollado representaciones basadas en el *estándar* de la IEEE, mediante el cual este tipo de números siguen siendo codificados en forma binaria pero no a partir de una representación en formato decimal natural sino a partir de una representación en formato exponencial.

Este tipo de pre-codificación del número real permite con tal solo 32 bits representar cualquier número real de precisión simple. De los 32 bits, los primeros 8 se aplican para la representación del exponente y el resto de los otros 24 bits se utilizan para otorgar precisión en la mantisa. Ante el beneficio que otorga el seguir manteniendo una representación binaria el costo de lo anterior es el proceso de decodificación ya que el cromosoma deberá ser interpretado de su representación en binario a su representación exponencial y luego finalmente a su valor en formato decimal natural.

---

[71] Whitley, D. (1998), *Representation issues in Neighborhood Search and Evolutionary Algorithms,* John Wiley and Sons, University of Sussex, England, Capítulo 3, pag 39-57.

[72] Mathias, K. (1994), *Transforming the search space with gray coding,* Proceedings of the IEEE International Conference on Evolution Computation, Piscataway New Jersey, EUA, pag 213-218.

5.11 Trabajos previos de investigación: aplicación de los algoritmos genéticos para optimización en problemas de logística.

Es interesante comprobar que el historial de la aplicación de los Algoritmos Genéticos en el campo de los problemas de logística es realmente incipiente en términos relativos. La revisión bibliográfica realizada nos lleva a identificar que la cantidad de publicaciones hasta el año 2002 registradas como aplicación total de los Algoritmos Genéticos es de 16,524 documentos. De igual modo de estas publicaciones tan solo 584 hacen referencia a su aplicación en la Logística, es decir únicamente el 3.5% del total. A continuación en la Figura 5.14 se evidencia dicha situación en la cual se utiliza una escala logarítmica en la que se muestra la cantidad de publicaciones año con año.

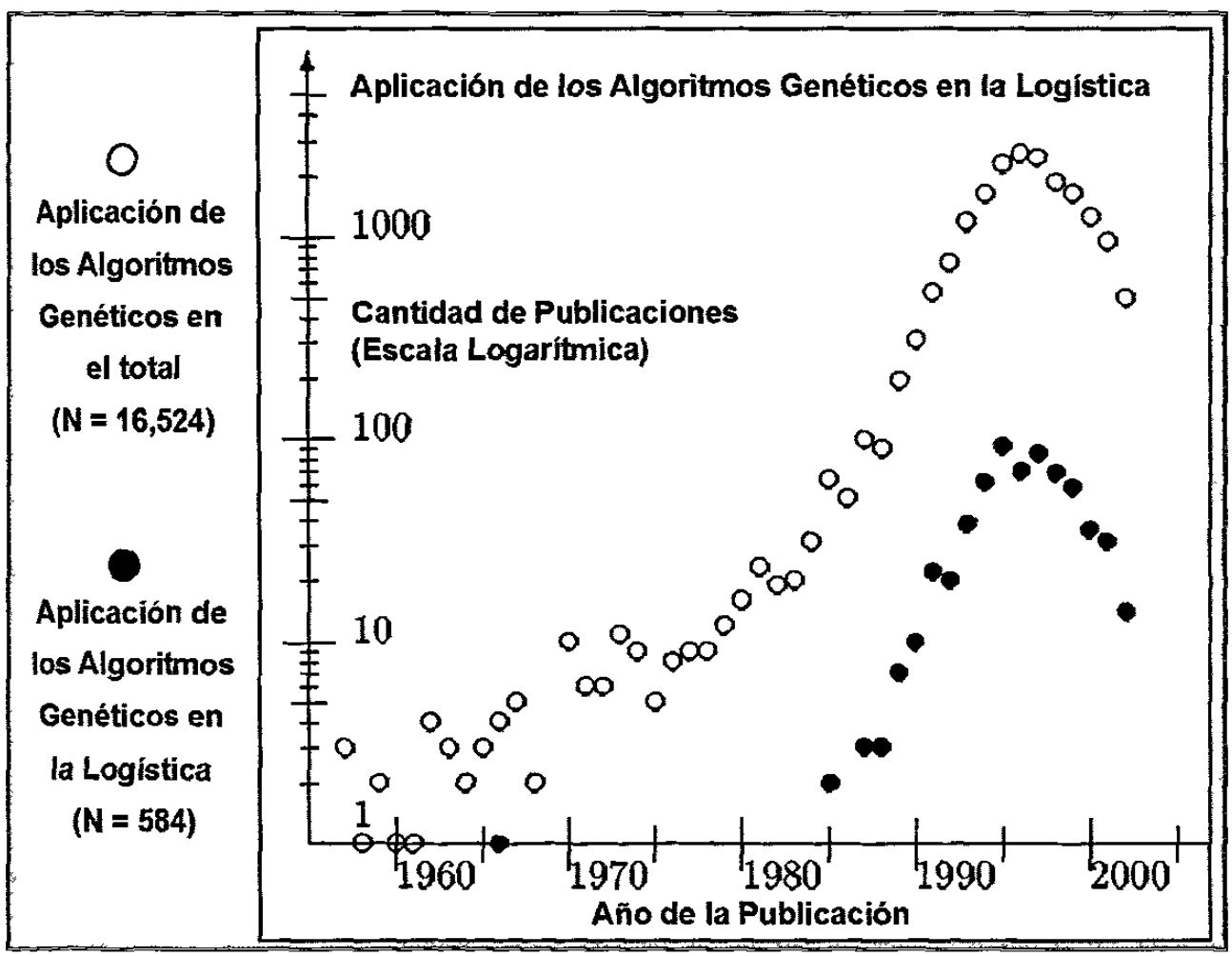


Figura 5.14 Relación entre la cantidad de aplicaciones de los Algoritmos Genéticos en el campo de la Logística con respecto al total[73].

[73] Alander, Jarmo. (2003), *An Indexed Bibliography of Genetic Algorithms in Logistics*, Department of Engineering, University of Vaasa, Finland, pag 11.

Como complemento a la figura anterior, podemos estratificar la cantidad de publicaciones que han sido desarrolladas de este mismo universo. En la tabla 5.2 se muestra la estadística por año.

Tabla 5.2 Cantidad de publicaciones hechas por año de Algoritmos Genéticos en la Logística [74].

AÑO	FREC
1966-1986	3
1987	3
1988	3
1989	7
1990	10
1991	22
1992	20
1993	38
1994	61
1995	92
1996	69
1997	85
1998	67
1999	57
2000	35
2001	31
2002	14
TOTAL	617

En la tabla 5.3 se muestra la estadística por país de origen en la cual las primeras 2 columnas hacen referencia al total de las aplicaciones y las últimas 2 columnas a las aplicaciones de los algoritmos genéticos únicamente en el campo de la logística.

Tabla 5.3 Publicaciones por país de los Algoritmos Genéticos en la Logística respecto al total [75].

	TOTAL		LOGISTICA	
PAIS	FREC	%	FREC	%
Estados Unidos	4,829	29%	155	27%
Japón	2,320	14%	62	11%
Gran Bretaña	1,838	11%	54	9%
Alemania	1,266	8%	44	8%
Finlandia	534	3%	30	5%
China	803	5%	26	4%
Francia	441	3%	23	4%
Italia	505	3%	23	4%
Corea del Sur	408	2%	21	4%
Canada	262	2%	15	3%
Australia	422	3%	14	2%
Taiwan	363	2%	14	2%
España	276	2%	13	2%
Grecia	102	1%	8	1%
India	264	2%	8	1%
Isreal	110	1%	7	1%
Singapur	136	1%	7	1%
Holanda	181	1%	7	1%
Brazil	134	1%	6	1%
Croacia	26	0%	6	1%
Otros	1,304	8%	41	7%
TOTAL	16,524	100%	584	100%

[74] Alander, Jarmo. (2003), Idem, pag 11.
[75] Alander, Jarmo. (2003), Idem, pag 12.

Concentrando ahora la atención en las aplicaciones de los Algoritmos Genéticos en el campo de la Logística, podemos ahora clasificar su frecuencia en función al tema particular. A continuación en la Tabla 5.4, se pueden apreciar la contribución de ha tenido los algoritmos genéticos en diversos aspectos relacionados a la función de la logística.

Tabla 5.4 Publicaciones por Tema de Investigación de los Algoritmos Genéticos en la Logística[76].

TEMA DE INVESTIGACION	FREC
Problema del TSP	175
Telecomunicaciones	88
Control y Paralelismo	71
Transportación	62
Ruteo	45
Otros Logística	28
Secuenciación	25
Estrategias Hibridas	22
Técnicas de Cruzamiento	18
Tráfico	14
Redes Neuronales	12
Lógica Confusa	9
Estrategias Evolutivas	9
Tamaño de Población	6
TOTAL	584

Finalmente en la Tabla 5.5, podemos revisar la clasificación de las publicaciones de acuerdo al tipo de documento que haya sido desarrollado.

Tabla 5.5 Publicaciones por Tema de Investigación de los Algoritmos Genéticos en la Logística[77].

TIPO DE DOCUMENTO	FREC
Artículos (Proceedings)	339
Artículos (Journals)	199
Tésis de Doctorado	19
Tésis de Maestría	6
Libros de Texto	5
TOTAL	568

Es interesante mencionar que la aplicación de los Algoritmos Genéticos en problemas reales de logística, ha tenido un desarrollo híbrido. Es decir, no han sido aplicados de manera holística, sino que han sido implementados como parte componente de una metodología más extendida en la cual se incluyen diversos algoritmos de pre-procesamiento y post-procesamiento. Existen implementaciones

[76] Alander, Jarmo. (2003), Idem, pag 10.
[77] Alander, Jarmo. (2003), Idem, pag 9.

metodológicas que en una primera fase, utilizan heurísticas de simplificación de rutas[78]. También es posible el uso de procesos de reinicialización una vez detectada alguna convergencia local.

Existen metodologías en las cuales, la fase correspondiente al proceso evolutivo, genera un porcentaje de la población inicial utilizando técnicas tipo GRASP, dejando que el resto de la población se genere de forma naturalmente aleatoria. La generación de soluciones iniciales a través de técnicas GRASP, permite obtener soluciones bastante buenas con un mínimo esfuerzo computacional[79]. La única desventaja de insertar en la población inicial un por ciento de soluciones de este tipo, es que esto irremediablemente lleva a una convergencia local en iteraciones tempranas del algoritmo. No obstante, esta situación puede ser resuelta a través de la aplicación de operadores de reinicialización[80].

En lo referente a la técnica de codificación de la solución en el cromosoma, existen diversas formas de implementarlo. Una de estas es la representación basada en el orden, esto es, a través de listas de tamaño variable con los nodos del camino cerrado que conforman la solución. Cada solución comienza incorporando a una ruta vacía un nodo aleatorio como primer elemento y continúa añadiendo al azar el nodo Vk del subconjunto V de nodos adyacentes al último nodo incorporado. Este procedimiento finaliza cuando todos los nodos están incorporados a la ruta terminada. Sea (c1, c2,..., cn) la ruta obtenida. Para cerrar la ruta, entonces se adiciona a la ruta el nodo (cn+1,..c1) tal que (cn,cn+1,..,c1) obtenga la ruta de distancia mínima entre cn y c1[81].

Tocante a la función de desempeño o "fitness", ésta típicamente esta representada por la distancia de la ruta. Al igual, el operador de selección mayormente utilizado es el de torneo (tournament en inglés), en el cual se van identificando dos soluciones de forma aleatoria y seleccionando la mejor. Este proceso se repite dos veces con la población dada, de forma tal que la mejor solución será seleccionada dos veces mientras que la peor es eliminada[82].

En lo que concierne a la fase de reproducción, lo más adecuando es utilizar operadores de cruzamiento

---

[78] Ling, Wang (1999), *An effective hybrid optimization strategy for job-shop scheduling problems*, Computers and Operations Research. Vol. 28, pag 585-596.
[79] Feo, T, (1995), *Greedy Randomized Adaptive Search Procedures*, Journal of Global Optimization, Vol. 6, pag 109-133.
[80] Gulnara, M, (2002), *Un enfoque híbrido basado en metaheurísticas para la solución del cartero rural*, Congreso latinoamericano de Investigación de Operaciones, Chile, pag 3-7.
[81] Hakim, Al (2001), *An analogue genetic algorithm for solving job shop scheduling problems*, International Journal of Production Research, vol. 39, No 7, pag 1537-1548.
[82] Coello, Carlos. (1995), *La importancia de la representación en los Algoritmos Genéticos*, Tecnologías de Información y Estrategias de Negocios, Año 3, No. 17, Enero de 1995, pag 5-11.

que estén personalizados específicamente para el problema en cuestión. La razón de lo anterior es debido a que un operador de cruce que no esté bien planeado, puede llegar a generar frecuentemente organismos que no satisfagan los requerimientos de una ruta para el problema[83]. Aunque en tales casos es posible aplicar procedimientos reconstructivos a través de la incorporación de material genético a partir del último nodo que pudiera estar a salvo en el cromosoma, no obstante, lo mejor siempre será crear un operador de cruzamiento que evite generar organismos infactibles en la población. Indistintamente en el producto del cruzamiento de dos padres se preservan los dos mejores entre los dos padres y los dos hijos.

En lo que toca al operador de mutación, idealmente esté debe estar inmerso en el procedimiento para la formación de los hijos. No obstante en algunos otros casos, el operador de mutación se maneja en forma individual con la finalidad de que el algoritmo genético rápidamente converja y pueda así acceder deliberadamente a fases especializadas que estén incluidas en la metodología general[84].

Hablando de fases de post-procesamiento respecto al algoritmo genético, una estrategia de implementación puede ser en la cual una vez generada una nueva población, se busca entonces mejorar cada elemento de la población utilizando rutinas de simplificación de rutas. Estas rutinas de simplificación pertenecen a la fase de búsqueda local y pueden ser dirigidas a través de técnicas basadas en GRASP para mejorar cada ruta construida[85].

Otra estrategia de post-procesamiento al algoritmo genético puede ser implementada a partir de una población de soluciones elite obtenida durante la ejecución de la fase evolutiva. En estos casos, es posible aplicar a cada organismo de la población elite, un operador genético de inversión modificado. Este operador genético de inversión modificado solo requiere de un padre para generar a un hijo. El operador de inversión antes descrito, invierte un sub-camino de la ruta completa, la cual es formada entre dos puntos de ruptura seleccionados como parte del proceso, produciendo de esta manera una nueva solución[86]. La nueva ruta formada no requiere de un proceso de verificación, ya que dicho operador preserva mucho de los enlaces entre los genes del cromosoma evitando el "rompimiento" de

---

[83] Palominos, Pedro. (2001). *Influencia del Tipo de Cruza y Tamaño de la Población en la Calidad de la Solución de un Algoritmo Genético,* Revista del Instituto de Investigación Operativa , Volumen 6, N°3, pag 3-8.
[84] Gruttner, E. (2002), *Algoritmos Genéticos en recorridos óptimos de líneas de transporte público,* Departamento de Ingeniería y Ciencias de la Computación, Universidad de Concepción, Chile, pag 8-15.
[85] Colomer, José. (1999), *Los modelos de cuatro etapas: utilidad y limitaciones,* Universidad Politécnica de Valencia, Valencia, España, pag 6-20.
[86] Gendreau, M. (1992), *New Insertion Post-optimization Procedures for the Traveling Salesman Problem,* Operations Research, Vol 40, pag 1086-1094.

la ruta. Lo anterior hace efectivo su aplicación en un proceso de mejora de soluciones, aunque exista la posibilidad de que el hijo generado no sea mejor que el padre. Así entonces, en cada iteración de la fase de post-procesamiento se va aplicando el operador de inversión modificado al 100% de la población, manteniendo siempre la mejor solución entre el padre y su propio hijo.

En lo referente a la estrategia de finalización generacional de las poblaciones, hay algunas en las cuales cuando se detecta una convergencia a un óptimo local se aplica el operador de reinicialización, generando una nueva población inicial. Lo anterior permite acceder a las diversas regiones del espacio solución a ser explorado a lo largo del algoritmo general[87].

Como es posible constatar, existe una diversidad interesante de estrategias a poder ser implementadas en un Algoritmo Genético. Para el caso de nuestro problema de investigación, es importante mencionar que como parte del proceso de inicialización, se incorpora un procedimiento particular para generar la población inicial. En términos generales nuestro procedimiento parte de un nodo designado como origen el cual entonces es ubicado como primer gen en el cromosoma. Luego a partir de una matriz de adyacencia, se busca aleatoriamente un vecino a este nodo[88]. Luego entonces, este vecino se coloca en el segundo gen y así sucesivamente se repite el proceso para cada nodo hasta alcanzar el nodo destino. Este procedimiento se aplica tantas veces como cromosomas se ocupen para completar los organismos que se requieran para cubrir la población inicial.

Este procedimiento "dirigido" para la generación de la población inicial a partir de una matriz de adyacencia previamente construida, ciertamente también puede ser tomado en cuenta como estrategia para el desarrollo de un operador de cruzamiento dirigido[89]. La ventaja de esta estrategia "dirigida" es que evita caer al algoritmo genético propuesto en óptimos locales. Lo anterior se logra computacionalmente hablando a través de identificar como inválidas aquellas rutas que no contengan alguna particularidad o propiedad esperada en la conformación del contenido genético. En nuestro caso, se logra implementar lo anterior a través de la aplicación de factores de penalización para todas aquellas secuencias genéticas que aparecieran en el cromosoma que se estuvieran alejando de alguna propiedad deseada en la matriz de adyacencia.

---

[87] Bäck, T. (1998), *On The Behavior Of Evolutionary Algorithms In Dynamic Environments*, Proceedings of the Fifth IEEE Conference on Evolutionary Computation, IEEE Press, Pag. 446-451.
[88] Hansen, P. (1998), *Variable Neighborhood Decomposition Search*, University of Montreal, pag 53-98.
[89] Mladenovic, N. (1995), *A Variable Neighborhood Algorithm: a New Metaheuristic for Combinatorial Optimization*, Abstract of papers presented at Optimization Days, Montreal, pag 12.

5.12 Ventajas y desventajas del algoritmo genético como estrategia de solución.

A continuación enumeramos algunas de las ventajas que se tiene al aplicar los Algoritmos Genéticos en problemas de optimización matemática:

1. No se necesitan conocimientos específicos sobre el problema que se busca resolver.
2. Paralelismo implícito: operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales.
3. Robustez: resultan menos afectados respecto a las heurísticas tradicionales para el manejo de convergencias prematuras hacia óptimos locales (soluciones sub-optimas).
4. Son fáciles de implementar en los sistemas computacionales con arquitectura en paralelo.

En la Figura 5.15, se describe la forma mediante la cual dichas ventajas resultan ser preponderantes respecto a la solución de los problemas NP-Hard en lo general y del problema de investigación en lo particular.

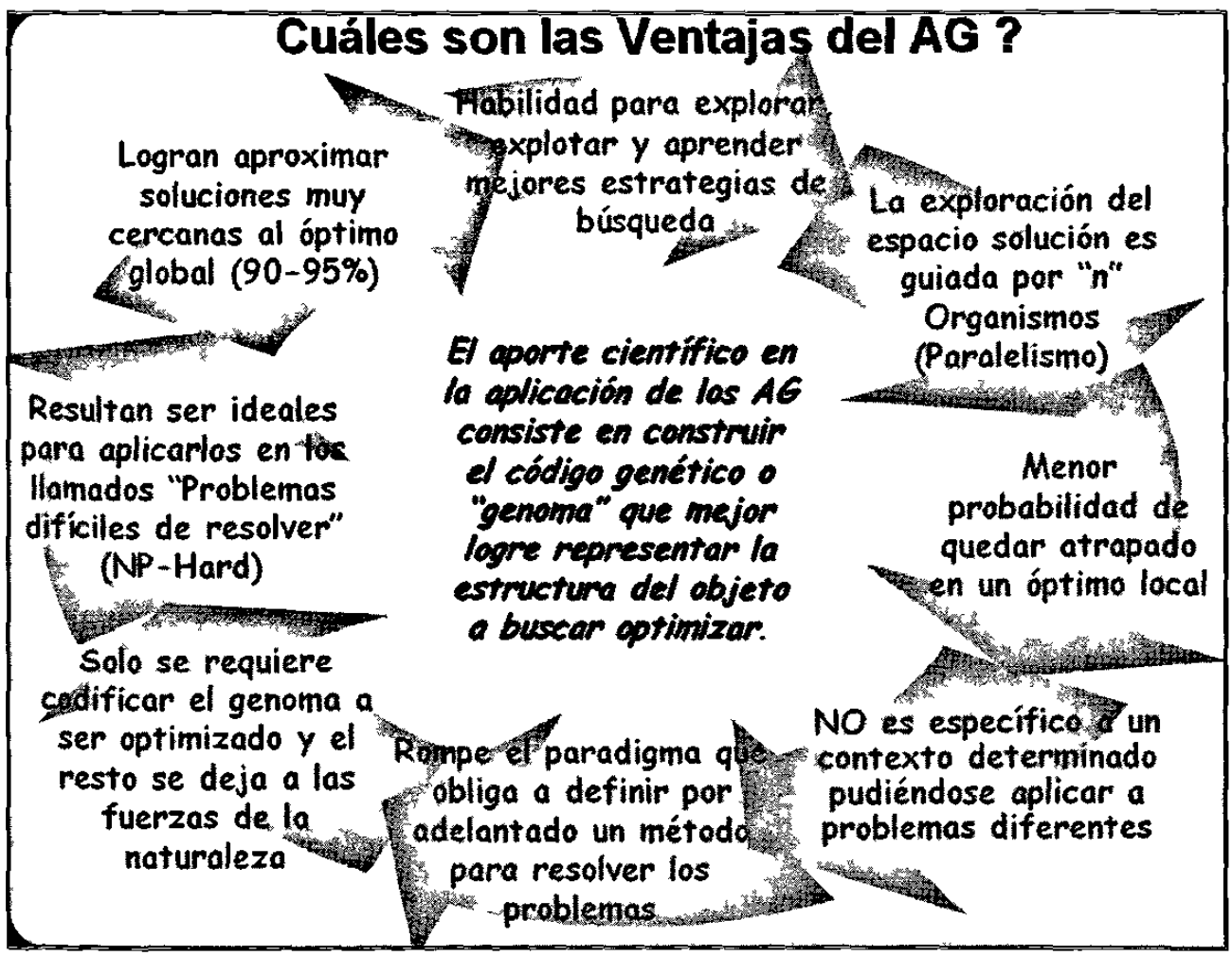


Figura 5.15 Ventajas en la aplicación de los algoritmos genéticos.

En confrontación con lo anterior, a continuación se enumeran algunas de las desventajas que se tiene en su aplicación:

1. Se aplican operadores de convergencia probabilísticos, en vez de determinísticos.
2. Están fuertemente en dependencia respecto a sus parámetros de operación ya antes mencionados, tales como el tamaño de la población, número de generaciones, el porcentaje de mutación, el porcentaje de cruce, etc.
3. Dependiendo de las características propias del problema, pueden llegar a tardar mucho en converger, o quizás peor aún nunca llegar a converger en absoluto[90].

Para finalizar el presente apartado, citaremos algunos de las conclusiones obtenidas en otros proyectos de investigación previos en donde se desarrollan análisis comparativos del desempeño de los algoritmos genéticos respecto al uso de otras meta-heurísticas para la solución de problemas de secuenciación en general.

Reeves en 1995 desarrolla un análisis comparativo entre un algoritmo genético y un algoritmo basado en enfriamiento simulado. La investigación de Reeves concluye que el algoritmo genético obtiene mejores soluciones en un tiempo menor que su contraparte[91]. Por otro lado, Croce en su investigación concluye que el algoritmo genético obtiene mejores resultados que la meta-heurística Tabu-search con la única atenuante del tiempo computacional[92].

Finalmente podemos citar a Cheng, el cual en su investigación se enfoca en comprobar el porqué es superior la aplicación del algoritmo genético con respecto al uso de otras meta-heurísticas. Cheng, expone en sus conclusiones que enfriamiento simulado y tabu-search desarrollan un búsqueda uni-direccional a través del uso de una sola solución como candidato. En contrate, continua Cheng, el algoritmo genético desarrolla una búsqueda multi-direccional en paralelo a través del uso de múltiples soluciones potenciales. Es precisamente en ese sentido, que el algoritmo genético hace uso de los mejores organismos en la población para explorar diferentes secciones del espacio solución simultáneamente[93].

---

[90] Coello, Carlos. (1995), *Introducción a los Algoritmos Genéticos*, "Tecnologías de Información y Estrategias de Negocios, No. 17", México, pag 5-11.

[91] Reeves, C. (1995), *Genetics algorithms flow shop sequencing*, Computers & Operations Research, Vol 22, EUA, pag 5-13.

[92] Croce, F. (1995), *A genetic algorithm for the job shop problem*, Operations Research, Vol 22, EUA, pag 15-24.

[93] Cheng, R. (1997), *Genetic algorithm and Engineering design*, John Wiley and Sons, New York EUA, pag 7.

5.13 Variantes adicionales en los algoritmos genéticos.

Para nuestro problema de investigación, aunque ya hemos comentado que se trata de optimización discreta, la problemática de la codificación binaria sigue estando presente. En este sentido, la mayor parte de las investigaciones se han entonces venido concentrando en la necesidad de desarrollar operadores de cruce especializados que logren mantener las bondades de la codificación binaria pero que a la vez eviten generar soluciones inválidas durante el proceso de reproducción[94].

Investigaciones han propuesto el uso de cromosomas con una representación matricial y binaria la cual permite realizar operaciones de cruza y de mutación que no son posibles realizar en los Algoritmos Genéticos convencionales. Adicional a la ventaja antes descrita, el mecanismo de selección para la cruza hace que la población final no converja hacia un contenido genético similar de organismos, evitando así la posibilidad de caer en óptimos locales que redundan en una degeneración de la población. Por otra parte, el aplicar un criterio seudo-elitista en la selección de individuos para cada nueva población, permite una convergencia en el valor de la función de adaptación del mejor individuo encontrado.

Dentro de la fase de selección de la nueva población es posible que el algoritmo emplee el principio del espacio poblacional extendido que permite que tanto padres como descendientes tengan las mismas posibilidades de competir por la supervivencia. De esta manera el algoritmo genera una nueva población de igual tamaño a la original, combinando elitismo y aleatoriedad en la formación de ella. En una primera instancia, se eligen los individuos más adaptados, el mejor de cada población (inicial, hijos y mutantes). Enseguida, para completar el número constante de individuos, se seleccionan representantes de cada población de manera aleatoria.

Una última variante del Algoritmo Genético que se va a exponer, es aquella que resulta cuando es necesario el tratar con cromosomas no de tamaño fijo como sucede en el caso natural, sino cuando el tamaño del cromosoma puede ir variando (longitud variable) de acuerdo a los diferentes estados posibles durante el proceso de búsqueda de la solución. La necesidad para manejar cromosomas de longitud variable, existe con la finalidad de capacitar al algoritmo para ir cambiando el nivel de precisión a medida que éste avanza en el proceso de búsqueda. No obstante, también hubo otra razón después, y esta proviene de la necesidad de ir agregando o disminuyendo de manera dinámica

[94] Deb, K. (1995), *Simulated Binary crossover in evolutionary optimization*, "Complex Systems No. 9", EUA, pag 115-148.

elementos (o variables) del cromosoma a medida que se va convergiendo hacia una solución en el espacio de búsqueda.

Así fue como nació la versión "desordenada" del Algoritmo Genético (Messy Genetic Algorithms) [95]. La idea fundamental de esta variante es la de empezar con cromosomas de longitud corta que permitan de manera gradual y al mismo tiempo relativamente más rápido identificar aquellos bloques esquemáticos que mejor características provean a los organismos de la población para de esta manera gradualmente propagar estos esquemas hacia cromosomas de mayor longitud. Así pues, la versión "desordenada" del Algoritmo Genético se desarrolla en dos fases.

La primera es la llamada fase primordial que servirá la identificar aquellos esquemas que servirán como bloques constructores a propagar a su vez en la segunda fase. La segunda fase se le denomina, fase de yuxtaposición. El gran asunto en esta variante corresponde en determinar que tan largos deben ser los esquemas a ser generados en la primera fase. La dificultad de lo anterior obedece a que si son demasiado pequeños se puede caer en el error de que no contengan suficiente material genético para el proceso de propagación y mejoramiento en la fase de yuxtaposición. Por el otro lado, si acaso son demasiado largos, pues entonces se redundaría en un Algoritmo Genético simple de solo una fase. Las investigaciones reportadas a la fecha en esta variante del Algoritmo Genético, reportan relativamente pocos casos de éxito [96].

---

[95] Goldberg, D. (1996), *Don't Worry, be messy*, "Proceedings of the 9th International Conference on Genetic Algorithms", The MIT Press, Cambridge Mass, EUA, pag 24-30.
[96] Chowdury, M. (1997), *Messy Genetic Algorithms based new learning method for fussy controllers*, "Proceedings of the IEEE International Conference on Industrial Technology", The MIT Press, Shangai China, pag 110-130.

5.14 Comentarios finales.

En los 90's, una gran parte de la investigación en esta área se ha concentrado en el desarrollo de mejoras al desempeño de los algoritmos genéticos. Se han propuesto nuevas técnicas de representación, selección y cruza. Por ejemplo, el uso de los códigos de Gray y la codificación dinámica han superado algunos de los problemas asociados con la representación de valores reales mediante cadenas binarias. También se han propuesto técnicas adaptativas que varían dinámicamente los parámetros de control (porcentajes de mutación y cruza), en contraposición con el esquema estático tradicional. Otras innovaciones notables son los algoritmos genéticos distribuidos y los algoritmos genéticos paralelos [97].

No obstante el avance relativo en la aplicación de los Algoritmos Genéticos, el fundamento matemático de estos sigue siendo un área abierta de investigación. Este interés se ha reflejado recientemente en México, a través del centro de investigación LANIA (Laboratorios Nacionales de Informática Avanzada).

Con el presente capítulo damos por terminado el marco conceptual de la tesis. En el capítulo 3 definimos a la logística como área de aplicación y algunos de sus problemas más representativos desde donde establecimos de manera genérica hacia donde enfocaríamos el problema de investigación. Luego en el capitulo 4 definimos el concepto NP-Hard en relación al problema de investigación previamente establecido. En dicho capítulo se establecieron las técnicas de optimización más prominentes utilizadas para dar solución a los problemas tipo NP-Hard. Finalmente, en este capítulo 5 establecimos el marco teórico conceptual referido a la técnica de optimización basada en Algoritmos Genéticos, la cual está siendo propuesta como técnica de solución para el problema de investigación.

En el siguiente capítulo 6 nos abocaremos a definir formalmente y matemáticamente el problema de investigación. En el capítulo 7 haremos el planteamiento de los objetivos e hipótesis y en el capítulo 8 el diseño experimental.

[97] Srinivas, M. (1999), *Genetic Algorithms: A Survey,* IEEE Computer, EUA, pag 17-26.

Es más valioso conocer algunas de las preguntas que todas las soluciones.

James Thurber

6. Planteamiento del problema de investigación

6.1 Introducción al problema de investigación.

En la primera parte del documento de tesis (capítulo 1 y 2) se expusieron los antecedentes del proyecto, mediante los cuales, se introduce al lector en el tema de la logística así como en el factor educativo como elemento habilitador para la aplicación de la investigación de operaciones. La segunda parte correspondió al marco conceptual (capítulos 3 al 5). En el capítulo 3, se expusieron de manera general algunos de los problemas en la práctica de la logística. En el capítulo 4 se exponen de manera particular los fundamentos matemáticos necesarios para atender los problemas de logística de ruteo. Finalmente en el capítulo 5 se exponen los fundamentos teóricos referentes al procedimiento meta-heurístico propuesto para dar solución al tipo de problemas tratados particularmente en el capítulo 4.

A partir de ahora damos inicio a la 3era y última parte del documento de tesis el cual corresponde al desarrollo del proyecto de investigación. En el presente capítulo 6 se hará el planteamiento formal del problema de investigación. En el capítulo 7 se formularan los objetivos, justificaciones e hipótesis del proyecto. En el capítulo 8 se expondrá el diseño experimental para probar la hipótesis y finalmente en los capítulos 9 y 10 se expondrán los resultados y las conclusiones del experimento respectivamente.

Los problemas de optimización discreta se diferencian de los de optimización continua en que los primeros son aquellos que involucran variables cuyas soluciones necesariamente requieren ser del tipo "entero" debido al tipo de fenómenos que inherentemente tratan. La propiedad de integralidad de las variables involucradas no es trivial ya que esto representa el corazón mismo del problema redundando en la condición de intratabilidad de los problemas del tipo "NP-Hard". Aunque este último aspecto esta fuera del tratamiento del proyecto de investigación, es importante precisar que la mayor parte de los tipos de problemas combinatorios redundan en esta misma característica de complejidad.

Todos estos problemas "NP-Hard" poseen como característica común el que ninguno puede ser resuelto de manera eficiente a través de algún algoritmo conocido, de existir, entonces podría propiciar

la solución para varios problemas de su mismo tipo de la misma manera[98]. La condición de intratabilidad que se propaga a lo largo de todos los miembros del grupo de problemas "NP-Hard", ha propiciado que la comunidad científica se halle enfrentada en encontrar un método (algoritmo) para hacerle frente. Nuestro problema de investigación pertenece a esta clase de problemas.

La administración del proceso de distribución físico presenta diversas variantes a ser atendidas en la toma de decisiones. Al nivel estratégico podemos referenciar decisiones cuyo objetivo es dar respuesta a la ubicación óptima de plantas y centros de distribución. En el nivel táctico podemos hablar de decisiones referentes a la determinación óptima del tamaño y configuración de la flota de distribución. El presente proyecto de investigación apunta hacia la administración del nivel operativo de la distribución el cual, como más adelante se definirá, se concentra en optimizar la distancia, tiempo y/o costo del proceso de distribución.

Aunque más adelante se establecerán las características que diferencian y justifican nuestro proyecto de investigación, resulta inevitable referenciar nuestro problema de investigación a esfuerzos científicos anteriores. La mejor similitud que se tiene respecto a investigaciones previas, se logra al referenciar nuestro proyecto al problema de entrega y recolección con ventana de horario (Pickup and delivery problem with time windows, PDP-TW). En general, la investigación en esta área del conocimiento se ha enfocado principalmente en la atención de modelos simplificados cuyo objetivo ha sido el servir como herramientas y bloques de construcción para la solución de instancias del mundo real [99].

En términos matemáticos, podemos decir que nuestro problema PDP-TW resulta ser un problema extremadamente difícil de resolver en términos del esfuerzo computacional requerido, aún para instancias pequeñas del problema. Más aún, Savelsberg demuestra que tan solo encontrar una solución factible (no necesariamente óptima) representa ser un problema difícil de solucionar (NP-Hard) [100]. Nuestro problema PDP-TW se puede definir como la conjunción de 3 problemas de optimización combinatoria:

1. El problema de ruteo (Routing) del TSP.
2. El problema de secuenciamiento (Scheduling) del SJSS: debido a las restricciones de

---

[98] Parker, R; Rardin, R. (1998) *Discrete Optimization*, Academic Press, New York EUA, pag 67.
[99] Cook, W; Rich, Jennifer. (1999), *A parallel cutting-plane algorithm for the vehicle routing problem with time windows,* Computational and Applied Mathematics Rice University, Houston Texas EUA, pag 2.
[100] Savelsberg, M. (1995), *Local search in Routing Problem with Time Windows,* Annals of Operations Research, Rotherdam Holanda, pag 285-305.

precedencia y las ventanas de horario que requieren ser consideradas.

3. El problema de empacamiento (Packing): por las restricciones de entrega y recolección que requieren ser tomadas en cuenta respecto a la capacidad de carga del vehículo.

Como se puede apreciar, nuestro problema PDP-TW es una variante mucho más general que el problema básico del TSP. La razón de lo anterior es básicamente a que en el PDP-TW aparecen muchas más restricciones que tomar en consideración. Así por ejemplo partiendo del PDP-TW:

1. Si se eliminan las restricciones de entrega y de recolección, entonces nuestro PDP-TW se convierte en un problema de ruteo de vehículos con restricciones de ventana de horario (Vehicle Routing Problem with Time Windows, VRP-TW).
2. Si al VRP-TW se le eliminan las restricciones de capacidad de carga del vehículo, entonces el VRP-TW se convierte en un problema de ruteo con restricciones de ventana de horario (Traveling Salesman Problem, TSP-TW).
3. Más aún, si al TSP-TW se le eliminan las restricciones de ventanas de horario, entonces el TSP-TW se convierte en el problema básico del TSP.

Por si acaso hubiera duda de la complejidad del PDP-TW, podemos referirnos a un problema cercano como es el TSP-TW con la finalidad de buscar simplificar nuestro problema original. Tsitsiklis en 1992, investiga la variante más simplificada del TSP-TW. Esta resulta ser aquella en la que los tiempos de proceso asociados a cada uno de los nodos de la red, tienen asignado un valor igual a cero. Tsitsiklis en su investigación demuestra que para esta versión simplificada del TSP-TW aún obtener una solución apenas factible, ya no digamos la óptima, resulta ser igualmente un problema NP-Hard[101].

En la década de los 90's, se han logrado grandes avances en el aspecto teórico para la solución del problema referido al PDP-TW. Ciertamente uno de los mayores avances en el tema ha sido la incorporación de características del mundo real tales como la consideración de las ventanas de horario así como las restricciones de precedencia que hasta antes habían sido "relajadas" debido a su complejidad matemática. Cuando hablamos de la relajación de un problema, nos referimos a omitir de manera temporal las restricciones de integralidad de las variables discretas involucradas en el problema. Más adelante en el apartado 8.2, se atenderá de manera específica este tema.

[101] Tsitsiklis, J. (1992), *Special cases of traveling salesman and repairman problems with time windows,* "Networks No. 22", EUA, pag 263–282.

6.2 Planteamiento formal del problema de investigación:

Ruteo de distribución para un vehículo con entrega y recolección de producto con restricciones de ventana de horario negociables (SPDP-sTW).

El problema del PDP-TW puede ser a su vez visto en dos variantes principales. Savelsbergh en 1995, distingue el PDP-TW en su variante para un solo vehículo SPDP-TW (Ruteo de distribución para un vehículo con entrega y recolección de producto con restricciones de ventana de horario), de aquella otra variante para vehículos múltiples MPDP-TW. El primer caso se trata de un TSP-TW restrictivo mientras que el segundo se trata de un VRP-TW restrictivo. Obviamente el calificativo de restrictivo, proviene de la necesidad de considerar en el problema la operación de entrega y de recolección así como la capacidad de carga del vehículo en cuestión. Nuestro proyecto se concentra en el primer caso, es decir en el SPDP-TW. Más aún, se trata de un SPDP-sTW debido a la existencia de ventanas de horario negociables para cada cliente, donde la "s" que antecede al "TW" proviene del inglés "soft". Así pues, hemos finalmente ubicado a nuestro problema de investigación taxonómicamente hasta su nivel más específico. Podemos ahora apuntar el siguiente planteamiento para nuestro problema. El problema de investigación será entonces planteado formalmente en términos de dos componentes. El primero será la definición del objetivo del problema y el segundo se referirá a las condiciones de operación para el problema. A continuación se exponen ambos:

I. Objetivo del problema:

El objetivo del problema es determinar la ruta óptima para un vehículo de distribución. Una ruta se define como la secuencia de llegada a cada uno de los clientes saliendo a partir de un centro de distribución y regresando al mismo al final de la ruta. Se define una ruta óptima como aquella que logre visitar todos los clientes de la manera más eficiente posible en términos del costo (la distancia o el tiempo) y a la vez lo haga considerando el aspecto de las ventanas de horario definidas para dar servicio y atención a cada cliente.

II. Condiciones de operación para el problema:

1. Se requiere obtener la ruta óptima que debe cubrir un vehículo de carga con capacidad finita y estando localizado en un centro de distribución.
2. El vehículo, saliendo desde el centro de distribución, debe atender a un conjunto de clientes geográficamente dispersos y luego regresar al punto de origen (centro de distribución).

3. Entre cada uno de los clientes se tiene definida una matriz de costo que resulta de establecer el nivel de gasto o distancia requerida para trasladarse de un cliente a otro.
4. Entre cada uno de los clientes se tiene definida una matriz de tiempo discreto que establece el número de secuencias de arribo que se requieren para trasladarse de un cliente a otro.
5. Para cada uno de los clientes existe un requerimiento definido por los siguientes elementos:
    a. Un volumen de producto a entregar.
    b. Un volumen de producto a recoger.
6. La cantidad de tiempo requerido para realizar ambas operaciones (entrega y recolección) en cada cliente se considera razonablemente el mismo por tratarse de cargas de trabajo similares.
7. Las ventanas de horario identificadas para cada cliente están definidas por una hora de apertura y por una hora de cierre la cual puede tener diferente amplitud dependiendo de las características de cada cliente.
8. La amplitud de la ventana de horario en cada cliente es igual a la diferencia entre la hora de cierre y la hora de apertura para la atención del cliente.
9. Las ventanas de horario identificadas en cada cliente pueden ser negociables (o re-definidas) de acuerdo a la conveniencia económica en el recorrido de la ruta que pueda presentarse al haber otros clientes geográficamente cercanos y cuyas ventanas de horario sean diferentes.
10. Las visitas a ser hechas a cada uno de los clientes, deberán ser aplicadas rigurosamente dentro de la ventana de horario que haya sido definida en el punto anterior.
11. No esta permitido llegar antes de la hora de apertura del cliente. Es decir, no se pueden considerar tiempos de espera en el punto de venta hasta el tiempo de la apertura del cliente de acuerdo a la ventana de horario.
12. Tampoco es factible llegar después de la hora de cierre del cliente. En tal caso la ruta se considera inválida y es necesario evaluar otras opciones.
13. Por lo mencionado en los puntos 11 y 12, en sustitución se recurre a re-negociar las ventanas de horario que hagan falta y así aprovechar al máximo el tiempo operativo de la ruta (ver punto 9).
14. De acuerdo a la secuencia de ruteo establecida para el vehículo, se tendrá una descarga de producto a entregar en el cliente y una carga a recoger del cliente. En todo momento deberá de ser respetada la capacidad de carga del vehículo.

Hasta aquí el planteamiento formal del problema de investigación. A continuación, en el siguiente apartado procederemos a la modelación matemática del mismo. Más adelante se expondrán las referencias bibliográficas de trabajos de investigación que han abordado previamente el tema.

6.3 Modelación matemática del problema de investigación.

El desarrollo del modelo matemático para un problema es un buen punto de inicio. De hecho a partir de una formulación hecha en términos de programación lineal, es posible llegar a estar cerca de la solución de muchos tipos de problemas. No obstante, debemos advertir que no resulta así en el caso de nuestro problema de investigación por tratarse de un problema de optimización discreta NP-Hard. Aunque para un problema de programación lineal existe una cantidad infinita de soluciones, no obstante existen ya los métodos para explotar la estructura de este tipo de problemas y lograr convertir este espacio de soluciones infinito en uno finito. En el corazón de estos métodos se tiene al método "Simplex".

Por otro lado, para los problemas de optimización discreta se tienen espacios de solución que aunque son inmensos resultan ser finitos. Paradójicamente, para el caso de esta clase de problemas no se dispone del mismo avance algorítmico tal como el que se verifica en el caso de los problemas de programación lineal. La contraparte de la programación lineal es la programación mixta entera (MIP). Esta fue introducida en la década de los 60's[102]. Se tuvieron intentos de aplicación en aquel entonces, sin embargo en ese momento el software y el hardware computacional para la solución de este tipo de problemas era apenas incipiente. Fue probablemente, la falta de capacidad y de disponibilidad de software y hardware lo que durante décadas propició la falta de aplicación.

No obstante lo anterior, podemos ahora mencionar que la situación en cuanto a estas capacidades de software y de hardware computacional han cambiado dramáticamente sobre todo a partir de 1999, año en cual se desarrollaron nuevas estrategias y mejoras tanto en los algoritmos de programación lineal así como también en los algoritmos de programación mixta entera[103].

En nuestro problema del SPDP-sTW, se requiere la construcción completa de una di-gráfica (o gráfica bi-dimensional) determinada por $G=(V,A)$, donde:

1. V esta constituido por el conjunto de los nodos de la red de distribución $\{1,...,n\}$.
2. A esta constituido por el conjunto de los arcos que forman la red de distribución $\{(i,j): i,j \in V\}$.

---

[102] Pawda, Juan. (2000), *Modelos de Investigación de Operaciones, 1era.Ed*, Limusa, México, pag 73.
[103] Bixby, Robert. (1999), *MIP: Theory and practice closing the gap*, Ilog Cplex Division, Department of Computational and Applied Mathematics, Rice University, Houston EUA, pag 3.

3. Se tiene definida una matriz $Ci,j \geq 0$ que determina el costo para trasladarse de cada nodo "i" a cada nodo "j" $\{:(i,j) \in A\}$ y definiendo que $Ci,i = \infty$ para cada $i \in V$.
4. Además se tiene definida una matriz $Ti,j \geq 0$ que determina el tiempo para trasladarse de cada nodo "i" a cada nodo "j" $\{:(i,j) \in A\}$ y definiendo que $Ti,i = \infty$ para cada $i \in V$.
5. El objetivo es formar un ciclo Hamiltoniano (tour o recorrido cerrado) de extensión o costo mínimo que logre recorrer a cada uno de los nodos visitándolos una vez a cada uno de ellos. Matemáticamente tenemos entonces que se requiere obtener una $G=(V,A)$ de tal manera que $|A| = n$ asegurando que cada uno de los nodos $v1,v2, \ldots vn \in V$ y buscando como objetivo minimizar $\sum(i,j) \in A * Cij$.

A continuación se muestra la modelación matemática detallada de nuestro problema de investigación.

Variables de Entrada (1 de 2):

Di = Demanda de producto entrante hacia el nodo "i".

$$\begin{cases} = 0 & \text{El nodo "i" solo sirve como mecanismo de transferencia topológica.} \\ \neq 0 & \text{El nodo "i" SI es un cliente para logística de entrega.} \end{cases}$$

Oi = Devolución o retorno de producto saliente desde el nodo "i".

$$\begin{cases} = 0 & \text{El nodo "i" solo sirve como mecanismo de transferencia topológica.} \\ \neq 0 & \text{El nodo "i" SI es un cliente para logística inversa.} \end{cases}$$

Restricciones topológicas de la red (Simétrica o Asimétrica):

Cij = Costo Tiempo requerido para ir desde el nodo "i" hacia el nodo "j"

E = Capacidad de carga de la unidad de transporte.

Variables de Entrada (2 de 2):

Restricciones de Ventana de Tiempo (Time Window):

Ai = Tiempo de llegada más temprano permitido para el nodo "i".

Donde: Ai = 0, para los nodos de transferencia

o para nodos sin restricciones de TW.

Pi = Tiempo de atención / procesamiento para el nodo "i".

Donde: Pi = 0, para los nodos de transferencia

Bi = Tiempo de llegada más tarde permitido para el nodo "i".

Donde: Bi = +Inf, para los nodos de transferencia

o para nodos sin restricciones de TW.

Variables de Salida (1 de 2):

$$X_{ijk} \geq 0, \leq 1, ent \quad \begin{cases} 0 \rightarrow NO\ se\ va\ desde\ el\ nodo\ "i"\ hacia \\ \quad\ el\ nodo\ "j"\ en\ la\ secuencia\ "k" \\ 1 \rightarrow SI \end{cases}$$

$T_j \geq 0$ {*Tiempo de llegada al nodo " j"*

$W_j \geq 0$ {*Tiempo de procesamie nto en el nodo " j"*

$S_j \geq 0$ {*Tiempo de salida del nodo " j"*

$Z_j \geq 0$ {*Tiempo de traslado hacia el nodo " j"*

Variables de Salida (2 de 2):

$AA_j \geq 0$ {*Tiempo de llegada al nodo "j" antes de la hora de apertura.*

$DA_j \geq 0$ {*Tiempo de llegada al nodo "j" después de la hora de apertura.*

$AC_j \geq 0$ {*Tiempo de llegada al nodo "j" antes de la hora de cierre.*

$DC_j \geq 0$ {*Tiempo de llegada al nodo "j" después de la hora de cierre.*

$$Q_j \geq 0, \leq 1, ent \quad \begin{cases} 0 \rightarrow NO\ se\ cumple\ el\ "Time\ Window"\ del\ cliente\ "j" \\ 1 \rightarrow SI \end{cases}$$

$IN_j \geq 0$ {*Carga en " kgs" que se va a entregar en el cliente " j"*

$OUT_j \geq 0$ {*Carga en " kgs" que se va a recojer en el cliente " j"*

$F_j \geq 0$ {*Carga en " kgs" del camión a la salida del cliente " j"*

Restricciones (1 de 4):

$$\sum_{i=1}^{n1}\sum_{j=1}^{n2} X_{ijk} = 1 \quad \left\{\begin{array}{l}\textit{para cada secuencia solo puede}\\ \textit{existir un solo movimiento desde un}\\ \textit{nodo "i" hacia un nodo "j"}\end{array}\right\}$$

para $k = 1..n3$

$$\sum_{i=1}^{n1}\sum_{k=1}^{n3} X_{ijk} \geq Y_j \quad \left\{\begin{array}{l}\textit{para cada nodo de entrada "j" al cual se tenga}\\ \textit{que llegar, deberá ser proveniente de cualquier}\\ \textit{nodo "i" y en cualquier secuencia "k"}\end{array}\right\}$$

para $j = 1..n2$

$$\sum_{i=1}^{n1} X_{ijk} = \sum_{i=1}^{n1} X_{jik+1} \quad \{\textit{Eliminación de Sub-Tours}\}$$

para $j = 1..n2,$

$k = 1..n3$

Restricciones (2 de 4):

$$T_j = S_{j-1} + Z_j \quad \left\{\begin{array}{l}\textit{para cada nodo "j", su tiempo de llegada}\\ \textit{es igual a l tiempo d e salida d el nodo an terior}\\ \textit{más el tie mpo de tra slado haci a el nodo "j".}\end{array}\right\}$$

para $j = 1..n2$

$$S_j = T_j + W_j \quad \left\{\begin{array}{l}\textit{para cada nodo "j", su tiempo de salida}\\ \textit{es igual a su tiempo de llegada más el}\\ \textit{tiempo de proceso en el nodo "j".}\end{array}\right\}$$

para $j = 1..n2$

$$W_j = \sum_{i=1}^{n1}\sum_{k=1}^{n3} X_{ijk} * P_j \quad \left\{\begin{array}{l}\textit{Tiempo de proceso en el nodo "j",}\\ \textit{sin importar desde cual nodo "i" provenga}\\ \textit{y en que secuencia "k" lo haga.}\end{array}\right\}$$

para $j = 1..n2$

$$Z_j = \sum_{i=1}^{n1}\sum_{k=1}^{n3} X_{ijk} * C_{ij} \quad \left\{\begin{array}{l}\textit{Tiempo de traslado hacia el nodo "j",}\\ \textit{desde el nodo "i" y en la secuencia "k"}\end{array}\right\}$$

para $j = 1..n2$

Restricciones (3 de 4):

$A_j - T_j = AA_j - DA_j$ { *tiempo de llegada al nodo" j" antes o después de la hora de apertura del "Time Window"* }

para $j = 1..n2$

$B_j - T_j = AC_j - DC_j$ { *tiempo de llegada al nodo" j" antes o después de la hora de cierre del "Time W indow"* }

para $j = 1..n2$

$AA_j + DC_j$			$1 - Q_j \leq AA_j + DC_j$ *para* $j = 1..n2$
LI	*LS*		
1	∞	$Q_j = 0$	$\infty(1 - Q_j) \geq AA_j + DC_j$
0	*0*	$Q_j = 1$	*Cump liniento d el Time Wi ndow para el Nodo "j "*

Restricciones (4 de 4):

$IN_j = \sum_{i=1}^{n1} \sum_{k=1}^{n3} X_{ijk} * D_j$ { *Cantidad de carga a entregar en el nodo "j".* }

$OUT_j = \sum_{i=1}^{n1} \sum_{k=1}^{n3} X_{ijk} * O_j$ { *Cantidad d e carga a recojer en el nodo " j".* }

$F_j = F_{j-1} - IN_j + OUT_j$ { *La carga en "kgs" con la cual se sale del cliente "j" es igual a la carga con la cual se llega, menos la q ue se va a entregar más la que se va a recojer.* }

$\therefore F_j \leq E$ { *La carga en "kgs" a la salida del cliente "j" , no debe exceder a la capacidad de carga del camión.* }

Funciones Objetivo (1 de 2)

Enfoque al cumplimiento de las Ventanas de Horario:

$$FO_{max} : \sum_{j=1}^{n2} Q_j \quad \left\{ \begin{array}{l} \textit{Maximizar la cantidad de clientes con} \\ \textit{cumplimiento en el "Time Window"} \end{array} \right\}$$

$$FO_{min} : \sum_{j=1}^{n2} AA_j + DC_j \quad \left\{ \begin{array}{l} \textit{Minimizar el tiempo total de} \\ \textit{incumplimiento del "Time Window"} \\ \textit{de todos los clientes en el sistema.} \end{array} \right\}$$

$$FO_{min} : MAY \quad \left\{ \begin{array}{l} \textit{Minimizar el máximo incumplimiento de todos} \\ \textit{los "Time Window" del Sistema.} \end{array} \right\}$$

$$FO_{min} : S_n \quad \{\textit{Minimizar el tiempo de recorrido de toda la ruta}\}$$

Funciones Objetivo (2 de 2)

Enfoque al factor económico de la ruta

$$FO\ min = \sum_{i=1}^{n1} \sum_{j=1}^{n2} \sum_{k=1}^{n3} X_{ijk} * C_{ij} \quad \left\{ \begin{array}{l} \textit{Minimizar la cantidad de distancia, tiempo} \\ \textit{o costo económico implicado para recorrer} \\ \textit{la totalidad de los clientes de la ruta.} \end{array} \right\}$$

El Proyecto de Investigación se centra en este tipo de consideración.

6.4 Aproximaciones para la solución del SPDP-sTW vía algoritmos basados en el TSP.

A partir del apartado 6.4 al 6.7, el postulante expone la revisión bibliográfica en lo referente a los trabajos de investigación previos que han sido desarrollados para resolver el problema de investigación. En principio, es conveniente puntualizar que dicha exposición no se presentó como parte del marco conceptual debido a la especificidad del problema de investigación que recién hemos formalmente planteado en el apartado 6.2. Por tal motivo, estos apartados (6.4 al 6.7), pueden ser vistos como parte complementaria del marco conceptual pero ya ahora enfocado al problema especifico de investigación.

Al revisar cada uno de los apartados (6.4 al 6.7) se podrá comprobar que aún en este momento, el postulante hará su exposición bibliográfica partiendo de variantes similares al problema específico referido al SPDP-sTW (TSP en el apartado actual), hasta llegar a la variante específica en el apartado 6.6 y finalmente para llegar a la exposición de lo que el postulante considera, el estado del arte en cuanto a la técnica de solución empleada, la cual se expone en el apartado 6.7.

Para el problema de investigación SPDP-sTW, una buena referencia para establecer una estrategia de ataque es a partir de las investigaciones que se hayan desarrollado con problemas de investigación parecidos. Garey y Johnson demostraron en 1989, que aún la tarea de encontrar una solución factible (ciclo Hamiltoniano) para el problema del TSP resulta ser un problema NP-Hard, no obstante esto resulta ser en términos relativos más fácil que lograr encontrar su solución óptima lo cual resulta por consecuencia aún más difícil [104]. En términos generales podemos decir que el problema TSP ha sido resuelto mediante métodos de solución exacta con razonable éxito de acuerdo al tamaño de las instancias que fueron experimentadas.

El método de solución exacta mayormente empleado ha sido históricamente el método de Ramificación y Corte (Branch & Cut en inglés, de aquí en adelante se le referenciará con las siglas BC). Así entonces, resulta sustentable establecer al algoritmo BC, como el método que será utilizado para calcular la cota inferior que será utilizada para contrastar la solución que será ofrecida por la propuesta de investigación doctoral basada en el algoritmo genético. Dicha definición del BC como cota inferior será expuesta más adelante en el apartado 7.3.

---

[104] Garey, M; Johnson, D. (1989), *Computers and Intractability: a Guide to the Theory of NP-Completeness,* Freeman Ed, San Francisco EUA, pag 67-89.

Una de las propuestas de investigación basadas en el BC para la solución del TSP considera la posibilidad de transformar el mismo problema del TSP en un problema de asignación (léase AP). Lo anterior se logra a través del relajamiento de las restricciones de conectividad (se omiten las restricciones para la eliminación de sub-tours) en el problema básico del TSP y convirtiéndolo en un problema de AP. Lo anterior equivale a no incluir en el sistema de restricciones $\boldsymbol{Ax} \leq \boldsymbol{b}$. todos aquellos cortes (o hiperplanos) que aseguren la eliminación de subtours en el circuito Hamiltoniano. Así, el problema del AP simplemente trata de encontrar el conjunto de sub-tours (tours no conectados) de tal forma que se asegure incluir a todos los nodos incluidos en V al mínimo costo.

Lo interesante del enfoque basado en la solución del problema AP aplicado al TSP (y también al PDP-TW), radica en que en el primero existe una propiedad poco común entre los problemas de redes y es que en principio las restricciones de integralidad no requieren ser impuestas ya que su solución mediante programación lineal siempre arrojará implícitamente soluciones enteras. Lo anterior aplicado al problema básico del TSP, equivaldría a disponer de un método de solución basado tan sólo en programación lineal dejando de requerir la aplicación de un algoritmo de enumeración parcial basado en el BC. Así, si la solución obtenida en el problema del AP mediante la relajación del problema original incluye un solo ciclo Hamiltoniano (cerrado), pues entonces se tiene ya la solución óptima. Desafortunadamente esto rara vez ocurre en la práctica.

La dificultad del enfoque anterior se presenta al intentar integrar los cortes en el sistema $\boldsymbol{Ax} \leq \boldsymbol{b}$. que vayan restringiendo la aparición de sub-tours en la solución. Normalmente lo anterior no sería un problema insalvable ya que son bien conocidos los algoritmos requeridos para generar estas restricciones (o hiper-planos). El problema se presenta debido a que al agregar dichas restricciones se pierde para siempre la propiedad inherente del AP que permitía solucionar sus variables enteras mediante tan solo la aplicación de programación lineal.

En cualquier caso y habiendo asumido la perdida de la propiedad anterior, es importante considerar que la solución ofrecida por el planteamiento del AP mediante programación lineal, resulta ser la cota inferior para el problema original del TSP. Esto quiere decir, que en la misma corrida anterior y sin requerir de esfuerzo computacional adicional, es posible disponer de la matriz de costos reducida (o también llamada solución dual) con la cual es posible verificar la conveniencia relativa (penalización adicional) que cada uno de los arcos (i,j) tienen al ser incluidos en la solución del problema.

Lo anterior se puede explicar de una manera aún más formal diciendo que la cota inferior ***LB*** para un arco ***xij = 1*** se recalcula como: ***LB (para xij=1) = LB + ĉij, donde ĉij*** es el costo reducido o penalización implicada por agregar el arco ***xij*** en la solución final correspondiente al ciclo hamiltoniano. Por lo tanto resulta obvio decir, que aquellas variables ***xij*** que tengan calculado un costo reducido ***ĉij*** menor, serán las mismas que tengan mayor probabilidad de aparecer en la solución final. Desafortunadamente respecto a la exposición desarrollada hasta aquí, es necesario decir que según el trabajo de Dell'Amico y de Martello en 1997, se concluye que en términos generales las cotas obtenidas del problema original mediante su relajación vía la aplicación del AP no resultan ser eficientes en el sentido de estar ajustadas al problema original del TSP, matemáticamente hablando [105]. Por tanto, es predecible que ocurra lo mismo para nuestro problema de investigación SPDP-sTW.

Como complemento a la estrategia anterior desarrollada por Dell'Amico y Martello, en 1996 Balas y Simonetti, implementaron una metodología basada en el trabajo previo que Lawler y Lenstra desarrollaron en 1986 para la generación de cortes que lograban restringir la aparición de sub-tours en el TSP (sub-tours elimination constraints) [106]. Esta última propuesta, no establecía su estrategia a partir de una relajación basada en el problema del AP sino tan solo relajaba las restricciones para la eliminación de los sub-tours.

Una última investigación fue desarrollada en 1999 cuando Focacci y Milano implementaron una relajación del TSP mediante un AP pero incluyeron una estrategia diferente para implementar los cortes requeridos para restringir los sub-tours. En lugar de incluir los cortes como parte del sistema de restricciones $\boldsymbol{Ax} \leq \boldsymbol{b}.$, decidieron hacerlo como parte de la función objetivo a través de la aplicación de técnicas de relajación de Lagrange [107].

---

[105] Dell 'Amico, M; Martello, S. (1997), *Linear assignment: Annotated Bibliographies in Combinatorial Optimization,* Wiley & Sons editors, EUA, pag 355–371.
[106] Balas, E; Simonetti, N. (1996), *Linear time dynamic programming algorithms for some new classes of restricted TSP's*, "Management Science Research Report 617", Graduate School of Industrial Administration, Carnegie Mellon University, EUA.
[107] Focacci, F; Milano, M. (1999), *Solving Tsp with time windows with constraints*, "INFORMS Journal of Computing, ICLP 99", International Conference on Logic Programming, EUA.

6.5 Aproximaciones para la solución del SPDP-sTW basadas en el TSP-TW y en el VRP-TW.

No hay duda de que el problema TSP-TW es mucho más parecido a nuestro problema de investigación que el TSP. Brevemente podemos resumir que varios algoritmos han sido propuestos para el TSP-TW. Ellos van desde la aplicación de heurísticas tales como las de inserción de Lin-Kernighan. Otros algoritmos están basados en la aplicación de métodos de solución exacta basados en la aplicación de BC y más aun otras en la aplicación de la programación dinámica con relajación de estado de transición. De lo que si no hay duda es que el TSP-TW es un problema NP-Hard, lo cual implica que no exista un algoritmo que pueda resolver el problema en un tiempo de ejecución polinomial y que al mismo tiempo la programación dinámica tradicional esté propensa a fallar [108].

Ahora bien, resulta que problema más parecido al SPDP-sTW es el VRP-TW. Ambos son considerados problemas relevantes en la investigación actual y con una herencia compartida respecto al problema del TSP. Así pues, identificar que se ha desarrollado en el estudio del VRP-TW es una buena estrategia. En 1984 Solomon desarrolló 87 instancias de prueba para el VRP-TW. De todas ellas, la más grande es de 100 clientes [109]. De hecho, de 17 instancias que aún permanecían sin poder ser resueltas, 10 de ellas fueron apenas resueltas en 1999 por Cook y Rich de la Universidad de Rice en Houston [110]. No obstante todo el esfuerzo que se ha desplegado en términos de investigación en el VRP-TW en estos últimos años, aún existen 7 instancias que no han podido ser resueltas en la literatura científica.

Es importante precisar que la estrategia de solución implementada por Cook y Rich se basa en un algoritmo cuya base es programación lineal y que se apoya en un motor de búsqueda el cual a su vez está apoyado en un generador de cortes (o hiperplanos) derivados de un algoritmo de BC. El algoritmo de BC implementado tiene la particularidad de que incluye un procedimiento para la generación de columnas aplicadas de manera incremental y progresiva sobre el problema maestro. A este enfoque de generación incremental de columnas, se le denomina en inglés Branch & Price por tratarse de una derivación del método BC tradicional. Esto último se explica más adelante de manera breve.

---

[108] Fisher, Marshall. (1995), *Overview over optimization models in transportation,* Handbooks in Operations Research and Management Science, North Holland, EUA, Pag 10.

[109] Solomon, M. (1984), *On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints,* "Report 83-05-03", The Wharton School, University of Pennsylvania EUA.

[110] Cook, W; Rich, Jennifer. (1999), *A parallel cutting-plane algorithm for the vehicle routing problem with time windows,* Computational and Applied Mathematics Rice University, Houston EUA, pag 5.

La estrategia utilizada por Cook y Rich para la identificación de las restricciones (o cortes) se deriva de las heurísticas de eliminación de sub-tours de diversos tipos tales como:

1. k=1: desarrollados por Dantzig, Fulkerson, y Johnson en 1954.
2. k=2: desarrollados por Kohl, Desrosiers, Madsen, Solomon en 1992.
3. "k-path" para k>=3: desarrollado por Karger en 1993 [111].

El enfoque consiste en iniciar con un problema maestro limitado en cuanto a la cantidad de rutas que van a ser evaluadas en la primera iteración (recordemos que el VRPTW trata de un problema con múltiples rutas). Así de esta manera, se aplica un procedimiento recursivo en el cual después de ejecutar el problema maestro mediante programación lineal, se utilizan entonces los costos reducidos de los arcos ***(i,j)*** (valores duales) a fin de ir progresivamente incorporando el resto de los nodos y por ende de las rutas en el problema maestro. Cada nueva ruta que se va incorporando, representa una nueva columna a ser adicionada en la base numérica del problema. A este método en el que se combina la utilización de generación de cortes a la vez que los costos reducidos de las variables, se le denomina "Método de ramificación y costo" (o en inglés Branch & Price, BP). El método BP ha resultado ser una buena alternativa para resolver los problemas referentes al VRP-TW. Mayor detalle respecto al método puede ser revisado en la obra de Cook [112].

Es oportuno finalizar el apartado mencionando que básicamente en el método BP, cada una de las columnas que se van agregando a la base numérica del problema resulta ser una ruta asociada a cada uno de los vehículos que están siendo integrados de manera gradual al problema principal. Dado que nuestro problema de investigación SPDP-sTW está enfocado en dar tratamiento a un solo vehículo, por tanto es predecible que el BP no resulte ser una estrategia de solución conveniente a ser propuesta.

---

[111] Karger, D. (1993), *Global min-cuts in RNC and other ramications of a simple min-cut algorithm,* "Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms", EUA, pag 84-93.
[112] Cook, W. (1999), *A parallel cutting-plane algorithm for the vehicle routing problem with time windows,* Computational and Applied Mathematics Rice University, Houston Texas, pag 2.

6.6 Revisión algorítmica referente a trabajos de investigación previos para el problema PDP-TW.

Las investigaciones en el problema PDP-TW en términos generales iniciaron hace apenas 10 años. Aunque más adelante se establecerán las diferencias especificas que delimitan nuestro problema de investigación (SPDP-sTW) respecto al problema general del PDP-TW, a continuación se muestra una breve cronología de las investigaciones más relevantes realizadas con respecto al tratamiento de este problema. Esto tiene el propósito de establecer los diversos enfoques que han sido explorados para dar solución al problema en cuestión:

1. El problema del PDP-TW fue apenas por primera vez atendido en el año de 1993 por el trabajo de Dumas, Desrosiers y Soumis. Ellos propusieron varias estrategias para atacar el problema mediante una formulación matemática basada en los problemas típicos de flujo de redes .

2. Más adelante tenemos el trabajo de Applegate y Bixby, los cuales en 1994 se dieron a la tarea de investigar y explotar la estructura matemática del problema con la finalidad de lograr obtener el sistema de restricciones o cortes que mejor lograba resolver este tipo de problemas.

3. Bruggen, Lenstra y Schuur en 1993 experimentan con dos alternativas para la solución del problema. El primer experimento fue a través de la aplicación de un procedimiento de búsqueda local basado en el método de intercambio de Lin-Kernigham. El segundo se desarrollo a partir de la aplicación de la meta-heurística "Enfriamiento Simulado" (Simulated Annealing) desarrollado en 1985 por Kirkpatrick y Lawler. Ambos experimentos fueron probados exitosamente con instancias de no más allá de 50 nodos y con ventanas de horario estrechas. "Arriba de 50 nodos y con ventanas de horario amplias, los resultados no fueron tan buenos y la principal desventaja empezó a ser el tiempo computacional ya que el procedimiento resultó ser sumamente lento requiriendo arriba de 10 minutos de tiempo computacional" [113].

[113] Mitrovic, Snezana. (1998), *Pickup and Delivery Problem with Time Windows,* "Technical Report SFU CMPT TR 1998-12", Canada, pag 31.

4. Van Eijl en 1995 propuso un algoritmo de cortes basado en el BC. Sus resultados computacionales fueron probados en instancias hasta de 15 nodos y con tiempos de ejecución computacional muy elevados en términos relativos, aunque no se reporta en la investigación algún valor específico de tiempo [114].

5. Coth y Vigo en 1995-1996 presentan otro algoritmo también basado en la aplicación de un procedimiento meta-heurístico. En esta ocasión se aplica el procedimiento desarrollado por Glover en 1987 denominado "Búsqueda Tabu" (o Tabú Search). Los resultados son modestos aunque el principal problema seguía siendo que el procedimiento requería de considerables tiempos de ejecución computacional. El algoritmo fue implementado en dos fases, la primera fase correspondía a la aplicación de una heurística de inserción. La segunda fase corresponde a la fase de mejoramiento mediante la aplicación de la meta-heurística "Tabu Search". Los tiempos de ejecución reportados para instancias clasificadas como de "gran escala" (arriba de 100 nodos) fueron de alrededor de una hora [115].

6. Resulta obligado referenciar las aportaciones de Dumas, Desrosiers y Solomon, los cuales en 1995 presentan un algoritmo basado en la aplicación del método de descomposición de Dantzig-Wolfe (1956). Este método de descomposición (o también llamado de "generación de columnas"), tiene su utilidad en la posibilidad de integrar las teorías de programación lineal de Dantzig y Wolfe en la obtención de cotas más ajustadas para la solución de problemas de programación entera. Lo anterior redunda en la obtención de un árbol de búsqueda mucho más pequeño a tener que ser explorado por los algoritmos de BC. En términos generales, se le ha referenciado a esta propuesta algorítmica como la mejor alternativa disponible para aproximar soluciones al problema. Los experimentos computacionales reportan instancias de entre 100 y 200 nodos requiriendo tiempos computacionales de entre 20 y 30 minutos[116].

---

[114] Eijl Van, C. (1995), *A polyhedral approach to the delivery man problem,* "Technical Report 95–19", Department of Mathematics and Computer Science, Eindhoven University of Technology, The Neatherlands, pag 12-14.
[115] Mitrovic, Snezana. (1998), *Pickup and Delivery Problem with Time Windows,* "Technical Report SFU CMPT TR 1998-12", Canada, pag 32-33.
[116] Dumas, Y; Desrosiers, J; Gelinas, E; Solomon, M. (1995), *An algorithm for the traveling salesman problem with time windows,* "Operations Research 43(2)" , EUA, pag 23–25.

7. En 1996 Balas y Simonetti, presentan un algoritmo de solución exacta basado en programación dinámica. Su estrategia se basa primordialmente en la implementación de algoritmos para reducción del espacio solución vía relajamiento de los estados de transición típicos en la aplicación de las técnicas de programación dinámica. La única consideración es que el algoritmo solo funciona adecuadamente en instancias de hasta 40 o 50 nodos y con poca incidencia en el traslape de las ventanas de horario. Los tiempos reportados en el experimento son arriba de 2 horas de ejecución computacional[117]. Dichas investigaciones apuntan a concluir que para poder aplicar el algoritmo de solución exacta basado en programación dinámica se requiere que exista poca probabilidad de que se presenten conflictos entre las ventanas de horarios de los clientes a una misma hora. Se concluye en el reporte que la principal desventaja de la programación dinámica es la inherente influencia que existe en la forma en como se implemente la discretización del tiempo en el algoritmo lo cual limita su aplicabilidad debido a las restricciones de recursos computacionales.

8. En 1997 Mingozzi, Bianco, y Ricciardelli propusieron una metodología híbrida basada en la aplicación de BC para la definición de las acotaciones del espacio solución pero complementándolo con una variante de la programación dinámica en la cual se aplica una relajación de los estados de transición para explotar el espacio solución obtenido previamente en la fase de BC. Aquí se presentan instancias resueltas hasta de 120 nodos aunque no se incluyen los resultados de los tiempos computacionales[118].

9. Ascheuer, Jünger y Reinelt en el año 2000 aplican una estrategia diferente. En esta ocasión ellos utilizan un algoritmo generador de cortes (hiperplanos) basado en el BC. El algoritmo fue aplicado en una variante del TSP pero solo con restricciones de precedencia. Es bien sabido que las restricciones de precedencia y las restricciones de ventana de horario son bastante similares. La estrategia basada en el BC fue exitosa (tiempos debajo de 20 minutos). No obstante, en este experimento no fue posible identificar cual sería el resultado al aplicar el algoritmo propuesto a un problema modificando las restricciones de

---

[117] Balas, E; Simonetti, N. *Linear time dynamic programming algorithms for some classes of restricted tsp's*; Technical Report MSRR No. 617; Carnegie Mellon University; Pittsburgh EUA 1997; pag 10-11.

[118] Mingozzii, A; Bianco, L; Ricciardelli, S. (1997), *Dynamic programming strategies for the travelling salesman problem with time windows and precedence constraints,* "Operations Research No. 45", EUA, pag 277.

precedencia por las del ventana de horario [119]. Dichos resultados se comprobarían un año después cuando Ascheuer, Fischetti y Grotschel desarrollarían un experimento para ese propósito específico (ver siguiente inciso 10).

10. El último trabajo revisado en la investigación bibliográfica fue desarrollado en el año 2001 por Ascheuer (Alemania), Fischetti (Italia), y Grotschel (Alemania). Esta investigación se centra en la aplicación de una metodología basada en el BC pero aplicándolo para una variante asimétrica del TSP (es decir en el ATSP). Ellos inician el planteamiento de su investigación sobre la base de una amplia revisión bibliográfica de los trabajos previos hechos sobre el problema del ATSP-TW. Como es de esperarse, el trabajo de ellos se concentra en el tratamiento de las ventanas de horario (TW) en el ATSP. Los resultados de su investigación se presentarán en el siguiente apartado 6.7.

[119] Ascheuer, N; Jünger, M; Reinelt, G. (2000), *A branch & cut algorithm for the asymmetric Traveling Salesman Problem with precedence constraints,* "Computational Optimization and Applications 17(1)", EUA, pag 61–84.

6.7 Identificación del estado del arte para el problema de investigación.

Como ya se mencionó antes, el problema del TSP-TW es un problema relativamente cercano en definición al PDP-TW. Nuestro punto de partida será el último trabajo presentado en la cronología expuesta del apartado anterior. Ascheuer, Fischetti, y Grotschel en el 2001 mencionan en su trabajo que resulta interesante la poca atención que se le ha brindado a este problema en particular. Especialmente mencionan, que los algoritmos de solución exacta basados en el BC han tenido un protagonismo menor tanto para la versión simétrica (TSP-TW) así como para la versión asimétrica (ATSP-TW).

Como antecedente en este rubro mencionan a Baker, el cual en la década de los 80's resolvió mediante algoritmos de solución exacta de BC instancias de hasta 50 nodos con ventanas de horario moderadamente cerradas (matemáticamente hablando, entre más cerradas sean las ventanas de horario más pequeño es el espacio solución a ser explorado y por tanto más fácil es el problema en resolver). Además estas instancias estaban caracterizadas por ventanas de horario con un porcentaje bajo de traslape[120]. No se dispone de registros acerca de la velocidad computacional que se obtuvo en el experimento de Baker tomando en cuenta el avance de los procesadores matemáticos de aquella época.

De la revisión anterior, ellos parten de la existencia hasta la fecha de tres modelos basados en programación mixta entera (MIP), y solucionados mediante la aplicación de algoritmos de BC. Hasta antes de su investigación, ellos afirman no tener conocimiento de cual de estos tres modelos es el que ofrece mejor perspectiva computacional. Así pues, su proyecto de investigación consistió en determinar un análisis comparativo entre estos tres para comprobar cual era el mejor modelo. Ascheuer, Fischetti, y Grotschel precisan que para el TSP-TW, no resulta fácil comparar diferentes algoritmos para su solución ya que en principio no se dispone de instancias que puedan servir como "benchmark" estándar del problema en cuestión.

Habitualmente la dificultad computacional para la solución de las instancias de los problemas ha sido tipificada en términos del tamaño (cantidad de nodos) del problema. Sin embargo es obvio precisar que la dificultad de este tipo de problemas depende fuertemente de la estructura de las ventanas de horario que se definan. En general, el resultado de los experimentos con el TSP-TW hechos por Ascheuer, Fischetti, y Grotschel, arrojaron que este problema es particularmente difícil de resolver para aquellas instancias en las que la cantidad de nodos activos que contengan restricciones de ventana de

[120] Ascheuer, N; Fischetti, M; Grotschel, M. (2001), *Solving ATSP with time windows by branch-and-cut*, Springer-Verlag, EUA.

horario estén por arriba del 50%. En lo referente al tamaño de los problemas que fueron probados en el experimento solo uno de los tres modelos revisados pudo resolver instancias de más de 69 nodos [121].

Una vez verificado cual de los tres modelos es el mejor, entonces Ascheuer, Fischetti, y Grotschel desarrollaron una versión especializada sobre la base del mejor modelo de los tres. En esta versión, ellos implementaron una serie de al menos 15 procedimientos heurísticos con la finalidad de mejorar aún más el desempeño del algoritmo original. Se corrieron instancias de hasta 233 nodos. La instancia de 69 nodos requirió de 5.95 minutos de tiempo computacional. El resto de todas las instancias mayores ocuparon más de 5 horas en la solución computacional (no se especifica cuanto más de 5 horas); excepto una de 127 nodos que pudo resolverse en menos de 5 horas.

Ascheuer, Fischetti, y Grotschel afirman que es muy difícil predecir que tan difícil será una instancia particular del TSP-TW. Como ejemplo (y paradoja) de lo anterior muestran una instancia de 127 nodos que ocupo 3:49.82 horas de tiempo computacional, mientras que una de 43 nodos no pudo resolverse en menos de 5 horas. Esta instancia en particular de 43 nodos, resultó bastante difícil ya que en 5 horas de ejecución computacional, apenas se alcanzo un porcentaje de optimalidad relativa del 90.84% con más de 109 mil ejecuciones de programación lineal (llamadas al método "Simplex") y más de un millón de hiperplanos generados por el algoritmo BC. Ascheuer, Fischetti, y Grotschel, en base a las experiencias computacionales, concluyen que las instancias del ATSP-TW en el límite de hasta 50 o 70 nodos, pueden ser resueltas hasta la solución óptima por el algoritmo BC.

A continuación en la Tabla 6.1, se presentan los resultados de dicho experimento. La segunda columna de la tabla corresponde al número de nodos que contienen cada una de las instancias que fueron aplicadas en el experimento. La última columna de la tabla corresponde a la cantidad de minutos computacionales que fueron requeridos en el experimento para que el algoritmo BC, lograra resolver cada una de las instancias respectivamente. Como puede ser comprobado en la última columna de la tabla, es posible darse cuenta que existen varias instancias que fueron utilizadas en el experimento, para las cuales no fue posible converger en una solución óptima aún después de 5 horas de ejecución computacional.

[121] Ascheuer, N; Fischetti, M; Grotschel, M. (2001), *Solving ATSP with time windows by branch-and-cut*, Springer-Verlag, Germany, pag 297-299.

Tabla 6.1 Resultados del experimento para la solución del ATSP-TW a través de BC [122].

			Solution		Root			BC-Trll				
	n	A	Opt	gap	bounds	gap	Qual	#N	Lev	#cuts	#LP	CPU
rbg010a	12	54	149	0 00	[148, 149]	0 68	99 32	2	1	22	14	0:00 12
rbg017	17	122	148	0.00	[148, 150]	0.00	100.00	4	2	43	43	0:00 82
rbg017.2	17	200	107	0 00	107	0.00	100.00	0	0	2	2	0:00 03
rbg016a	18	79	179	0 00	[177, 179]	1 13	98 87	2	1	8	8	0:00 20
rbg016b	18	167	142	0 00	[133, 142]	6.77	93 66	76	13	288	329	0:08 80
rbg017a	19	176	146	0 00	146	0 00	100 00	0	0	3	5	0:00.12
rbg019a	21	71	217	0.00	217	0.00	100.00	0	0	0	1	0:00 03
rbg019b	21	211	182	0.00	[180, 185]	2.78	98 88	820	29	623	1645	0:54 57
rbg019c	21	229	190	0.00	[182, 190]	4 39	95.56	58	9	360	325	0:08.72
rbg019d	21	156	344	0.00	[343, 344]	0.29	99 71	2	1	40	31	0:00 75
rbg021	21	229	190	0.00	[182, 190]	4.40	95 60	58	9	360	325	0:08.75
rbg021.2	21	237	182	0.00	182	0.00	100.00	0	0	32	10	0:00 22
rbg021 3	21	256	182	0 00	[178, 190]	6 74	97 75	340	17	788	869	0:27.15
rbg021.4	21	264	179	0 00	[177, 190]	7.34	98.87	72	10	189	237	0:05 82
rbg021.5	21	268	169	0.00	[167, 169]	1 20	98 80	76	10	199	264	0:06.63
rbg021.6	21	358	134	0 00	[133, 134]	0 75	99 24	2	1	55	54	0:01 38
rbg021.7	21	375	133	0.00	[128, 133]	3 91	96.09	24	6	131	166	0:04.30
rbg021.8	21	380	132	0.00	[129, 136]	5 43	97 67	254	17	369	672	0:17.40
rbg021.9	21	380	132	0.00	[128, 138]	7.81	96.87	320	15	620	948	0:26.12
rbg020a	22	95	210	0 00	210	0.00	100 00	0	0	0	1	0:00 20
rbg027a	29	487	268	0.00	[266, 268]	0.75	99 24	6	3	174	59	0:02.25
rbg031a	33	388	328	0.00	328	0 00	100 00	0	0	97	37	0:01 70
rbg033a	35	421	433	0.00	433	0.00	100 00	0	0	45	24	0:01 85
rbg034a	36	535	403	0.00	[401, 403]	0 50	99 50	2	1	48	16	0:00 98
rbg035a	37	477	254	0.00	254	0 00	100 00	0	0	121	34	0:01.83
rbg035a.2	37	940	166	0 00	[158, 215]	36.08	94.94	96	15	1253	698	1:04 80
rbg038a	40	486	466	0 00	[466, 474]	0.00	100.00	13204	40	20586	38855	70:32.23
rbg040a	42	539	386	0.00	[355, 393]	10.70	91 28	1756	26	2007	5605	12.31 82
rbg041a	43	628	[382, 417]	9 16	[361, 418]	15 79	84.49	23396	35	46846	109402	—*
rbg042a	44	762	[409, 435]	6.35	[394, 444]	12 69	89 59	22300	43	49238	99990	—*
rbg048a	50	1288	[455, 527]	15 82	[454, 527]	16.08	83 92	25222	49	103883	77604	—*
rbg049a	51	1083	[418, 501]	19 86	[408, 503]	23.28	77 20	17486	52	52679	61295	—*
rbg050a	52	1629	414	0 00	[414, 430]	0.00	100.00	6	2	392	121	0:18.62
rbg050b	52	1175	[453, 542]	19 65	[447, 548]	22 59	78.74	8600	25	30094	37337	*
rbg050c	52	1396	[509, 536]	5.30	[507, 539]	6 31	94.28	25184	35	99795	94976	—*
rbg055a	57	765	814	0 00	[813, 814]	0 12	99 88	2	1	229	68	0:06.40
rbg067a	69	843	1048	0.00	[1047,1048]	0.10	99 90	2	1	176	56	0:05.95
rbg086a	88	927	[1049, 1052]	0 28	[1042,1062]	1.92	99.04	12208	30	7317	26088	—*
rbg092a	94	1367	[1102, 1111]	0.81	[1084,1111]	2.49	97 51	8828	39	12502	27938	—*
rbg125a	127	1824	1410	0.00	[1402,1412]	0.71	99.42	56	6	654	293	3:49.82
rbg132	132	1575	[1348, 1400]	3.86	[1323,1400]	5 82	94.17	7628	32	5630	20294	—*
rbg132.2	132	3126	[1069, 1125]	5.24	[1053,1128]	7.12	93.16	4336	26	5001	13939	—*
rbg152	152	2125	[1770, 1792]	1.24	[1759,1792]	1.87	98 12	5038	28	7263	13732	—*
rbg152.3	152	6191	[1525, 1594]	4.53	[1521,1596]	4.93	95 20	2558	37	9340	8817	—*
rbg172a	174	2837	[1787, 1897]	6 15	[1777,1897]	6.75	93.24	3434	33	7752	11139	—*
rbg193	193	3050	[2388,2452]	2.68	[2386,2452]	2 76	97.23	2790	28	6666	8254	—*
rbg193.2	193	6031	[1981, 2093]	5.65	[1969,2093]	6 29	93 70	1726	21	7542	7602	—*
rbg201a	203	3287	[2159, 2296]	6.34	[2158,2296]	6.39	93 60	3282	35	3611	7395	—*
rbg233.2	233	7588	[2152, 2304]	7 06	[2146,2304]	7.36	92.64	1106	31	10520	5111	—*
rbg233	233	3766	[2647, 2786]	5.25	[2635,2786]	5 73	94 26	1200	25	11927	7254	—*

—* : time limit of 5 CPU hours exceeded

Lo antes concluido por Ascheuer, Fischetti, y Grotschel para el problema del TSP-TW no es trivial, ya que esto mismo y con mayor razón aplica para nuestro caso particular del SPDP-sTW. Es decir, el SPDP-sTW requiere más consideraciones a tomar en cuenta que las que requiere el problema del ATSP-TW.

[122] Ascheuer, N; Fischetti, M; Grotschel, M. (2001), *Solving ATSP with time windows by branch-and-cut*, Springer-Verlag, Germany, pag 259-310.

Finalmente, es importante precisar que la variante de nuestro proyecto de investigación referida a la condición de tener que considerar ventanas de tiempo negociables (restricciones de ventana de tiempo suaves) para cada cliente, es una variante relativamente poco explorada en el ámbito de la investigación bibliográfica. Tenemos como caso más reciente el trabajo conjunto de Pesant, Gendreau, Potvin, y Rousseau desarrollado en 1999. Es importante precisar que el trabajo de ellos proviene de una rama de la investigación completamente diferente a las que hemos venido revisando hasta ahora. Es decir no se trata de una metodología basada en BC ni tampoco en programación dinámica o metaheurísticas. Se trata de un enfoque relativamente nuevo y de la corriente europea denominada Programación Restrictiva (Constraint Programming, CP).

No obstante, es probado mencionar que CP ha tenido dificultades para dar soluciones competitivas respecto a sus contrincantes (por ejemplo el BC). El trabajo de Focacci y Milano prueba lo anterior. Ellos en el 2000, propusieron una metodología híbrida basada en el CP y en el BC para resolver el problema básico del TSP. Al final ellos concluyen que aunque su metodología superaba por mucho las soluciones ofrecidas por otras investigaciones basadas en implementaciones puras del CP, no obstante seguían estando lejos de ser comparables respecto a la calidad de las soluciones ofrecidas por los robustos esquemas de corte basados en el BC[123]. En realidad la contribución de ellos fue proponer un esquema basado en el CP el cual ofrecía como ventaja la flexibilidad de que con pocas modificaciones, el mismo modelo básico podría dar tratamiento a la consideración de ventanas de horario simple y a la vez ventanas de horario múltiples ofreciendo de esta manera la posibilidad de poder negociar con el cliente.

A continuación en la Tabla 6.2, se resume la revisión bibliográfica que se ha venido exponiendo en los apartados 6.5 al 6.7 respecto a los estudios de investigación previos realizados para atender la solución computacional de problemas similares al nuestro. Con dicha tabla resumen damos por concluida la exposición de los proyectos de investigación previos desarrollados en el tema.

[123] Focacci, Filippo; Milano, Michela. (2000), *Solving TSP with Time Windows with Constraints*, Dip. Ingegneria, University of Ferrara, Italia, pag 4.

Tabla 6.2 Resumen comparativo de los estudios de investigación previos.

Investigador	Año	Taxonomía del Problema	Algoritmo	Dimensión	Ventanas de Horario	Desempeño Computacional
Baker	1988	ATSP-TW	BB	≤50 nodos	Cerradas, Poco Traslape	
Dumas, Desrosiers y Soumis	1993	PDP	Flujo de Redes			
Bruggen, Lenstra y Schuur	1993	PDP-TW	Herrística Lin-Kernigham y Enfriamiento Simulado	≤50 nodos	Cerradas, Poco Traslape	Instancias fuera de estas condiciones requerían ≥10 Mins
Applegate y Bixby	1994	TSP-TW	BB			
Van Eijl	1995	VRP	BB	≤15 nodos	Abiertas	
Coth y Vigo	1995	PDP-TW	Meta-Heurística Tabu Search	50 ≤n ≤100 nodos		Instancias para n ≥100 nodos requerían ≥1 Hora
Dumas y Solomon	1995	VRP-TW	Método de Descomposición de Dantzig-Wolfe	100 ≤n ≤200 nodos	Cerradas, Poco Traslape	Instancias para 100 ≤n ≤200: 20 ≤mins ≤30
Balas y Simonetti	1996	TSP-TW	Programación Dinámica	40 ≤n ≤50 nodos	Cerradas, Poco Traslape	≥2 Horas
Mingozzi, Bianco, y Ricciardelli	1997	TSP-TW	Hibrido: BB y Programación Dinámica	≤120 nodos		
Ascheuer, Jünger y Reinelt	2000	ATSP-TW	BB con generación de hiperplanos	50 ≤n ≤70 nodos	Abiertas y con bajo % de incidencia de TW	10 ≤mins ≤20
Ascheuer, Fischetti, y Grotschel	2001	ATSP-TW	Hibrido: BB con Heurísticas de Intercambio	70 ≤n ≤233 nodos	Cerradas, Poco Traslape	5 mins ≤Tiempo ≤5 hrs

6.8 Comentarios finales: delimitación y diferenciación de la variante del problema de investigación (SPDP-sTW).

Para las conclusiones del presente capítulo buscaremos a la vez cubrir los siguientes 4 objetivos:

a. Resumir las propiedades específicas a ser consideradas en el problema de investigación.
b. Delimitar el problema de investigación.
c. Diferenciar el problema respecto a otros proyectos de investigación previos.
d. Contribuir a ofrecer elementos para la justificación del proyecto de investigación

A continuación se enumeran las características relevantes de nuestro problema de investigación:

1. El algoritmo identificará aquellos clientes que por su cercanía geográfica convenga, en su caso, renegociar su ventana de horario. Este elemento resulta sumamente útil en lo referente a las aplicaciones de ruteo en la práctica de las empresas, ya que de esta manera se evita generar recorridos de ruta costosos sobre la base de un supuesto nivel de servicio en el cliente que potencialmente resulta mejor re-negociar.

2. De lo expuesto en el punto anterior, es requerido desarrollar un análisis para medir el nivel de incidencia con el cual es conveniente renegociar las ventanas de horario con los clientes y así entonces verificar el nivel de esfuerzo requerido por parte de la empresa y a la vez el nivel de apego a las ventanas de horario originalmente deseada por el cliente.

3. La mayor parte de las investigaciones en el PDP-TW se centran en aquellos casos en los que se tiene definida una sola actividad a realizar en cada nodo, es decir se recoge producto en un lugar y se entrega el producto en otro. En el caso del proyecto de investigación nuestro, nos estamos concentrando en aquella situación en la cual se llega con un cliente al cual se le tiene que entregar un pedido. Más aun a este mismo cliente se le tiene que recoger todo aquel producto que ya no vaya a ocupar.

4. La devolución de producto mencionado en el punto anterior, puede obedecer a varias razones tales como:
    i. Producto defectuoso o fuera de la fecha de caducidad.
    ii. Producto devuelto a solicitud del cliente.

iii. Empaque retornable (o también denominada Logística Inversa).

5. La mayor parte de las investigaciones en el VRP-TW se enfocan solo en asegurar que la capacidad de carga del vehículo utilizado, sea suficiente para albergar la suma total del producto que se va a entregar a largo de la ruta. Sin embargo, en nuestro problema ocurre que en cada uno de los clientes a ser visitados en la ruta, existe la posibilidad de entregar pero también de recoger producto, lo cual naturalmente ocupa un volumen de carga.

6. La circunstancia anterior origina que cada vez que se termine la operación con un cliente, sea necesario re-calcular el saldo del espacio disponible para cargar en el vehículo. Esto quiere decir que la restricción de capacidad de carga debe ser verificada dinámicamente a la salida de cada uno de los clientes a los cuales se va a llegar. La situación anterior invalida la posibilidad de manejar tan solo restricciones de agregación a nivel de ruta tal y como sucede en el caso del VRP-TW.

7. Los problemas de PDP-TW generalmente consideran una sola carga a poder ser atendida por evento. Es decir, desde el momento en que se carga el producto hasta el momento en que se descarga, el vehículo solo está sirviendo a un solo cliente. En nuestro problema de investigación sucede que mientras se ejecuta el proceso de carga y descarga, el contenido del vehículo está ofreciendo servicio al producto que a muchos clientes se les va a entregar o simultáneamente a muchos clientes se les acaba de recoger.

8. Las ventanas de horario que nuestro proyecto de investigación requiere manejar tanto para la entrega de producto como también para la recolección del mismo en el sitio especificado por el cliente, son normalmente mucho más amplias que lo que habitualmente se maneja en los casos típicos del problema PDP-TW.

9. La razón de lo anterior se explica primero debido a que en nuestra instancia particular a atender, los clientes normalmente están dispuestos a ser atendidos en períodos de tiempo muy amplios ya que en condiciones especiales existe la posibilidad de ocupar varias eventos de entrega de producto al día.

10. Por el otro lado, las instancias típicas del PDP-TW revisadas en la bibliografía se centran en aplicaciones muy orientadas al tratamiento de embarques y recibos programados sobre la base del manejo de citas ya muy acotadas en cuanto a la amplitud de tiempo en las cuales se están comprometiendo para su ejecución en la práctica operativa.

11. En la práctica de las empresas pueden llegar a existir diversos objetivos en consideración al momento de tener que enfrentar un problema relacionado al PDP-TW. En términos generales existen 3 objetivos principales:
    a. Minimizar el tiempo total requerido para atender el total de los clientes de la ruta (esto equivale a minimizar el "makespan" en los problemas de programación de la producción).
    b. Minimizar el costo total de los traslados requeridos para la atención de la ruta.
    c. Minimizar la falta de nivel de servicio o insatisfacción que puedan tener los clientes por llegar a atenderlos fuera de la ventana de horario comprometida previamente.

12. Como puede verificarse, el primer objetivo tiene que ver más con un sentido de eficiencia de operación enfocado a que la ruta termine su jornada de trabajo lo antes posible. De acuerdo a la empírica del postulante, pocas veces este objetivo es una necesidad real en la práctica de las empresas de distribución en ruta lo cual desincentiva su experimentación en nuestro caso.

13. Más aún como puede verificarse en el segundo objetivo, es de esperarse que si se logra obtener un recorrido con un costo mínimo, también es de esperarse que se obtenga al menos un buen indicador en cuanto a la duración de las jornadas de trabajo de las rutas.

14. El tercer objetivo pudiera ser interesante explorar en aquellos casos en lo que sea una posibilidad el poder manejar ventanas de horario "flexibles". Un ejemplo de lo anterior sería aquel caso en el que por cada cierta cantidad de minutos que se llegue fuera de la ventana de horario, se tendría que incurrir en una penalización (o un costo de oportunidad) por la aplicación de un descuento aplicado en el precio del servicio (o del producto) ofrecido al cliente.

.5. Puede entonces verificarse que resulta innecesario utilizar el enfoque anterior en aquellos casos en los que se trate de ventanas de horario "negociadas" y mucho menos "fijas", ya que en estos casos lo que se busca es dar tratamiento a las ventanas de horario a través de restricciones a cumplir y no de objetivos a lograr.

16. Así pues, decidimos emplear el segundo objetivo que es el de minimizar el costo total de los traslados requeridos para la atención de la ruta. El costo incurrido de los traslados en una ruta puede ser una función lineal de la distancia o del tiempo requerido para dicha operación.

17. En ocasiones resulta ser más útil relacionarlo con el tiempo, sobre todo cuando se trata de rutas con recorridos urbanos donde existen áreas con un tráfico intenso. Así por ejemplo, si una ruta gasta $1000 por día en mano de obra, combustible y mantenimiento, pues entonces aquellos traslados entre clientes que ocupen mayor cantidad de tiempo también de manera proporcional tendrán una participación mayor en el costo.

18. Es importante puntualizar en que no es lo mismo hablar de una ventana de horario fija que de una ventana de horario cerrada. Así por ejemplo puede existir un cliente con una ventana de horario fija que este dispuesto a recibir su atención en el rango de las 10:00 hrs. hasta las 16:00 hrs. Esta ventana de horario se dice que es fija porque si no se cumple entonces ya no puede darse el servicio. Sin embargo también puede percibirse que la ventana de horario es bastante amplia.

Con todo lo expuesto en este apartado, no pretendemos establecer que nuestro problema de investigación (SPDP-sTW) sea más difícil o fácil de resolver que cualquier otra variante del PDP-TW o del VRP-TW. Simplemente se trata de un problema con una variante diferente la cual conviene ser atendida y de ahí que se haya establecido como objetivo al inicio del apartado el dar una justificación para la investigación del problema.

Así entonces con esta exposición bibliográfica, concluimos la revisión de todos aquellos trabajos de investigación que previamente se han desarrollado en lo referente al tratamiento de problemas similares a nuestro SPDP-sTW. Pasamos ahora en el siguiente capítulo 7, al planteamiento de los objetivos e hipótesis del proyecto de investigación.