



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN

***Planeación de rutas Libres de
Obstáculos utilizando Algoritmos
Genéticos***

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO MECÁNICO ELECTRICISTA

ÁREA INGENIERÍA INDUSTRIAL

P R E S E N T A

ERNESTO ALCÁNTARA CONCEPCIÓN

ASESOR: DR. JULIO SOLANO GONZÁLEZ



2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Al pueblo de México

Que con su trabajo me permitió acceder incluso a la educación superior

A todos mis compañeros

Que día a día luchan por la defensa de la educación pública y gratuita a todos los niveles

A la Universidad Nacional Autónoma de México

Por contribuir en mi formación como profesionista y ciudadano

Al Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas

Por permitirme participar y desarrollar este proyecto

A mi madre

Por su empeño y dedicación para que todos mis hermanos y yo cursáramos una carrera universitaria

A mis hermanos

Que con sus conocimientos ayudaron a mi formación.

ÍNDICE

INTRODUCCIÓN.	i
CAPÍTULO 1. LA ROBÓTICA.	1
Robots.	1
Componentes de un robot Manipulador	2
Ligas o uniones	2
Muñecas	3
Clasificación de los robots	4
Manipuladores	5
Robots de repetición o aprendizaje.	5
Teleoperadores	5
Exoesqueletos	5
Robots controlados por computadora	5
Robots inteligentes	6
Micro-robots	6
Los robots en la actualidad	7
CAPÍTULO 2. PLANEACIÓN DE RUTAS LIBRES DE COLISIÓN.	11
Aspectos básicos de la planeación de rutas	11
Problema básico para la planeación de movimientos	12
Métodos de planeación de rutas	18
Mapa de ruta (Roadmap)	18
Gráfica de visibilidad (Visibility graph).	18
Diagramas de Voronoi.	19
Camino libre (Freeway net).	20
Descomposición en celdas (Cell Decomposition).	21
Campos potenciales (Potential Field).	23
CAPÍTULO 3. ALGORITMOS GENÉTICOS	25
La optimización.	25
Componentes de un algoritmo genético	29
Esquemas de reproducción.	31
Métodos de selección usados.	34
Ranking.	34
<i>Ruleta (roulette wheel)</i>	35

<i>Estocástico universal</i>	35
Cruza	35
Muta	36
Teorema del esquema.	36
Algoritmos genéticos paralelos.	40
Granja	41
Migración.	41
Modelo de difusión	42
CAPÍTULO 4. GENERACIÓN DE RUTAS LIBRES DE OBSTÁCULOS CON ALGORITMOS GENÉTICOS	44
Planteamiento del problema.	44
Los efectos de la diversidad de la población y la presión de selección sobre los AG's.	45
Resultados	47
CAPÍTULO 5. RESULTADOS Y ANÁLISIS DEL PROBLEMA	55
Primer método	55
Segundo método.	58
ANALISIS	67
<i>Análisis del primer método</i>	67
<i>Análisis del segundo método.</i>	68
CONCLUSIONES	70
BIBLIOGRAFÍA	72
CÓDIGO FUENTE	80

INTRODUCCIÓN.

A través de los años y con la evolución de la tecnología, los seres humanos han logrado crear máquinas para realizar actividades que muchas veces están fuera de su alcance, por ejemplo, la posibilidad de volar y la de estudiar otros planetas, etc. La creación de robots que realicen múltiples tareas sin intervención alguna, que substituyan a los seres humanos en la realización de algunas actividades, que les permitan llevar una vida más placentera, es un sueño que no sólo se ha hecho realidad, sino que se ha desarrollado, y sus avances parecen estar cada vez más cerca de integrarse a la cotidianidad.

Por otra parte, en un futuro cercano, al realizar actividades autónomas, los robots se moverán en distintos ambientes y se enfrentarán a situaciones inesperadas o a problemas no previstos.

Llevar a cabo cada una de estas tareas requiere del desarrollo de nuevas tecnologías y algoritmos específicos, que permitan a los robots, al igual que a las computadoras, interpretar y representar la realidad en términos matemáticos, resolviendo los problemas que se les presenten (Nilsson, 1985). Las tecnologías desarrolladas que han permitido que los robots se desempeñen de manera autónoma, abarcan una amplia gama de materias, desde el tipo de material con el que son construidos, el diseño de uniones entre partes, los dispositivos que utilizan para realizar sus tareas, la unidad central de procesamiento (CPU), entre otros, áreas que, sin embargo, aún tienen un largo camino por andar. Un ejemplo de estas tecnologías indispensables para el diseño de los robots son los sensores, que tienen como objetivo describir el medio ambiente y/o localizar personas u objetos que se encuentran a su alrededor.

El uso de algoritmos permite integrar al robot las tecnologías de manera apropiada, para transformar el mundo real en símbolos y operaciones matemáticas (Nilsson, 1985), que le permiten realizar tareas complejas, como tomar objetos, producir algún elemento de manera automatizada o moverse; ya que a través de la creación de dichos algoritmos se diseñan paso a paso las rutas o criterios de decisión que determinan el movimiento del robot.

Sin embargo, el diseño de un robot implica planear rutas de tal manera que alcance sus objetivos, ésta no es una tarea sencilla, debido a que por lo regular se encuentran obstáculos en el camino que es necesario sortear. Se concluye que la planeación de rutas es uno de los retos más importantes a vencer para lograr que los robots sean realmente autónomos.

Planear una ruta consiste en la búsqueda de un camino de un punto inicial a uno final, sin intersectar con ningún obstáculo. El problema que se presenta es que la ruta o camino que debe seguir el robot se conforma por 'n' segmentos o puntos intermedios. A cada segmento o punto intermedio que tomará el robot para lograr su objetivo, lo definiremos como una tarea a ejecutar. La solución puede encontrarse al elegir una entre dos alternativas para determinar la ruta que ha de tomar el robot, ya sea:

- a) Buscando los puntos intermedios que conformarán la ruta (encontrando las coordenadas cartesianas).
- b) Generando los 'n' segmentos de recta por separado (viendo el camino como un todo y construyendo la ruta a partir de secciones de rutas).

Cada uno de estos puntos o segmentos de recta formarán una ruta a seguir. Para encontrar esta serie de puntos o rectas intermedias, se requiere de múltiples pasos previos, utilizando técnicas y herramientas matemáticas para transformar el entorno que lo rodea, e ir descubriendo los obstáculos a su paso.

Por ejemplo: un robot al desplazarse en una situación real, digamos una fábrica, deberá resolver un sinnúmero de situaciones similares a las que enfrentamos los seres humanos al caminar de un lugar a otro; tendrá que interactuar con obstáculos que se encuentran a su alrededor, algunos en movimiento y otros fijos, existe la posibilidad que su ubicación y forma sean desconocidas, en otros casos se enfrentará a accidentes de trabajo o situaciones inesperadas. Para resolver algunos de estos casos se han diseñado algoritmos específicos para la situación, por ejemplo, un robot que encuentra un obstáculo que no conoce deberá seguir una serie de instrucciones sencillas, tales como ir hacia atrás, adelante, izquierda o derecha, para evadirlo (Mitchell; Keating; Kambhampati, 1994).

La planeación de rutas integra una serie de algoritmos, como el anterior, para enfrentarse a diversas de situaciones, sin embargo, integrar a un robot un algoritmo para enfrentar cada situación resulta ineficiente e incluso imposible de prever cada una de las situaciones en las que se involucre el robot.

El desplazamiento de un robot en una superficie dada, tiene dos componentes principales: el primero es el movimiento del robot y las fuerzas que lo afectan; y, el segundo es la superficie en la que se moverá.

Las fuerzas como la gravitatoria, las corrientes de aire o agua en el mar, por ejemplo, tendrán que ser tomadas en cuenta para el diseño de los robots, por otro lado, la superficie en la que se desplaza un robot se subdivide a su vez en la forma del suelo (pendiente, distancia, escalones), y la cantidad y distribución de obstáculos, lo que convierte a éste en un problema de geometría, dinámica y búsqueda de rutas óptimas.

Considerar un desplazamiento en estas condiciones podría resultar trivial para una persona que se desplaza a diario de un lugar a otro, y ha aprendido a evadir los obstáculos que se presentan, pero resulta un reto mayor para un robot (Grabowski, 2003).

Los algoritmos de rutas óptimas no se limitan exclusivamente a la robótica, son y pueden ser aplicados a muchas áreas de investigación, como la aviación, la exploración de terrenos de difícil acceso ya sea en otros planetas, en las profundidades del mar, dispositivos para guía de misiles, entre otros, logrando, por ejemplo, enviar robots con materiales o personas de un lado a otro de una empresa, a otros sitios o planetas, sin la supervisión humana.

En este sentido, los robots se enfrentarán a diferentes escenarios como, son el espacio sideral, lugares montañosos, rocosos o fangosos; tan distintos como puede haber en la Tierra u otros planetas. Los robots interactuarán con las diferentes fuerzas que afectan el espacio de desplazamiento (p. ej. gravitatorias). Con el fin de que el robot sea capaz de identificar esta amplia gama de escenarios y actuar en consecuencia, los investigadores han representado a estos escenarios como espacios geométricos complejos. Para realizar esto se han hecho confluír diferentes tecnologías y algoritmos que utilizan formas diversas de representar al mundo real. Al respecto, una referencia obligada es Latombe, que hace una revisión de los algoritmos más utilizados para la planeación de rutas, entre los que incluye gráficas de visibilidad, diagramas de Voronoi, caminos libres, descomposición en celdas, campos potenciales (Latombe, 1991). Éstas transforman la topología y describen el espacio que rodea al robot, por donde este puede pasar libremente.

En el campo de la planeación de rutas, actualmente, se están utilizando combinaciones de los métodos tradicionales para lograr algoritmos más eficientes, (Barraquand; Langlois; Latombe, 1992), (Choset; Yeong, 2001). Por otra parte, han aparecido algunas investigaciones que utilizan técnicas de inteligencia artificial para la optimización de rutas, como son los algoritmos genéticos (Solano; Jones, 1993).

Los algoritmos genéticos forman parte de lo que se conoce como computación evolutiva. La computación evolutiva incorpora ideas y mecanismos encontrados en la naturaleza para resolver problemas de búsqueda y optimización. Las ideas esenciales de la computación evolutiva fueron aportadas en el año de 1965, por Rechenberg. Más adelante, John Holland y sus estudiantes comenzaron a realizar investigaciones con algoritmos genéticos (1968-1971), las áreas fundamentales de su trabajo fueron observar los abstractos y rigurosos procesos de adaptación de sistemas naturales y su simulación en computadora.

Los algoritmos genéticos (AG) son métodos sistemáticos de búsqueda y optimización de problemas, que utilizan los principios de los mecanismos de selección natural y de genética, para resolver problemas. La selección natural y la genética son conceptos originados en biología desde el siglo XIX, con las investigaciones de Charles Darwin, acerca de la evolución de las especies (Darwin, 1859); y de Gregor Mendel, respecto de la germinación de chícharos en condiciones controladas (Mendel, 1863).

Según la teoría de Darwin y el principio de selección natural, los individuos con mayor aptitud al medio dejarán más descendencia y transmitirán sus genes a las generaciones futuras. Para determinar quiénes son los individuos más aptos en el AG, cada uno de estos será evaluado. La evaluación será la unión entre el AG y el mundo externo. Esta evaluación se realiza a través de una función que representa de forma adecuada el problema. La función de evaluación, también conocida como función objetivo, es para el AG, lo que el medio ambiente es para un ser vivo. Los seres vivos se representan en el AG, como las posibles soluciones (Cavalcanti, 1999). Cada una de estas posibles soluciones podrán ser generadas aleatoriamente o agregarse a soluciones que se sabe, de antemano, que son

resultados factibles. A los individuos prometedores se les permitirá seguir reproduciéndose; así, estos individuos favorecidos se cruzarán con otros dentro de la población, lo que generará nuevos miembros, el proceso se repite muchas veces con el paso del tiempo. Cada nueva generación contendrá individuos que podrán mejorar o empeorar, y la expectativa es que la media de la población se mejore en cada una de las generaciones, y en el caso análogo, llegue así a una solución muy exacta o próxima del problema.

Los algoritmos genéticos se han utilizado en la solución de problemas de optimización. Algunos de estos usos son: Optimización de funciones matemáticas, el problema del agente viajero, optimización de circuitos, distribución de plantas o espacios de trabajo, de negocios (Michelewics, 1996), así como en otras áreas de investigación, como la química, aeronáutica, eléctrica (Montagnon, 1999), telecomunicaciones, juegos, robótica, acústica, etc. Con el tiempo, los algoritmos genéticos se han modificado para mejorar su desempeño (Eksin; Erol, 2001) o para mejorar su aplicación en problemas específicos o en combinación con otras herramientas de inteligencia artificial.

En el campo de la robótica las investigaciones realizadas con algoritmos genéticos, para la generación de rutas libres de obstáculos, realizan la búsqueda de rutas, tomando 2 ó 3 dimensiones, mediante los métodos expuestos por Latombe (Latombe, 1991), comparando los resultados en función de su efectividad, por ejemplo, tiempo de ejecución, distancia recorrida y otros. En la Tabla A-1 se presenta un resumen de las principales investigaciones concernientes a este tema. En dicha Tabla, se muestran, entre otros los desarrollos de I. Ashiru, C. Czarnecki (Ashiru et al, 1995), Nishimura, Sugawara, Yoshihara K Abe (Nishimura et al, 1999), P. Vadakkepat, K. Chen T and W. Ming-liang (Vadakkepat et al, 2000), los cuales utilizaron el método 'campos potenciales'. También se aprecia en esta Tabla que uno de los métodos usados para la resolución de este problema es dividir el espacio de configuración en celdas o dividir el camino en grafos.

En este contexto se realizó esta tesis. En los primeros tres capítulos se describe el marco conceptual, se incluyen los algoritmos utilizados en la planeación de rutas libres de obstáculos, además de las herramientas que se utilizaron posteriormente en la generación de estas rutas. En los capítulos 4 y 5 de la tesis se presenta dos alternativas originales para la planeación de rutas, basados en algoritmos genéticos.

Objetivo de la tesis

El objetivo de este trabajo es la generación de rutas óptimas, libres de obstáculos usando algoritmos genéticos.

Contribución

Actualmente los investigadores han encontrado una gran variedad de formas para solucionar la planeación de rutas, sin embargo, los ambientes complejos en los que los robots se mueven demandan algoritmos cada vez más eficientes.

Encontrar el camino más corto (optimización) entre un punto y otro es uno de los problemas de la planeación de rutas, aunado a esto, se construye una representación geométrica del robot y del entorno que los rodea. El robot se considera como una partícula en el espacio multidimensional y los objetos a su alrededor se transforman de acuerdo a los movimientos y número de articulaciones del robot, así como su morfología. Los algoritmos tradicionales de búsqueda de rutas óptimas que se realizan son costosos en términos del número de operaciones matemáticas necesarias para encontrar las soluciones, aunado a que son lentos, resultan de difícil implementación en una aplicación real, el uso de los algoritmos genéticos resulta novedoso y son algoritmos intrínsecamente paralelos, lo que puede reducir el tiempo de ejecución de los mismos en arquitecturas paralelas. Además de obtener buenos resultados en problemas de optimización, como el que representa la planeación de rutas.

En esta tesis se presentan dos alternativas originales para la planeación de rutas. En la primera, el enfoque es tomar las rutas como un todo, generando rutas completas dentro del AG; la segunda se genera la ruta a partir de círculos, se comienza con un círculo de radio 'r' y se busca el punto más cercano al punto objetivo dentro de esa circunferencia, después se aumenta el diámetro y nuevamente se busca el punto más cercano dentro de la nueva circunferencia, y así sucesivamente, hasta llegar al punto final u objetivo. Cuidando siempre de evitar colisiones o choques con los obstáculos.

Muchos de los algoritmos utilizados requieren que el espacio de trabajo sea conocido e incluyen un procesamiento posterior del espacio de trabajo. Este procesamiento consiste en poner marcas o dividir el espacio de trabajo en segmentos para asegurar el libre paso del robot, hasta su destino (ej. gráficas de visibilidad, diagramas de Voronoi), realizando un paso previo para preparar el espacio de trabajo. Esto, en una situación real, donde no se conoce el medio, resulta difícil de implementar, con el trabajo aquí presentado, sólo es necesario conocer el punto de salida y el de llegada, y el robot generará la ruta completa. Otro de los métodos más utilizados es el de "campos potenciales", alejando al robot de los obstáculos.



Figura I.1 El TX8, robot llevado por la UNAM a la Robo Cup en Alemania 2006

A continuación se describe en forma general los capítulos que integran este trabajo de tesis:

Capítulo 1.- LA ROBÓTICA

Contiene una introducción a la historia de la robótica, tipos y clasificación de los robots y las partes que lo componen. Se presenta contexto general del tipo de robots existentes y el futuro de éstos.

Capítulo 2.- PLANEACIÓN DE RUTAS LIBRES

Se da una breve introducción de las técnicas más utilizadas en la planeación de rutas, describiendo las tareas esenciales que un robot debe realizar para la planeación de rutas, se plantea el problema y las consideraciones iniciales a tomar en cuenta. Este problema resulta complejo porque existen diversos elementos que intervienen en la planeación y que a lo largo de este capítulo se expondrán, así como las bases científicas en las que se apoya este trabajo.

Capítulo 3.- ALGORITMOS GENÉTICOS

En la primera parte se hace una revisión histórica del surgimiento de los algoritmos genéticos; en la segunda parte se expone la definición de optimización, que es uno de los objetivos principales de los algoritmos genéticos; por último, se describe el funcionamiento de los algoritmos genéticos y los operadores genéticos comúnmente utilizados para la resolución de problemas de búsqueda y optimización.

Capítulo 4.- GENERACIÓN DE RUTAS LIBRES DE OBSTÁCULOS CON ALGORITMOS GENÉTICOS.

En este capítulo se hace el análisis de los efectos producidos por la modificación de los parámetros (tamaño de la población, probabilidades de mutación y selección) en combinación con los métodos de selección expuestos en el capítulo anterior.

Capítulo 5.- RESULTADOS Y ANALISIS DEL PROBLEMA

Se desarrolla la investigación, se describe la implementación de los algoritmos genéticos en la planeación de rutas libres de obstáculos, exponiendo las ecuaciones utilizadas, analizando los resultados obtenidos; además, así como la conclusión de este trabajo.

CAPÍTULO 1. LA ROBÓTICA.

Construir dispositivos mecánicos que liberen de la realización de actividades peligrosas, tediosas o repetitivas que requieran de un gran esfuerzo, ha sido una tarea constante durante miles de años. Algunas de las referencias más antiguas de estos dispositivos se encuentran en la mitología Griega y oriental. Por ejemplo, en el siglo II A.C., Hero de Alejandría construyó aves mecánicas que aleteaban, gorgojeaban y comían; en la Edad Media y el Renacimiento son muchas las referencias de hombres mecánicos contruidos, principalmente por el gremio de los relojeros, destinados principalmente para las cortes o para su exhibición en ferias. También fueron famosos el león animado y los autómatas de funcionamiento cíclico, gobernado por tambores de púas contruidos por Leonardo da Vinci.

Tal vez una de las más complicadas y versátiles máquinas, contruidas con mecanismos de reloj, fue el increíble autómata que escribía y dibujaba, contruido alrededor de 1805 por Henri Mailladet, que tenía una gran memoria, además de movimientos precisos.

En la actualidad estas máquinas reciben el nombre de “robot”, palabra derivada del checo (robotik = siervo), utilizada, inicialmente por Karel Capek en 1920, en una obra de teatro llamada R.U.R (Robots Universales de Rossum).

Isaac Asimov, uno de los mejores escritores de ciencia ficción y a quien se le acredita como el primero que utilizó la palabra robótica (Asimov, 1920)

Es común encontrar la afirmación de que el padre de la robótica industrial fue, en 1960, George Devol, quien, buscando construir una máquina automática, definió como características fundamentales:

- a) La Flexibilidad, en su adaptación a diversos trabajos y herramientas.
- b) Y la Sencillez de manejo.

Desarrolló lo que se considera como el primer robot industrial, al que se le incorporó una computadora como parte fundamental.

Los robots, a través de los años, han evolucionado y en la actualidad existen, por ejemplo, robots equipados con sensores que describen su medio ambiente o detectan personas u objetos que se desplazan a su alrededor, este tipo de robots son de gran utilidad en plantas industriales, donde se integran a los procesos de producción y pueden conocer la ubicación de los obstáculos y utilizar esta información para desplazarse de un punto a otro sin colisiones

Robots.

De acuerdo con la Real Academia Española un robot se define como: trabajo, prestación personal, máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas sólo a las personas. Y *el Robot Institute of America lo define como un manipulador multifuncional y*

reprogramable, diseñado para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos programados y variables que permitan llevar a cabo diversas tareas (Ollero, 2001). Para realizar cualquier tarea útil el robot debe interactuar con el entorno, el cual puede incluir dispositivos de alimentación, otros robots y, lo más importante, las personas. La robótica abarca no solamente el estudio del robot en sí, sino también las interfaces entre él y sus alrededores.

En resumen podemos decir que, un robot es una máquina, manipulador multifuncional y reprogramable capaz, de manejar o mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos programados y variables para realizar tareas que antes eran exclusivas de las personas.

Los Robots Móviles

Los robots móviles son máquinas autómatas capaces de moverse dentro de su entorno, sin estar fijos en una posición.

Para moverse dentro de su entorno utilizan diferentes tipos de locomoción mediante ruedas o patas que les confieren diferentes características y propiedades respecto a la eficiencia energética, dimensiones, cargas útiles y maniobrabilidad.

También existen robots aéreos y submarinos que encuentran aplicación en la inspección, recopilación de datos o mantenimiento de instalaciones en entornos naturales de difícil acceso.

Los robots móviles los podemos entonces clasificar como:

- Ackerman (cuatro ruedas).
- Triciclo.
- Direccionamiento diferencial.
- Skid steer (varias ruedas en cada lado del vehículo).
- Pistas de deslizamiento (tipo Oruga).
- Sincronías.
- Locomoción con patas.
- Robots submarinos y aéreos.

Manipuladores

Los robots Manipuladores son esencialmente brazos articulados, de forma más precisa un manipulador industrial es una cadena cinemática abierta, formada por un conjunto de eslabones o elementos de la cadena, interrelacionados mediante articulaciones o pares cinemáticos (Ollero, 2001).

La Robot Institute of America define a un robot industrial como un manipulador programable multifuncional diseñado para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos variados, programados para la ejecución de distintas tareas. El estándar ISO (ISO 8373:1994, Robots industriales

manipuladores Vocabulario) define un robot industrial como un manipulador programable en tres o más ejes multipropósito, controlado automáticamente y reprogramable.

Componentes de un robot Manipulador

La parte mecánica de un robot la integran una serie de elementos o eslabones unidos mediante articulaciones que permiten un movimiento relativo entre cada eslabón consecutivo. Debido a que la mayor parte de los robots industriales guardan cierta similitud con el brazo de un ser humano, a veces se hace referencia a ellos con el nombre de cuerpo, brazo, codo y muñeca (Figura 1.1).

Exteriormente un robot industrial tiene una base en donde está soportado y un brazo que tiene en la punta una pinza, en la base del robot se encuentran los motores que le permiten moverse. Los robots cuentan también con sensores para cumplir con dos objetivos: uno para monitorear las acciones realizadas, y otros que el de perciben al mundo que lo rodea.

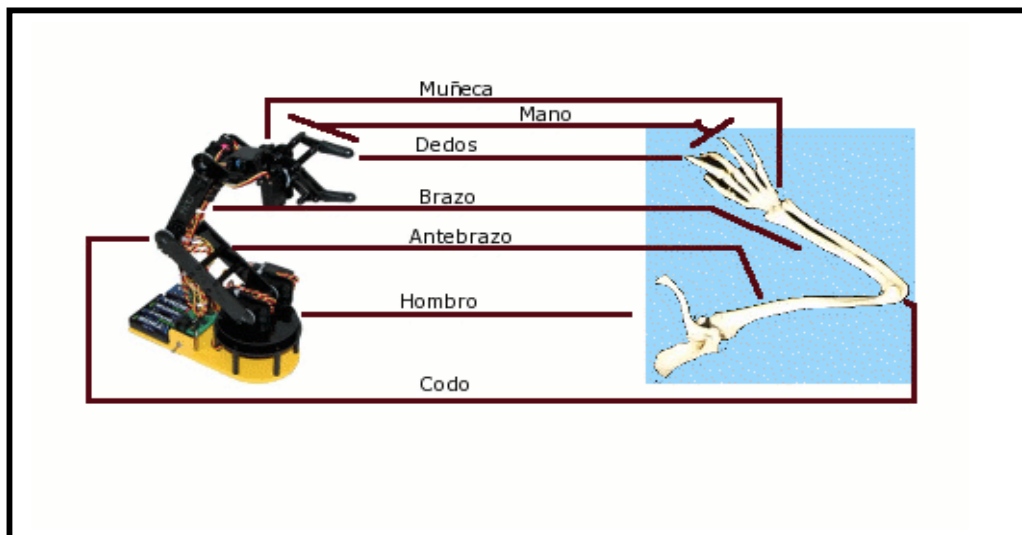


Figura 1.1.-Partes de un robot industrial y su nombre con referencia al cuerpo humano.

Ligas o uniones

Las secciones individuales de un brazo robot son llamadas ligas, éstas pueden rotar o deslizarse. Estas articulaciones son partes rígidas que soportan las cargas llevadas por los robots, por lo general están huecas para reducir el peso y proveer de espacio para el cableado, las mangueras u otros sistemas que sean necesarios para el control del robot.

En la Figura 1.2, se muestra el diagrama del robot PUMA (Programable Universal Machine for Assembly) 600, desarrollado por la compañía UNIMATION para mostrar las ligas y uniones del mismo

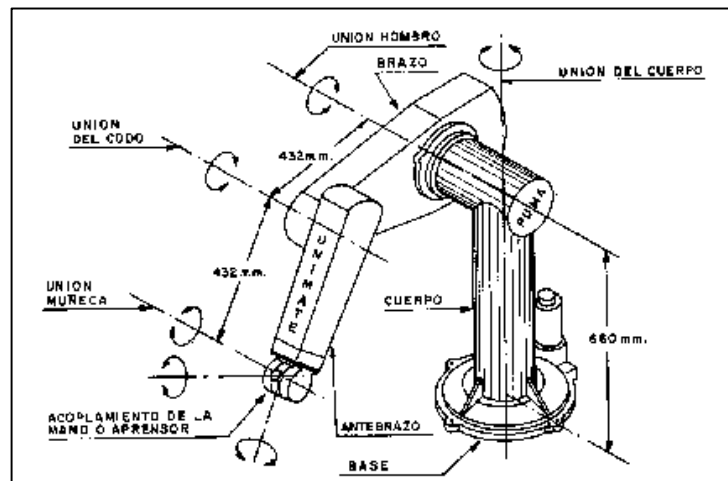


Figura 1.2 Articulaciones del robot PUMA 600 de UNIMATION.

Muñecas

Son montadas al final del brazo y muchas tienen dos o tres uniones, usualmente son creadas como una sola unidad y dotan al robot de movilidad para poder sujetar o girar.

El sujetador final o mano (griper) es un dispositivo que se une a la muñeca del brazo del robot con la finalidad de activarlo para la realización de una tarea específica. La razón por la que existen distintos tipos de sujetadores finales es, precisamente, por las funciones que realizan. Se dividen en dos grandes categorías clasificadas en: pinzas y herramientas.

Donde las *pinzas* han sido diseñadas para que el robot cargue y descargue objetos, transporte materiales y ensamble piezas. Los tipos de pinzas más comunes pertenecen al tipo llamado pivotante, como la que se muestra en la Figura 1.3



Figura 1.3 Pinza de un robot.

Los dedos de la pinza giran en relación con los puntos fijos del pivote. De esta manera, la pinza se abre y se cierra. Otro tipo de pinzas se denominan de movimiento lineal. En este caso, los dedos se abren y se cierran, ejecutando un movimiento paralelo entre sí. Al momento de diseñar una pinza deben tomarse en cuenta la forma y peso de la pieza de trabajo, así como el tipo de movimiento que harán los dedos. Con estos lineamientos se puede asegurar una buena sujeción, de tal forma que la pinza no modifique o dañe la estructura de la pieza. Una regla general es que la pinza debe sujetar a la pieza de trabajo en su centro de gravedad, esto ocasiona que se anulen los momentos que se pudieran generar por el peso de la pieza de trabajo.

La segunda categoría de los sujetadores, se refiere a las herramientas. Una herramienta se utiliza como actuador final en aplicaciones en donde se exige al robot realizar alguna operación sobre una pieza de trabajo. Estas aplicaciones pueden ser: soldadura por puntos, soldadura por arco, pintura y operaciones con taladro, entre otras. En cada caso, la herramienta particular estará unida a la muñeca del robot para realizar la operación.

En la Figura 1.4 podemos observar una mano robótica la cual se utiliza para soldar, manipular y empaquetar piezas dentro de una fábrica.



Figura 1.4 Brazo robot industrial para soldadura por arco, manipulación y empaquetado

Clasificación de los robots

A los robots se les clasifica de acuerdo con el tipo de control, por el nivel de compatibilidad, y su configuración mecánica. Para diferenciarlos, se presenta una clasificación general:

Manipuladores

Son sistemas mecánicos multifuncionales, con un sencillo sistema de control, que permite gobernar el movimiento de sus elementos, de los siguientes modos:

- Manual: Cuando el operario controla directamente la tarea del manipulador.
- De secuencia fija: Cuando se repite, de forma invariable, el proceso de trabajo preparado previamente.
- De secuencia variable: Se pueden alterar algunas características de los ciclos de trabajo.

Robots de repetición o aprendizaje.

Son manipuladores que se limitan a repetir una secuencia de movimientos, previamente ejecutada por un operador humano, haciendo uso de un controlador manual o un dispositivo auxiliar. En este tipo de robots, el operario en la fase de enseñanza, puede utilizar una pistola de programación con diversos pulsadores o botones, palancas de mando, maniquí, o bien, desplazar directamente la mano del robot. Los robots de aprendizaje son los más conocidos, hoy en día, en los ambientes industriales y el tipo de programación que incorporan, recibe el nombre de "gestual".

Teleoperadores

El término teleoperador fue usado alrededor de 1966, por Edwin Johnson, el cual los describió como manipuladores con servocontroles que no estaban conectados directamente. Los teleoperadores son dirigidos por humanos por medio de un control remoto que usa una palanca, teclados o exoesqueletos, éstos se usan en trabajos peligrosos, como el manejo de residuos radiactivos o en el espacio.

La comunicación con el robot puede ser por conexión óptica (infrarrojo), ondas de radio o vía satélite, por voz, láser, entre otros.

Exoesqueletos

Los exoesqueletos se utilizan en la operación o control de los robots y simulan los movimientos del manipulador. El operador utiliza una estructura ligera que se adapta al cuerpo humano y es sensible a los movimientos, convirtiéndolos en señales eléctricas que son interpretadas por el manipulador.

Robots controlados por computadora

Son manipuladores o sistemas mecánicos multifuncionales, controlados por una computadora, que habitualmente suele ser un microprocesador.

En este tipo de robots, el programador no necesita mover realmente la máquina, cuando la prepara para realizar un trabajo. El control por computador dispone de un lenguaje específico, compuesto por varias instrucciones adaptadas al robot, con las que se puede confeccionar un programa de aplicación, utilizando sólo la terminal de la computadora. A esta programación se le denomina textual y se crea sin la intervención del manipulador.

Robots inteligentes

Son similares a los del grupo anterior, pero, además, son capaces de relacionarse con el mundo que les rodea a través de sensores y tomar decisiones en tiempo real (auto programable).

Micro-robots

Existen numerosos robots de formación o micro-robots con fines educativos, de entretenimiento o investigación a un precio muy accesible, cuya estructura y funcionamiento son similares a los de aplicación industrial.

La Asociación Francesa de Robótica Industrial (AFRI) clasifica los robots de acuerdo con dos criterios; por el tipo de control y por el tipo de programación que requieren. La Tabla 1.1 muestra que existen cuatro tipos de robots de acuerdo con el manipulador que utilizan y tres de acuerdo con su programación.

Clasificación de acuerdo con el tipo de control que utilizan.	
Tipo de robot	Clasificación
Tipo A	Manipulador con control manual o telemando.
Tipo B	Manipulador automático con ciclos preajustados; regulación mediante fines de carrera o topes; control por PLC; accionamiento neumático, eléctrico o hidráulico.
Tipo C	Robot programable con trayectoria continua o punto a punto. Carece de conocimiento sobre su entorno.
Tipo D	Robot capaz de adquirir datos de su entorno, readaptando su tarea en función de éstos.
Clasificación de los robots industriales en generaciones.	
1° Generación.	Repite la tarea programada secuencialmente. No toma en cuenta las posibles alteraciones de su entorno.
2° Generación.	Adquiere información limitada de su entorno y actúa en consecuencia. Puede localizar, clasificar (visión) y detectar esfuerzos y adaptar sus movimientos en consecuencia.
3° Generación.	Su programación se realiza mediante el empleo de un lenguaje natural. Posee la capacidad para la planificación automática de sus tareas.

Tabla 1.1 Clasificación de los robots según AFRI.

La clasificación realizada por T.M. Cancel, presentada en la Tabla 1.2, incluye cinco generaciones de robots relacionados con el año en que se prevé su aparición, tipo de control, el grado de movilidad y el uso que se le dará a cada uno de éstos (Engelberger, 1979).

Clasificación de los robots según T M Knasel.				
Generación	Nombre	Tipo de control	Grado de movilidad	Usos más frecuentes
1 (1982)	Pick & place	Fines de carrera, aprendizaje	Ninguno	Manipulación, servicio de máquinas
2 (1984)	Servo	Servocontrol, trayectoria continua, programa condicional	Desplazamiento por vía	Soldadura, pintura
3 (1989)	Ensamblado	Servos de precisión, visión, tacto,	Guiado por vía	Ensamblado, desbardado
4 (2000)	Móvil	Sensores inteligentes	Patas, ruedas	Construcción, mantenimiento
5 (2010)	Especiales	Controlados con técnicas de Inteligencia artificial	Andante, saltarín	Militar, espacial

Tabla 1.2. Clasificación de los robots, según T.M.Knasel.

Los robots en la actualidad

A partir de los años 70's, ha existido un creciente interés en la construcción de robots, pero el número necesario de operaciones para imitar el comportamiento humano o animal resulta, aun hoy, extremadamente alto y complejo, esto provocó, en un principio, que las investigaciones se enfocaran en áreas, como la conducta, la navegación y la planeación de rutas.

Una de las industrias que se ha visto más beneficiada con la construcción de robots es la industria automotriz, donde muchos de los procesos repetitivos han sido remplazados por robots. Pero los robots están siendo utilizados en todos los ámbitos de la vida, por ejemplo, en la limpieza de residuos tóxicos, exploración espacial, minería, búsqueda y rescate de personas, localización de minas terrestres. Compañías en Japón como Sony y Honda, están comenzando a vender robots humanoides, como mascotas; y, han diseñado y comercializado robots con forma de perro, por ejemplo: Aibo, de Sony, Figura 1.5.



Figura 1.5. Robot AIBO ERS-7 de sony

Los robots industriales en uso, según la International Federal of Robotics en 2004, eran 86,200 y se prevé que para 2007 sean 106,300. Es de destacar que los países con mayor número de robots dentro de las fábricas son Japón y la República de Corea con 47,900 robots, que representan el 58%, mientras que Estados Unidos tiene un 15%. Ver Tabla 1.3 (AFI, 2005)

Actualmente los robots se construyen con formas de artrópodos e insectos, que ofrecen gran flexibilidad y han probado adaptabilidad a cualquier ambiente. Una tendencia de los robots es que sus componentes son cada vez más baratos y pequeños, miniaturizándose los componentes electrónicos que se utilizan para controlarlos, mediante el uso de la nanotecnología. (Xiao 2002)

El avance de la ciencia y de las computadoras ha hecho posible que muchos robots sean diseñados en simuladores, mucho antes de que sean construidos e interactúen en ambientes físicos reales. Las investigaciones que existen se encaminan a la implementación de software con herramientas de inteligencia artificial que les darán mayor autonomía. Estas investigaciones, en conjunto con la creación de microprocesadores programados en plataformas libres (Linux), suponen una explosión en el uso de los robots, y es fácil prever que en un futuro no muy lejano será posible tenerlos en los hogares.

Instalaciones y Robots Industriales multipropósito en almacén en 2002-2004 y el pronóstico para 2004-2007

País	Instalaciones/Año				Robots almacenados al final de año			
	2002	2003	2004	2007	2002	2003	2004	2007
Japón	25,373	31,588	33,200	41,300	350,169	348,734	352,200	349,400
USA	9,955	12,693	128,000	15,900	103,515	112,390	121,300	145,100
Unión Europea	26,096	27,114	28,800	34,400	233,769	249,200	266,100	325,900
Alemania	11,862	13,381	14,100	16,300	105,212	112,693	121,500	151,400
Italia	5,470	5,198	5,500	6,400	46,881	50,043	53,100	63,400
Francia	3,012	3,117	3,300	3,000	24,277	26,137	28,400	35,900
Reino Unido	750	1,111	120	1,500	13,651	14,015	14,600	16,300
Australia	570	365			3,521	3,602		
Bonalux	654	715			8,708	9,052		
Dinamarca	249	288			1,853	2,078		
Finlandia	376	387			3,151	3,407		
Portugal	138	135			1,282	1,367		
España	2,420	2,031			18,352	19,847		
Suiza	495	386			6,881	6,959		
Resto de Europa	582	922	1,000	1,300	11,009	11,409	11,900	14,200
Republica Checa	87	498			1,022	1,445		
Hungría	61	35			211	216		
Noruega	80	48			664	684		
Polonia	128	60			622	548		
Rusia	21	9			5,000	5,000		
Eslovaquia	24	1						
Eslovenia	25	31						
Suiza	156	240			3,490	3,480		
Asia/Australia	5,123	6,695	7,200	8,900	60,427	65,419	69,900	78,500
Australia	392	533			3,192	3,571		
Rep Coreana	3,998	4,660			44,265	47,845		
Singapur	53	48			5,299	5,273		
Taiwán	680	1,454			7,971	8,730		
Otros países	1,466	2,764	3,200	4,500	11,216	13,620	16,500	27,200
Subtotal: Japón y Rep. Corea	39,224	45,528	47,900	58,700	375,671	404,193	485,700	590,900
Total: Incluido Japón y Rep. Corea	68,595	81,776	86,200	106,300	770,105	800,772	886,200	997,700

Fuente, UNECE, IFR y National robot associations

Tabla 1.3 Número de robots industriales en uso y su estimación para 2007, según la International Federation of Robotic.

CAPÍTULO 2. PLANEACIÓN DE RUTAS LIBRES DE COLISIÓN.

Muchas investigaciones en el campo de la robótica se han enfocado a la realización de tareas complejas, tales como tomar objetos o producir algún elemento de manera automatizada. Para realizar estas tareas los robots necesitan desplazarse de un lugar a otro y enfrentar a obstáculos que se mueven o que permanecen estáticos, situaciones inesperadas, lugares desconocidos, la interacción con otros robots y/o realizar múltiples tareas. Los mecanismos de decisión necesarios para desplazar un robot provienen de diversas áreas del conocimiento como son: las ciencias de la computación, la teoría de control, y el procesamiento de señales e imágenes, entre otras. Estas áreas del conocimiento se combinan para transformar el mundo real en símbolos y operaciones matemáticas, un lenguaje entendible por los robots (Nilsson, 1985), en este caso los símbolos y operaciones, representan espacios geométricos complejos.

El estudio de la planeación de rutas es relativamente nuevo. J.P. Loumout *et al* mencionan que las primeras investigaciones respecto de la planeación de rutas se reportaron en congresos formales sobre Inteligencia Artificial (IA). Estas primeras investigaciones datan de finales de los 60's del siglo pasado y fueron realizadas por NJ Nilsson (Nilsson, 2005). En 1977, Alan M. Thompson construyó "gráficas de visibilidad" a partir de los vértices de los obstáculos; durante 1979, Giralt descompone este ambiente en celdas convexas libres de obstáculos (Loumout et al, 1998). A finales de los 70's, Lozano-Pérez introduce el concepto de "espacio de configuración" que, como veremos más adelante, considera al robot como un punto en el espacio. Este concepto abrió el camino a la investigación de algoritmos más complejos y es muy utilizado en las investigaciones de planeación de rutas (Lozano, 1979), en los noventa existen aproximaciones que utilizan algoritmos genéticos y la distancia euclidiana (Solano, 1993). Algunos de los trabajos más recientes se enfocan a la aplicación de las investigaciones realizadas en condiciones reales (Bracho, 2000).

Aspectos básicos de la planeación de rutas

La planeación de rutas es la búsqueda de una ruta o camino libre de obstáculos que conecta el punto de partida con el de llegada, donde el espacio en que se mueve el robot contiene una distribución compleja de los obstáculos.

Como se muestra en la Figura 2.1, Solano et al (Solano, 1993) menciona que para la generación de una ruta, un robot autónomo debe realizar en general cinco tareas

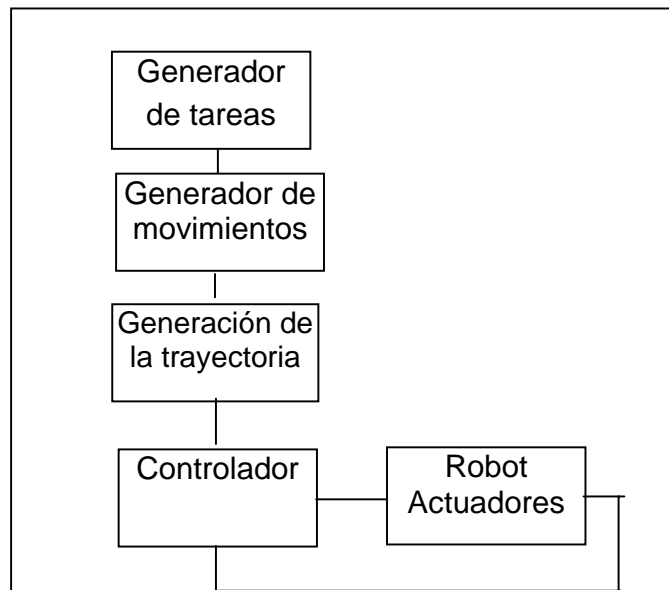


Figura 2.1 Robot autónomo¹

Donde:

- *Generador de tareas de alto nivel.*- Recibe la información acerca de los movimientos que requiere el robot y la transformación en comandos. Este componente debe incluir mayores niveles de razonamiento y capacidades de planeación.
- *Generador automático de movimientos.*- Este elemento utiliza la representación del mundo (modelos geométricos del robot y su medio) para generar una búsqueda en el espacio, utilizando alguna de las técnicas de generación de rutas libres de obstáculos.
- *Generador de trayectoria.*- Controla y transforma un camino en una trayectoria (o algoritmo) que puede ser ejecutada por el robot, es decir, genera la velocidad y la aceleración con que el robot debe recorrer el camino generado, automáticamente.
- *Controlador.*- Procesa las fuerzas que ejercen los actuadores en tiempo real para ejecutar el movimiento deseado.
- *Robot Actuadores.*- Ejecutan las instrucciones enviadas por el Controlador

Problema básico para la planeación de movimientos

El movimiento de un robot consiste en el desplazamiento (en el caso de un brazo o herramienta) desde un punto hasta otro. Inicialmente se conocen los extremos (la posición de partida y la posición final), pero no a las posiciones intermedias (trayectoria). La planeación de rutas busca encontrar estos puntos intermedios, de tal modo que el robot logre librar los obstáculos que se le presenten.

¹ J Solano et al (1993)

Para planear las rutas o trayectorias que seguirá el robot, suponemos lo siguiente (Latombe, 1991):

- El robot es un cuerpo rígido.
- El espacio debe ser conocido (*workspace*).
- Se ignoran las propiedades dinámicas del robot, con objeto de diseñar una planeación geométrica de la ruta y no una planeación del movimiento del robot.

Entonces:

- a) Un robot "A" es un objeto rígido moviéndose en un espacio euclidiano W , R^N , donde $N=2$ ó 3 (dimensiones)
- b) B_1, \dots, B_q son objetos rígidos sobre W , llamados obstáculos
- c) Donde la formas geométricas de A, B_1, \dots, B_q y su localización dentro del plano son conocidas. Asumiendo además que no se consideran las limitantes cinemáticas de los cuerpos

Para ello, existe una posición inicial de A (punto en el espacio W) y una posición final, con una orientación de A en W , donde el problema a resolver es: generar una secuencia de puntos y orientaciones² (trayectoria) evitando colisionar con los obstáculos B_i , hasta llegar al final. Figura 2.2

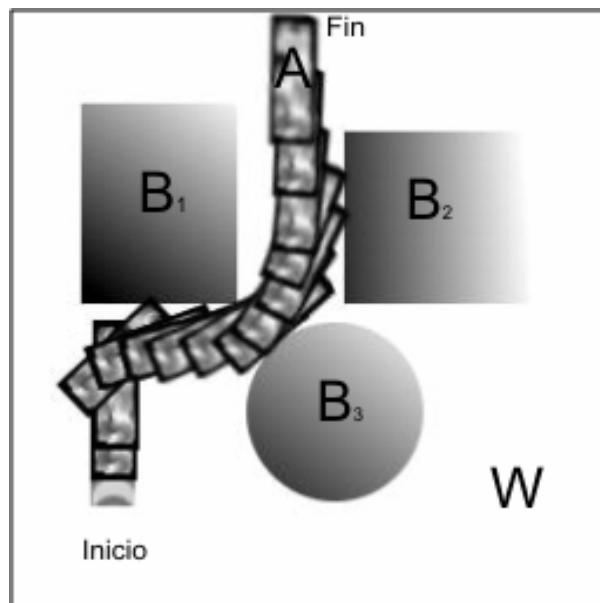


Figura 2.2 El objeto o robot se desplaza por el espacio desde su posición inicial hasta la final, siguiendo una secuencia de pasos y orientaciones.

²Se entiende como orientación a la posición inicial del robot, cuando éste puede rotar o girar sobre su propio eje, en el sentido de las manecillas del reloj, hasta una dirección determinada. Como se ve en la figura 2.2.

Muchos algoritmos de planeación de movimiento, así como algunos métodos generales de planeación, utilizan conceptos que definen el espacio y movimiento del robot. Así, un *espacio de configuración* es la representación geométrica del robot y la ubicación de obstáculos en un espacio de estudio, el robot es visto como una partícula en un espacio multidimensional, y los objetos a su alrededor, se transforman de acuerdo a la forma del robot. El problema se vuelve más sencillo de entender, con sólo tener que mover una partícula en el espacio.

El termino *configuración de un obstáculo*, se refiere a la descripción de cualquier posición que adopte este objeto con referencia al espacio en que se encuentra.

Para explicar estos conceptos, supongamos un robot A que se moverá en un espacio bidimensional o espacio de trabajo y que contiene al conjunto de obstáculos $S=\{B_1, \dots, B_n\}$, asumiremos que A tiene forma de polígono que se moverá en vectores (X,Y) por lo que se denota como $A(x,y)$. Si el robot se encuentra en $A(0,0)$, tiene vértices en $(-1,-1), (1,-1), (1,1), (0,3), (-1,1)$, entonces se puede decir que los vértices de $A(6,6)$ están en $(7,3), (7,5), (6,7), (5,5)$ y $(5,3)$, y es posible tomar todos los vértices que lo componen de la forma $A(x,y)$ para simplificar. (Figura 2.3 a)

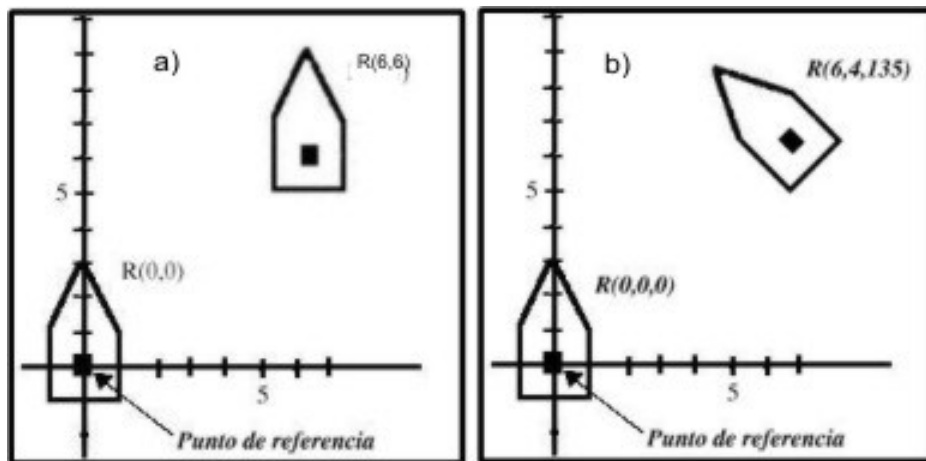


Figura 2.3 Se muestra la posición inicial del robot en $A(0,0)$ y la posición final en $A(6,6)$.

Ahora supongamos que el robot se mueve sobre su eje (orientación), β , tenemos ahora un parámetro extra que describe la orientación del robot. Se tiene ahora $A(x,y, \beta)$ el cual describe la posición en (x,y) y la rotación como un ángulo β , tomando como referencia a las manecillas del reloj. Así, tenemos una posición inicial en $A(0,0,0)$. Como se muestra en la Figura 2.3 b)

En general, la posición está dada por el número de parámetros que corresponden al número de grados de libertad³ del robot (N). Este número es dos, si el robot sólo puede trasladarse de un punto a otro, y es tres si el robot puede rotar y trasladarse. El número de parámetros que se requieren para estudiar un robot en tres dimensiones resulta muy alto.

El entorno del robot A, usualmente se le denomina *espacio de configuración* y se denota como $C(A)$. Un punto p, en este espacio de configuración, corresponde a

³ Ver apéndice A.

una cierta posición $A(p)$ del robot en el espacio de trabajo, que es el entorno donde se mueve el robot.

En el caso en que el robot puede rotar y trasladarse en el espacio de configuración significa que el robot se mueve en un espacio de tres dimensiones, donde un punto (x,y, β) corresponde a una posición $A(x,y, \beta)$ en el espacio de trabajo.

Si se requiere trasladar el robot sin rotar sobre su eje, se tiene un espacio euclidiano de dos dimensiones, con lo que se observa que el espacio de configuración es idéntico al espacio de trabajo. A partir de esto, se pueden distinguir: a) El espacio de trabajo es el entorno donde el robot se mueve (el mundo real) y b) El espacio de configuración que refleja la percepción del robot de su entorno.

Un robot en forma de polígono en el espacio de configuración se representa como un punto en el espacio de configuración, y cualquier punto en el espacio de configuración corresponderá a algún punto en el espacio de trabajo.

Un punto en el espacio de configuración representa cierta posición dentro del espacio de trabajo. Si se observa es claro que no todos los puntos en el espacio son válidos. Algunos de éstos cruzan con los obstáculos (estos puntos provocarían colisiones), por lo que son denominados áreas prohibidas o no válidas y denotadas por $CPr()$ ($Cforb()$). El resto del espacio corresponde a áreas libres de colisiones, donde el robot no intercepta ningún obstáculo, se le conoce espacio de configuración libre o espacio libre. Y se denota como $CLi()$ ($Cfree()$), Figura 2.4 *i)* y *ii)*

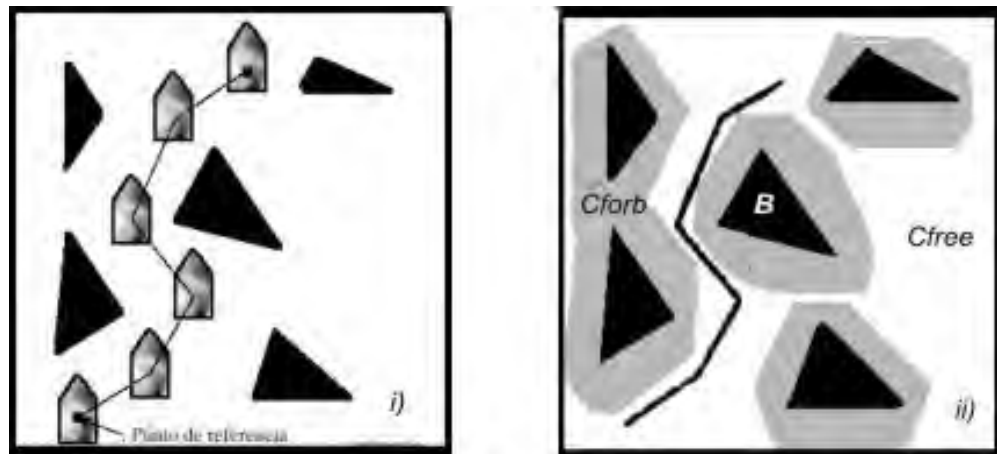


Figura 2.4 Espacio de trabajo y su correspondiente en el espacio de configuración, las zonas marcadas en gris son las zonas prohibidas.

Cada una de las rutas trazadas en el espacio de trabajo tiene su ruta correspondiente en el espacio de configuración y viceversa. En la Figura 2.4 *i)* se muestra el robot en el espacio de trabajo, el cual se traslada del punto inicial al punto de llegada a través del espacio libre de obstáculos, mientras que en la 2.4 *ii)*, muestra el espacio de configuración, donde se observan las áreas prohibidas

CPr (color gris) donde el robot y el área libre están en blanco. El camino indicado corresponde al camino libre de obstáculos.

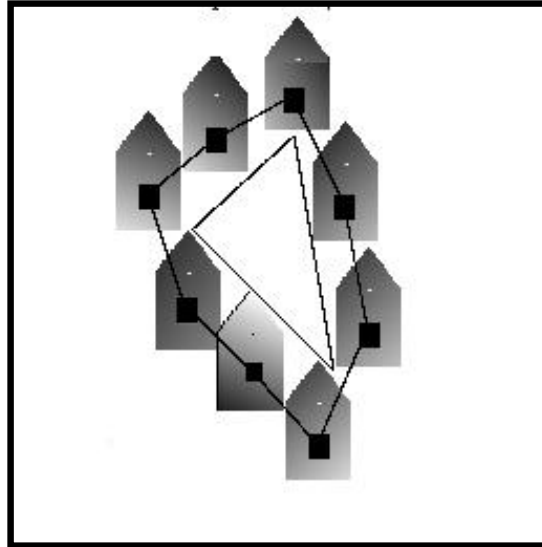


Figura 2.5 Construcción del espacio de configuración

La Figura 2.5 muestra cómo se construye el espacio de configuración, el robot visualiza el conjunto de obstáculos y delimita las áreas que no son válidas. Construyendo áreas como las mostradas en la Figura 2.4 ii).

Para su estudio, se requiere formalizar los conceptos de lo que ya se ha hablado. Así, el espacio de configuración queda definido como: Si A es un objeto rígido, moviéndose en un espacio euclidiano $W = \mathbb{R}^N$, $N=2$ ó 3 , y $B_1 \dots B_n$, son objetos rígidos en W , Los B_i son obstáculos, si A forma un subconjunto de W , y los obstáculos $B_1 \dots B_n$ son conjuntos cerrados de W , entonces la configuración de A es la descripción de todas las posiciones de cualquier punto en A con respecto a F_w ; donde F_w es el sistema cartesiano. El subconjunto de W ocupado por A , como la configuración de q , es $A(q)$. Un camino de una configuración inicial q_{init} a una configuración final q_{fml} es un camino continuo, $\tau: [0,1] \rightarrow C$ con $\tau(0) = q_{init}$ y $\tau(1) = q_{fml}$.

El *espacio de trabajo* contiene un número finito de obstáculos B_i , donde $i=1, \dots, n$. Cada uno de estos obstáculos B_i , representados en C , son llamados $C_{obstáculo}$:

$$C(B_i) = \{q \in C \mid A(q) \cap B_i \neq \emptyset\}$$

La unión de todos los $C_{obstáculo}$ forman la $C_{región\ obstáculo}$,

$$\bigcup_{i=1}^n C(B_i)$$

Por lo tanto, el sistema será :

$$C_{Li} = C \setminus \bigcup_{i=1}^n C(B_i) = \{q \in C : A(q) \cap (\bigcup_{i=1}^n B_i) = \emptyset\}$$

Donde C_{Li} es definido como el espacio libre. El espacio libre de obstáculos entre dos configuraciones es cualquier camino continuo $\tau: [0,1] \rightarrow C_{Li}$

Grado de dificultad

Para la realización de este trabajo, se asume que los obstáculos son fijos, y que es un robot único, que tiene la característica de ser un simple objeto rígido, pero pueden existir varios escenarios donde un robot interactúe con múltiples objetos y otros robots que se mueven a su alrededor, lo que incrementa su dificultad de resolución

El grado de dificultad de la planeación de rutas en robótica depende de un par de factores: si la información de los obstáculos en su entorno son conocidos (ej. Tamaño y/o localización, entre otros.), y si éstos están en movimiento o estáticos, mientras el robot se mueve. La Tabla 2.1 muestra las posibles combinaciones de dichos factores.

	Obstáculos estáticos	Obstáculos en movimiento
Conocidos	Caso I	Caso II
Desconocidos	Caso III	Caso IV

Tabla 2.1 Diferentes escenarios en la planeación de rutas

El escenario más simple es el Caso I, donde todos los obstáculos se encuentran inmóviles, y los detalles acerca del obstáculo se conocen antes de la planeación de rutas. Los Casos II y III implican un nivel de complejidad mayor hasta llegar al Caso IV, que representa las situaciones a las que se enfrentaría el robot en la vida cotidiana

El trabajo que aquí se desarrolló se enfoca hacia una situación donde los obstáculos son conocidos y se encuentran estáticos; este problema es usualmente resuelto a través de los siguientes pasos:

- Definir una gráfica que represente la estructura geométrica del ambiente
- Realizar un algoritmo de búsqueda que conecte el nodo que contiene el punto de partida y el de llegada, con lo que se obtiene una ruta libre de obstáculos

La gráfica que contiene la estructura geométrica depende del método usado, los seis métodos más usados para la planeación de rutas serán presentados en la siguiente sección.

Métodos de planeación de rutas

Existen diversos algoritmos de aproximación para la planeación de rutas. Latombe presentó una excelente revisión de los métodos utilizados (Latombe, 1991). Según Latombe estos métodos de aproximación se clasifican en: Mapa de ruta (Roadmap), Descomposición en celdas (Cell Decomposition) y Campos potenciales (Potential Field). En esta sección se dará una revisión de los métodos existentes para dar una visión de las investigaciones desarrolladas y que son objeto de estudio en recientes años (Prahlad, 2003), (Ramakrishnan, 2001), (Atkar, 2001).

Mapa de ruta (Roadmap)

La aproximación por mapas de ruta consiste en crear una serie de conexiones, a partir del entorno conocido, espacio libre (CLi), ubicando una red unidimensional de caminos. Una vez construida, se utiliza para crear una serie de rutas estándar. Reduciéndose el problema a solo conectar el punto inicial y final a esta red, buscando una ruta apropiada entre estos dos puntos.

Existen varios métodos basados en esta idea en general, estos métodos tienen distintos tipos de mapa de ruta.

Gráfica de visibilidad (Visibility graph).

El método de gráfica de visibilidad (Vgraph) no es un método de graficación directo, donde sus nodos representan el punto de partida (Pini), el punto de llegada (Pfnl) y todos los vértices de los obstáculos. Este método considera las conexiones, como todos los segmentos que conecten dos nodos y que no cruzan con el interior de ningún obstáculo, las otras uniones en la gráfica son Pini y Pfnl. La Figura 2.6 muestra un ejemplo.

El método de gráfica de visibilidad es un método no directo de graficación, restringido por las siguientes condiciones:

- I) Los nodos son Pini, Pfnl y los vértices de los obstáculos, (éstos tienen una distancia finita)
- II) Dos nodos en la gráfica son conectados por una liga, si y sólo si cualquier unión de estos segmentos es un borde del obstáculo. O si estas líneas se encuentran en el espacio libre, excepto cuando se tienen dos puntos finales.

La mayoría tienen aplicaciones en espacios de configuración de dos dimensiones con obstáculos en forma de polígonos

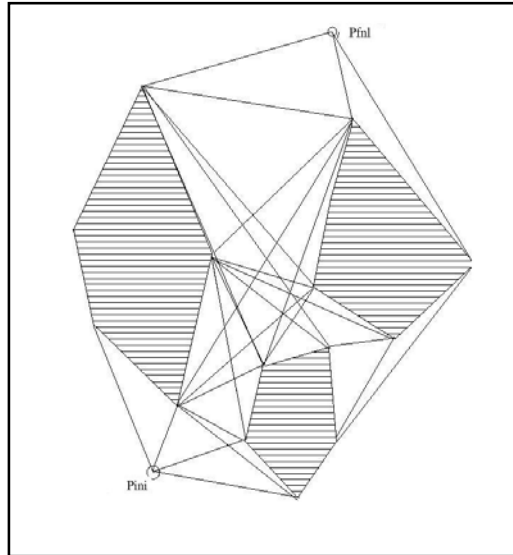


Figura 2.6 Gráfica de visibilidad.

Diagramas de Voronoi.

Los diagramas de Voronoi (DV) son una estructura geométrica que representa puntos o bordes de objetos. El plano se divide, asignando a estos puntos o bordes un área cercana, pero no es la única forma, ya que el espacio puede ser dividido, localizando puntos equidistantes de dos o más puntos o bordes de los objetos, incluyendo el borde del espacio de trabajo. El DV consiste en una red de líneas y segmentos de líneas parabólicas. La ventaja de estos diagramas consiste en que minimizan el espacio entre los obstáculos y el robot. Cuando los obstáculos son poligonales, el diagrama de Voronoi contiene segmentos rectos y parabólicos. Ver Figura 2.7

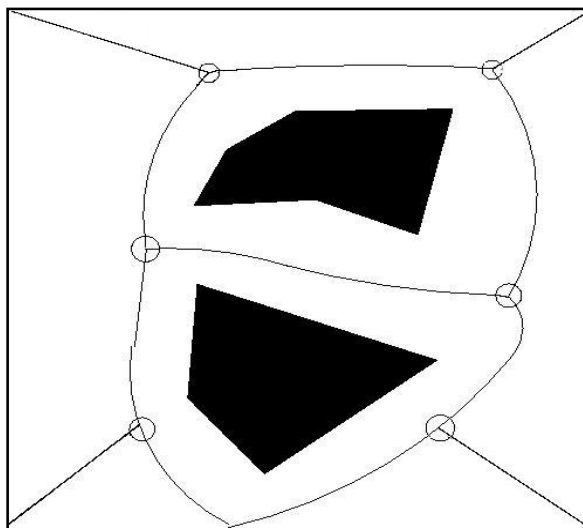


Figura 2.7 Diagrama de Voronoi.

El algoritmo para la planeación de la ruta, en esencia, consiste en encontrar la secuencia de segmentos en el diagrama de Voronoi, tal que conecten el punto inicial con el punto final (retracción). Dicha secuencia forma la ruta buscada.

A continuación se describe el algoritmo:

- Calcular el diagrama de Voronoi.
- Calcular punto inicial y final.
- Encontrar la secuencia de segmentos $\{s_1, \dots, s_p\}$ tal que el punto inicial y el final pertenezcan a s_p .
- Si se encuentra dicha secuencia, mostrar la ruta. Si no indicar condición de error.

Al igual que los grafos de visibilidad, este método trabaja en entornos totalmente conocidos y con obstáculos modelados mediante polígonos. Sin embargo, también existen versiones para la utilización del mismo con obstáculos inesperados (Ji Yeong et al, 2004).

Camino libre (Freeway net).

Este método es aplicado a objetos poligonales que tienen movimiento de translación y rotación en un espacio de trabajo poligonal de dos dimensiones. Este método es similar al usado en la retracción del espacio libre en los diagramas de Voronoi.

El método de camino libre consiste en la extracción de Figuras geométricas llamadas camino libre del *espacio de trabajo*, conectando estas Figuras geométricas dentro de una gráfica llamada *red de camino libre*. Un camino libre es un cilindro rectilíneo generalizado (CRG), donde el eje es creado, se conserva la descripción de las orientaciones de A, y donde el punto de referencia O_A se mueve a lo largo de éste.

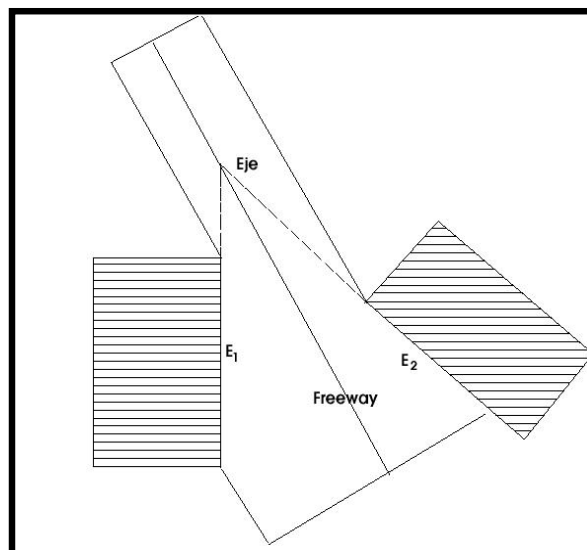


Figura 2.8 Un camino libre (freeway) es una red de cilindros rectilíneos generalizados.

La construcción de un CRG se realiza a partir de las aristas de los distintos obstáculos que se encuentran en el entorno. Para que un par de aristas E_i y E_j , pertenecientes a los obstáculos B_1 y B_2 , respectivamente, puedan formar un cilindro generalizado, deben cumplir las siguientes condiciones:

- i) Para $i, j \in \{1, 2\}$. Las dos extremidades de E_i deben estar por completo en la región opuesta en la que se encuentra situada E_j . Este criterio es simétrico.
- ii) El producto integrado escalar de los vectores normales salientes del obstáculo que contiene cada arista debe resultar negativo.

Si se cumplen estas condiciones significa que ambas aristas se encuentran enfrentadas, y por tanto se puede construir un CRG con ellas. Una vez detectadas dos aristas que pueden formar un CRG, el siguiente paso será construirlo. El proceso para alcanzar este cometido, se encuentra descrito en la Figura 2.8

El primer paso es el cálculo del eje del CRG, el cual se define como la bisectriz del ángulo formado por el corte de las rectas que contienen las aristas E_1 y E_2 que cumplen las condiciones i) y ii). Por ambos lados de dichas aristas se construyen segmentos rectilíneos paralelos al eje, con origen en los vértices de las aristas implicadas y con extremo señalado por la proyección del primer obstáculo que corta el eje, repitiendo este proceso con todos los obstáculos.

Descomposición en celdas (Cell Decomposition).

Es quizá uno de los métodos más estudiados en la planeación de rutas. Consiste en descomponer el espacio libre del robot en regiones, llamadas celdas. Tal que una trayectoria entre dos configuraciones en una celda sea fácilmente generada.

Para el método de descomposición en celdas se precisa la resolución de dos problemas: la descomposición del espacio libre en celdas y la construcción de un grafo de conectividad

Los métodos de *Descomposición en celdas* pueden clasificarse dentro de métodos exactos y aproximados.

La Figura 2.9 representa el método de descomposición exacta en un espacio bidimensional. El espacio libre es descompuesto en celdas trapezoides o triangulares, estas celdas son construidas generando rayas verticales en los vértices de los obstáculos. Dos celdas son adyacentes si comparten alguna parte. La ruta libre se encuentra al conectar el punto inicial y el final a través de los puntos intermedios, buscando todos los puntos consecutivos que describan la ruta más corta.

La “*Gráfica de conectividad*” (*Connectivity graph*) es la representación de celdas adyacentes, es decir, los nodos en la gráfica son celdas en el espacio de

configuración, estos nodos son conectados sólo si éstos corresponden a una celda adyacente, Figura 2.9.

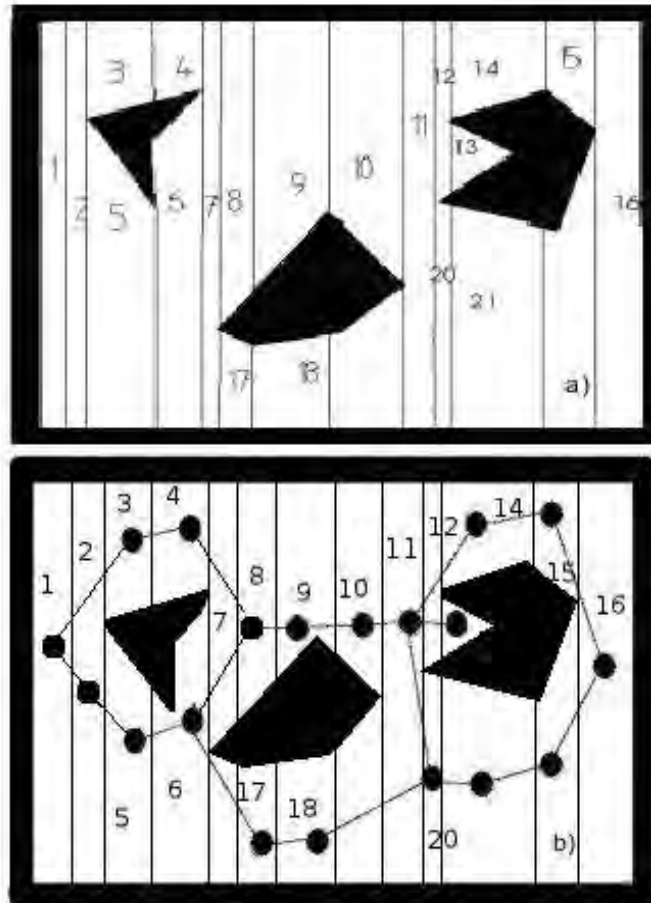


Figura 2.9 Método de descomposición de celdas exacto, a) Muestra la descomposición del espacio en regiones; b) Muestra los puntos intermedios.

En la Figura 2.10 se observa el método de descomposición de celdas aproximado, donde el espacio es descompuesto recursivamente en pequeños rectángulos, esta descomposición genera cuatro nuevos rectángulos (esta descomposición es llamada quadtree, porque puede ser representada como un árbol de grado cuatro, si las líneas están libres de obstáculos la descomposición se detiene, si no es así, ésta se sigue descomponiendo en cuatro rectángulos sucesivamente, la descomposición tiene una resolución predeterminada.

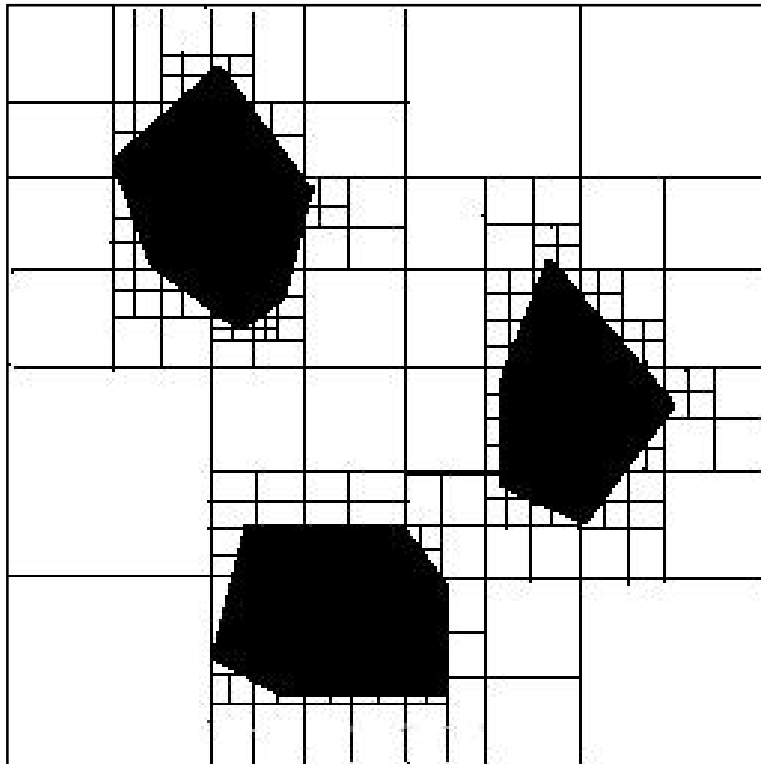


Figura 2.10 Método descomposición de celdas aproximado, quadtree.

Campos potenciales (Potential Field).

El robot es representado como un punto en el espacio de configuración, esta partícula se mueve bajo la influencia de un campo potencial artificial producido por los obstáculos y la posición final o punto de llegada. La función de campos potenciales determina el movimiento del robot, el robot; se moverá en términos de dos diferentes campos potenciales. El punto de llegada o posición final por lo general generará un campo potencial de atracción, y los obstáculos generarán un campo de repulsión, el cual empujará o alejará al robot. El camino entre el punto inicial y el final es generado al seguir el gradiente o pendiente negativa.

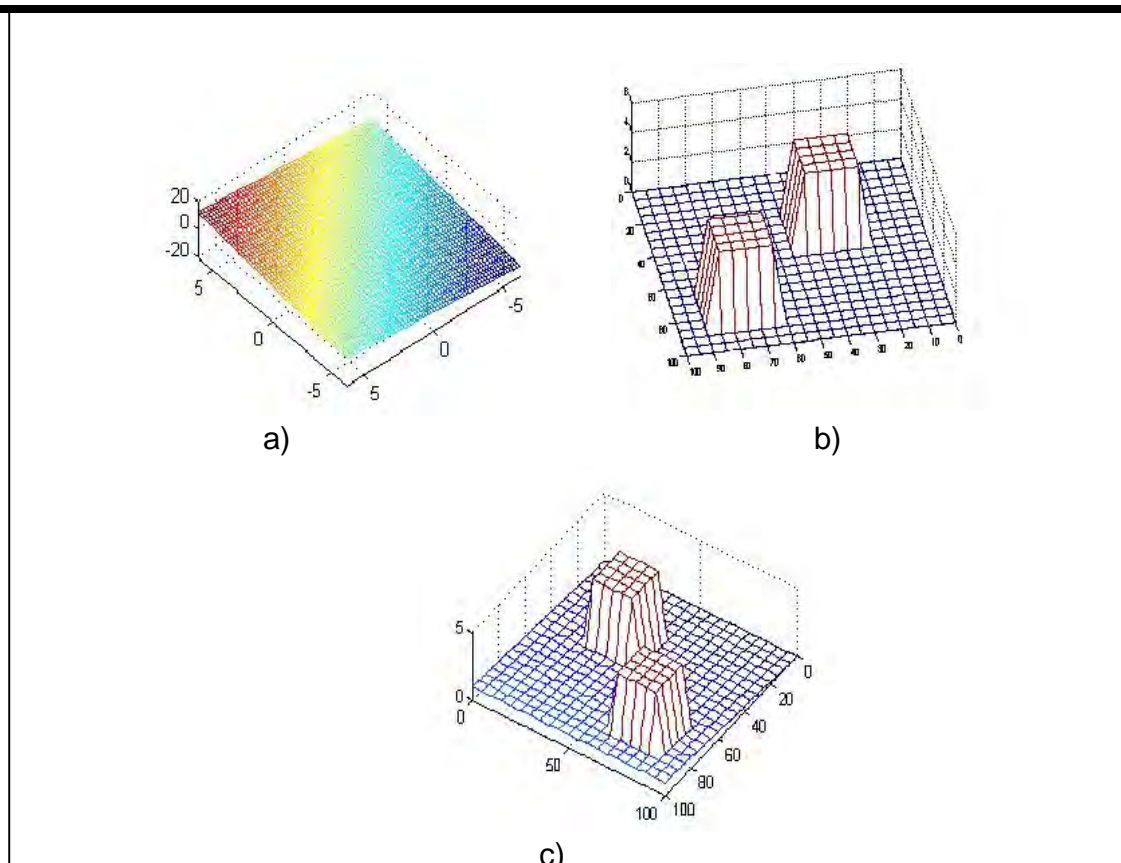


Figura 2.11 Muestra las fuerzas de atracción y repulsión y sus combinaciones. Las fuerzas de repulsión son cero a cierta distancia del obstáculo y tienden a infinito, cuando la distancia hacia el obstáculo tiende a cero. El camino es construido siguiendo, las fuerzas negativas del gradiente.

La desventaja de este método es que puede quedar atrapado en mínimos locales, es decir, el robot queda en lugares que no son la posición destino, donde el potencial resulta nulo. Una situación de este tipo puede hacer que el robot quede atrapado en una posición que no sea el destino, o bien, debido a la naturaleza discreta del método girar alrededor de ella. Solucionar este conflicto implica definir ciertas funciones potenciales que eviten la aparición de mínimos locales, lo cual resulta arduo, si bien existen soluciones que lo aseguran en entornos donde los obstáculos están modelados mediante círculos (Rimon, Koditschek, 1988). Otra solución para evitar caer en un mínimo local se encuentra en el uso de un algoritmo de búsqueda en grafos. Para ello se divide el entorno mediante el uso de una rejilla. Cada celda tiene almacenado un valor que indica su potencial. O bien crear mecanismos que permitan que logre salir del mínimo local.

La planeación de rutas, como ya se ha expuesto, es un problema complejo al que este trabajo propone darle solución a través de la utilización de Algoritmos Genéticos, como se expone en los siguientes capítulos (4 y 5).

CAPÍTULO 3. ALGORITMOS GENÉTICOS

Es este capítulo se hace una revisión de los algoritmos genéticos para describir su funcionamiento y las diferentes partes que lo componen, en el capítulo 5 adaptaremos los algoritmos genéticos para la planeación de rutas.

Los algoritmos genéticos (AG) son métodos sistemáticos de búsqueda de soluciones y optimización, que simulan los mecanismos de selección natural y de la genética. La selección natural y la genética son conceptos originados en biología desde el siglo XIX, con las investigaciones de Charles Darwin sobre evolución de las especies (1859) y Gregor Mendel (1863) sobre germinación de chícharos en condiciones controladas

Las ideas esenciales de la computación evolutiva engloban las técnicas que simulan la evolución natural que fueron aportadas en el año de 1965 por Rechenberg. Más adelante, John Holland (1968-1971) y sus estudiantes realizaron investigaciones con algoritmos genéticos; las áreas fundamentales de su trabajo fueron abstractas y rigurosos procesos de adaptación de sistemas naturales y su simulación por computadora

Los algoritmos genéticos se han utilizado en la solución de problemas como: optimización de funciones matemáticas, el problema del agente viajero, optimización de circuitos, distribución de plantas, espacios de trabajo, o de negocios (Michalewicz, 1994), así como en otras áreas de investigación como química, aeronáutica y eléctrica (Montagnon, 1999), telecomunicaciones, juegos, robótica, acústica, entre otros. Con el tiempo los algoritmos genéticos se han modificado para mejorar su desempeño (Eksin, Erol, 2001) o para mejorar su aplicación a problemas específicos y en combinación con otras herramientas de inteligencia artificial.

La optimización.

La optimización tiene sus orígenes en la Segunda Guerra Mundial, cuando existía la necesidad de asignar recursos a las diferentes operaciones militares de forma efectiva. Es por esto que los gobiernos estadounidense e inglés hicieron uso de los científicos para resolver los problemas estratégicos y tácticos que se presentaban. Al término de la guerra varios de estos científicos aplicaron sus conocimientos a la industria y a los gobiernos. Muchos de los esfuerzos en el campo de la investigación de operaciones se enfocaron a la mejora de las técnicas utilizadas, surgiendo así, la programación lineal, la programación dinámica, teoría de inventarios y teoría de juegos.

El principal objetivo de la optimización es la obtención del mejor resultado a un problema, minimizando o maximizando los recursos existentes. Por ejemplo, dentro de una empresa se quieren reducir los costos de producción y obtener las mayores ganancias, ¿Qué alternativa sería la más factible dentro de la empresa, sin afectar los intereses de cada uno de los departamentos que la componen?.

Para contestar esta cuestión es necesario tomar en cuenta todos los factores que influyen respecto de la operación de la empresa, como son los trabajadores, la competencia y los proveedores. Algunos de estos factores representan restricciones y ayudan a delimitar las soluciones o alternativas factibles de las que no lo son. Al delimitar el problema se tiene un universo de soluciones más pequeño, donde, se busca la mejor solución al problema. Después de identificar cuál es el objetivo y delimitar el problema, es necesario expresar en términos matemáticos la esencia del problema, a ésta se le denomina como función objetivo. Esta función objetivo está construida por modelos matemáticos (ecuaciones y desigualdades) establecidas en términos de variables

Es decir, dada una función $f(x)$ -función de costo-, encontrar el valor de un vector de entrada, tal que la salida de la función sea un mínimo o un máximo, cumpliendo el vector de entrada una serie de restricciones. La *función de costo* es la función que determina cuán válido es el vector de acuerdo con al problema que se resuelva.

Los problemas de optimización no siempre están formulados de una forma clara, la forma de la función F no se conoce, y debe aproximarse mediante polinomios o combinaciones de funciones conocidas, en estos casos se trata de encontrar los coeficientes de los polinomios o funciones que hacen que la función calculada esté lo más cercana a la función objetivo. En estos casos, se habla de *optimización paramétrica*

Otro tipo de optimización de problemas se denomina *optimización combinatoria*, en este caso se trata de optimizar $F(c)$, donde C es una combinación de diferentes elementos que pueden tomar un número finito de valores; pueden ser combinaciones con o sin repetición, e incluso permutaciones. Este tipo de optimización es la que se encarga la computación evolutiva y los algoritmos genéticos de resolver.

La planeación de rutas podría plantearse como un problema de optimización donde el objetivo principal es minimizar la distancia entre un punto inicial y el punto donde queremos llegar, esta búsqueda resulta compleja, debido a que se requiere analizar una gran cantidad de soluciones posibles para seleccionar la óptima y por ello se ha decidido trabajar con algoritmos genéticos, para generar o planear rutas óptimas. Por otro lado los AG (Algoritmos Genéticos) son intrínsecamente paralelos, por lo cual resultan de fácil programación en estructuras paralelas (clusters) lo que reduce el tiempo de ejecución del programa de búsqueda y los hace muy atractivos, en comparación con otros métodos de optimización.

Goldberg menciona que los algoritmos genéticos tienen 4 importantes diferencias con los métodos tradicionales de optimización, (Goldberg, 1988);

Los cuales son:

- 1.Los AG's trabajan con el código de los parámetros, y no con los parámetros.
- 2.Los AG's trabajan con una serie de puntos, no con un solo punto.
- 3.Los AG's utilizan la información salida de la función objetivo, no utiliza derivadas u otro conocimiento auxiliar.

4. Los AG's usan reglas probabilísticas, no reglas determinísticas.

Algoritmo genético.

Como parte de este trabajo se programó un Algoritmo Genético Simple (AGS) como el propuesto por Goldberg.

Los algoritmos genéticos, utilizan analogías con conceptos de la selección natural (Tabla 3.2), en la naturaleza los cromosomas contienen cadenas de ADN (Ácido desoxirribonucleico) que en los AG's son conjuntos de cadenas de unos y ceros. Estas cadenas son decodificadas dentro de la célula para formar proteínas que modifican el comportamiento y/o composición de la célula. En la naturaleza es esencial, tanto la secuencia de la cadena de caracteres nucleótidos como la posición en que se encuentran de la cadena (locus), el mínimo conjunto de cadenas de nucleótidos que se expresarán en una proteína se llama gen; la decodificación de la secuencia de ADN se lleva a cabo por el RNAm (mensajero) el cual realiza una copia del ADN (RNAt -transferencia), esta copia se utiliza para producir proteínas.

Una falla en este proceso en la naturaleza puede resultar grave, ya que existen organismos que contienen un solo gen. Asimismo los cambios en el ADN reciben el nombre de mutaciones. Debido a que la información en un gen se codifica en la secuencia específica de bases, una mutación es un cambio en la secuencia de bases.

Aunque la mayor parte de las mutaciones son inocuas o neutrales, las mutaciones son esenciales para la evolución, ya que finalmente todas las variaciones genéticas se originan como cambios aleatorios en la secuencia del ADN. Una forma de cambiar el ADN y que ayuda a la evolución de las especies es la recombinación, ya sea artificial o sexual.

Para entender el funcionamiento de los AG's conviene analizar el estudio del comportamiento de una población de ratones, presentado por Jorge Soberón en su libro: *Ecología de Poblaciones*, donde se observan factores que son determinantes en la evolución de las especies (Soberón, 1989).

Imaginemos una isla en medio de un lago, que se encuentra cubierta de pastos y algunos arbustos, poblada por insectos y una sola especie de ratones, que es visitada ocasionalmente por tecolotes de las orillas del lago. Existen cuatro causas que producen cambios en el número de ratones de la población de la isla: nacimientos, muertes, emigración e inmigración. Las dos últimas se ignorarán para no complicar el modelo. Baste ahora recordar que las tareas vitales de conseguir alimento, eludir a los depredadores, resguardarse de las condiciones climáticas extremas y conquistar una pareja son los principales problemas que tienen que enfrentar los seres vivos para sobrevivir.

Estos problemas influyen directamente en el número de muertes, y las fluctuaciones de la población están dadas por la relación entre decesos y nacimientos de ratones. Si el invierno es muy crudo morirán más ratones que en un año favorable, si los tecolotes del litoral descubren que existen muchos ratones

en la isla la visitarán más y la población de ratones disminuirá, o bien, si la población de insectos aumenta, afectarán la producción de semillas y los ratones tendrán el riesgo de sufrir escasez de alimento y por lo tanto la mortalidad aumentará. En suma, cualquier factor que influya en el clima, en la competencia con otras especies o el actuar de los depredadores, determinará la población de ratones.

Ahora, ¿quién determina respecto de quién sobrevive a la acción de los tecolotes, ante la hambruna o a las inclemencias del clima? ¿Cuáles serán los individuos que obtendrán pareja y contribuirán con la siguiente generación? Lo que pudo observar Charles Darwin (Darwin, 1859), es que no todos los individuos de una especie tienen la misma probabilidad para reproducirse, llegar a la edad adulta y mantenerse en ella (esto es por que no todos los seres vivos son idénticos). Los ratones pueden parecer iguales a simple vista, pero si se observan, unos estarán más gordos, otros más peludos, comerán de diferentes tipos de granos, etc. A estas características individuales Charles Darwin las llamó “descendencia con variación”. Es decir, que todos los seres vivos tienen descendencia con características parecidas, pero no idénticas.

Esta variabilidad entre los individuos de una misma especie trae como consecuencia que la mortalidad o el éxito reproductivo, no están determinados puramente por el azar. Entonces, si existe un ratón con mucho pelo, resistirá mejor el invierno crudo y dejará descendencia, a diferencia del que no lo tiene, que posiblemente morirá.

Este proceso de interacción entre las distintas variables de un modelo natural determinado lleva a condiciones que favorecen a ciertos individuos con respecto a otros. Si llevamos este proceso a un modelo matemático que lo simule, donde los ratones (del ejemplo) son expresiones matemáticas, y lo que buscamos es un punto o solución (que en el ejemplo serían ratones con ciertas características físicas o habilidades) existente en la población o “nacido” de ella, podremos resolver cualquier problema de búsqueda u optimización. A este modelo se le llamó algoritmo genético.

En la computación evolutiva se utiliza un símil de los términos usados en la genética para describir la estructura de los individuos de la población que son manipulados por el algoritmo. Normalmente se utiliza la representación binaria por ser sencilla, fácil de manipular, de ser transformada en números reales y, además, porque facilita la explicación de teoremas. En ocasiones, no es posible usar una representación binaria y es necesario usar un alfabeto con más símbolos, cualquiera que sea la representación utilizada, ésta debe abarcar todas las posibles soluciones del problema que se quiere resolver.

A continuación presentamos una Tabla 3.2 comparativa de los términos (Tabla 3.2) utilizados en las ciencias naturales y los utilizados en los AG :

NATURAL	ALGORITMO GENÉTICO
Cromosoma	Conjunto de características o rasgos.
Gen	Rasgo o característica (cadena de números)
Alelo	Valor de la característica (expresado en números)
Locus	Posición del rasgo dentro del conjunto de características.
Genotipo	Estructura
Fenotipo	Decodificación de la estructura
Epítasis	No lineal

Tabla 3.2 Comparación entre términos Naturales y Algoritmos Genéticos. (Holland, 1975)¹.

Un algoritmo genético puede verse como un proceso continuo que repite ciclos de evolución, teniendo un criterio de parada, como se muestra en la Figura 3.3. Los AG's algoritmos son métodos estocásticos, esto dificulta la especificación de un criterio de parada, una población puede quedar estática por varias generaciones antes de encontrar un individuo con mejores aptitudes al medio. Una práctica común en los algoritmos genéticos es detener el algoritmo después de un número determinado de ciclos y ver la calidad de los individuos de la población con respecto a la definición del problema, si las soluciones no resultan satisfactorias el algoritmo puede ser reiniciado.

Componentes de un algoritmo genético

En términos generales los pasos a seguir por un algoritmo genético simple son los siguientes:

- Transformar el problema en una expresión matemática.
- Decodificar las posibles soluciones en números binarios a números decimales.
- Evaluar los individuos para determinar la factibilidad de reproducción de cada uno (adaptación al medio artificial), calculando las condiciones de cada individuo de la población en la función objetivo (función de costo).
- Seleccionar los descendientes de la siguiente generación.
- Reproducir los individuos seleccionados por medio de operadores genéticos:
 - Selección.
 - Cruza.
 - Mutación.
- Convertir a la población resultante en población inicial, y
- Repetir el ciclo un número predeterminado de generaciones o aplicar algún criterio de convergencia o terminación.

¹En la terminología de las ciencias naturales, se dice que los cromosomas están compuestos por genes, que pueden tener algún número de valores llamados alelos. En genética, la posición de un gene determina alguna característica particular de algún animal, por ejemplo el color de los ojos. En los algoritmos genéticos se dice que las cadenas están compuestas por características, que adquieren diferentes valores, dependiendo de la posición donde se encuentran.

En la Figura 3.3 se observa un algoritmo genético simple:

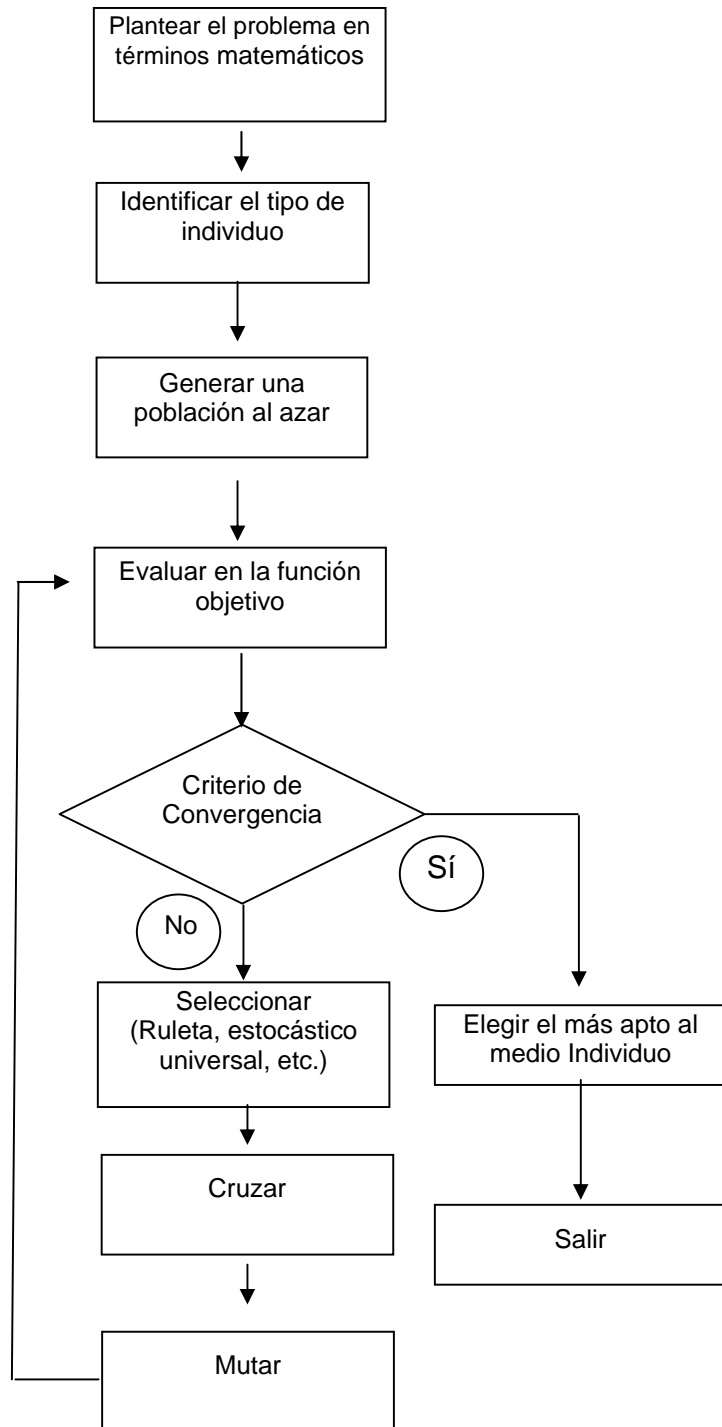


Figura 3.3 Funcionamiento de un algoritmo genético

Ahora es posible explicar las componentes que integran un algoritmo genético para entender su funcionamiento.

Los algoritmos genéticos requieren que las posibles soluciones de una población sean codificadas en cadenas de símbolos o letras que generalmente son un código binario con cadenas de unos y ceros. Por ejemplo, en la Tabla 3.1 tenemos que cada parámetro que compone a un individuo P1 puede ocupar las posiciones del 0 al 2, P2 ocupará las posiciones del 3 al 5, etc. El número de bits usado en cada parámetro determinará el número de características de cada individuo.

P1	P2	P3
001	010	01010

Tabla 3.1 Codificación binaria de los parámetros de una solución factible.

Cada una de estas estructuras es evaluada por medio de la función objetivo, asignándole una medida generalmente llamada aptitud (*fitness*, medida de desempeño o adaptación al problema). Las probabilidades de sobrevivir a la siguiente generación están dadas por su estructura basándose en su valor particular, es decir, en el ejemplo de los ratones, cada ratón tiene características distintas, unos son más peludos, otros son gordos, flacos, con gustos alimenticios distintos, que le dan un mayor desempeño y mayores probabilidades de sobrevivir dentro del medio en que la población de roedores se encuentra. En un algoritmo genético cada ratón estaría representado con un mayor o menor número de ceros, dependiendo de estas características.

En el AG's quién vive y quién muere lo determina el operador de selección utilizado, lo que desplazará a ciertos individuos y hará que predominen otros, los cuales dejarán descendencia con mayores probabilidades de sobrevivir.

En la programación de algoritmos genéticos, las nuevas generaciones de estructuras son creadas al seleccionarlas de acuerdo con las probabilidades asignadas en la presente generación, teniendo varios operadores que ayudan a simular el proceso de selección llamados **operadores genéticos**.

Esquemas de reproducción.

La selección en algoritmos genéticos toma a individuos para la reproducción, la selección se basa en la aptitud de los individuos: los individuos más aptos tienen mayor probabilidad de reproducción. En los AG's existen varios métodos de selección, como el método de selección de la ruleta, donde los individuos con mayor "aptitud" son más favorecidos. En el proceso de reproducción los individuos son copiados de acuerdo con su valor o aptitud al ser evaluados en la función objetivo, es decir, si el valor de un individuo es muy apto al ser evaluado en la función objetivo, las probabilidades de sobrevivir a la siguiente generación son muy altas. En el operador de reproducción de ruleta, a cada individuo se le asigna una sección circular proporcional a su valor de aptitud. Como se muestra en la Tabla 3.2 a la cadena de números (representación en números binarios de los individuos) después de ser evaluada en la función objetivo, se le asignará un porcentaje proporcional a su valor.

No	Cadena de unos y ceros	Valor de la cadena de unos y ceros después de ser evaluado En la función objetivo $f(x)=x^4$	% del total
1	01011	121	24.30
2	01010	100	20.10
3	01110	196	39.36
4	01001	81	16.26
Total		498	100.00

Tabla 3.2 Distribución del porcentaje de selección de acuerdo a su valor en la función objetivo

Después se le asignará una porción de acuerdo con su porcentaje en una ruleta o círculo, se gira la ruleta y se elige al individuo asociado a esa sección o casilla, dependiendo del valor donde queda la pelota. Siguiendo con el ejemplo, al igual que los ratones con mucho pelo sobreviven mejor a los inviernos crudos, en la selección de la ruleta, estos individuos tendrán mayor valor y por lo tanto, mayor probabilidad de ser seleccionados (sobrevivirán) y por supuesto dejarán descendencia (heredarán sus características genéticas) para generaciones posteriores, que para el AG es lo que interesa.

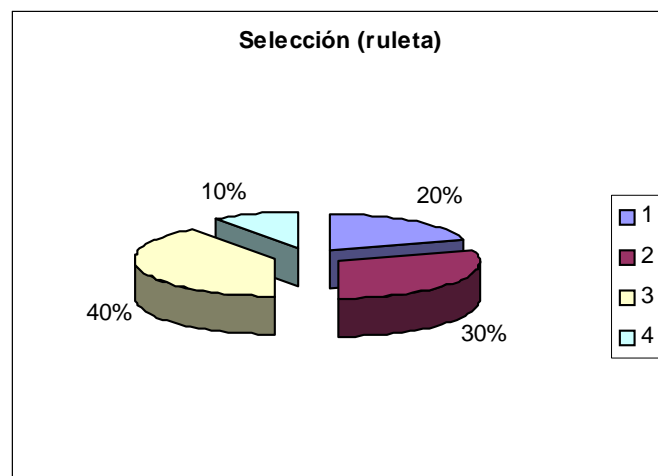


Figura 3.1. Representación de los porcentajes asignados a cada individuo de acuerdo con su valor en la función objetivo

Otros factores importantes son las probabilidades de cruce y de mutación que ayudan a la evolución de las poblaciones. Veamos cómo se incluyen en un AG estos factores.

Retomando el ejemplo de esta sección, suponemos que dos ratones (padre y madre), uno con mucho pelo y el otro con poco pelo, pero que han logrado sobrevivir a las inclemencias del tiempo por su capacidad de conseguir alimento, lo que les permite tener una reserva de grasa para sobrevivir, se cruzan, el resultado será que los ratones hijos tendrán características del padre, de la madre

o de ambos, heredando de cada progenitor atributos únicos que les ayudarán o no a sobrevivir bajo las condiciones existentes.

En los algoritmos genéticos al aumentar o disminuir las probabilidades de cruce se crean individuos con mejores o peores aptitudes para enfrentar al medio que los de la generación anterior, y sus probabilidades de ser seleccionados y heredar sus características genéticas disminuirán o aumentarán dependiendo del caso.

Por último, en la naturaleza al mutar un individuo, éste puede adquirir alguna característica que lo haga sobresalir sobre los demás individuos, dándole probabilidades a favor o en contra. Así, al modificar las probabilidades de mutar, se introducen nuevos genes a la población. Su objetivo es prevenir que todas las soluciones de la población se encuentren en óptimos locales para resolver el problema, vale mencionar que si aumentamos demasiado esta variable, la población tendría demasiadas variaciones y no podría adaptarse.

En la Figura 3.2 se muestra que al aumentar el factor de mutación (círculos) se logra un rápido acercamiento al óptimo, y ayuda al AG a salir de óptimos locales, mientras que, al tener una mutación baja (estrellas) el algoritmo tarda más generaciones en llegar al óptimo y es menos estable; otra de las ventajas de éste operador es la introducción de un mayor grado de diversidad en la población.

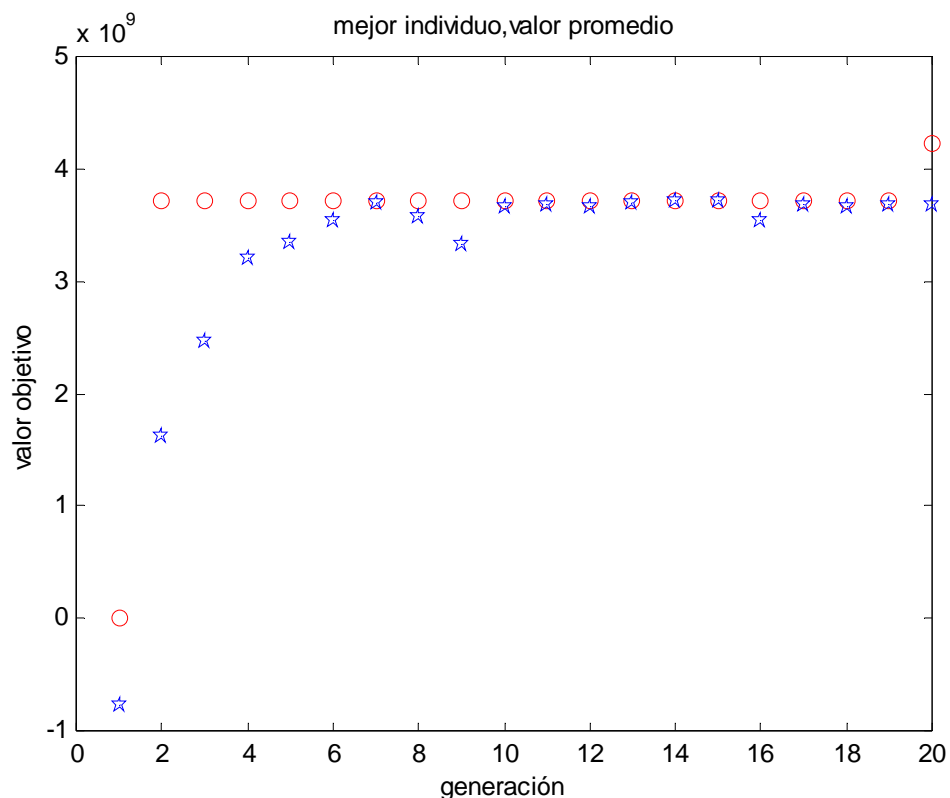


Figura 3.2 Efecto de la mutación

Métodos de selección usados.**Ranking.**

Baker introdujo la idea de selección por clasificación (Ranking) (Baker, 1985). El proceso de selección priorizada clasificará a la población según una posición jerarquizada, dependiendo de la aptitud del individuo, al seleccionar a los individuos de mayor a menor grado, con asignación de un valor máximo como límite máximo de descendencia, tomando al área bajo la curva como el tamaño de la población (Michalewicz, 1996).

O bien:

$$fitness(pos) = 2 - PS + (2) \frac{(PS - 1)(Pos - 1)}{Nind - 1} \dots\dots (3.1)$$

Donde:

- $Nind$ = Número de individuos
- Pos = Posición de un individuo en la población
- PS = Presión de selección

El método de clasificación Lineal permite valores de presión de selección entre uno y dos, [1.0, 2.0]

Otra forma de clasificación usa una distribución no lineal, el uso de la clasificación no lineal permite una presión de selección mas fuerte que el método lineal, calculándose como sigue:

$$fitness(pos) = \frac{(Nind)(X^{(Pos-1)})}{Sum(X^{i-1}) - 1}; = 1 : Nind \dots\dots (3.2)$$

X es calculada como la raíz de un polinomio

$$0 = (PS - 1)(X^{Nind-1}) + PS(XNind - 2) + \dots + (PS)(X) + PS \dots\dots (3.3)$$

La clasificación no lineal permite que los valores de la presión de selección se encuentren entre [1.0, Nind-2.0].

Con este método, se superan algunos problemas ocasionados por la asignación de probabilidades de acuerdo a su capacidad de adaptación.

Ruleta (roulette wheel)

El operador de reproducción de la ruleta, es cuando a cada individuo se le asigna una sección circular proporcional a su valor capacidad de adaptación. Una tirada de la ruleta y escoger al individuo asociado a esa sección o casilla donde caiga la pelota. Como se muestra, en la Tabla 3.2 y Figura 3.1. (Goldberg, 1989)

Es decir

$$P_i = \frac{F(x)}{\sum F(x)} \dots \dots \dots (3.2)$$

Donde:

$F(X)$ es el valor después de ser evaluado en la función objetivo

Es un método muy sencillo, pero ineficiente en la medida que aumenta el tamaño de la población, su complejidad es $O(n^2)$. Presenta además el inconveniente de que al momento de encontrar un individuo con súper-cromosomas que dominan el proceso de selección, provocará que la competencia entre individuos se vuelva débil, observándose solo una búsqueda aleatoria, o bien se estabilizará en algún máximo o mínimo local (convergencia prematura).

Estocástico universal

Aquí, los individuos son ordenados de manera similar a la ruleta, con la diferencia que aquí hay tantos punteros como individuos a seleccionar. Entonces, si consideramos que N es el número de individuos a seleccionar, tenemos que la distancia entre cada puntero es igual a $1/N$, y la posición del primer puntero está dada por una tirada al azar en un rango dado $[0, 1/N]$.

Cruza

El operador de cruza, consiste en el intercambio de material genético entre dos cromosomas o individuos.

Para llevar a cabo el cruza se escogen a dos individuos al azar (Figura 3.4), a quienes se denominará padres, además se escoge aleatoriamente un punto para hacer un intercambio de cromosomas (Figura 3.5).

$A_1=010101010101000$ $A_2=010101101011011$
--

Figura 3.4 Individuos escogidos al azar.

Supongamos que escogemos un punto al azar (el cual está indicado por "|") cortamos el código genético en ese punto y se intercambian sus cromosomas (representados por 0 y 1), el resultado de la cruce serán dos nuevos individuos que formarán parte de la nueva generación.

$$\begin{array}{l} A_1=0101010101010|11 \\ A_2=010101110101110|00 \end{array}$$

Figura 3.5 Individuos después del cruce.

Muta

La mutación se considera un operador básico, que proporciona un pequeño elemento de aleatoriedad en la vecindad (entorno) de los individuos de la población, esto altera el cromosoma de manera aleatoria. En general, consiste en cambiar uno o varios de los genes por un valor distinto. Este operador introduce nuevo material genético a la población. Como se observa en la Figura 3.6

$$\begin{array}{l} A_1=0101010101010|1|1 \\ A_2=010101110101110|0|1 \end{array}$$

Figura 3.6 Proceso de mutación.

En este operador se elige un individuo de la población al azar, se escoge un gene a mutar, y se lanza un volado, si cae dentro de las probabilidades de mutar, el gen será cambiado.

Teorema del esquema.

El Teorema del esquema se desarrollo con el objetivo de proporcionar un análisis matemático formal del comportamiento de un AG (Holland, 1992), que plantea las similitudes que existen en una población de cadenas y su aptitud al medio (evaluación en la función objetivo) que guía u orienta la búsqueda. El concepto de esquema se usa para descubrir esta información contenida en la población y explicar como la aprovechan los algoritmos genéticos.

Un esquema es un patrón que describe un subconjunto de cadenas con similitudes en ciertas posiciones de las cadenas. Se define un símbolo "*" que indica que no está determinado el alelo. Si usamos un alfabeto binario, el conjunto de caracteres sobre los que está definido un esquema $H = 1^*11^*10$ representa a los individuos $\{(1\ 0\ 1\ 1\ 0\ 1\ 0), (1\ 0\ 1\ 1\ 1\ 1\ 0), (1\ 1\ 1\ 1\ 0\ 1\ 0), (1\ 1\ 1\ 1\ 1\ 1\ 0)\}$. Donde los asteriscos indican una posición no importante para la definición del

esquema. Se observa que mediante el concepto de esquema, se están definiendo regiones del espacio total de soluciones. Así, en una representación binaria de longitud L , cada individuo de la población pertenece al orden de 2^L esquemas o regiones del espacio de búsqueda de la solución del problema

La finalidad de los esquemas es proporcionar de forma sencilla las similitudes de las cadenas que forman una cierta población a partir de un alfabeto finito, obteniéndose una mayor información y guiándolos hacia una mejor búsqueda.

La longitud de un esquema es la distancia entre la primera y la última posición especificadas del esquema. Se denota como $d(H)$, Ejemplo: $d(10***1**) = 5$

Estas propiedades son útiles en el desarrollo del teorema fundamental de los algoritmos genéticos que están contenidas en el teorema del esquema. Este teorema indica la expectativa del número de copias de un esquema en particular bajo la reproducción, cruce y mutación, y está dada por la siguiente expresión matemática:

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{f} \left[1 - p_c \frac{\delta(H)}{l-1} - P_m o(H) \right] \dots \dots (3.3)$$

Donde:

1. P_c y P_m son las probabilidades de cruce y mutación, respectivamente
2. l es la longitud de la secuencia (individuo)
3. f es el valor promedio de la población.
4. $o(H)$ es el orden de especificación del esquema H
5. δ es la distancia entre las posiciones exteriores del esquema
6. $f(H)$ es el valor promedio del esquema que se calcula con:

$$f(H) = \frac{\sum_{s_i \in H} f(s_i)}{m(H, t)} \dots \dots (3.4)$$

Este teorema de los esquemas se comprende más fácilmente si se consideran cada uno de los términos aisladamente se puede hacer el supuesto de que las tres operaciones – reproducción, cruce y mutación- son independientes, donde:

- El producto $m(H, t) \cdot f(H)/f$, representa el número esperado de ejemplares obtenidos en la operación de selección. El efecto de la reproducción es realizar un crecimiento de forma exponencial de aquellos esquemas mejor adaptados, dependiendo de la relación entre la media de los representantes y la media de la población.

- El término $1 - p_c \cdot d(H)/(L-1)$, representa la probabilidad de supervivencia de un esquema respecto de la operación de cruce, es decir, la probabilidad de que el cruce no ocurra dentro de la longitud especificada del esquema. Se comprueba que los esquemas, con una adaptación superior al promedio y con longitud corta, reciben un incremento exponencial del número de cadenas en la próxima generación.
- El término $(1 - p_m)^o(H)$, representa la probabilidad de supervivencia de un esquema con respecto a la operación de mutación. El operador mutación no cambia demasiado las conclusiones anteriores.

Estos dos últimos operadores son necesarios porque la reproducción no introduce nuevos esquemas. Por esto se usa el operador de cruce, que crea nuevos esquemas, intercambiando información de forma aleatoria, y el operador de mutación que introduce variedad dentro de la población.

A continuación se describen los efectos de los esquemas de reproducción y operadores genéticos en el comportamiento de los Algoritmos Genéticos

Esquema de reproducción

Para determinar el efecto de la reproducción en el número esperado de esquemas en una población, se parte de un tiempo t , dado como m ejemplos de esquemas en la población $A(t)$ donde $m=M(H,t)$

Durante la reproducción de una cadena A , se selecciona con una probabilidad $P_i=f_i/\sum f_i$. Con esto se espera tener $M(H,T+1)$ representaciones del esquema H en la población en el tiempo $t+1$ dada por la ecuación:

$$M(H,T+1)(n)(f(H)/ \sum f(H)) \quad (3.5)$$

y

$$F(H) = \sum_{i \in H} \frac{f_i}{M(H,t)} \dots \dots (3.6)$$

Donde:

$f(H)$ es la aptitud media del esquema H dada por: la aptitud media de la población.

La siguiente ecuación permite hacer una última transformación:

$$\bar{f} = \frac{\sum_{j=1}^n f_j}{n} \dots \dots (3.7)$$

$$m(H, t+1) = m(h, t) * \frac{F(H)}{f} \dots\dots (3.8)$$

Operador de Cruza.

Sea P_d la probabilidad de destrucción de un esquema por la cruza de un punto:

$$P_d(H) = \frac{\delta(H)}{L-1} \dots\dots (3.9)$$

por lo tanto:

$$P_s(H) = \frac{\delta(H)}{L-1} \dots\dots (3.10)$$

es la probabilidad de supervivencia de ese patrón

Siendo P_c la tasa de aplicación de la cruza y considerando que el par de un cromosoma recupera parte de un patrón destruido por la cruza, tenemos:

$$P_s \geq 1 - p_c \left(\frac{\delta(H)}{L-1} \right) \dots\dots 3.11$$

Combinando el efecto de la reproducción y cruza se obtiene;

$$m(H, t+1) \geq m(H, t) \left(\frac{F(H)}{f} \right) \left[1 - p_c \left(\frac{\delta(H)}{L-1} \right) \right] \dots\dots 3.12$$

El efecto combinado de la cruza y la reproducción se obtienen multiplicando el número esperado de esquemas para la reproducción por la probabilidad de supervivencia bajo la cruza p_s .

Operador de mutación.

El operador de mutación se aplica a cada una de las posiciones de una cadena independiente con probabilidad de sobrevivencia por la influencia de este operador de $(1-P_m)$. Un esquema en particular sobrevive cuando cada una de las posiciones de $O(H)$ en el esquema sobreviven. La probabilidad de supervivencia para este operador se obtiene multiplicando $(1-P_m)$, $O(H)$ veces de $(1-P_m)^{O(H)}$. Para valores de P_m pequeños ($P_m \ll 1$), la probabilidad de supervivencia de un esquema se aproxima a $1 - O(H)(P_m)$. A partir de esto se llega a la expresión que representa el número esperado de copias que va a recibir la generación siguiente de un esquema H , en particular, bajo la influencia de los tres operadores genéticos (reproducción, cruza y mutación)

$$m(H, t+1) \geq m(H, t) \left(\frac{f(H)}{f} \left[1 - p_c \left(\frac{\delta(H)}{L-1} \right) (1 - p_m(O(H))) \right] \right) \dots 3.13$$

Donde:

- F.- Resultado de la evaluación de la función.
- H.- Esquema.
- M.- Número de esquemas.
- δ .- Definición de longitud.
- O.- Orden.
- PC.- Probabilidad de cruza.
- Pm.- Probabilidad de mutación.
- L.- Longitud de la cadena.

En resumen, la consecuencia que se extrae de la expresión es que los esquemas cortos, de bajo orden, con adaptación por encima de la media incrementarán su número de representantes de forma exponencial en las siguientes generaciones. Esta afirmación constituye el enunciado del Teorema de los esquemas.

Algoritmos genéticos paralelos.

Los algoritmos genéticos son fácilmente programados en arquitecturas paralelas, lo que permite evaluar a cada uno de los individuos independientemente. El atractivo del estudio de los algoritmos genéticos en arquitecturas paralelas es la reducción de tiempo de ejecución con respecto a otros modelos de optimización tradicionales, los cuales están diseñados para ejecutarse en arquitecturas secuenciales (solo un procesador) y su implementación en arquitecturas paralelas resulta difícil. Existen diferentes modelos de algoritmos genéticos paralelos para diferentes tipos de problemas, usualmente se utilizan las siguientes arquitecturas Granja (Farming), difusión y la migración. Nowostawski los clasifica en ocho arquitecturas (Nowostawski et al, 1999):

- Maestro esclavo
 - Sincrónicos.
 - Asincrónicos.
- Poblaciones estáticas con migración.
- Subpoblaciones entrelazadas con migración.
- Algoritmos genéticos paralelos controlados.
- Algoritmos genéticos paralelos desordenados.
- Métodos híbridos.

A continuación vamos a revisar algunos de los principales métodos para la implantación de algoritmos genéticos:

Granja

El prototipo granja es un proceso capataz que coordina a un número “n” de procesos esclavos. El maestro controla los operadores genéticos (la selección, el cruce, y la mutación) y la función objetivo en paralelo, asignando tareas a los esclavos. Como se muestra en la Figura 3.7.

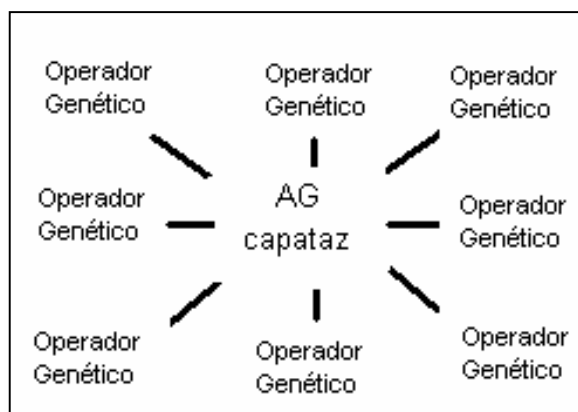


Figura 3. 7 Modelo de capataz-esclavo²

Migración.

El modelo de migración (Figura 3.8) divide a la población en “n” subpoblaciones. Estas subpoblaciones evolucionan independientemente en un número determinado de generaciones. Al término de este tiempo un número de los individuos de las subpoblaciones son distribuidos entre las demás (migración). Tanto el número de individuos intercambiados como el método de selección utilizado determinan en mucho la diversidad que tendrán las subpoblaciones

La implementación del modelo de migración no sólo muestra una mejora en el tiempo (*speedup*), sino también un menor número de evaluaciones de la función objetivo comparado con un algoritmo genético

La selección de los individuos en la migración puede tomar varias formas siendo las principales:

1. Sorteo uniforme, se toman los individuos al azar para la migración

² GA significa Algoritmo Genético

2. Por su aptitud, se toman a los mejores individuos de cada una de las subpoblaciones

Existen muchas arquitecturas en las que se puede implementar la migración de los individuos en las subpoblaciones, por ejemplo: Entre todas las subpoblaciones, (Topología de red- sin restricciones), en una topología de anillo, y una topología de vecindad.

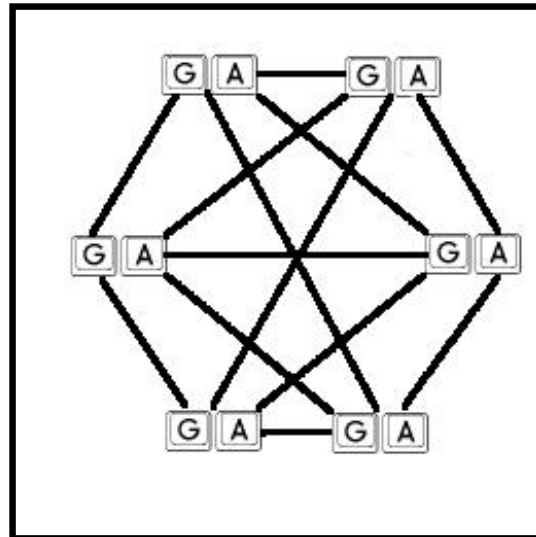


Figura 3.8 Modelo de Migración sin restricciones

La estrategia más utilizada es la migración sin restricciones, en ésta los individuos pueden migrar a cualquiera de las subpoblaciones. Estos procesos reciben a los individuos de la migración y reemplazan a los individuos que componen al proceso.

En este algoritmo de migración se tienen dos parámetros nuevos que son:

- Emigrante, que corresponde al porcentaje de la población de individuos seleccionados para la migración de cada subpoblación.
- Miginterval, que determina el número de generaciones entre cada migración.

A estos parámetros se les da los valores de:

$$\text{Emigrante} = 1/N_{\text{subpob}} \text{ y } \text{miginterval} = 1.$$

Modelo de difusión

En el modelo de difusión cada nodo calcula una o varias funciones, estos a su vez pueden intercambiar información con los nodos más cercanos. Pudiendo éstos tomar los individuos seleccionados de sus vecinos para después ejecutar la cruce y la mutación, produciéndose un traslape en las subpoblaciones. Cada una de estas partes está constituida por más de un elemento. Así, el número de posibles individuos para contribuir en la generación de nuevos individuos dentro de una subpoblación será mayor al número de nuevos miembros que ésta debe generar (Figura 3.9).

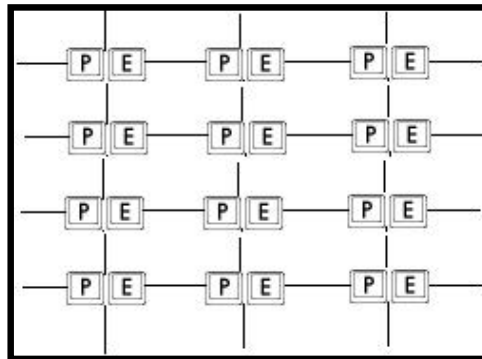


Figura 3.9 Modelo de difusión.

CAPÍTULO 4. GENERACIÓN DE RUTAS LIBRES DE OBSTÁCULOS CON ALGORITMOS GENÉTICOS

En el campo de la robótica las investigaciones realizadas con algoritmos genéticos para la generación de rutas libres de obstáculos, realizan la búsqueda de rutas, tomando 2 ó 3 dimensiones, retomando los métodos expuestos por Latombe (Latombe, 1991), utilizando algoritmos genéticos y comparando los resultados en función de su efectividad, por ejemplo, tiempo de ejecución y distancia recorrida. En la Tabla A-1 se presenta un resumen de las principales investigaciones concernientes a este tema. En dicha Tabla se muestran, entre otros, los desarrollos de I. Ashiru, C. Czarnecki (Ashiru et al, 1995), Nishimura, Sugawara, Yoshihara Abe (Nishimura et al, 2000), Vadakkepat, Chen and Ming-liang (Vadakkepat et al, 2000), lo cuales utilizaron el método Campos Potenciales. También se puede apreciar en esta Tabla que uno de los métodos usados para la resolución de este problema es dividir el espacio de configuración en celdas o dividir el camino en grafos.

A continuación se expone el planteamiento del problema de la planeación de rutas y las consideraciones tomadas para la realización de este trabajo.

Planteamiento del problema.

Este trabajo tiene como objeto generar soluciones al problema de rutas óptimas libres de obstáculos, utilizando algoritmos genéticos (AG), como se dijo antes, los algoritmos genéticos constituyen un campo de investigación importante, ya que son una herramienta muy poderosa de optimización. Una ventaja que ofrecen es que permiten realizar búsquedas robustas en espacios de dimensiones elevadas, en las que no es necesario saber mucho acerca del problema y no se tienen grandes restricciones al plantear la función a optimizar.

En los algoritmos genéticos la función objetivo resulta fundamental para la resolución de los problemas, ya que determina que individuos en la población tienen mejores características, y por consecuencia, mejores oportunidades para sobrevivir y reproducirse. Por lo que es esencial buscar una función objetivo que lleve a la población por un camino óptimo y factible.

Para realizar el planteamiento de la función objetivo del AG, se consideraron los siguientes elementos metodológicos:

Se utilizó el “espacio de configuración” y se visualizó el robot como un punto en el espacio como se explica en el segundo capítulo de esta tesis, proponiendo dos formas de resolver el problema de la planeación de rutas libres de obstáculos:

- Primer método.- Se construyeron rutas completas con el fin ponerlas a competir entre sí.

- Segundo método.- Se generó la ruta a partir de círculos. Se comenzó con un círculo de radio 'r', y se buscó el punto más cercano al punto objetivo dentro de esa circunferencia, después se aumentó el diámetro y nuevamente se buscó el punto más cercano dentro del nuevo círculo, y así sucesivamente, hasta llegar al punto final u objetivo. Cuidando que no se tuviera ninguna colisión o choque con los obstáculos.

En estos dos casos los mejores individuos se definieron como rutas que no pasarán por algún obstáculo (resultados factibles); es decir, aquellas que se encuentran sobre el espacio libre de obstáculos y los peores individuos, como aquellos que pasaran por algún obstáculo o se salieran del espacio de configuración delimitado (resultados no factibles).

Una forma de discriminar entre las rutas factibles y no factibles es aplicar un método muy utilizado ante esta situación, el cual consiste en penalizar con un valor muy grande todas aquellas rutas que cruzan por algún obstáculo (rutas no factibles).

Se utiliza con el fin de hacer esta discriminación la siguiente función (Michalewicz, 1996):

$$\text{Eval}_{NF} = \text{Eval}(p) + Q(P): \dots \dots \dots (4.1)$$

Donde:

- Q(P) penalización para los individuos (p) no factibles.
- Eval_{NF} evaluación de los individuos no factibles
- Eval(p) evaluación del individuo

En este tipo de optimización se requiere dar valores adecuados a las soluciones para evitar que rutas que cruzan por algún obstáculo tengan valores semejantes a rutas libres de éstos. Tomando este criterio se penalizó con un valor muy alto, que discriminara entre las rutas libres de obstáculos y aquellas que no lo eran, teniendo en cuenta que al realizar este tipo de penalización se puede perder material genético valioso.

Para el desarrollo del programa que permitió el análisis, diseño e implementación del AG, así como para la resolución del problema de la planeación de rutas libres de obstáculos, se utilizó el software de aplicación Matlab 7.2

Los efectos de la diversidad de la población y la presión de selección sobre los AG's.

Los parámetros principales que determinan el comportamiento y desempeño de los AG's son: la diversidad de la población y la presión de selección. Estos factores se encuentran fuertemente relacionados: el incremento en la presión de

selección y la disminución la diversidad en la población, y viceversa (Michalewicz, 1996). La forma óptima de establecer estos parámetros ha sido ampliamente estudiada desde el inicio de los algoritmos genéticos, y aún hoy, no existe una solución a este problema. El Dr. Coello en sus apuntes de algoritmos Genéticos, hace una revisión de las investigaciones que existen al respecto, expone diversas investigaciones; una de ellas son las de Goldberg que intenta determinar el tamaño de población, otra es la de Jong, Grefenstette donde varían las probabilidades de mutación, cruza y el tamaño de la población, así como las de Schaffer, Srivas y Patnaik, Davis que realizaron investigaciones acerca de AG's, en los que se auto-adaptan los parámetros con el tiempo y el comportamiento de la población, Holland, Bäck con mutaciones variables (Coello, 2004).

En este trabajo se hace un análisis para encontrar la combinación entre las diferentes variables que conforman los AG's, con el fin de obtener el mejor desempeño para la resolución del problema de planeación de rutas. Se utilizaron tres métodos conocidos; el método de selección Ruleta (*Roulette Wheel*) el método de selección estocástico universal y el método de selección Ranking, descritas en el capítulo anterior.

Para medir el grado de eficiencia de un algoritmo genético es necesario realizar varios experimentos bajo las mismas condiciones para observar su comportamiento al paso de varias generaciones. Al correr el algoritmo cambiamos la probabilidad de mutación y la probabilidad de cruza. En la Tabla 4.1 se muestran las probabilidades de mutación, cruza y tamaño de poblaciones utilizadas, con estos valores se realizan todas las combinaciones posibles para observar el comportamiento del algoritmo genético. Para cada combinación de estos valores se corrió el programa 20 veces y se observó el comportamiento del programa, dejando 1,000 generaciones por corrida. En los algoritmos genéticos, como en cualquier técnica de optimización, es importante llegar a una buena respuesta en un número pequeño de iteraciones.

Para observar el comportamiento del algoritmo genético, se calcularon las medias de la evaluación de todos los individuos, así como las medias de los mejores individuos en cada ciclo, durante varios experimentos, para describir el comportamiento del algoritmo.

Tamaño de población	% de mutación	%de cruza
50	0.001	0.85
100	0.01	0.9
150	0.1	0.95
200	0.2	1

Tabla 4.1 Probabilidades de mutación, cruza y tamaño de población.

Los algoritmos genéticos, como ya se mencionó antes, son métodos de búsqueda y optimización de problemas, y su campo de aplicación es muy amplio, en este experimento se evaluó el AG con la función Rosenbrock

$$f(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \quad (4.2)$$

Donde:

- $\{X_i / x \in \mid, -5.12 < x > 5.12\}$

En la función de Rosenbrock se tiene un solo punto óptimo de dimensiones “n”, es decir en $x=(1,1,\dots,1)$ Y resulta muy difícil de encontrar, Figura (4.1) y ecuación 4.2

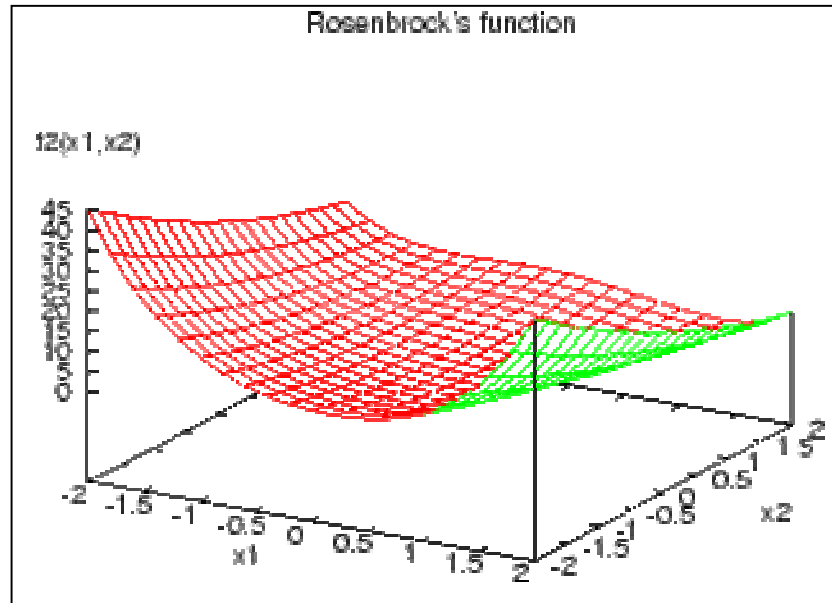


Figura 4.1.- Función de Rosenbrock¹

Resultados

Las poblaciones con el método de selección de ruleta y estocástico universal tienen un comportamiento pobre, al no converger hacia el óptimo y presentan un comportamiento poco estable, como se observa la Figura 4.2. La asignación de la probabilidad de selección de acuerdo con el desempeño (*fitness*) de cada individuo, provoca una pérdida de la diversidad, al encontrar individuos con una fortaleza muy grande y estancándose en óptimos locales. Una solución a este problema consistiría en aumentar la probabilidad de mutación. Este operador permite la introducción de nuevo material cromosómico en la población, tal y como sucede con sus equivalentes biológicos; sin embargo, una probabilidad de

¹ German Aerospace Center, Enero 2004

mutación muy alta provoca que el algoritmo genético se convierta en una búsqueda aleatoria.

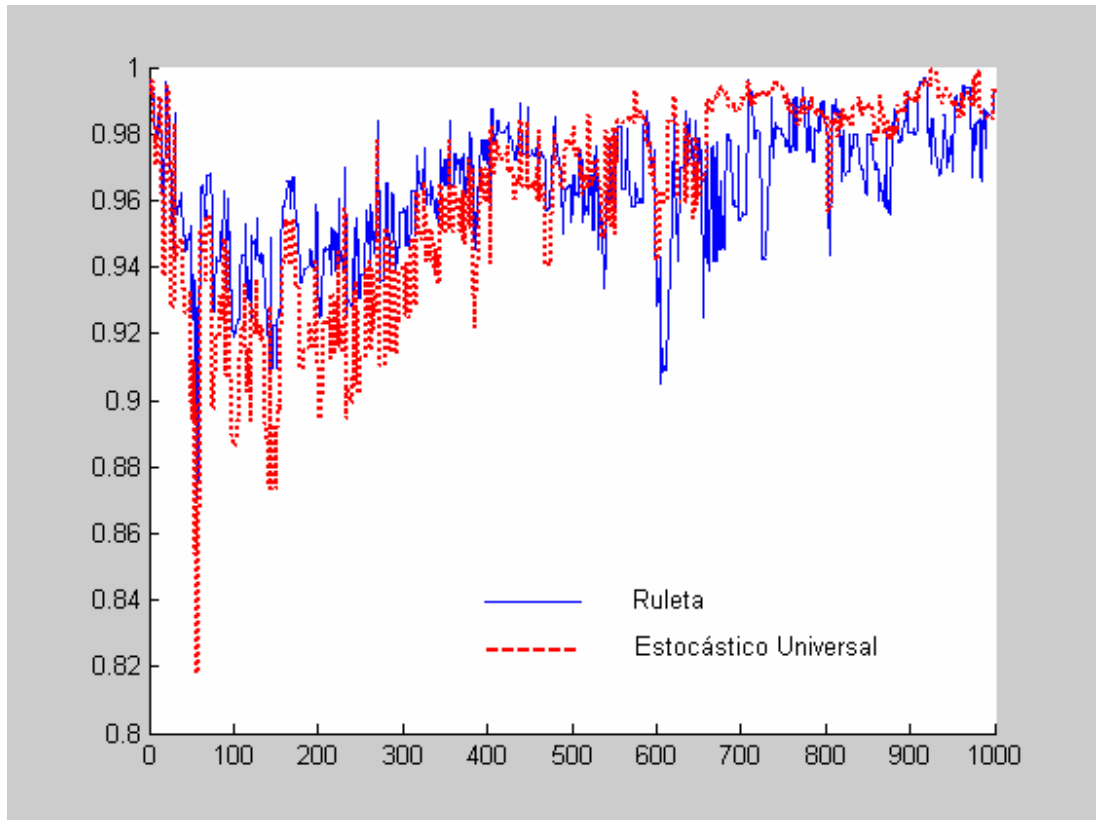


Figura 4.2 Comportamiento del algoritmo genético al utilizar los métodos de Ruleta y estocástico universal

Para disminuir el problema de la asignación de probabilidades de selección con base en la fortaleza, se utilizó el Ranking. Este método clasifica a la población según una posición jerarquizada, dependiendo de la aptitud del individuo, asignando un valor a los individuos de mayor a menor grado y aumentando la competencia entre mismos, conservando una mayor diversidad entre la población. El aumentar la competencia entre los individuos de la población permite que éstos, que contienen información valiosa, se conserven a través de las generaciones y mejoren el desempeño. Este método converge hacia el resultado óptimo de una manera más ordenada (Figura 4.3). Para este experimento se utilizó el ranking para asignar las probabilidades de selección y después se utilizaron los métodos de la ruleta y estocástico universal para seleccionar a los individuos

En la Figura 4.3 se muestran los resultados obtenidos del algoritmo genético de los métodos de selección estocástico universal y ruleta, en combinación con ranking, la combinación que obtuvo mejor comportamiento, convergiendo con el valor óptimo en un número menor de generaciones fue Ranking-estocástico universal en comparación con los obtenidos con Ranking - ruleta

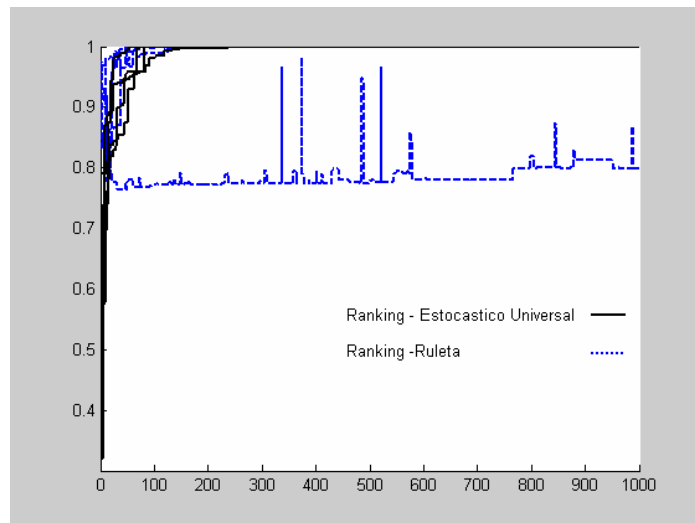


Figura 4.3.- Comparativa entre Ranking con estocástico Universal y Ranking con Ruleta

Las probabilidades de cruce de 0.85 a 0.95, en combinación con las probabilidades de mutación de 0.1 y 0.2, muestran una excelente convergencia hacia el valor óptimo como se observa en la Figura 4.4a. y se obtienen pobres resultados cuando la probabilidad de mutación es baja (0.001 y 0.01) Figura 4.4b.

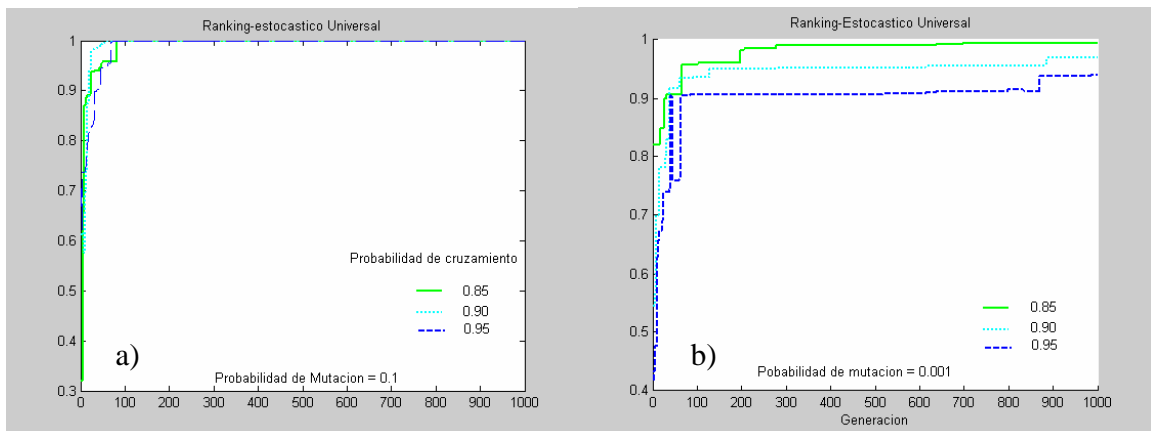


Figura 4.4 a) Ranking - estocástico Universal con probabilidad de mutación de 0.1, b) Ranking - Estocástico Universal, fijando la probabilidad de mutación en 0.001

El número de generaciones que son utilizadas para llegar al óptimo depende de muchos factores, uno de los factores importantes es el tamaño de la población, el tener poblaciones grandes ayuda a mejorar la convergencia hacia el óptimo en un número menor de generaciones. En la Figura 4.5 se ve el resultado del algoritmo genético con una población de 200 individuos, mejorando el desempeño con respecto a los algoritmos con poblaciones más pequeñas.

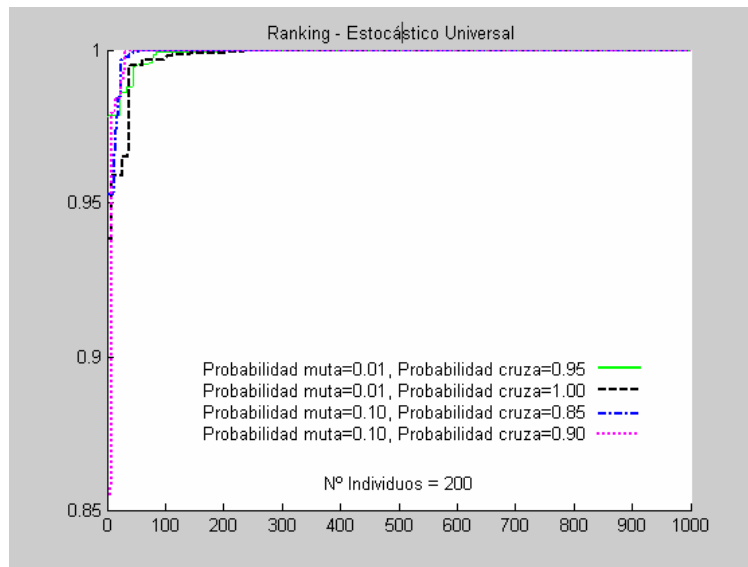


Figura 4.5. Ranking – Estocástico Universal con poblaciones de 200 individuos, las poblaciones grandes mejoran la eficiencia del algoritmo genético.

Como se ha observado en este estudio, podemos concluir que el uso del método Ranking en combinación con el estocástico universal produce una convergencia más rápida hacia el óptimo. Y la combinación que obtuvo un mejor desempeño para la optimización de la función, y la cual será utilizada para la planeación de rutas libres de obstáculos, fue la que tiene un tamaño de población = 200 probabilidades de mutación = 0.10 y probabilidades de cruce = 0.90, como se observa en la Figura 4.5.

Se ha demostrado con investigaciones anteriores (Coello, 2004), (Solano et al 1993) que la modificación de los parámetros que componen el algoritmo genético modifica el comportamiento, haciéndolo más o menos eficiente. Es por esto que esta área ha sido estudiada con anterioridad por su importancia en la generación de buenos resultados, en un número pequeño de generaciones, por lo que es importante seguir estudiando el comportamiento y los efectos que genera en las poblaciones.

En las siguientes Tablas se observan algunos de los resultados obtenidos en este trabajo, cada columna representa una corrida con las probabilidades de mutación y de cruce que se muestran en la parte inferior de la Tabla, donde la utilización del Ranking mejora sustancialmente la optimización de la función.

Generación	Resultados de la combinación Ranking-estocástico											
980	2178.4	2633.5	1551.0	5171.8	2821.9	1699.8	2963.1	5489.5	11313	1671.4	1284.5	2864.3
981	2178.4	2633.5	1551.0	5171.8	2821.9	1699.8	2963.1	5490.0	11313	1671.4	1284.5	2864.3
982	2178.4	2633.5	1551.0	5171.8	2821.9	1699.8	2963.1	5490.0	11313	1671.5	1284.5	2864.3
983	2178.4	2247.9	1551.0	4915.1	2821.9	1698.3	2963.1	5490.0	11313	1671.5	1284.5	2864.3
984	2178.4	2247.9	1551.0	4915.1	2822.2	1698.3	2963.1	5490.0	11313	1671.5	1284.5	2864.3
985	2178.4	2247.9	1551.0	4915.1	2822.2	1698.3	2963.1	5489.8	11313	1671.5	1284.5	2864.3
986	2178.4	2259.6	1544.7	4874.3	2822.2	1698.3	2963.1	5489.8	11313	1671.4	1284.5	2864.3
987	2178.4	2259.6	1544.7	4874.3	2822.2	1698.3	2963.1	5489.8	11313	1671.4	1284.5	2864.3
988	2178.4	2259.6	1544.7	4874.3	2822.2	1698.3	2963.1	5489.8	11313	1671.4	1284.5	2864.3
989	2178.4	2305.3	1544.7	4874.3	2822.2	1698.3	2963.1	5489.8	11313	1671.4	1284.5	2864.3
990	2178.4	2305.3	1544.7	4874.3	2821.9	1698.3	2963.1	5489.8	11313	1671.5	1284.5	2864.3
991	2178.4	2305.3	1544.7	4874.3	2821.9	1698.3	2963.1	5489.8	11313	1671.5	1284.5	2864.3
992	2178.4	2305.3	1544.7	4860.7	2821.9	1698.3	2963.1	5489.8	11314	1671.5	1284.1	2864.3
993	2178.4	2305.3	1544.7	4860.7	2821.9	1698.3	2963.1	5489.8	11314	1671.5	1284.5	2864.3
994	2178.4	2305.3	1544.7	4860.7	2821.9	1698.3	2963.1	5489.8	11315	1671.5	1284.5	2864.3
995	2178.4	2305.3	1544.7	4789.6	2821.9	1698.5	2963.1	5489.8	11314	1671.5	1284.1	2864.3
996	2178.4	2305.3	1544.7	4789.6	2822.2	1698.5	2963.1	5489.8	11314	1671.5	1284.1	2864.3
997	2178.4	2305.3	1542.7	4789.6	2822.2	1698.0	2963.1	5489.8	11314	1671.4	1284.1	2864.3
998	2178.4	2305.3	1542.7	4789.6	2822.2	1698.3	2963.1	5489.8	11314	1671.4	1284.1	2864.3
999	2178.4	2305.3	1542.7	4789.6	2822.2	1698.3	2963.1	5489.8	11314	1671.4	1284.1	2864.3
1,000	2178.4	2238.3	1542.7	4830.4	2822.2	1698.3	2963.1	5489.8	11314	1671.4	1284.1	2864.3
Prob Muta	0.001	0.001	0.001	0.01	0.01	0.01	0.01	0.1	0.1	0.1	0.1	0.2
Prob Cruza	0.90	0.95	1.00	0.85	0.90	0.95	1.00	0.85	0.90	0.95	1.00	0.85

Tabla 4.2 Resultados de las últimas 20 generaciones, utilizando el ranking -estocástico universal con las probabilidades de mutación y cruza

Generación	Resultados de la combinación Ranking ruleta											
980	2,75E+17	1,08E+15	5,15E+16	3,89E+16	1617.6	3,10E+09	5589.2	10057	2821.5	1277.8	22214	3014.8
981	2,74E+17	6,08E+16	5,15E+16	3,89E+16	1617.6	3,10E+09	5589.2	9901	2821.5	1277.8	22195	3014.8
982	2,74E+17	6,08E+16	5,15E+16	3,89E+16	1617.6	3,09E+09	5589.2	10051	2815.2	1281.2	22195	3014.8
983	2,74E+17	6,08E+16	5,15E+16	3,89E+16	1617.6	3,08E+09	5259.9	10051	2815	1254.4	22195	3014.8
984	2,74E+17	5,32E+16	5,15E+16	3,89E+16	1611.9	3,09E+09	5265.1	10057	2617.5	1254.4	22195	3014.8
985	2,74E+17	6,08E+16	5,15E+15	3,89E+16	1617.8	3,09E+09	5565.8	10057	2815	1277.5	22185	3015.5
986	2,74E+17	6,08E+16	5,15E+15	3,89E+16	1617.8	3,09E+09	5642.4	10057	2815	1255.6	22185	3068.9
987	2,74E+17	1,08E+16	5,13E+16	3,89E+16	1617.8	3,09E+09	5642.4	10057	2653.6	1277.5	22200	3068.9
988	2,74E+17	8,27E+15	5,15E+15	3,89E+16	1607.8	3,09E+09	5518.3	10057	2629.5	1261	21583	3068.9
989	2,75E+17	1,06E+16	5,15E+15	3,89E+16	1591.8	3,09E+09	5312.7	10057	2611.8	1219.6	21583	3067.8
990	2,75E+17	1,06E+16	5,15E+15	3,89E+16	1599.1	3,09E+08	5312.7	10057	2618.4	1219.6	21583	3067.8
991	1,93E+17	1,06E+16	5,15E+15	3,89E+16	1617.8	3,09E+08	5380.9	10044	2618.4	1219.6	22070	3067.8
992	1,93E+17	6,50E+14	5,15E+15	3,89E+16	1612.2	3,09E+09	5380.9	10050	2619.7	1223.1	22069	3,06E+03
993	1,93E+17	6,50E+14	5,15E+15	3,89E+16	1617.8	3,07E+09	5380.9	10057	2617.8	1074.8	22070	3033.6
994	1,93E+17	1,08E+16	4,97E+16	3,89E+16	1615.3	3,07E+09	5380.9	9998.9	2660.9	1223.1	21565	2956.6
995	1,93E+17	4.25e+012	4,97E+16	3,89E+16	1615.3	3,07E+09	5380.9	10028	2661.2	1223.1	21515	2791.1
996	1,93E+17	4.25e+012	2,22E+15	3,89E+16	1608.9	3,07E+09	5265.1	10028	2661.3	1218.6	21547	2791.1
997	1,93E+17	4.25e+012	2,22E+15	3,89E+16	1590.1	3,10E+09	5265.1	10028	2619.7	1215.9	21547	2791.1
998	1,93E+17	4.25e+012	4,80E+16	3,89E+16	1590.1	3,10E+09	5265.1	10028	2654.7	1218.5	21547	2,84E+03
999	1,93E+17	3,99E+16	4,80E+16	3,88E+16	1599.1	3,10E+09	5265.1	10024	2654.7	1218.7	21551	2,84E+03
1,000	1,93E+17	4.25e+012	4,80E+16	3,88E+16	1599.1	3,09E+09	5265.9	10024	2654.7	1218.7	21551	2,84E+03
Prob Muta	0.001	0.001	0.001	0.01	0.01	0.01	0.01	0.1	0.1	0.1	0.1	0.2
Prob Cruza	0.9	0.95	1	0.85	0.9	0.95	1	0.85	0.9	0.95	1	0.85

Tabla 4.3 Resultados de las últimas 20 generaciones, utilizando Ranking - ruleta con las probabilidades de mutación y cruza

Generación	Resultados Ruleta											
980	7,00E+19	4,29E+19	4.72e+015	1,09E+20	1,03E+19	1,31E+18	7,52E+18	2,38E+18	6,69E+18	2,93E+18	6,31E+17	1,21E+18
981	7,00E+19	4,29E+19	4.72e+015	1,09E+20	1,07E+19	1,34E+18	5,98E+18	2,36E+18	6,34E+18	2,93E+18	6,39E+18	1,22E+18
982	7,00E+19	4.29e+015	3,73E+18	1,09E+20	1,12E+19	1,22E+18	7,52E+18	2,35E+18	6,33E+18	2,83E+18	7,03E+18	1,22E+18
983	6,99E+19	4,29E+19	3,73E+18	1,09E+20	1,16E+19	1,22E+18	5,97E+18	2,38E+18	6,69E+17	2,84E+18	7,99E+18	1,24E+17
984	7,00E+18	4,29E+19	3,73E+18	1,09E+20	1,16E+19	1,11E+18	2,27E+18	2,38E+18	7,99E+18	2,90E+18	7.19e+014	1,24E+18
985	7,00E+18	4,28E+19	3,73E+19	1,09E+20	1,16E+19	2,45E+18	5,88E+18	1,19E+18	6,89E+18	2,90E+18	4,79E+18	1,22E+18
986	6,98E+19	3,88E+19	3,73E+19	1,09E+20	2,06E+19	2,42E+18	5,88E+18	1,19E+18	7,62E+18	2,84E+18	3,17E+18	1,24E+18
987	7,19E+19	4,28E+19	4,04E+18	1,09E+20	1,39E+19	2,45E+18	6,75E+18	1,19E+19	7,88E+18	2,86E+18	9,62E+18	8,33E+17
988	7,18E+19	4,28E+19	1,44E+18	1,09E+20	7,21E+17	2,45E+18	6,54E+18	1,59E+19	1,32E+19	2,75E+18	1,05E+19	9,51E+16
989	7,18E+19	4,28E+19	1,44E+18	1,09E+20	1,34E+19	2,45E+18	6,60E+17	1,10E+19	1,26E+19	2,76E+18	5,43E+18	9,49E+17
990	7,18E+19	4,25E+18	4,23E+18	9,47E+19	4,23E+18	2,45E+18	6,75E+18	1,10E+19	1,55E+18	2,86E+18	5,43E+18	9,47E+17
991	6,63E+19	4,25E+18	4,23E+18	9,47E+19	3,96E+18	1,13E+18	4,94E+17	1,10E+19	1,51E+19	1,02E+18	5,43E+18	9,47E+17
992	6,83E+18	4,25E+19	1,41E+19	9,46E+19	1,59E+19	1,13E+18	2,34E+18	1,10E+19	1,44E+19	9,11E+17	6,43E+18	9,49E+17
993	7,00E+19	4,25E+19	1,41E+19	9,67E+19	1,56E+19	1,09E+18	2,34E+18	1,58E+19	1,06E+19	9,12E+17	1,05E+19	9,53E+17
994	7,00E+19	4,25E+19	1,35E+19	9,67E+19	2,22E+19	1,07E+18	2,35E+18	1,58E+19	1,03E+19	9,12E+17	1,05E+18	9,53E+17
995	7,22E+19	4,29E+19	1,41E+19	9,67E+19	2,22E+19	1,07E+18	1,71E+18	3,47E+19	1,01E+19	9,12E+17	1,10E+19	3,55E+18
996	7,22E+18	4,29E+19	1,41E+19	1,07E+20	1,46E+19	2,54E+18	1,66E+18	3,43E+18	1,03E+19	2,52E+18	7,09E+18	3,55E+18
997	7,21E+19	4,29E+19	1,41E+19	1,08E+20	1,77E+19	3,64E+18	1,66E+18	3,20E+19	1,03E+19	2,52E+18	6,45E+18	3,55E+18
998	7,21E+19	4,29E+19	1,41E+19	1,08E+20	2,37E+19	3,64E+18	1,66E+18	2,65E+19	1,04E+19	2,52E+18	6,51E+18	3,44E+18
999	6,59E+19	4,17E+19	1,41E+19	1,08E+20	2,75E+19	2,34E+18	1,59E+18	3,04E+19	1,04E+19	2,52E+18	5,25E+18	5,70E+18
1,000	7,18E+19	4,29E+18	8,43E+17	1,08E+20	1,49E+19	2,34E+18	1,65E+18	3,08E+19	1,25E+18	7,33E+16	4,56E+18	7,44E+18
Prob Muta	0.001	0.001	0.001	0.01	0.01	0.01	0.01	0.1	0.1	0.1	0.1	0.2
Prob Cruza	0.9	0.95	1	0.85	0.9	0.95	1	0.85	0.9	0.95	1	0.85

Tabla 4.4 Resultados de las últimas 20 generaciones, utilizando la ruleta con las probabilidades de mutación y cruza

Generación	Resultados estocástico universal											
980	6,99E+19	4,22E+19	1,35E+19	9,18E+18	8,96E+18	7,86E+17	7,36E+18	3,05E+17	5,14E+18	4,29E+16	6,24E+18	1,05E+17
981	6,98E+19	4,22E+19	1,35E+19	9,18E+18	1,01E+19	8,12e+013	5,83E+18	2,88E+17	5,14E+18	4,29E+16	6,32E+18	1,05E+18
982	6,99E+19	4,22E+18	3,61E+18	9,18E+19	1,00E+19	8,12E+17	7,38E+18	2,82E+17	5,14E+18	3,35E+17	6,95E+17	1,05E+17
983	6,98E+19	4,22E+18	3,57e+014	9,18E+19	1,03E+19	8,12E+17	5,83E+18	3,07E+17	5,14E+18	3,49E+17	7,91E+18	1,07E+18
984	6,98E+19	4,22E+18	3,61E+18	9,18E+19	1,03E+19	8,12E+17	2,28E+18	3,07E+17	6,44E+17	4,06E+17	7,12E+18	1,07E+18
985	6,98E+19	4,22E+19	3,61E+18	9,18E+19	1,03E+19	2,15E+18	5,90E+18	3,26E+16	6,46E+17	3,88E+17	4,72E+18	1,05E+18
986	6,98E+19	3,82E+19	3,61E+18	9,18E+19	1,03E+19	2,11E+18	5,89E+18	3,26E+17	6,19E+18	3,36E+17	3,10E+18	1,08E+18
987	7,17E+19	4,21E+19	3,61E+18	9,18E+19	9,72E+18	2,15E+18	6,77E+18	3,06E+17	6,46E+18	3,37E+17	9,55E+18	6,73E+17
988	7,16E+19	4,21E+19	1,01E+18	9,18E+19	3,07E+18	2,15E+18	6,55E+18	4,30E+18	1,17E+19	3,49E+17	1,04E+19	7,91E+17
989	7,16E+19	4,21E+19	1,01E+18	9,18E+19	3,15e+014	2,15E+18	6,61E+18	4,30E+18	1,17E+19	3,56E+17	5,36E+18	7,89E+16
990	7,16E+19	4,18E+19	3,79E+18	9,18E+19	3,68E+18	2,15E+18	6,61E+18	4,30E+18	1,46E+19	3,56E+17	5,36E+18	7,89E+17
991	6,62E+19	4,18E+19	3,79E+18	9,18E+19	3,41E+18	8,30E+17	4,82E+18	4,30E+18	1,41E+19	3,55E+17	5,36E+18	7,89E+17
992	6,81E+19	4,18E+19	1,37E+19	9,17E+19	3,41E+18	8,30E+17	2,22E+17	4,30E+18	1,35E+19	2,41E+17	6,40E+18	7,89E+17
993	6,98E+18	4,18E+19	1,37E+19	7,98E+19	3,08E+18	7,93E+17	2,22E+17	4,20E+18	9,70E+18	2,42E+17	1,04E+19	7,93E+17
994	6,98E+18	4,18E+19	1,31E+19	7,98E+19	9,73E+18	7,75E+17	2,23E+18	4,20E+18	9,36E+18	2,42E+17	1,04E+19	7,94E+17
995	7,20E+19	4,22E+19	1,37E+19	7,98E+19	9,73E+18	7,75E+17	1,57E+18	2,31E+19	9,35E+18	2,42E+17	1,09E+19	7,97E+17
996	7,20E+19	4,22E+19	1,37E+19	9,02E+19	2,13E+18	2,24E+18	1,51E+18	2,27E+19	9,38E+18	1,19E+16	7,00E+18	7,97E+17
997	7,20E+19	4,22E+19	1,37E+19	9,09E+19	2,13E+18	3,46E+18	1,51E+18	2,27E+19	9,40E+18	1,18E+17	6,37E+18	7,97E+17
998	7,20E+19	4,22E+19	1,37E+19	9,09E+19	8,09E+18	3,46E+18	1,51E+18	2,43e+015	9,44E+18	1,18E+17	6,43E+18	7,02E+17
999	6,58E+19	4,10E+19	1,37E+19	9,09E+19	8,09E+18	2,15E+18	1,45E+18	2,43e+015	9,44E+18	1,18E+17	5,17E+18	7,97E+17
1,000	7,16E+19	4,22E+19	7,99E+18	9,09E+19	8,09E+18	2,15E+18	1,50E+18	2,46E+19	1,14E+19	1,18E+17	4,47E+18	6,43E+17
Prob Muta	0.001	0.001	0.001	0.01	0.01	0.01	0.01	0.1	0.1	0.1	0.1	0.2
Prob Cruza	0.9	0.95	1	0.85	0.9	0.95	1	0.85	0.9	0.95	1	0.85

Tabla 4.5 Resultados de las últimas 20 generaciones, utilizando el estocástico universal con las probabilidades de mutación y cruza

CAPÍTULO 5. RESULTADOS Y ANÁLISIS DEL PROBLEMA

Al realizar el algoritmo para generar rutas libres de obstáculos, se pensó en obtener rutas completas y muy flexibles que pudieran realizar la búsqueda en el espacio de configuración, partiendo de la idea de que una ruta es la sucesión de segmentos de recta o pasos intermedios. A partir de esta idea se construyó el algoritmo siguiente:

Primer método

La solución que se propone es encontrar el número de rectas o segmentos de recta que, sin pasar por los obstáculos, generen la ruta más corta. Para realizar un planteamiento formal se requiere de las siguientes definiciones:

Segmento de recta.

Definición 1: El segmento de recta t de extremos A y B es el conjunto de puntos

$$\{t(B - A) + A \mid 0 \leq t \leq 1\} \quad (5.1)$$

Los puntos A y B son los extremos del segmento y corresponden al intervalo $[0,1]$ donde $t=0$ y $t=1$, respectivamente.

t es un número real positivo.

Para formar los segmentos de recta que conforman la ruta, necesitamos dos puntos, estos puntos corresponden a los extremos del segmento de recta, para lo cual, el espacio de configuración quedó dividido en $N \times M$, donde la N corresponde al eje de las X , la M corresponde al eje de las Y , donde cada coordenada (X, Y) corresponde a un posible punto o extremo de recta.

Ahora se puede definir la trayectoria de N segmentos:

$$\{t_1(B - A) + A\} \cup \{t_2(C - B) + B\} \cup \dots \cup \{t_n(D - C) + C\} \quad (5.2)$$

Es decir,

$$\begin{aligned} & \{t_1(B_1 - (x_i + y_i)) + (x_i + y_i)\} \\ & \{t_2(B_2 - B_1) + B_1\} \\ & \{t_3(B_3 - B_2) + B_2\} \\ & \cdot \\ & \cdot \\ & \cdot \\ & \{t_n((x_{f_i} + y_f) - B_{n-1}) + B_{n-1}\} \end{aligned}$$

La función objetivo se puede formular como:

$$F(x) = \sum_{i=0}^n \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \dots\dots 5.3$$

* A esta función objetivo se le agrego la función de penalización que esta descrita en el capitulo anterior.

Como se ve en la Figura 5.1 se generan "n" puntos intermedios que conforman la ruta completa, midiendo la distancia del punto inicial al siguiente punto generado al azar, así sucesivamente hasta el punto objetivo.

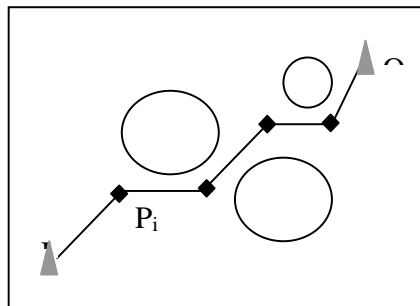


Figura 5.1 Representación de la generación de rutas con 'n' puntos intermedios.

A partir de estas fórmulas se trabajó en un espacio de configuración con un obstáculo en forma de T debido a las dificultades que presenta este tipo de obstáculo, e implementando esto en el algoritmo genético, se generó una ruta como se muestra en la Figura 5.2. Esta ruta se conformó por tres segmentos de recta, con lo cual se obtuvo una distancia de 179.2715, lográndose una buena aproximación, pero no la óptima.

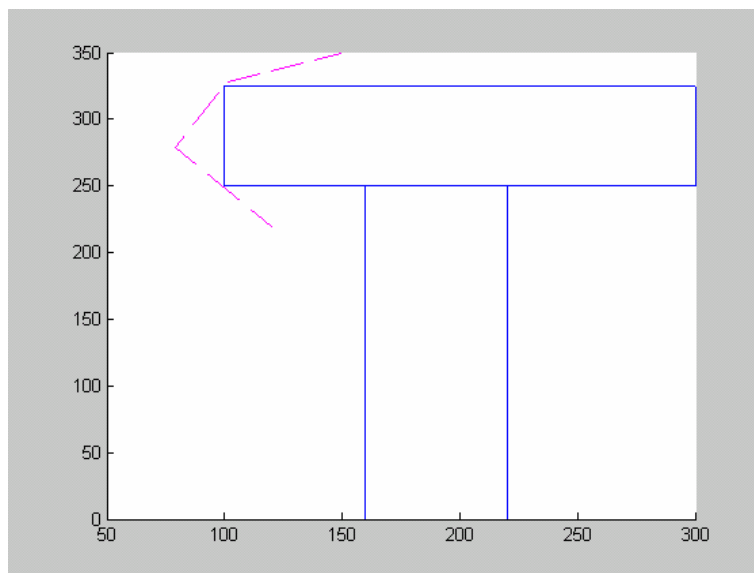


Figura 5.2 Resultado del primer método con 3 segmentos de recta.

En el ejemplo de la Figura 5.3, se conformó una ruta con seis segmentos y se observó que la distancia entre el punto inicial y final es mayor que la anterior. En este ejemplo la distancia fue de 256.5552, al aumentar el número de rectas que conforman la ruta completa, la búsqueda se vuelve más compleja y se obtienen varias soluciones posibles para un mismo problema, sin llegar a un óptimo.

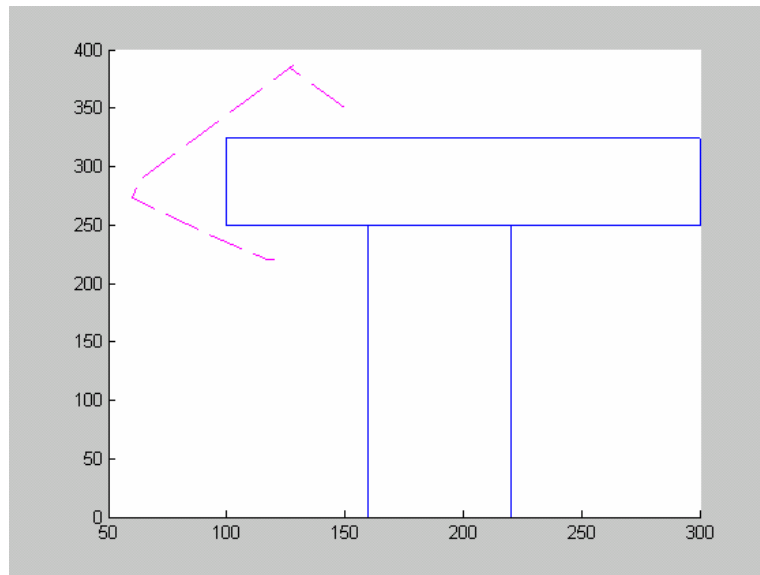


Figura 5.3 Utilización de la primer método con 6 segmentos de recta

Al utilizar un algoritmo genético sencillo e implementarle la función objetivo arriba expuesta, se observó que al hacer la cruce de los individuos se podría perder material genético, y se decidió, para obtener mejores rutas con un mayor número de segmentos de recta, modificar la forma de cruzar para que sólo se cortaran segmentos de recta completos, Figura 5.4.

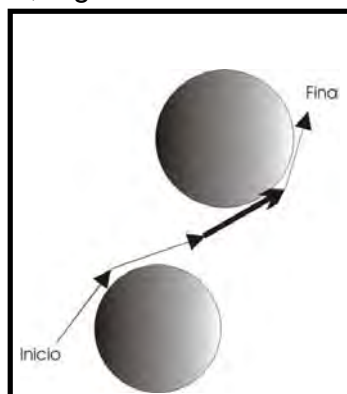


Figura 5.4. Cada segmento está representado por un vector, y cada segmento, a su vez, puede ser cortado y pegado para formar una nueva ruta.

El resultado fue similar al observado anteriormente, concluyéndose que la forma en que se planteó el problema resulta de un alto grado de complejidad, y que se pueden encontrar un número muy alto de soluciones para un mismo problema, lo que hace muy difícil la búsqueda, Figura 5.5 .

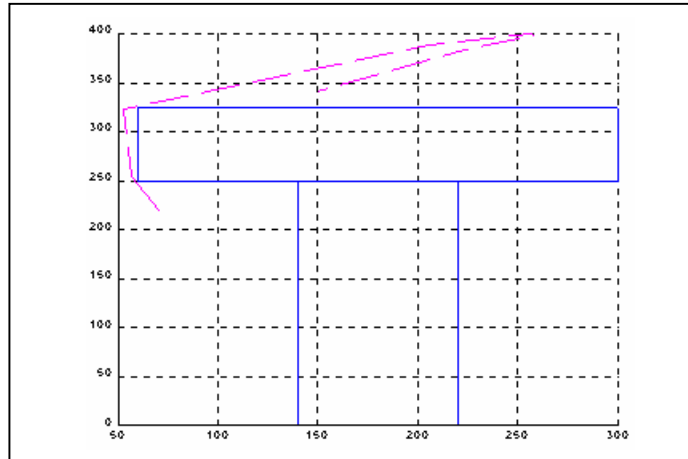


Figura 5.5. Algoritmo modificando la cruz con seis segmentos de recta.

Segundo método.

Otra alternativa que se desarrolló para la planeación de trayectorias libres de obstáculos con algoritmos genéticos fue la siguiente;

Este método se origina en una posición inicial $I(x, y)$, el cual se considera como centro de un círculo de radio “ r ”. La búsqueda se hace a lo largo de la circunferencia, buscando la coordenada más cercana al punto de llegada $f(x,y)$. Al terminar la búsqueda de este punto el radio se incrementa en r , para hacer la búsqueda del punto más cercano en la siguiente circunferencia, esta operación se repite hasta llegar el punto final $f(x,y)$.

El primer método queda como sigue:

$$d = \sqrt{(x_g - (r \cos \theta + x_i))^2 + (y_g - (r \sin \theta + y_i))^2} \dots\dots(5.4)$$

- x_g, y_g Posición final
- x_i, y_i Posición inicial
- R Es igual al radio de la circunferencia o la distancia de la posición inicial al círculo bajo consideración
- θ Ángulo definido por las X-axis y la posición bajo consideración

* A esta función objetivo se le agregó la función de penalización que esta descrita en el capítulo anterior, la cual consiste en sumar un valor muy grande a todas aquellas rutas que cruzan por algún obstáculo (rutas no factibles).

Utilizando para esto la siguiente función (Michalewicz, 1996):

$$\text{Eval}_{NF} = \text{Eval}(p) + Q(P) : \dots \dots \dots (4.1)$$

Donde:

- Q(P) penalización para los individuos (p) no factibles.
- EvalNF evaluación de los individuos no factibles
- Eval(p) evaluación del individuo

En la Figura 5.6 se muestra cómo se construye la ruta libre de obstáculos, generando circunferencias y en cada circunferencia se busca el punto más cercano al punto objetivo

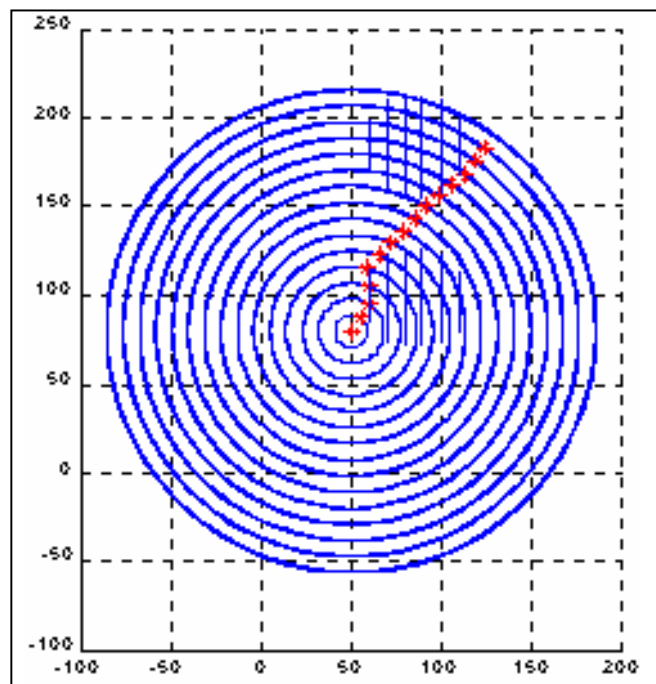


Figura 5.6 Construcción del algoritmo, utilizando el segundo método, con la ayuda de círculos.

Al generar rutas con el segundo método se obtienen rutas más cortas que con el método anterior. En la Figura 5.7, el robot se encuentra con dos obstáculos en forma de círculos. Los círculos son representados como líneas horizontales de penalización que corresponden al espacio discreto en el que se puede mover el robot obteniendo una ruta cercana al óptimo.

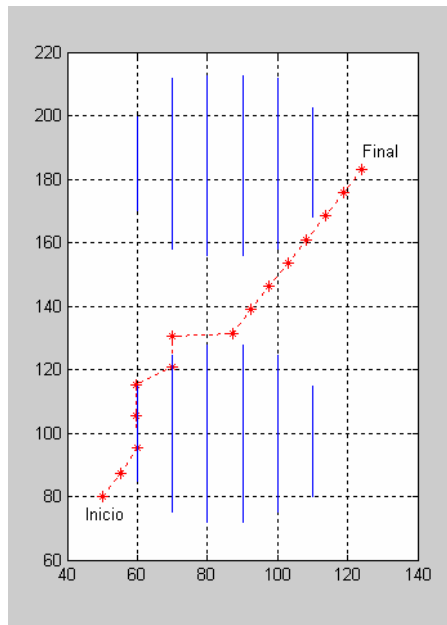


Figura 5.7 Resultado utilizando el segundo método.

El algoritmo se corrió 20 veces, midiendo la distancia recorrida en cada uno de los intentos, para corroborar la mejora en las rutas generadas se obtuvo la distancia promedio de 1,372.944

El algoritmo genera muy buenas rutas al encontrarse obstáculos difíciles de evadir, como lo es el obstáculo en forma de T, que resulta difícil de librar, ya que el robot al toparse con el obstáculo tiene que regresar para buscar otro camino, como se muestra en la Figura 5.8

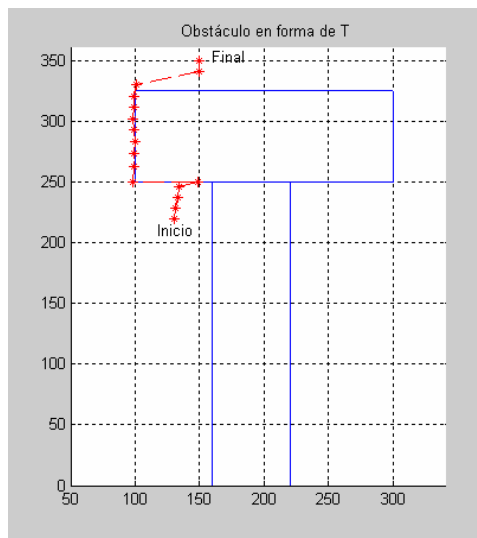


Figura 5.8, Segundo método con un obstáculo en forma de T.

Con este método se obtuvieron mejores resultados comparados con el método anterior obteniendo rutas más cortas, garantizando que en este punto de llegada en particular lograra dar la vuelta y construir una ruta libre de obstáculos cercana al óptimo, otra de las ventajas que presenta este algoritmo es el número de operaciones que realiza con respecto al anterior. Esto resulta significativamente menor, sobre todo cuando la cantidad de segmentos con que se divide la ruta en el método anterior aumenta.

Como se verá más adelante esta forma generar las rutas tiene algunos problemas al tratar de construir rutas si el punto de llegada esta del lado opuesto de donde inicia.

Se hizo una segunda aproximación, tomando la función objetivo anterior se sumó la distancia del punto donde se encuentra el robot y la distancia de la circunferencia hacia el punto objetivo, logrando mejorar las distancias (Solano et al 1993).

Para este caso la función objetivo en términos del algoritmo genético es la siguiente:

$$d = \sqrt{(x_g - (r \cos \theta + x_i))^2 + (y_g - (r \sin \theta + y_i))^2} + \sqrt{(x_{prev} - (r \cos \theta + x_i))^2 + (y_{prev} - (r \sin \theta + y_i))^2} \dots (5.5)$$

Donde :

x_g, y_g	Posición final
x_i, y_i	Posición inicial
r	Es igual al radio de la circunferencia o la distancia de la posición inicial al círculo bajo consideración
θ	Ángulo definido por las X-axis y la posición bajo consideración
X_{prev} y Y_{prev}	Define la posición previa.

Con la segunda aproximación, en el segundo método se mide la distancia más cercana al punto objetivo y la distancia del punto donde está ubicado y el punto objetivo. En la Figura 5.9, se muestra que con esta segunda aproximación se obtiene una mejora sustancial con respecto a la anterior.

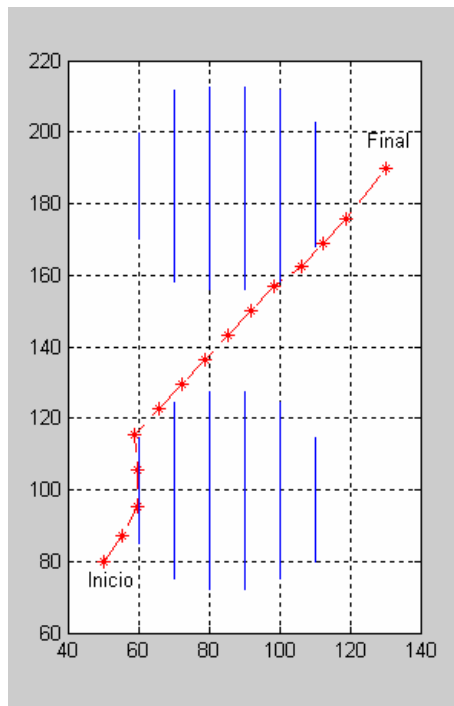


Figura 5.9 Segundo método utilizando la segunda aproximación

Las distancias recorridas por el robot son menores que con la primera aproximación, con esta función objetivo la distancia promedio fue 1,321.340

En las Figuras 5.9, 5.10, se observa que entre cada una de las rutas generadas existen pequeñas diferencias, esto se debe a que el algoritmo genético no tiene ningún otro criterio que conserve a los mejores individuos de cada generación, como puede ser el elitismo¹, lo que provoca que no siempre se alcance la ruta óptima, esto se corrige al aumentar el número de generaciones del algoritmo genético.

¹ Elitismo, conserva a los mejores individuos de cada generación en los algoritmos genéticos.

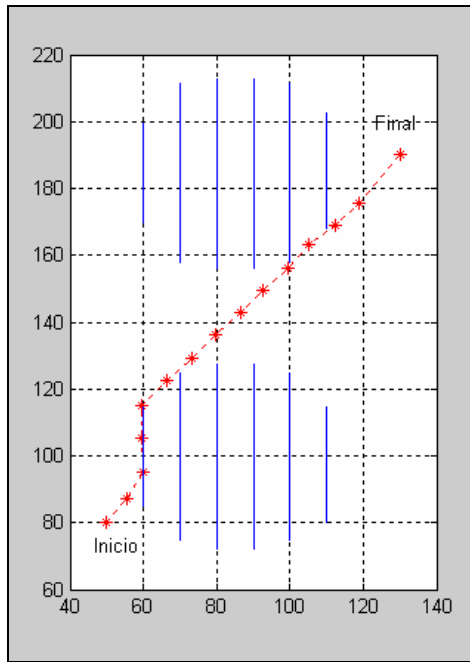


Figura 5.10, ruta generada con el segundo método con la segunda aproximación con obstáculos en forma de círculos.

Al igual que en el primer método el algoritmo se puso con un obstáculo en forma de T, este obstáculo resulta particularmente difícil, ya que el robot sólo tiene una salida posible y tiene que rodear todo el obstáculo para encontrar la ruta hacia al punto objetivo, como se observa en la Figura 5.11.

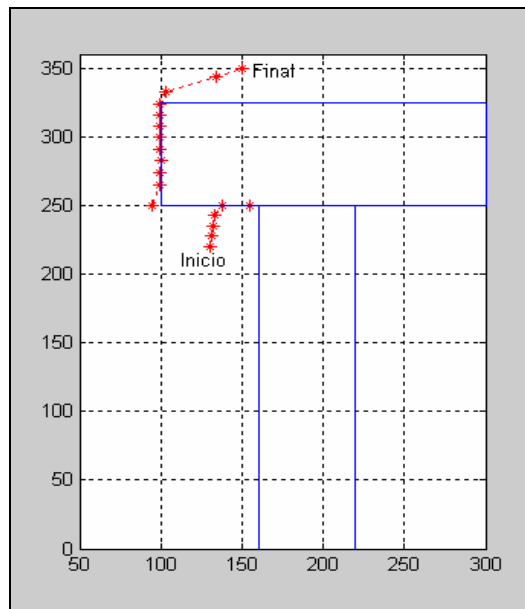


Figura 5.11, ruta generada a partir del segundo método con la segunda aproximación con un obstáculo en forma de T.

La diferencia entre la primera y la segunda aproximación, como ya fue expuesto anteriormente, es que en la primera aproximación no se considera la distancia entre el punto donde está el robot al inicio de la ruta y el punto objetivo, Figura (5.12 A); en la segunda se mide la distancia entre el robot y la circunferencia más cercana al punto objetivo (5.12 B). Esta diferencia hace que el robot tome una ruta más corta (Figura 5.12), recorriendo una distancia de 158.0081 (primera aproximación) y de 101.6739 (segunda aproximación), por lo que el robot recorre 56.3342 menos.

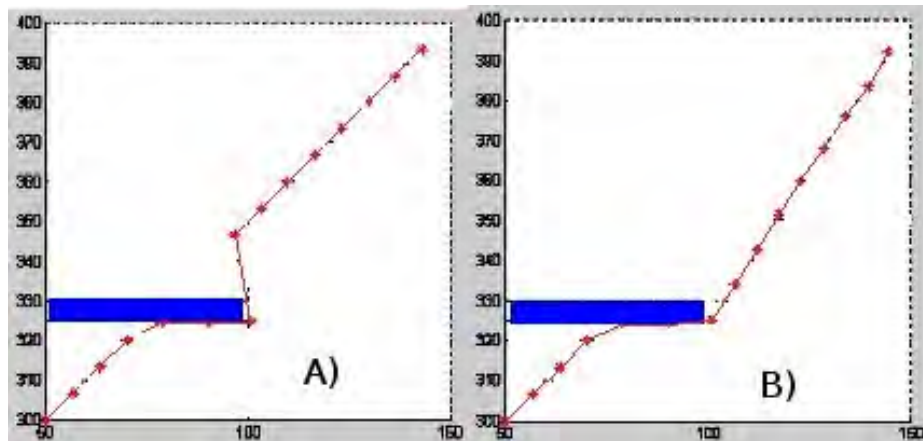


Figura 5.12 A) Ruta generada con el segundo método, B) Ruta generada con el segundo método y la segunda aproximación.

Por lo que, la segunda aproximación se puede considerar como la mas adecuada para alcanzar el punto objetivo. En esta aproximación, las circunferencias son generadas en una posición inicial $I(x, y)$ con un radio "r",; con un incremento en "r" cada vez que encuentra un punto mas cercano al objetivo $f(x,y)$, hasta que la distancia "r" es igual a la distancia lineal entre $I(x, y)$ y $f(x,y)$. En el caso de poder evadir el obstáculo antes de la distancia $I(x, y) - f(x,y)$ el algoritmo no puede encontrar la solución o ruta, Figura 5.13.

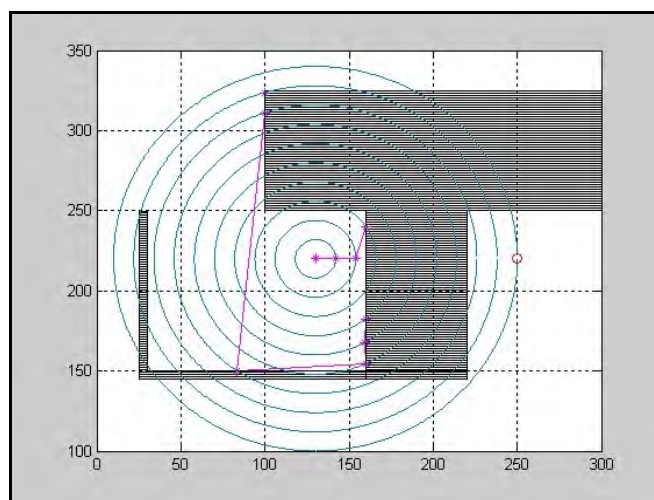


Figura 5.13. El robot se queda varado al llegar a la última circunferencia generada.

Para solucionar dicho problema, se incrementó “r” tantas veces como el robot necesite para evadir el obstáculo, ampliando el rango de búsqueda del robot, como se observa en la Figura 5.14. Donde el algoritmo se ve provisto de más puntos (circunferencia) para encontrar la ruta optima.

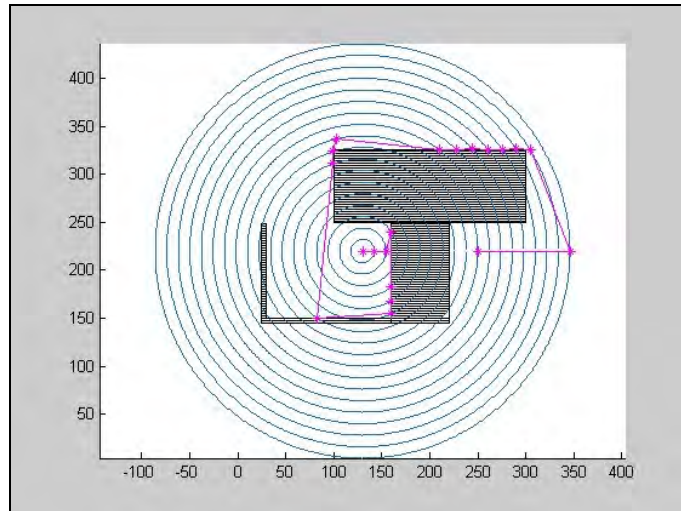
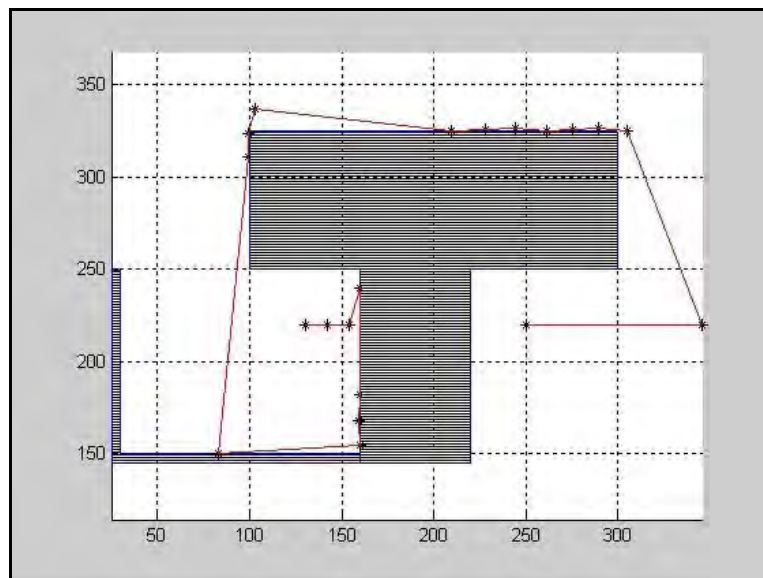


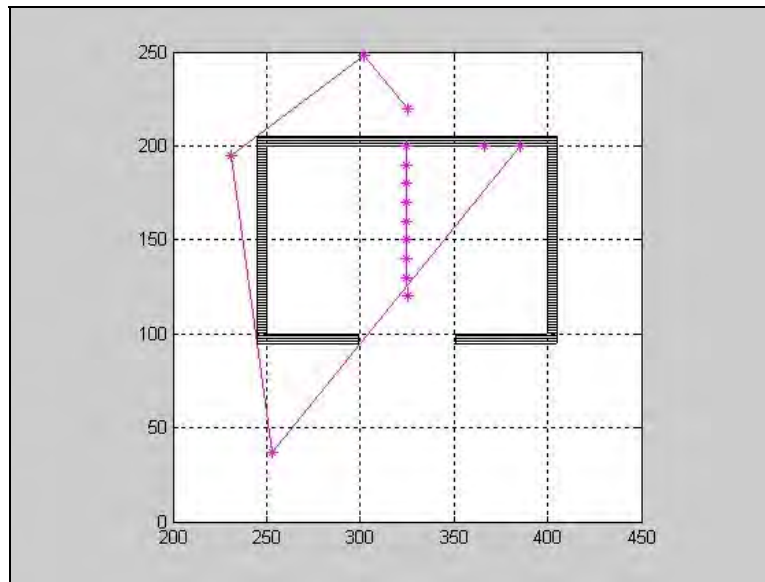
Figura 5.14 ruta generada al aumentar el numero de circunferencias sobre un obstáculo en forma de T.

En la Figura 5.15, se muestra la ruta encontrada por el algoritmo omitiendo la representación de los círculos de iteración generados en la búsqueda del punto objetivo. Cabe señalar que el algoritmo, al utilizar incrementos de “r”, no detecta los obstáculos presentes en las rutas fijadas para los primeros cálculos, esto es que la visión del robot es corta, por lo que da la apariencia de no encontrar la ruta, solo hasta que “r” es suficientemente grande encuentra la salida y por ende la ruta.



5.15 Ruta generada con un obstáculo en forma de T.

De esta forma es posible encontrar la ruta óptima para librar obstáculos complejos distintos, mediante el algoritmo propuesto (ruta 2 modificada), como el mostrado en la Figura 5.16, donde el robot solo tiene una salida y rodea los obstáculos hasta encontrar la ruta que lo llevara al objetivo deseado, igual que lo mencionado anteriormente, solo hasta que la visión del robot ó "r" es lo suficientemente grande, para encontrar la salida.



5.16. Ruta generada cuando el robot se encuentra dentro de un cuadro.

ANALISIS

Análisis del primer método

Una ruta está compuesta por “n” segmentos de recta, cada segmento se conforma por dos puntos, los cuales representan los extremos del segmento, así, las rutas desde el punto inicial hasta el final son constituidas por la unión de cada uno de los puntos intermedios.

La labor del algoritmo genético será buscar la combinación de puntos intermedios que recorran la menor distancia y libren el o los obstáculos.

La planeación de rutas a través de este método se torna complejo, debido a que no existe un conjunto único de rectas que de solución al problema; es decir, al generar la población inicial, se generan las coordenadas que forman rectas en el espacio de configuración, sin embargo, estas coordenadas pueden o no formar parte de un mismo segmento de recta, generando un problema con múltiples soluciones, por el aumento en el número de puntos intermedios que conforman la ruta completa y generando un número mayor de soluciones que satisfacen el problema o bien encuentran el punto objetivo, resultando muy complejo el calculo de la ruta libre de obstáculos.

Ante problemas complejos como serían los obstáculos en forma de T, este método no logra encontrar rutas libres de obstáculos (Figura 5.17) y por tanto nunca llega al punto objetivo, es más, no parece encontrar puntos lógicos que lo lleven en el sentido correcto.

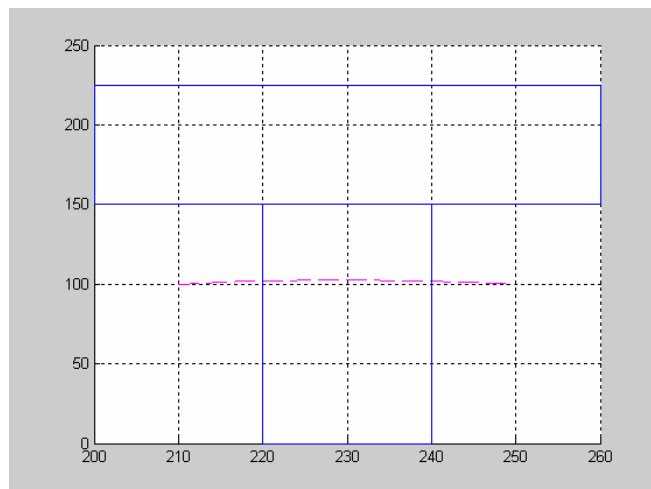


Figura 5.17.- El algoritmo no encuentra una ruta óptima.

Por otro lado, el primer método genera rutas óptimas cuando dentro de la población inicial existen puntos ubicados dentro del "espacio libre de obstáculos" y el número de rectas que lo conforma es pequeño, ver Figura 5.18.

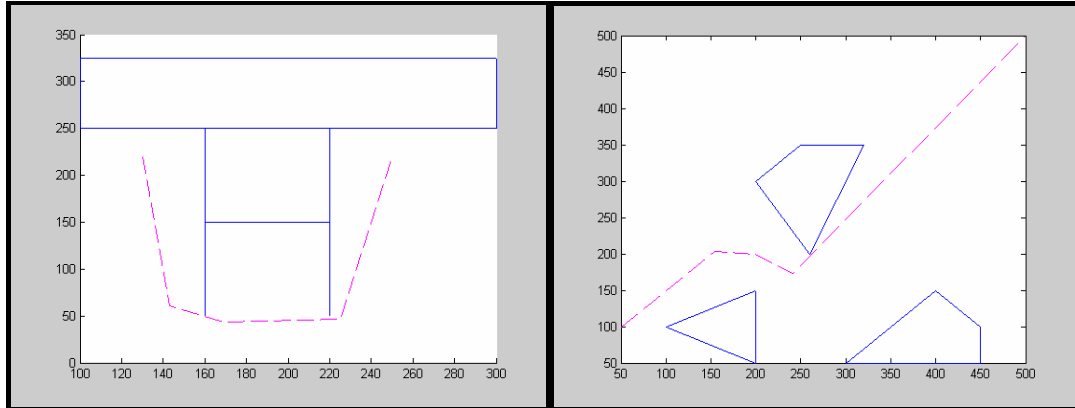


Figura 5.18.- El algoritmo logra encontrar una ruta libre de obstáculos.

Análisis del segundo método.

En la mayoría de los métodos de planeación de rutas, el espacio de configuración es dividido en pequeñas partes, de tal forma que el robot puede buscar la ruta óptima dentro del área delimitada, de manera similar a los métodos de descomposición de celdas, el método que aquí se propone, divide el espacio en circunferencias, primero generando una circunferencia de radio "r", a través de la cual el robot busca el punto más cercano al punto objetivo, cuando es encontrado, el radio de la circunferencia se aumenta, generando así una ruta, esto puede provocar que no se encuentre el punto objetivo, pero siempre dependerá de la morfología del obstáculo a librar y el punto de partida y el punto final.

Esto se explica dado que el robot no tiene una visión del entorno total que lo rodea, por lo que únicamente indaga la ruta mas cercana dentro del área delimitada por la circunferencia, cuando la circunferencia ("r") aumenta la visión del robot también aumenta.

La razón fundamental por la que, en un principio, las circunferencias (visión) son pequeñas, es para que la generación de rutas sea más flexible, al no conocerse la morfología del sitio se debe sortear con los obstáculos inmediatos antes de poder buscar el punto objetivo, adaptándose, en forma creciente a ambientes más complejos.

En la Figura 5.18 se aprecia, que las trayectorias obtenidas en espacios de configuración complejos son cercanas al óptimo, y no se tiene como en el caso anterior una restricción en el número de rectas que lo componen.

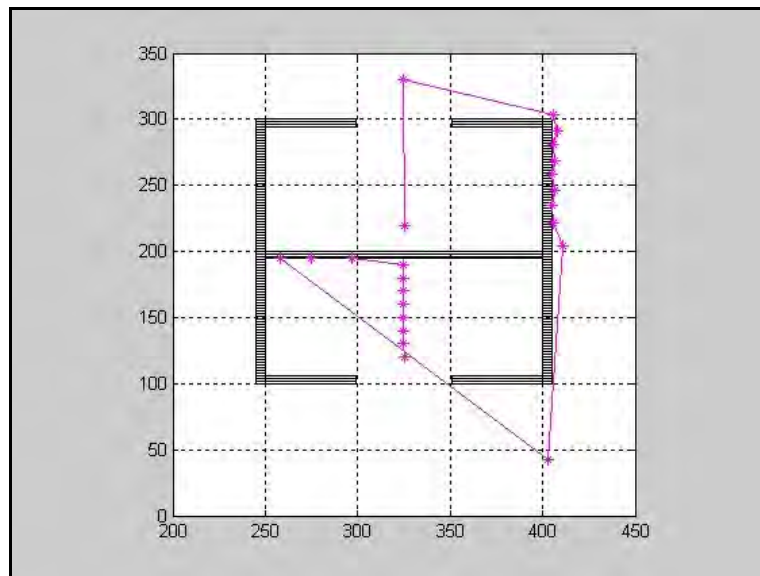


Figura 5.18. Ruta generada cuando los puntos de inicio y de llegada están encerrados en cuadrados

Con esta función objetivo se observa que las rutas son más próximas al óptimo, a diferencia de la primera función propuesta, dando como resultado trayectorias mucho más robustas y corrigiendo los problemas que se presentaron con el método del J Solano y D.I Jones². Además se observa que los obstáculos son rodeados obteniendo una trayectoria válida óptima.

J Solano y D.I Jones 1993²

Op. cit

CONCLUSIONES

En el campo de la planeación de rutas ya existen varias alternativas, pero los métodos aquí presentados complementan los ya existentes y ofrecen una alternativa viable para muchas aplicaciones.

La utilización de algoritmos genéticos en la planeación de rutas libres de obstáculos permite, para el primer método, trayectorias satisfactorias, teniendo como principal desventaja, que para cada trayectoria válida (si aumentamos el número de segmentos de recta que lo componen) el número de posibles soluciones al problema se incrementa sustancialmente, es decir, podría generarse un gran número de rectas que a su vez integren un segmento de recta o trayectoria deseada. No genera rutas óptimas, aunque sí factibles.

El problema se vuelve complejo, ya que no existe un conjunto único de rectas que solucione el problema de planeación de rutas, es decir, al generar la población inicial, se generan coordenadas que forman rectas, estas coordenadas pueden o no formar parte de una misma recta, obteniéndose así un problema con muchas soluciones, ya que al aumentar el número de puntos intermedios que conforman la ruta, el número de posibles soluciones también aumenta, por lo que la búsqueda de la ruta libre de obstáculos resulta compleja.

En resumen, se presentan las ventajas y desventajas de utilizar este método:

Ventajas

1. Buscar entre todas las posibles soluciones en el espacio.
2. Rutas más flexibles.
3. Rutas óptimas con pocos segmentos de recta.

Desventajas:

1. Al aumentar el número de segmentos de recta se hace más ineficiente.
2. Tiempo de ejecución muy alto.

En el segundo algoritmo se encontró que la función objetivo permite obtener mejores trayectorias que las generadas con el primer método, dividiendo el espacio en circunferencias y buscando siempre el punto más cercano al punto objetivo, esta forma de buscar la ruta óptima permite tener una visión más amplia del entorno que lo rodea, por lo tanto bajo ciertas circunstancias el robot parece dar un rodeo alrededor de los obstáculos antes de encontrar la ruta hacia el punto objetivo.

Con la función objetivo propuesta encuentra trayectorias más cortas y con las siguientes características:

1. Rutas bien definidas y casi-óptimas
2. Sin restricciones en el número de rectas o puntos intermedios que componen la ruta.
3. Menor tiempo de ejecución

La planeación de rutas libres de obstáculos enfrenta todavía muchos retos de investigación, por lo que es importante continuar realizando investigación para encontrar algoritmos más eficientes. Los trabajos futuros podrían consistir en generar una combinación de las dos funciones objetivo antes expuestas y comprobar si es posible obtener mejores rutas. Tomar la segunda función objetivo, y seguir explorando otras formas de búsqueda para obtener mejores rutas

BIBLIOGRAFÍA

- Ackoff, R L (1975). *Fundamentos de Investigación de operaciones*, Editorial Limusa, México
- Ahuactzin, J M; Eg T; Bessiere P; Emazer (1992). *Using Genetic algorithms for robot motion planning*, Institut National Polytechnique de Grenoble, France.
- Ashiru I.; Czarnecki C, (1998) *Evolving communicating controllers for multiple mobile robot systems*. Department of Computer Science, De Montfort University, Leicester LE1 9BH, UK
- Asimov Isaac (1998). Fundación, Plaza & Janes editores. Edición original 1951
- Atkar P N, Choset H, Rizzi A, Acar E, (2001). *Exact Cellular Decomposition Of Closed Orientable Surfaces Embedded in R^3* , Carnegie Mellon University.
- Barahona A, Piñero D (1994). *Genética: la vida continuidad de la vida* Editorial Fondo de Cultura Económica.
- Benson S, Nilsson N J. (1995). *Reacting, Planning, And Learning in an Autonomous Agent*, Robotics Laboratory, Department of Computer Science, Stafford University, Stafford,
- Berg M De; Kreveld M; Movermars; Schwarzkopf O, (1997). *Computational Geometry Algorithms and Applications*, Ed Springer
- Bracho de R M, Moreno J A (2000). *Heuristic Algorithm for Robot path Planning Based on Real Space Renormalization*,
- Cavalcanti P M (1999), *Algoritmos Genéticos: Principios e aplicaciones*, ICA: Núcleo de Pesquisa em Inteliência Computacional Aplicada, Departamento de Engenharia Eléctrica, Pontificia Universidade Católica do Río de Janeiro
- Challou D, Boley D, Gini M, Kumar V, A (1995). Parallel Formulation of Informed Randomized Search for Robot Motion Planning Problems, Department of Computer Science, University of Minnesota,
- Choset H, Acar E, Rizzi A A, Luntz J (2000). *Exact Cellular Decomposition in terms of Critical Points of Morse Functions*, Carnegie Mellon University.
- Choset H, Keiji N, Lazar N A, (2003) *The Arc-Transversal Median Algorithm: A geometric Approach to Increasing Ultrasonic Sensor Azimuth Accuracy*, Carnegie Mellon University, Pittsburgh.
- Choset H, Nagatani K (2001). *Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization Without Explicit Localization*.
- Choset H, Yeong L J (2001), *Sensor-Based Construction of a Retract-Like Structure for a Planar REod Robot*.
- Coello C. 2004. Introducción a la computación evolutiva, CINVESTAV-IPN, Departamento de Ingeniería Eléctrica sección de Computación.

-
- Conner D C, Atkar P N, Rizzi A A, Choset H, (2002). *Development of Deposition Models for Paint Application on Surfaces Embedded in R3 for Use in Automated Path Planning*, Carnegie Mellon University, Pittsburgh. PA.
- Critchlow A J (1985). *Introduction to Robotics*, Macmillan Publishing Company.
- Darwin C. (1983). *El Origen de las especies*. (Editado originalmente en 1859) SARPE Ed. Madrid. 638 p.
- De la Cueva V; Ramos F, (1998)., *Cooperative Genetic Algorithms: A new Approach to Solve the Path Planning Problem for Cooperative Robotic Manipulators Sharing the same work Space*. División de Ingeniería y Ciencias.
- Eksin I; Erol O.K, (2001) *Evolutionary algorithm with modifications in the reproduction phase* Istanbul Teknik Universitesi, Elektrik-Elektronik Fak, Turkey
- Engelberger, J. (1979). *Robotics Today*, Robotics.
- Fujimura K, (1991). *Motion Planning in dynamic Environments* Ed Springer-Verlay Tokyo.
- Fuller G, (1981). *Geometría Analítica*, Ed. Compañía Editorial Continental, SA de CV, México.
- Gallardo D; Colomina O; Flórez F; Rizo R (1998). *Generación de trayectorias robustas mediante computación evolutiva*.
- Godfried T. Toussaint, (1992). "What is computational geometry," Proc. IEEE, vol. 80, No. 9, September, pp. 1347-1363.
- Goldberg D E, (1989). *Genetic Algorithms in search, optimization and machine learning*, The University of Alabama, Addison-Wesley publishing company.
- Gonzales K. s. fu R.C; Lee C.S.G, (1987). *Robotics, Control, Sensing, Vision, And Intelligence* McGraw-Hill
- Grabowski R; Khosla P; Choset H, (2003). *Autonomous Exploration Via Regions of Interest*, Carnegie Mellon University, Pittsburgh.
- Hernando M; Gambao E, (2002). *Visibility Analysis and Genetic Algorithms for Fast Robot Motion Planning*, Universidad Politécnica de Madrid, España.
- Hiller F; Lieberman G (1990). *Introduction to Operations Research*, ED MacGrwa-Hill, Quinta edición.
- Hisiang W K; Hsing C C; Der L J, (1995). *A Fuzzy Potential Approach whit the Cache Genetic Learning Algorithm for Robot Path Planning*, Department of Electrical Engineering, National Cheng Kung University, Taiwan, Department of Electrical Engineering Chang Gung College of Medicine And Technology Tao-Yuan, Taiwan,
- Hocaoglu C; C Sanderson A, (1996). *Planning Multi-Paths Using Appreciation in Genetic Algorithms*, Electronics Agile Manufacturing Research Institute, And Electrical, Computer, And Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, New York,

-
- Holland J, (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- Ibi A; Chis C, (1995). *Optimal Motion Planning for Mobile Robots Using Genetic Algorithms*, Department of Computer Science, De Montfort University Leicester UK.
- Kang D ; Hashimoto H; Harasima F, (1995), *Path Generation for Mobile Robot Navigation using Genetic Algorithm*, Institute of Industrial Science University of Tokyo,
- Kwong k. C; Alí M.S Z, (1993). *Genetic-Based Minimum-time Trajectory Planning of Articulated Manipulators with torque Constraints*, The Institution of Electrical Engineers
- Latombe J, (1991). *Robot Motion Planning*, Kluwer Academic Publisher.
- Latombe J, (1999). *Motion Planning: A journey of Robots, Molecules, Digital Actors, And Other Artifacts*, Stanford University, USA, .
- Laumond J.P; Sekhavat; Lamiroux (1998). *Guidelines in Nonholonomic Motion Planning for Mobile Robots*, Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique
- Laumond JP; Sekhavat S; Lamiroux F, (1998). *Robot Motion Planning and Control*, Capítulo 1 y 2,
- Lozano-Pérez T; Wesley M A, (1979). *An Algorithm for Planning Collision-free Paths among Polyhedral Obstacles*, IBM. Thomas J Watson Research Center
- Makino T; Yokoi H; Kakazu Y, (1999), *Development of a Motion Planning System for an Agricultural Mobile Robot*, Hokkaido University,
- Meng Q; Hamam Y, (1995). *Optimal Dynamic Control of Mobile Robot by Genetic Algorithm whit Symmetric Code-GASC*, Department of Computer Science, Yantai University, Yantai, China, Department Automatique ESIEE, Noisy-le-Grand, France,
- Michalewicz Z (1996). *Genetic algorithms + data structures = evolution programs*, Berlin Springer,
- Mitchell RJ ; Keating DA; kambhampati C, (1994). *Learning Strategy for a Simple Robot Insect*, University of Reading, UK
- Montagnon C, (1999). *Planificación de la expansión de sistemas de distribución vía Algoritmos Genéticos*, Pontificia Universidad Católica de Chile, Departamento de Ingeniería Eléctrica.
- Nilsson, N. *Fantasy, Computation, and Reality*, (1985). Unpublished version of the 1985 Jacob Bronowski Memorial Lecture given at New College, University of Toronto

Nishimura T; Sugawara K; Yoshihara I; Abe k. (1999). *A Motion Planning Method for Hyper Multi-joint Manipulator using a Genetic Algorithm*, Dept. of Electrical and Communication Engineering, School of Engineering, Tohoku University, Graduate School of Information system, the Univ. of Electro-Communications, Tokyo, System Development lab, Hitachi Ltb. Japan.

Norio B; Naoyuki K (1994). *Collision Avoidance Planning Of a Robot Manipulator by Using Genetic Algorithm A Consideration for the problem in which moving obstacles and/or several robots are included in the workspace*, Information Science, Osaka Kyoiku University, Asahigaoka, Kasahihara City 582 Japan, Hokkaido University, Sapporo, 060, Japan

Nowostawski M; Poli R, (1999) *Parallel Genetic Algorithm Taxonomy*, Information Science Department, University of Atago, School of Computer Science, The University of Birmingham.

Prahlad V; Kay C; Ming-Liang T and W, (2003). *Evolutionary Artificial Potential Fields and Their Applications in Real Time Robot Path Planning*, Department of electrical Engineering, The National University of Singapore.

Pulgarín A; Smith R (1998). *Algoritmo híbrido de algoritmos genéticos y búsqueda tabú aplicado a un problema de expansión*, Postgrado en aprovechamiento de recursos hidráulicos, Facultad de Minas, Universidad Nacional de Colombia.

Ramakrishnan R; Zein-Sabatto S, (2001). *Multiple Path Planning for a Group of Mobile Robot in 2-D Environment Using a Genetic Algorithms*, Tennessee State University, Nashville,

Roger T; Hong H; Chi Wong, (2001). *Robot path planning Using Genetic Algorithm*, Dept, Of Mechanical Engineering, University of Alberta

Schen S; Kiichi T, (1993). *Learning of Maze Using a Genetic Algorithm*, School of Science Waseda Univrsity, 3-4-1 Okubo, Shinjuku—Ku, Tokyo 160, Japan, IEEE

Schilling R J, (1990). *Fundamentals of Robotics, Analysis and Control*, ED Prentice Hall pp 362 -391.

Serkan A, Hakan T (2002), *Time-Optimal Trajectory Planning using A Smart Evolutionary Algorithm whit Fuzzy Inference System*, International Symposium on Intelligent Control, Vancouver, Canada,

Singh Sanjiv, Kelly Alonzo, (1996). *Robot Planning In the Space of Feasible Actions: Two Examples*, Field Robotics Center, Carnegie Mellon University,

Soberón J, (1989). *Ecología de poblaciones*, Editorial Fondo de Cultura Económica.

Solano J, Jones D I (1993). *Generation of Collison-free Paths, a Genetic Approach*, Departamento de Electrónica y Automatización, IIMAS, UNAM, School of Electronic Engineering and Computer Systems, University of Wales, Bangor, Great Britain. 1993

Toussaint G T, *What is Computational Geometry*, Computational Geometry Laboratory, School of Computer Science McGill University Montreal Canada.

Vadakkapat P; Chen T K; Ming-Liang W, (2000). *Evolutionary Artificial Potential Fields and Their Application in Real Time Robot Path Planning*. Department of Electrical Engineering. The National University of Singapore,

Vadakkapat P; Chen T K; Ming-Liang, (2000). *Evolutionary Artificial Potential Fields and Their Application in real Time-Robot path Planning* Department of Electrical Engineering, The National University of Singapore

Wu C.J; Lee T.T; Gruver W A. (1994) *Algorithm Path Planning for Robots Based On the Forbidden Rotation Range Center for Robotics and Manufacturing system*, University Kentucky

Xiao J; Xi N; Dulimarta; Tummala R.L, (2002). *Motion Planning of a Bipedal Miniature Crawling Robot in Hybrid ConFiguration Sapce*, Department of Electrical and Computer Engineering, Michigan State University,

Yeong L, Choset H, (2004) *Motion Planning for a Rod-shaped Robot in R^3 : Connecting the Rod-Hgvg using the Point-HGVG*, Carnegie Mellon University.

Referencias en Internet

<http://msl.cs.uiuc.edu/~lavelle/cs497/book/rplan/node4.html>

Robot Motion Planning and Visibility Graphs

<http://cs.smith.edu/~streinu/Teaching/Courses/274/Spring98/Projects/Philip/fp/contents.html> octubre 31 2002

Finding the Shortest Path in the Plane with Polygonal Obstacles

<http://www.cs.caltech.edu/~cesar/report/introduction.html>, octubre 31 2002

Introduction to Voronoi Diagrams

<http://dimacs.rutgers.edu/drei/96/classroom/voronoi/intro.html>, octubre 31 2002

Diagramas de Voronoi

<http://www.ma1.us.es/Miembros/almar/docencia/gctem2co.htm>

Robot Motion planning

http://www.dma.fi.upm.es/docencia/trabajosfindecarrera/programas/geometriacomputacional/Shortest_Path/html/E_Planif_Robots.htm

Robotics

<http://www.cs.dartmouth.edu/~brd/Teaching/AI/Lectures/Summaries/robotics.html>

German Aerospace Centre, Genetic algorithms and optimum robot design

<http://www.robotic.dlr.de/Holger.Weiss/garep/node4.html>, Enero 2004

International Federation of Robotics

<http://www.ifr.org/>

Motoman Robotics Ibérica

<http://www.motoman.es/detalle.asp?Referencia=UP20-6>

Real Academia Española - Diccionario de la Lengua Española

<http://www.rae.es/>

Nils J. Nilsson Publications (2005)

<http://ai.stanford.edu/users/nilsson/publications.htm>

Kaartinen Jukka (2005)

<http://cgm.cs.mcgill.ca/~athens/cs507/Projects/2002/JukkaKaartinen/>

Mathworks (2005)

<http://www.mathworks.com/access/helpdesk/help/toolbox/images/enhanc12.html>

Kenneth A. De Jong's Dissertation (2005)

http://cs.gmu.edu/~eclab/kdj_thesis.html

APÉNDICE A

Conceptos básicos.

El espacio de trabajo - El robot tiende a presentar una geometría fija y limitada. El espacio de trabajo es el límite de posiciones en espacio que el robot alcanza. Para un robot cartesiano (como una grúa arriba) los espacios de trabajo podrían ser un cuadrado, sin embargo, para los robots más sofisticados los espacios podrían ser de una forma esférica.

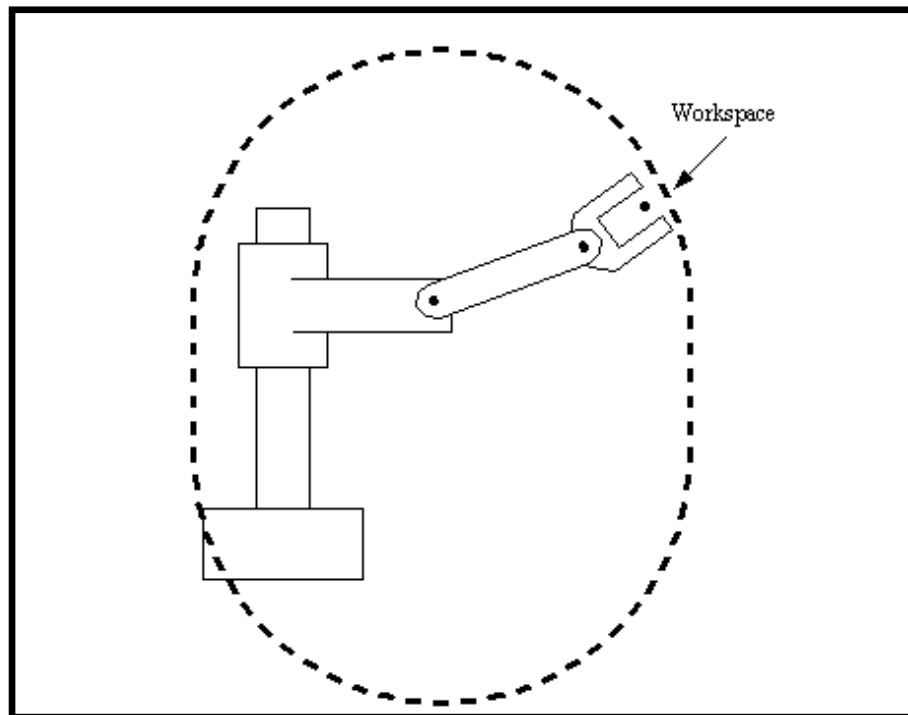


Figura A.1 Construcción del espacio de configuración para un brazo robot

El grado de Libertad (el gdl).- Cada junta en el robot introduce un grado de libertad. Cada gdl pueden ser un deslizador, del tipo rotatorio, u otro de actuador. Los robots tienen 5 o 6 grados de libertad típicamente, 3 de los grados de libertad permiten el posicionamiento en 3D espacio, mientras que en el otro se usan 2 o 3 para la orientación del efector del extremo. 6 grados de libertad son bastante para permitir al robot alcanzar todas las posiciones y orientaciones en 3D espacio. 5 gdl requiere una restricción a 2D espacio, el resto limita las orientaciones. Normalmente se usan 5 gdl, por ocuparse de herramientas como los soldadores del arco.

Tabla A-1

Artículo	Autores	Operadores Genéticos	Método	Otros	
Learning for a maze using genetic algorithm	Suchen su y Kiichi Tsuchiya	Algoritmo genético propuesto por Holland	Marcan el camino con cuadros blancos y negros para determinar donde puede caminar y donde no.		Waseda University
Path Generation for mobile robot Navigation Using genetic algorithm	Daehee Kang Hideki Hashimoto Fumio Harashima	Fitness tournament	Quad tree method Teoría de grafos		University of Tokyo
A fuzzy potential Approach whit the cache genetic learning for robot path planning	Kung Hsiang Chin Hsing Chen Jiann Der Lee		Potencial fiel	Fuzzy logic	National Cheng Kuang University
Genetic-based minimum-time trajectory planning of articulated manipulators with torque constraints	Kwong K Chan And Ali MS Zalzal		Un numero finito de puntos entre un punto a otro, tomando en cuenta la velocidad.		The Institution of Electrical Engineer
Collision Avoidance Planning of a Robot Manipulator by Using Genetic Algorithm	Norio Baba & Naoyuki Kubota		Divide el espacio en celdas y calcula su valor potencial.		Osaka Kyoiku University
Visibility Analysis and Genetic Algorithms for fast Robot Motion Planning	Miguel hernando Ernesto Gambao		Visibility Graph (Graficas de visibilidad)		Universidad Politécnica de Madrid
A Genetic Algorithm for Robust Motion Planning	Domingo Gallardo, Otto Colomina, Francisco Flórez, Ramón Rizo		distancia euclidiana y velocidad lineal y angular		Universidad de Alicante España
Multiple Path Planning For A group of mobile Robot in a 2D environment using genetic algorithms	R. Remakrishman And S Zein-Sabatto		Divide el espacio en celdas		Tennessee state University
Evolutionary Artificial potential fields and third application in real time robot path planning	Prahlad Vadakkepat, Kay Chen Tan and Wang Ming-liang		Potential fiel (campos potenciales)		Department of Electrical Engineering The National University of Singapore
Generation Of collision-free paths, A genetic approach	J Solano and D.I Jones		Distancia Euclidiana y rotación		
Using Genetic algorithms for robot motion planning	Juan Manuel Ahuactzin, El-Ghazali Talbi, Pierre Bessiere, Emmanuen Mazer		Cinemática Inversa		Institute National Polytechnique de Grenoble
A Motion Planning Method for a Hyper Multi-joint Manipulator using Genetic Algorithm	T Nishimura, K Sugawara, I. Yoshihara K Abe		Potential field and Collision-free sequence generation		Tohoku University

CÓDIGO FUENTE

Algoritmo Genético Simple (Goldbert)

```
rand('seed', 0)
clear all
clc
X=200;
Y=9;
Z=15;
NumGen=50;
Corin=[130 220];
corfnl=[150 350];
%Corin=[50 80];
%corfnl=[130 190];
Gen=0;
%varia=100;
pmuta=0.1;
pcruza=0.95;
vmin=0;
vmax=1;
bit=Y;
[cxyobst]=malla2;
[r]=radio(Corin,corfnl,Z);
r2=r;
prev=Corin;
ngn=5;
t=0;

t0=clock;
for i=1:Z
    if Gen==0
        [toc]=pena2(Gen,X,cxyobst,r,prev,Corin,corfnl);
        end
    prev
    if t~=0 & toc~=0
        r=r+r2;
        Gen=0;
        end
    prev1(i,:)=prev(,:);
    if toc==1
        %Generar una población
        [poblacion]= CPoblhiper(X,Y);
        A=poblacion;
        t=1;
```

```

while NumGen>Gen
%Decodifica el codigo binario
[AN]= decohiper(Y,A,X);
Gen=Gen + 1;

%Funcion Objetivo
|[ANC]=rosenbrock (X,Y,Z,AN);
|[ANC]=objetivo(X,Z,AN,Corin,corfnl,r,prev)
%Penalización
|[ANC]=pena(ANC,AN,X,cxyobst,r,prev,Corin);
%calculo de Promedio,vmaxPsel,SumaPsel,Pselect,PromPsel,Sumamed,vmaxmed,media.
|[Promedio,vmin]= valoreshiper(ANC);
|[mejor]=find(ANC<=vmin);

%PRESION DE SELECCION
|[Pselect]=fortaleza (ANC,Z);
%Ranking
|[Pselect]=ranki(ANC);
%Ruleta
|[B]=ruletahiperXY (Pselect,X,A,Y);
%estocastico
|[B]=estocasticohiperXY (Pselect,X,A,Y);

%Cruza
|[PCruz,PCor,B,ncruza]= cruzahiperX(X,Y,B,pcruza,bit);

%Mutacion
|[A,nmuta]= mutahiperX(X,Y,B,pmuta,bit);
%Graficas
|[ANCmej]=graficas(vmin,Gen,Promedio, cXy,ANC);
|[mejor]=find(ANC<=vmin);

if i~=0
hold on
grid

c1=r*cos(AN(mejor(1))*pi/180)+Corin(1,1);
c2=r*sin(AN(mejor(1))*pi/180)+Corin(1,2);

if Gen==NumGen
prev(1,1)=c1;
prev(1,2)=c2;
end

```

```

drawnow
plot (c1,c2,'g*')
plot(Corin(1,1),Corin(1,2),'r*')
plot(prev1(:,1),prev1(:,2),'r*')
end
mejor1(Gen)=AN(mejor(1));
fprintf('%5i %12.5g \n', Gen, vmin)
%end del while
end
else
if Gen==0
m3=(corfnl(1,2)-prev(1,2))/(corfnl(1,1)-(prev(1,1)));
else
m3=(corfnl(1,2)-Corin(1,2))/(corfnl(1,1)-(Corin(1,1)));
end
c1=r*cos(atan(m3))+Corin(1,1);
c2=r*sin(atan(m3))+Corin(1,2);
prev(1,1)=c1;
prev(1,2)=c2;
r=r+r2;
end
%Tiempo final
tiempo=etime(clock,t0);
%Salvar .mat
save(char(['pobla' num2str(i)]))
end
%Decodifica el codigo binario en decimal
function [AN]= decohiper(Y,A,X)
bit=Y;
k=2;
cont=Y;
for i=1:X
for j=1:Y:bit
st = num2str(A(i,j:Y));
bsca = find(st=='1' | st=='0');
b = st(bsca);
AN(i)=(bin2dec(b)*360)/511;
k=k-1;
end
k=2;
cont=Y;
end

```

%Genera una poblacion hiperdimensional binaria (1,0)

```
function [poblacion]= CPoblhiper(X,Y)
poblacion=rand(X,Y)<0.5;
```

%Funcion Objetivo

```
function [ANC]=objetivo(X,Z,AN,Corin,corfnl,r,prev)
dist=0;
%Ordenamiento de las cordenadas XY
for i=1:X
c1=r*cos(AN(i)*pi/180);
c2=r*sin(AN(i)*pi/180);
dist=(sqrt((corfnl(1,1)-(c1+Corin(1,1)))^2+(corfnl(1,2)-(c2+Corin(1,2)))^2))%...
%+(sqrt((prev(1,1)-(c1+Corin(1,1)))^2+(prev(1,2)-(c2+Corin(1,2)))^2));
ANC(i)=dist;
dist=0;
c1=0;
c2=0;
end
ANC=ANC';
```

%Penalización

```
function [ANC]=pena(ANC,AN,X,cxyobst,r,prev,Corin)

for j=1:X
c1=r*cos(AN(j)*pi/180)+Corin(1,1);
c2=r*sin(AN(j)*pi/180)+Corin(1,2);

for k=1:6
if prev(1,1)~=c1
m1=(prev(1,2)-c2)/(prev(1,1)-c1);
else
m1=inf;
end

if cxyobst(1,1,k)~=cxyobst(2,1,k)
m2=(cxyobst(1,2,k)-cxyobst(2,2,k))/(cxyobst(1,1,k)-cxyobst(2,1,k));
else
m2=inf;
end

if m1~=m2
p=[(prev(1,1)-c1) , -(cxyobst(2,1,k)-cxyobst(1,1,k))];...% (cxyobst(1,1,k)-c1);...
```

```

    (prev(1,2)-c2), -(cxyobst(2,2,k)-cxyobst(1,2,k)));%(cxyobst(1,2,k)-c2)];
p1=[(cxyobst(1,1,k)-c1);(cxyobst(1,2,k)-c2)];
t=p\p1;
if t(1,1)>=0 & t(1,1)<=1 & t(2,1)>=0 & t(2,1)<=1
    ANC(j)=ANC(j)+(realmax/100);
end
end
end
end
ANC=ANC';

```

%Fotaleza

```

function [Pselect]=fortaleza (ANC,Z);
SumaANC=sum(ANC);
vmaxPsel=0;
SumaPsel=0;
for k=1:Z
    Pselect=(1-(ANC(1,:)/SumaANC))/(sum (1-(ANC(1,:)/SumaANC)));
end

```

%seleccion de lo individuos (ruleta)

```

function [B]=ruletahiperXY (Pselect,X,A,Y);
SumPsel=0;
M=0;
for k=1:X
    SumPsel=SumPsel + Pselect(1,k);
    RM2=SumPsel-Pselect(1,k);
    rango(1,k)=RM2;
    SumPsel2(1,k)=SumPsel;
end

for k=1:X
    rand1=rand;
    for k=1:X
        if rango(1,k)<=rand1 & rand1<=SumPsel2(1,k)
            M=M+1;
            B(M,:)=A(k,:);
        end
    end
end
end

```

%Estocástico Universal

```

function [B]=estocasticohiperXY (Pselect,X,A,Y);
SumPsel=0;
M=0;
for k=1:X
    SumPsel=SumPsel + Pselect(1,k);
    RM2=SumPsel-Pselect(1,k);
    rango(1,k)=RM2;
    SumPsel2(1,k)=SumPsel;
end
n=0;
rand1=rand;
for k=1:X
    rand1=rand1+(1/X);
    if rand1>=1
        rand1=rand1-1;
    end
    n=find(rand1>=rango & rand1<=SumPsel2);
    for k=n
        M=M+1;
        B(M,:)=A(k,:);
    end
end

```

%Rankin funtion

```

function [Pselect]= ranki (ANC)

nm=1.5;
nn=2-nm;
w=[(1:size(ANC,2))', (ANC)];
q=[(1:size(ANC,2))' sortrows(ANC', 1) sortrows(w,2)];
for i=1:size(q,1)-1
    if q(i,2)==q(i+1,2)
        t=q(i+1:size(q,1),1)-1;
        q(i+1:size(q,1),1)=t;
        t=0;
    end
end
l=nm-(nm-nn)*(q-1)/(size(q,1)-1);
l(:,3)=q(:,3);
l=sortrows(l,3);
Pselect=l(:,1)/sum(l(:,1))';
Pselect=Pselect';

```



```

%Cruza
function [PCruz,PCor,B,ncruza]= cruzahiperX(X,Y,B,pcruza,bit);
ncruza=0;
PCruz = randperm(X); %Parejas de cruza
PCor = randperm(bit); %Define el punto de corte
final=0;
trans=0;
n=find(rand(1,X)<pcruza);
for k=1:2:2*size(n)
    %randn=rand;
    if X>=k+1
        final=PCor(1,k);
        if PCor(1,k)==1
            final=PCor(1,k)+1;
        end
        %for j=final:X;
        %if randn <= pcruza;
        ncruza=ncruza+1;
        trans=B(PCruz(1,k),final:bit);
        B(PCruz(1,k),final:bit)=...
        B(PCruz(1,k+1),final:bit);
        B(PCruz(1,k+1),final:bit)=...
        trans;
    end
end

```

```

%Mutacion
function [A,nmuta]=mutahiperX(X,Y,B,pmuta,bit);
nmuta=0;
e=find(rand(1,X)<pmuta);
for j=e
    nmuta=nmuta+1;
    MutacionY=randperm(bit);
    if B(j,MutacionY(1,1))==1
        B(j,MutacionY(1,1))=0;
    else
        B(j,MutacionY(1,1))=1;
    end
end
% end
end
A=B;

```