

**ALGORITMOS GENÉTICOS APLICADOS AL PLANEAMIENTO DE
TRAYECTORIAS DE ROBOTS MÓVILES. MODELAMIENTO Y SIMULACIÓN**

FABIO ALFONSO GONZÁLEZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
MAESTRÍA EN POTENCIA ELÉCTRICA
BUCARAMANGA
2004**

**ALGORITMOS GENÉTICOS APLICADOS AL PLANEAMIENTO DE
TRAYECTORIAS DE ROBOTS MÓVILES. MODELAMIENTO Y SIMULACIÓN**

FABIO ALFONSO GONZÁLEZ

**Trabajo de investigación desarrollado como
requisito para optar al título de MAGÍSTER
EN POTENCIA ELÉCTRICA.**

**Director :
JAIME BARRERO PÉREZ
Ingeniero Electricista, MSC.**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
MAESTRÍA EN POTENCIA ELÉCTRICA
BUCARAMANGA
2004**

*A Sandra, por la ayuda recibida...y es
quien escribe mis dedicatorias.*

AGRADECIMIENTOS

A DIOS quien me permite existir.

CONTENIDO

	pág.
INTRODUCCIÓN	1
1. ALGORITMOS GENÉTICOS. GENERALIDADES	3
1.1 INTRODUCCIÓN	3
1.2 COMPUTACIÓN EVOLUTIVA	4
1.3 PROBLEMAS DE OPTIMIZACIÓN COMBINATORIA	10
1.4 DEFINICIONES BÁSICAS	14
1.4.1 Conceptos Biológicos.	14
1.4.2 Conceptos de Computación Evolutiva.	17
1.5 EJEMPLO ILUSTRATIVO	21
2. PRINCIPIOS MATEMÁTICOS BAJO LOS CUALES OPERAN LOS ALGORITMOS GENÉTICOS	26
2.1 INTRODUCCIÓN	26
2.2 EFECTO DE LA SELECCIÓN EN UN AG	27
2.3 EFECTO DEL CRUCE EN LOS AG	29
2.4 EFECTO DE LA MUTACIÓN EN LOS AG	29
2.5 ESQUEMAS PROCESADOS EN FORMA ÚTIL	30
2.6 OTROS TÓPICOS DE INTERÉS EN EL ESTUDIO TEÓRICO DE LOS AG	32
3. EL PROBLEMA DEL PLANEAMIENTO DEL MOVIMIENTO	34
3.1 INTRODUCCIÓN	34
3.2 DEFINICIONES BÁSICAS	35
3.3 COMPLEJIDAD DEL PROBLEMA	40
3.4 APROXIMACIONES BÁSICAS A LA SOLUCIÓN DEL PROBLEMA	41
3.4.1 Mapas de Caminos.	41
3.4.2 Descomposición En Celdas.	44
3.4.3 Campos de Potencial.	45
3.4.4 Programación Matemática.	46
3.5 TÉCNICAS DE BÚSQUEDA	47
3.6 ROBOTS MÓVILES	48
4. MODELOS DEL ROBOT Y DEL AMBIENTE DE TRABAJO	52
4.1 INTRODUCCIÓN	52
4.2 MODELO DEL ROBOT MÓVIL	53
4.2.1 Modelo cinemático.	53
4.2.2 Modelo dinámico.	62
4.2.3 Modelo Simplificado.	64
4.2.4 Comprobación del modelo simplificado.	67
4.3 MODELO DEL ESPACIO DE TRABAJO	69

	pág.
5. METODOLOGÍA PARA APLICACIÓN DE LOS AG EN LA SOLUCIÓN AL PROBLEMA DEL PLANEAMIENTO DE MOVIMIENTO	72
5.1 INTRODUCCIÓN	72
5.2 PLANEAMIENTO DE CAMINO	72
5.2.1 Codificación del problema.	72
5.2.2 La función de evaluación.	77
5.2.3 Parámetros del AG.	79
5.3 PLANEAMIENTO DE TRAYECTORIA	80
5.3.1 Codificación del problema.	80
5.3.2 La función de evaluación.	82
5.3.3 Parámetros del AG.	83
5.4 HERRAMIENTA SOFTWARE PARA LA APLICACIÓN DE LOS AG EN EL PLANEAMIENTO DE CAMINO	85
5.5 HERRAMIENTA SOFTWARE PARA LA APLICACIÓN DE LOS AG EN EL PLANEAMIENTO DE TRAYECTORIA	87
6. ANÁLISIS DE LOS RESULTADOS OBTENIDOS	90
6.1 INTRODUCCIÓN	90
6.2 ESTUDIO DE LOS RESULTADOS OBTENIDOS EN CUANTO A PLANEAMIENTO DE CAMINO	90
6.3 ESTUDIO DE LOS RESULTADOS OBTENIDOS EN CUANTO A PLANEAMIENTO DE TRAYECTORIA	102
7. CONCLUSIONES	112
BIBLIOGRAFÍA	114
ANEXOS	119

LISTA DE TABLAS

	pág.
Tabla 1. Comparativo de las tres tendencias.	11
Tabla 2. Aplicación del problema del vendedor viajero.	12

LISTA DE FIGURAS

	pág.
Figura 1. Taxonomía de la Inteligencia computacional.	7
Figura 2. Comportamiento del orden en función de n.	13
Figura 3. Clasificación de las técnicas de búsqueda.	14
Figura 4. Algoritmo genético simple.	20
Figura 5. Función de Ackley	21
Figura 6. Ilustración del método de la ruleta pesada	23
Figura 7. Ilustración del cruce en un solo punto	24
Figura 8. Ilustración del proceso de mutación.	24
Figura 9. Resultados del desempeño del AG para la función de Ackley.	25
Figura 10. Ilustración de desplazamiento de ventana en un esquema particular.	31
Figura 11. Manipulador con dos grados de libertad.	37
Figura 12. Movimientos del manipulador con dos grados de libertad.	37
Figura 13. El cuerpo A en movimiento en el espacio de trabajo y su respectivo espacio libre y espacio de	38
Figura 14. Gráfica de visibilidad en el espacio de configuración bidimensional con obstáculos poligonales.	41
Figura 15. Brazo de dos grados de libertad ubicado entre obstáculos.	43
Figura 16. Representación en vías libres, sobre un espacio de trabajo con obstáculos en forma de	43
Figura 17. Método de las siluetas y la generación de una de ellas.	44
Figura 18. Método exacto de descomposición en celdas	45
Figura 19. Tendencias en la representación de objetos.	46
Figura 20. Representación de un espacio de trabajo mediante campos de potencial.	46
Figura 21. Tipos de actuadores para robots móviles.	49
Figura 22. Péndulo simple como ejemplo de un sistema holonómico.	54
Figura 23. Ejemplo de un sistema no holonómico.	54
Figura 24. Modelo del robot móvil tipo triciclo.	56
Figura 25. Caracterización de una rueda en sentido general.	57
Figura 26. Modelo cinemático del robot teniendo en cuenta el efecto de las tres ruedas.	60
Figura 27. Ubicación general del chasis del robot móvil.	65
Figura 28. Modelo cinemático del robot móvil, sin tener en cuenta los efectos de las dos ruedas traseras.	66
Figura 29. Perfiles de velocidad angular y su respectivo desplazamiento con beta inicial de cero	67
Figura 30. Perfiles de velocidad y su respectivo desplazamiento con beta inicial de cero y $\pi/4$.	68

	pág.
Figura 31. Representación de objetos.	69
Figura 32. Representaciones de obstáculos en el espacio de trabajo y trayectorias inválidas.	70
Figura 33. Ejemplo de la representación de espacio de trabajo con algunos obstáculos.	71
Figura 34. Ilustración de rejilla de 8x8 celdas con su respectivo punto de partida y llegada.	74
Figura 35. Cromosoma básico constituyente de un individuo para $n=8$ (elemento par-distancia).	75
Figura 36. Interpretación del par de dirección para monotonía en x.	75
Figura 37. Interpretación del par dirección para monotonía en y.	76
Figura 38. Ilustración de caminos con su respectiva codificación.	76
Figura 39. Comportamiento anómalo donde la distancia es menor al óptimo.	78
Figura 40. Representación de la codificación del problema.	81
Figura 41. Representación de los resultados de la codificación.	82
Figura 42. Perfiles para una determinada codificación en particular.	84
Figura 43. Herramienta para planeamiento de camino desarrollada con "GUI" de Matlab. 5.3.	86
Figura 44. Herramienta del Planeamiento de trayectoria desarrollada con "GUI" de Matlab. 5.3	88
Figura 45. Caso tipo de obstáculos.	90
Figura 46. Resultados del mejor individuo variando el tamaño de población entre 10 y 20 individuos, además de mantener la tasa de mutación y variar la tasa de cruce.	91
Figura 47. Resultados de la media poblacional variando el tamaño de población entre 10 y 20 individuos,	92
Figura 48. Resultados del mejor individuo variando el tamaño de población y la mutación.	94
Figura 49. Resultados de la media variando el tamaño de población y la tasa de mutación.	95
Figura 50. Resultados del mejor individuo y la media con tasa de cruce 20% y mutación 80%	96
Figura 51. Resultados del mejor individuo y la media con tasa de cruce 50% y mutación 1%.	98
Figura 52. Respuestas del algoritmo a obstáculos tipo.	99
Figura 53. Respuestas del algoritmo a diversas formas de obstáculos.	101
Figura 54. Comportamiento zigzagante para valores de aceleraciones Altos	102
Figura 55. Ejemplo de un camino obtenido mediante la herramienta para planeamiento de trayectoria.	103

	pág.
Figura 56. Grafica que relaciona la media para 500 generaciones con tamaño de población de 100.	104
Figura 57. Grafica que relaciona la media para 500 generaciones con tamaño de población de 50.	105
Figura 58. Resultados para el camino de referencia (diagonal).	105
Figura 59. Resultados para otros caminos de referencia.	107
Figura 60. Resultados para desplazamientos básicos de una celda a otra.	108

LISTA DE ANEXOS

	pág.
Anexo A. Detalles de la implementación del AG para maximizar la función de Ackley.	119

RESUMEN

TÍTULO.

ALGORITMOS GENÉTICOS APLICADOS AL PLANEAMIENTO DE TRAYECTORIAS DE ROBOTS MÓVILES. MODELAMIENTO Y SIMULACIÓN.*

AUTOR.

GONZÁLEZ, Fabio Alfonso.**

PALABRAS CLAVES.

Planeamiento de movimiento, Planeamiento de camino, Planeamiento de trayectoria, Robots Móviles, Algoritmos Genéticos.

DESCRIPCIÓN.

El presente trabajo de investigación tiene como objeto desarrollar una metodología que utilice los Algoritmos Genéticos (AG) como método de búsqueda en el planeamiento de trayectoria de robots móviles, específicamente sobre un tipo de robot con tres ruedas tipo triciclo, el cual se mueve en un ambiente bidimensional con obstáculos fijos.

La primera parte del trabajo consistió en una revisión del estado del arte en cuanto a los tópicos de computación evolutiva y en especial a una de sus técnicas de mas común uso como son los AG, las diferentes aplicaciones, sus fundamentos matemáticos y por último una revisión en cuanto a conceptos de robótica móvil contextualizado al tema de planeamiento de movimiento, además de los aspectos cinemáticos y dinámicos propios de estos sistemas con características no holonómicas.

Para aplicar los AG al problema de planeamiento de movimiento se requirió realizar el estudio del modelo de una arquitectura específica para un robot móvil como es el conocido como tipo triciclo, al cual se le estableció un modelo cinemático directo e inverso, además de un breve estudio en cuanto a su dinámica. En la solución al problema de planeamiento de movimiento se usaron los AG y el trabajo se dividió en dos partes: la referente al planeamiento de camino y la del planeamiento de trayectoria. Para cada una de estas se planteó una codificación, se estableció una función de evaluación y se propusieron los diferentes parámetros para el AG, el cual se sometió a un estudio de tipo exploratorio con el fin de comprobar su validez.

Como resultado del trabajo se realizó una herramienta software con Matlab, usando la GUI y Simulink; este último para el desarrollo y validez de los modelos estudiados.

* Trabajo de Investigación.

** Facultad de Ingenierías Físico-Mecánicas. Maestría en Ingeniería. Grupo CEMOS. E³T. UIS.
Director: MSC. Jaime Barrero.

ABSTRACT

TITLE.

GENETIC ALGORITHMS FOR THE TRAJECTORY PLANNING OF MOBILE ROBOT.
MODELLING AND SIMULATION.*

AUTHOR.

GONZÁLEZ, Fabio Alfonso.**

KEYWORDS.

Motion planning, Path planning, Trajectory planning, Mobile Robot, Genetic Algorithms.

DESCRIPTION.

The main objective in the present work is to develop a methodology that uses Genetic Algorithms (GA) as a search method for the planning of the trajectory of a mobile robot. A three wheels tricycle type robot which moves in a two-dimensional space with fixed obstacles was used.

In the first part of the work a revision of the state of the art for the topics of evolutionary computation was made, especially on the technique more commonly used, the GA. The different applications and mathematical foundation of the GA were analyzed. A review about concepts of mobile robotic to the planning topic on the way, besides the kinematic and dynamic characteristic of these systems with constraints non holonomic, is also included. To apply the GA to the motion planning problem it a study of the pattern of a specific architecture for a mobile robot as it is the type tricycle is required. To accomplish this task a forward and inverse kinematic model, and a brief study about of the dynamic system's was established .

To solve the motion planning problem a methodology that uses the GA as search strategy was divided in two parts, the path planning and the trajectory planning. For each one of these a coding and a fitness function was developed.

The computer code developed was run using the GUI of Matlab and Simulink; the latter for the development and validity of studied models.

*Investigation work

**Faculty of Physical-mechanical engineering's. Master in engineering. Group CEMOS. E³T. UIS.
Director: MSC. Jaime Barrero.

INTRODUCCIÓN

El planeamiento de trayectoria en robots móviles es un problema tan complejo pero a la vez tan interesante, que lleva al investigador al cual le apasione el tema, a indagar nuevos horizontes con el fin de estudiar novedosas propuestas que le permitan llegar a conclusiones interesantes en cuanto a su solución e implementación con elementos reales.

El ser humano siempre ha estado en una constante búsqueda del conocimiento, que lo lleva a explorar caminos que lo conducen hacia tecnologías que a simple vista parecen tener un alto grado de complejidad, pero que en definitiva lo único que hacen es imitar el comportamiento de la naturaleza. Existen gran variedad de algoritmos que son inspirados en sistemas y procesos estudiados en otras áreas o ciencias, que han sido los gestores de nuevas ideas, las cuales en forma *a priori* no son mejores que las que existían antes de la aparición de estas, pero el solo hecho de ser nuevas las posiciona como metodologías de un alto grado de popularidad. El uso de la inteligencia artificial se ha hecho en consecuencia cada vez mas difundido, no queriendo decir con esto que dicha ciencia sea improvisada, por el contrario ha demostrado ser una herramienta eficaz en la solución de problemas de alta dificultad, es decir problemas en los cuales a medida que se aumenta su complejidad su espacio de solución crece en forma exponencial; un ejemplo de lo anterior se encuentra en la robótica, específicamente en lo que se conoce como el problema del planeamiento de camino lo cual tiene un solo fin: lograr robots autónomos. Lo anterior ha provocado que las máquinas que actualmente se fabrican contengan un alto nivel de independencia de sus operadores y que cada vez los investigadores se sientan mas comprometidos en aunar esfuerzos para lograr como resultado final diseños que sean absolutamente autónomos e inteligentes. A pesar de estos esfuerzos los robots realmente inteligentes que actúan y toman decisiones por sí solos, aún se encuentran en los laboratorios operando bajo muchas restricciones y limitaciones mecánicas; es común encontrar robots automáticos preprogramados en una fábrica cumpliendo determinadas tareas las cuales tienen siempre características predecibles.

El actual trabajo muestra una metodología que utiliza los algoritmos genéticos en la solución del problema de planeamiento de movimiento. Es deseable que la presente idea también pueda ser de utilidad para otras líneas de investigación que encuentren en los algoritmos genéticos una herramienta mas para ayudar a encontrar soluciones a sus problemas NP duros o problemas del tipo de optimización combinatoria.

El presente texto se encuentra organizado de la siguiente forma: en el capítulo uno se presentan las generalidades de los Algoritmos genéticos ubicándolos en el contexto de una disciplina de la inteligencia artificial: la computación evolutiva; se

presenta con detalle cada una de sus características y sus diferencias con las otras tendencias en este campo como son las estrategias evolutivas y la programación evolutiva. Al final de este capítulo se presenta un ejemplo con el cual se ilustra la forma de operación de los algoritmos genéticos.

En el capítulo dos se presentan los principios matemáticos bajo los cuales operan los algoritmos genéticos desde un punto de vista lo mas estricto posible según las tendencias en el desarrollo de una teoría consistente que los justifica.

Para el capítulo tres se hace un desarrollo acerca del problema de planeamiento de movimiento introduciendo las definiciones básicas de muy común uso en la literatura especializada del tema, así como las notaciones comúnmente utilizadas para ello. Se presenta también la complejidad que aborda este problema describiendo en forma breve las distintas aproximaciones que se dan para su solución. Al final se describe lo que se entiende comúnmente como un robot móvil y algunos ejemplos de los distintos modelos desarrollados en este campo.

En el capítulo cuatro se presenta el modelo del robot, desarrollando las ecuaciones que lo modelan además de las consideraciones hechas acerca del ambiente de trabajo y la forma de modelar los obstáculos.

El capítulo cinco es una descripción de la metodología utilizada para resolver el problema de planeamiento de movimiento utilizando los algoritmos genéticos. Se hacen dos desarrollos, uno enfocado exclusivamente al planeamiento de camino en el cual el resultado que se desea obtener es un camino libre de colisiones. En la segunda parte se explica una metodología para la aplicación de los algoritmos genéticos en la obtención de las velocidades para cumplir con el camino planeado.

Por último, en el capítulo seis se muestran los resultados obtenidos y las propuestas que se hacen para trabajos futuros en este campo.

1. ALGORITMOS GENÉTICOS. GENERALIDADES

1.1 INTRODUCCIÓN

Los algoritmos genéticos(**AG**) son métodos adaptativos de exploración, los cuáles pueden ser usados en problemas de búsqueda y optimización; están basados en la mecánica de la selección y la genética natural las cuales rigen el comportamiento de los seres vivos. En las distintas poblaciones de especies que habitan la naturaleza, cada generación está sometida a un proceso de selección natural, en el cual el más adaptado tiene la mayor probabilidad de sobrevivir, según mencionan diversidad de artículos referenciando lo divulgado en 1859 por el naturalista británico Charles Darwin a través de un libro que cambiaría la manera de ver el mundo por los humanos: “*El origen de las especies por medio de la selección natural*”. En este libro Darwin propuso que los humanos, y de hecho todas las criaturas, no fueron puestas por Dios y hechas inmutables, por el contrario ellas evolucionaron a partir de otras criaturas. Para su época, la idea parecía prepóstera, pero el tiempo ha demostrado que él estaba en lo cierto, lo cual se demuestra en forma sencilla por los adelantos a los que ha llegado la tecnología pues actualmente es posible leer el ADN del ser humano y de otras criaturas, demostrando a través de ello que solo nos diferenciamos de otros organismos en la forma en que pensamos.

Con el transcurrir del tiempo, las criaturas cambian para adaptarse a su ambiente, sobrevivir y prosperar. Uno de los ejemplos más llamativos de esto son las Islas de Galápagos. Esta serie de islas es uno de los ejemplos más prominentes de evolución, selección natural y adaptación. El proceso que allí vivieron las especies demostró que los individuos no cambian por sí mismos, pero aquellos que mejor sobreviven, o tienen una aptitud más alta, sobrevivirán más tiempo y producirán más descendencia. Esto continúa ocurriendo con los individuos que se vuelven más aptos a su ambiente en cada generación. Fue esa mejora continua la que a los científicos de los computadores inspiró, uno de ellos el más prominente John Holland, a crear los AG¹. Un AG, se dice que está en la capacidad de imitar este proceso si está adecuadamente codificado. En la naturaleza, cada individuo de una determinada población, compite entonces con sus semejantes por recursos tales como alimento, agua y refugio. También compiten por reproducirse. Aquellos que tengan mas éxito en cuanto a sobrevivir y reproducirse tendrán un número relativamente mayor de descendientes; y por el contrario los que tengan bajo desempeño, tendrán pocos o ningún descendiente. Lo anterior visto desde el punto genético significa que los genes de los individuos mas adaptados se propagarán hacia un número creciente de individuos en cada generación sucesiva. La combinación de distintas características de los ancestros pueden producir en

¹ GOLDBERG, David. Genetic Algorithms in search, Optimization and Machine learning. Boston. Addison-Wesley, 2000. p. 2

algunas ocasiones descendientes “superadaptados”, con una mejor acomodación que cualquiera de los padres. De esta forma las especies evolucionan para llegar a estar mas adaptadas a su ambiente. Los AG usan una analogía directa del comportamiento natural. Trabajan con una población de “individuos”, donde cada uno representa una solución posible a determinado problema. A cada individuo se le asigna un puntaje asociado a su “adaptación”, según sea su proximidad a la solución del problema. Esto hace analogía en la naturaleza, a qué tan eficiente es un individuo en la competencia por los recursos. A estos “individuos”, se les da entonces la oportunidad de reproducirse con otros miembros de la población y sus descendientes comparten la información genética de sus dos padres. Los miembros menos adaptados de la población tienen por consiguiente menos probabilidad de reproducirse y entonces morirán mas rápidamente. La nueva generación obtenida contiene una proporción mas alta de las características que poseían los anteriores miembros mas adaptados de la población inicial². De esta forma en cada generación se espera que las buenas características que tiene cada individuo sean esparcidas mediante el intercambio y mezcla de ellas con las de otros individuos. De lo anterior se puede entonces concluir que los AG no son mas que una metáfora de un proceso natural como lo son también las redes neuronales, la lógica *fuzzy*, el recocido simulado (*simulated annealing*), la teoría del caos entre otros. Pero todas estas disciplinas se agrupan alrededor de un paradigma que ha venido mostrando gran auge y aceptación, como es la inteligencia artificial(IA); dentro de esta área aparecen los AG como una particularidad de lo que actualmente se conoce como computación evolutiva(CE). Por esta razón se hace necesario ubicar el estudio de los AG desde esta perspectiva que permita así compararlos con otras líneas como son las estrategias evolutivas (EE) y la programación evolutiva(PE). Este tema y un cotejo entre las diversas tendencias y los AG se presentan a continuación, además de una contextualización desde el punto de vista biológico para posteriormente contrastar con los modelos tomados en los patrones computacionales. Al final del capítulo se muestran todos los elementos que componen a un AG simple, de gran importancia para el presente trabajo, acompañándolo de un ejemplo con el fin de clarificar las ideas.

1.2 COMPUTACIÓN EVOLUTIVA

Para poder comprender con claridad este tópico sería importante establecer un marco histórico que muestre los pasos que ha seguido la comunidad científica para llegar hasta donde actualmente se encuentra en cuanto a técnicas heurísticas se refiere.* Con la limitante antes mencionada se procura mencionar acá algunos aspectos históricos, algo puntuales que permiten entender en parte

² BEASLEY, David. An overview of Genetic algorithms. En: Artículo de la IEEE, 1993; p.1

* Debido al poco espacio que aquí se tiene se pide al interesado que consulte las lecciones del curso “Computación Evolutiva” de Carlos Coello en el Laboratorio Nacional de Informática (lania) perteneciente al centro de investigación y estudios avanzados del Instituto Politécnico Nacional de México.

porque el pensamiento evolutivo actual gira en torno al Neo-Darwinismo, el cual establece que toda la vida en el planeta puede ser explicada a través de los siguientes procesos:

- ◆ Reproducción.
- ◆ Mutación.
- ◆ Competencia.
- ◆ Selección.

Estos elementos se convierten en el eje sobre el cual giran las tendencias de la computación evolutiva la cual desde 1930 de la mano de la evolución natural ha sido vista como un proceso de aprendizaje, específicamente con el trabajo de W.D. Cannon (*The Wisdom of the Body*). En 1950 el célebre matemático Alan Mathison Turing reconoció también una conexión "obvia" entre la evolución y el aprendizaje de máquina. A comienzos de 1960, el biólogo Alexander S. Fraser publicó una serie de trabajos sobre la evolución de sistemas biológicos sobre una computadora digital, dando la inspiración para lo que después se convertiría en el *algoritmo genético*. Aproximadamente en la misma época de Fraser, el estadístico inglés George E. P. Box propuso un enfoque evolutivo para la optimización de la producción industrial. R. M. Friedberg fue uno de los primeros científicos en intentar evolucionar programas de computadora (final de 1950). Sus experimentos no fueron muy exitosos, y originó una avalancha de críticas de parte de los investigadores de la inteligencia artificial clásica. G. J. Friedman fue tal vez el primero en proponer una aplicación de las técnicas evolutivas a la robótica: en su tesis de maestría que data de 1950, propuso evolucionar una serie de circuitos de control similares a las redes neuronales tal como se entienden hoy en día. Nils Aall Barricelli desarrolló las que probablemente fueron las primeras simulaciones de un sistema evolutivo en una computadora digital, entre 1953 y 1956. Sus experimentos siguieron los lineamientos de una disciplina bautizada a principios de 1980 como "vida artificial". Hans J. Bremermann fue tal vez el primero en ver la evolución como un proceso de optimización, además de realizar una de las primeras simulaciones con cadenas binarias que se procesaban por medio de reproducción (sexual o asexual), selección y mutación, en lo que sería otro claro predecesor del algoritmo genético.

Lawrence J. Fogel concibió el uso de la evolución simulada en la solución de problemas (sobre todo de predicción) hacia mediados de 1960. Su técnica se denominó "Programación Evolutiva". Peter Bienert, Ingo Rechenberg y Hans-Paul Schwefel desarrollaron hacia mediados de 1960 un método de ajustes discretos aleatorios inspirado en el mecanismo de mutación que ocurre en la naturaleza. Su técnica se denominó "estrategias evolutivas". John H. Holland desarrolló a principios de 1960 los "planes reproductivos" y "adaptativos" en un intento por hacer que las computadoras aprendan imitando el proceso de la evolución. Esta técnica sería después conocida mundialmente como el "algoritmo genético". Michael Conrad y H. H. Pattee se cuentan entre los primeros en simular un

ecosistema artificial jerárquico en el que un conjunto de organismos unicelulares estaban sujetos a una estricta ley de conservación de la materia que les inducía a competir por sobrevivir. Conrad propuso también en 1970 un "modelo de circuitos de aprendizaje evolutivo" en el cual especuló sobre la posibilidad de que el cerebro use el mismo tipo de mecanismos que usa la evolución para aprender. Su técnica fue uno de los primeros intentos por utilizar algoritmos evolutivos para entrenar redes neuronales.

Aunque los primeros intentos por evolucionar programas se remontan a 1950 y 1960, fue hasta 1980 en que se obtuvieron resultados satisfactorios. Hicklin (1986) y Fujiki (1986) usaron expresiones S en LISP para representar programas cuyo objetivo era resolver problemas de teoría de juegos. Nichal Lynn Cramer (1985) y posteriormente, John R. Koza (1989) propusieron (de forma independiente) el uso de una representación de árbol en la que se implementó un operador de cruce para intercambiar subárboles entre los diferentes programas de una población generada al azar (con ciertas restricciones impuestas por la sintaxis del lenguaje de programación utilizado).

La propuesta de Koza fue la que se impuso a la larga, y más tarde se denominó "Programación Genética", la cual hoy en día es muy popular y cuenta con una amplia gama de aplicaciones. Thomas S. Ray desarrolló a principios de 1990 un simulador muy original en el que evolucionaban programas en lenguaje ensamblador, los cuales competían por ciclos de CPU de una computadora, a la vez que intentaban reproducirse (o sea, copiarse a sí mismos) en la memoria de dicha computadora.

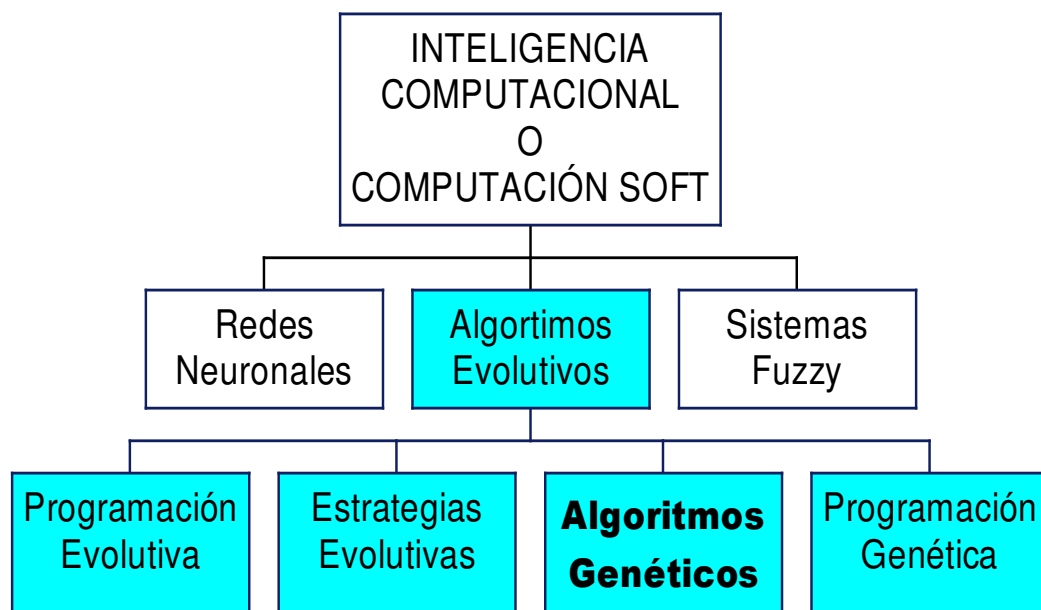
En el simulador de Ray, denominado "Tierra", se partía de un programa único con la capacidad de auto-replicarse, al que se denominaba "ancestro". Con base en este programa se generaban "criaturas" nuevas (segmentos de código), las cuales a su vez se podían subdividir para dar origen a nuevas criaturas. Tierra es uno de los pocos intentos por simular un ecosistema con el propósito expreso de observar los comportamientos que emergen de la dinámica evolutiva del mismo.

Como se puede deducir de esta breve reseña histórica tres son los paradigmas principales que conforman la computación evolutiva: Programación evolutiva, Estrategias Evolutivas y Algoritmos Genéticos. Una taxonomía de la computación *Soft*, también llamada por algunos autores inteligencia computacional, la cual alberga la CE se muestra en la figura 1.

A continuación se muestra un paralelo entre las diversas tendencias de la CE. La programación evolutiva nace con D. Fogel en 1960, quién define el objetivo de esta disciplina como el de lograr un comportamiento inteligente a partir de la simulación de la evolución. Inicialmente fue propuesta para operar sobre máquinas de estado finitas y las correspondientes representaciones discretas, pero actualmente ha tendido hacia problemas de optimización de parámetros continuos.

Los genes codifican un autómata finito o máquina de estados. Cada gen puede consistir en una terna del tipo "estado inicial + cadena de transiciones + estado final". En definitiva, cada gen es una regla que especifica las condiciones (cadena de transiciones) que se deben cumplir para llegar a un estado (estado final), suponiendo que la máquina se encuentra en un cierto estado (estado inicial). La cadena de transiciones corresponde con los sentidos o sensores del autómata (el suceso), y los estados corresponden con las acciones o actuadores. Esta disciplina enfatiza los nexos de comportamiento entre padres e hijos, en vez de buscar emular operadores genéticos específicos.

Figura 1. Taxonomía de la Inteligencia computacional.



Fuente: Autor del proyecto.

El algoritmo básico de la programación evolutiva es el siguiente:

1. Generar aleatoriamente una población inicial.
2. Aplicar el operador mutación.
3. Calcular la aptitud de cada hijo y usar un proceso de selección mediante el torneo (normalmente estocástico) para determinar cuáles serán las soluciones que se retendrán.
4. Volver al segundo paso, si no se ha cumplido el criterio de parada.

Algunas aplicaciones de la programación evolutiva son:

- ♦ Predicción.
- ♦ Generalización.

- ♦ Juegos.
- ♦ Control automático.
- ♦ Problema del viajero.
- ♦ Planeación de rutas.
- ♦ Diseño y entrenamiento de redes neuronales.
- ♦ Reconocimiento de patrones.

Las EE fueron desarrolladas en 1964 en Alemania para resolver problemas hidrodinámicos de alto grado de complejidad por un grupo de estudiantes de ingeniería encabezado por Ingo Rechenberg. Para describir las EE se usa la siguiente nomenclatura: $(\mu+\lambda)$ para el reemplazo determinista por inclusión; (μ,λ) para el reemplazo por inserción con μ como tamaño de la población y λ para el tamaño de la descendencia. La notación se refiere entonces al mecanismo de selección utilizado. En el primer caso, los μ mejores individuos obtenidos de la unión de padres e hijos sobreviven. En el segundo caso, sólo los μ mejores hijos de la siguiente generación sobreviven. La versión original (1+1)-EE usaba un solo padre y con este se generaba un solo hijo. Este hijo se mantenía si era mejor que el padre, o de lo contrario se eliminaba (a este tipo de selección se le llama *extintiva*, porque los peores individuos obtienen una probabilidad de ser seleccionado de cero). Rechenberg (1973) introdujo el concepto de población, al proponer una EE llamada $(\mu+1)$ -EE, en la cual hay μ padres y se genera un solo hijo, el cual puede reemplazar al peor padre de la población (selección extintiva). Schwefel (1975) introdujo el uso de múltiples hijos en las denominadas $(\mu+\lambda)$ -EEs y (μ, λ) -EEs. Rechenberg formuló una regla para ajustar la desviación estándar de forma determinística durante el proceso evolutivo, de tal manera, que el procedimiento convergiera hacia el óptimo. Esta regla se conoce como la "regla del éxito 1/5", y en palabras dice: "La razón entre mutaciones exitosas y el total de mutaciones debe ser 1/5. Si es superior, entonces debe incrementarse la desviación estándar y si es menor entonces disminuirse". Thomas Bäck (1996) derivó una regla de 1/7 para las (μ,λ) -EEs. Un aspecto característico de las EE es que las tasas de mutación de cada gen se codifican como parte del genoma, por lo que también se encuentran sujetas a evolución (a esto se le llama "auto-adaptación").

Los operadores de recombinación de las estrategias evolutivas pueden ser: a) *Sexuales*: el operador actúa sobre dos individuos elegidos aleatoriamente de la población de padres. b) *Panmíticos*: se elige un solo padre al azar, y se mantiene fijo mientras se elige al azar un segundo padre (de entre toda la población). Las EE simulan el proceso evolutivo al nivel de los individuos, por lo que la recombinación es posible. Asimismo, usan normalmente selección determinística. Aunque es difícil establecer un algoritmo básico para las EE ya que el estudio al que han sido sometidas en su desarrollo ha generado diversos cambios en su estructura, se puede plantear entonces a modo de ilustración el siguiente:

1. Establecer población inicial, generalmente un individuo.
2. Aplicar el operador de mutación.
3. Ejecutar el operador de cruce, si este se contempla.
4. Aplicar el proceso de selección, comparando los valores de adaptación de la función de evaluación de la descendencia con el de los padres y proceder a tomar los mejores.
5. Regresar al segundo paso si no se ha cumplido criterio de parada.

Algunas aplicaciones de las estrategias evolutivas son:

- ◆ Problemas de enrutamiento y redes.
- ◆ Bioquímica.
- ◆ Óptica.
- ◆ Diseño en ingeniería.
- ◆ Magnetismo.

Los AG (denominados originalmente "planes reproductivos") fueron desarrollados por John Holland a principios de 1960. Su motivación principal fue el aprendizaje de máquinas. El AG enfatiza la importancia del cruce sexual (operador principal) sobre el de mutación (operador secundario), y usa selección probabilística. La representación tradicional es la cadena binaria. A la cadena se le llama "cromosoma"; a cada posición de la cadena se le denomina "gen" y al valor dentro de esta posición se le llama "alelo". En este aparte no se hará mucho énfasis en esta terminología, más adelante se encuentra una explicación con mayor profundidad. Para poder aplicar el AG se requiere de cinco componentes básicos: una representación de las soluciones potenciales del problema; una forma de crear una población inicial de posibles soluciones (normalmente un proceso aleatorio); una función de evaluación que juegue el papel del ambiente con la cual se puedan clasificar las soluciones en términos de su "aptitud" (*fitness*); operadores genéticos que alteren la composición de los hijos que se producirán para las siguientes generaciones y valores para los diferentes parámetros que utiliza en sí el AG (tamaño de la población, probabilidad de cruce, probabilidad de mutación, número máximo de generaciones, etc.).³

El algoritmo básico para el AG es el siguiente:

1. Generar (aleatoriamente) una población inicial.
2. Calcular aptitud de cada individuo.
3. Seleccionar (probabilísticamente) con base en la aptitud individual.
4. Aplicar operadores genéticos (cruce y mutación) para generar la siguiente población.
5. Evaluar condición de parada, si no cumple regresar al segundo paso.

Algunas aplicaciones de los AG son las siguientes:

³ COELLO, Carlos. Apuntes del curso: "Computación Evolutiva". <http://delta.cs.cinvestav.mx/~ccoello>

- ◆ Optimización (estructural, de topologías, numérica, combinatoria, etc.)
- ◆ Aprendizaje de máquina (sistemas clasificadores).
- ◆ Bases de datos (optimización de consultas).
- ◆ Reconocimiento de patrones (por ejemplo, imágenes).
- ◆ Generación de gramáticas (regulares, libres de contexto, etc.).
- ◆ Planeación de movimientos de robots.
- ◆ Predicción.
- ◆ Problema del viajero.

La tabla 1 muestra un cuadro comparativo entre las tres tendencias mencionadas haciendo énfasis en las características que las clasifican como algoritmos evolutivos.

1.3 PROBLEMAS DE OPTIMIZACIÓN COMBINATORIA

El objetivo principal de cualquier técnica de optimización es encontrar el óptimo (o los óptimos) globales de cualquier problema. En matemáticas, existe un área que se ocupa de desarrollar los formalismos que permiten garantizar la convergencia de un método hacia el óptimo global de un problema (aunque efectivamente el término óptimo es un superlativo, en la literatura es común encontrar algunos calificativos para la palabra óptimo, debido al hecho que ciertos métodos catalogados de optimización fracasan en su tarea y obtienen respuestas aproximadas). Desafortunadamente, sólo en algunos casos limitados, puede garantizarse convergencia hacia el óptimo global.

Por ejemplo, para problemas con espacios de búsqueda convexos, las condiciones de Kuhn-Tucker⁴ son necesarias y suficientes para garantizar optimalidad global de un punto. En problemas de optimización no lineal, las condiciones de Kuhn-Tucker no son suficientes para garantizar optimalidad global. De hecho, todas las técnicas usadas para optimización no lineal pueden localizar al menos óptimos locales, pero no puede garantizarse convergencia al óptimo global a menos que se usen técnicas exhaustivas o que se consideren tiempos infinitos de convergencia. Existe una clase especial de problemas, en los cuales las variables de decisión son discretas y las soluciones suelen presentarse en forma de permutaciones. A estos problemas se les denomina de "optimización combinatoria". Un problema clásico en esta área ha sido el problema del vendedor viajero, este consiste en encontrar una permutación que represente el recorrido de una serie de ciudades de tal forma que todas sean visitadas minimizando la distancia total viajada. Este problema tiene la característica de tener un espacio de búsqueda de: $(n-1)!$, con lo cual se cataloga como un problema NP-duro.

⁴ GASS I. Saul. Linear Programming. Methods and applications. p.305

Tabla 1. Comparativo de las tres tendencias.

CARACTERÍSTICA	ALGORITMOS GENÉTICOS	ESTRATEGIAS EVOLUTIVAS	PROGRAMACIÓN EVOLUTIVA
Codificación	Sistema binario, típicamente.	Números reales	Números reales
Función de evaluación	Valor natural de la función objetivo.	Valor ajustado de la función objetivo.	Valor ajustado de la función objetivo.
Cruce	N puntos o uniforme. Principal operador.	Usado en forma discreta o intermedia.	No utilizado.
Mutación	Inversión de bits. Operador secundario.	Gaussiana. Principal operador.	Gaussiana. Único operador
Selección padres	Probabilística y proporcional a la aptitud.	Aleatoria uniforme	Determinística.
Selección supervivientes (descendencia)	Probabilística. Basada en la preservación. Todos los hijos reemplazan a los padres. (cuando no se usa elitismo)	Determinística. Extintiva basada en la preservación. Los hijos reemplazan a los padres, según su valor de aptitud.	Probabilística. Extintiva. Los hijos reemplazan a los padres, según su valor de aptitud.
Auto-adaptación	Ninguna.	En la desviación estándar.	Ninguna.
Soporte teórico	Teorema de los esquemas. Convergencia global para uso de elitismo.	Convergencia global para $(\mu+\lambda)$ -EEs.	Convergencia global para lo que se denomina meta-PE.
Efecto de la evolución	Genotipo.	Fenotipo.	Fenotipo.

Fuente: Autor del proyecto.

Diversos problemas de la naturaleza en mención se clasifican según el tiempo que requieran para ser resueltos en los dos siguiente tipos:

- P, si puede ser resuelto en tiempo polinomial mediante un algoritmo determinístico.
- NP, si puede ser resuelto en tiempo polinomial pero usando un algoritmo no determinístico.⁵

Existen problemas que no pueden resolverse usando un algoritmo que requiere tiempo polinomial. De hecho, en muchas aplicaciones prácticas, no es posible

⁵ GILL Mark –Zomaya Albert. Obstacle Avoidance in multirobot systems. p. 52

afirmar si existe una solución eficiente. Hay muchos problemas para los cuales el mejor algoritmo que se conoce requiere tiempo exponencial, este es el caso del problema del vendedor viajero. La tabla 2 ilustra su dificultad, suponiendo para dicho ejemplo un computador capaz de procesar $1 \cdot 10^6$ soluciones por segundo.

Para resolver un problema pueden existir varios algoritmos. Por tanto, es lógico elegir el “mejor”. Si el problema es sencillo o no hay que resolver muchos casos se podría elegir el más “fácil”. Si el problema es complejo o existen muchos casos habría que elegir el algoritmo que menos recursos utilice (tiempo de ejecución y espacio de almacenamiento). Generalmente, el más importante es el tiempo. Al hablar de la eficiencia de un algoritmo se refiere a lo “rápido” que se ejecuta. La eficiencia de un algoritmo dependerá, en general, del “tamaño” de los datos de entrada. A la hora de analizar un algoritmo es de interés fundamental, la forma en que este se comporta al aumentar el tamaño de los datos; es decir, cómo se incrementa su tiempo de ejecución. Esto se conoce como eficiencia asintótica de un algoritmo y permite comparar las distintas posibilidades que se den.

Tabla 2. Aplicación del problema del vendedor viajero.

CIUDADES	ESPACIO DE SOLUCIONES	TIEMPO	UNIDAD
4	6	0,000006	segundos
5	24	0,000024	segundos
6	120	0,00012	segundos
7	720	0,00072	segundos
8	5040	0,00504	segundos
9	40320	0,04032	segundos
10	362880	0,36288	segundos
11	3628800	3,6288	segundos
12	39916800	39,9168	segundos
13	479001600	7,98336	minutos
14	6227020800	1,729728	horas
15	8,7178E+10	1,009008	días
16	1,3077E+12	15,13512	días
17	2,0923E+13	242,16192	días
18	3,5569E+14	11,2787744	años
19	6,4024E+15	2,03017938	siglos
20	1,2165E+17	38,5734083	siglos
21	2,4329E+18	771,468166	siglos
22	5,1091E+19	16200,8315	siglos

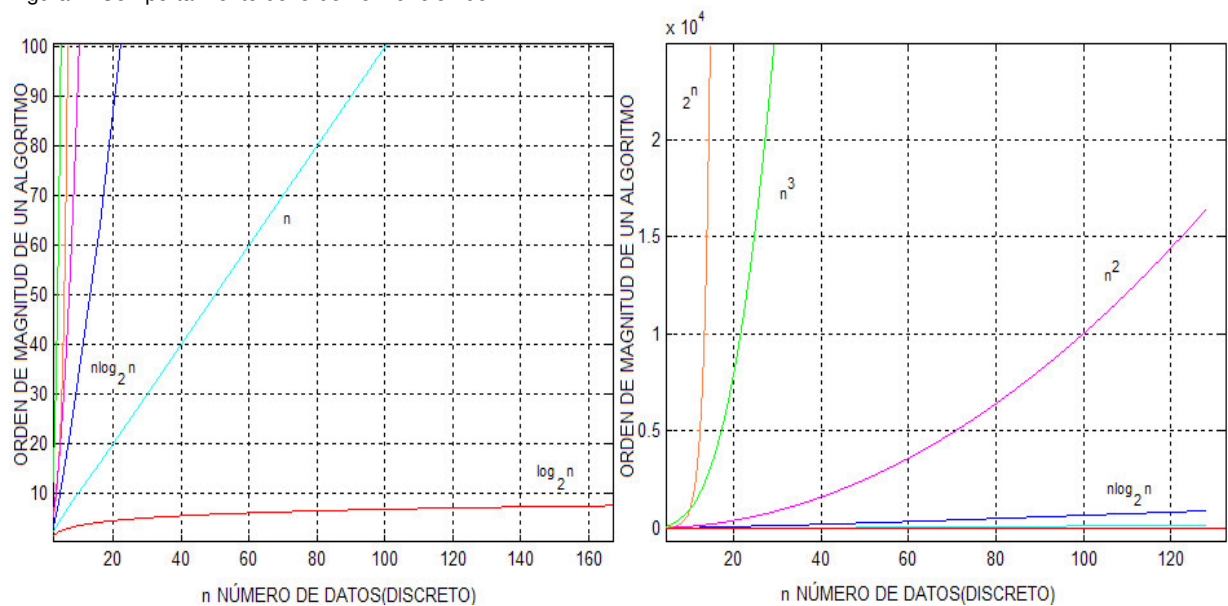
Fuente: COELLO, Apuntes del curso: “Computación Evolutiva”. <http://delta.cs.cinvestav.mx/~ccoello>

La notación asintótica se describe por medio de una función cuyo dominio es el conjunto de números naturales, N , y se denota por la letra O . Si un algoritmo tiene

complejidad $O(g(n))$ significa que al correrlo en una computadora con la misma información, pero con valores incrementales de n (número de datos discretos), los tiempos resultantes de ejecución serán siempre menores que $|g(n)|$. La figura 2 ilustra este hecho.

Los AG son una herramienta que se han utilizado con éxito de estos tipos de problemas de optimización combinatoria, por lo que se hace necesario ubicarlos dentro de las demás técnicas de búsqueda existentes como se muestra en la figura 3. Un problema que cumpla con lo anterior debe cumplir también con las siguientes consideraciones: el espacio de búsqueda (sus posibles soluciones) deben estar delimitadas dentro de un cierto rango el cual debe ser conocido; debe poderse definir una función de *aptitud* que indique qué tan buena o mala es una cierta respuesta; las soluciones deben codificarse de una forma que resulte relativamente fácil de implementar en la computadora.

Figura 2. Comportamiento del orden en función de n .



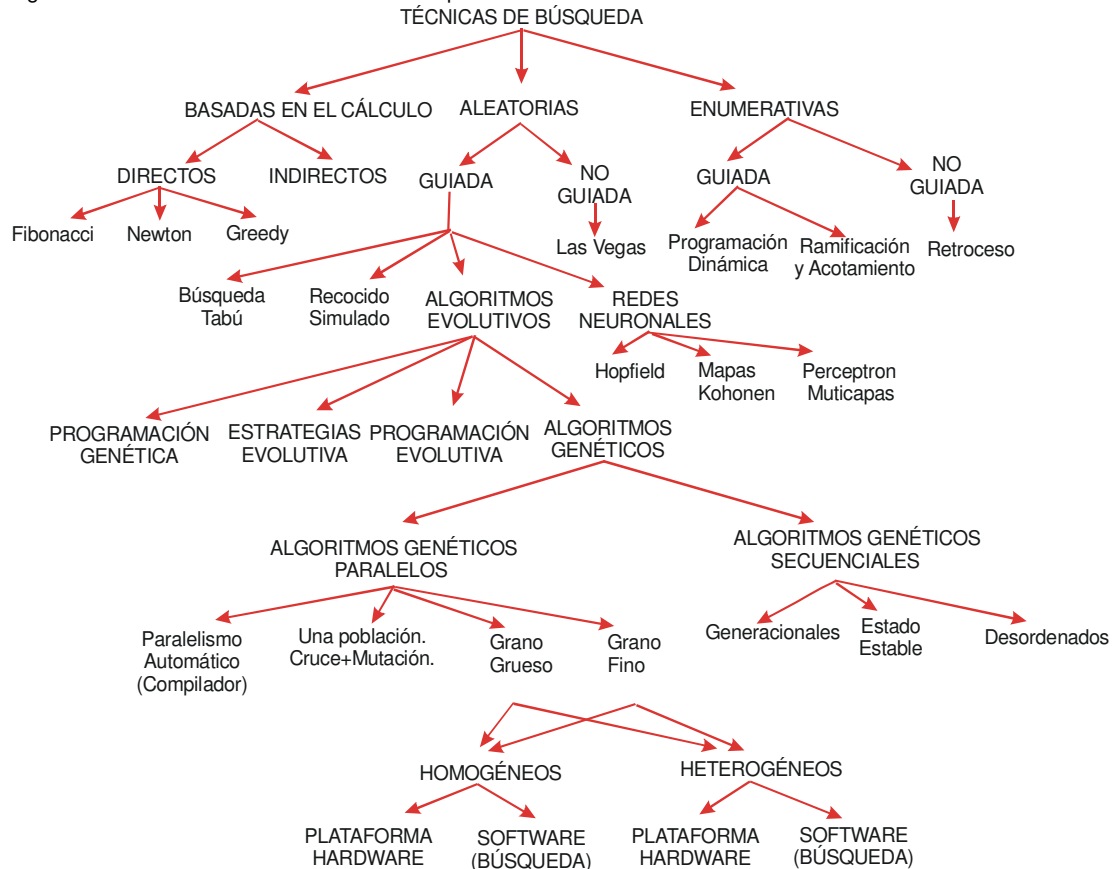
Fuente: Autor del proyecto.

El primer punto es muy importante, y lo más recomendable es intentar resolver problemas que tengan espacios de búsqueda discretos, aunque éstos sean muy grandes. Sin embargo, también se puede intentar usar la técnica con espacios de búsqueda continuos, pero preferentemente cuando exista un rango de soluciones relativamente pequeño⁶. El buen desempeño que tenga un AG depende de diversos factores por lo que es preponderante conocer cuando se pueden aplicar o

⁶ FORREST, Stephanie and MITCHELL, Melanie. ¿What makes a problem hard for a genetic algorithm?. Some anomalous results and their explanation. Machine Learning:1993, p.285-319.

no. Problemas con diversos parámetros o características que requieran ser combinadas en busca de la mejor solución, problemas con muchas restricciones o condiciones que no pueden ser representadas matemáticamente y problemas con grandes espacios de búsqueda, serían los de considerar para el uso de un AG, pues de lo contrario los métodos tradicionales serán mas eficientes.

Figura 3. Clasificación de las técnicas de búsqueda.



Fuente: Autor del proyecto.

1.4 DEFINICIONES BÁSICAS

1.4.1 Conceptos Biológicos.

Dadas las características sobre las cuales basan su modus operandi los AG, se hace necesario revisar algunos conceptos generales desde la óptica de la Biología. El ácido desoxirribonucleico (ADN) es el material genético fundamental de todos los organismos vivos. El ADN es una macro-molécula doblemente trenzada que tiene una estructura helicoidal. Ambos filamentos trenzados son moléculas de ácido nucleico lineales y sin ramificaciones, formadas de moléculas alternadas de desoxirribosa (azúcar) y fosfato. Un gen es una sección de ADN que

codifica una cierta función bioquímica definida, usualmente la producción de una proteína. Es fundamentalmente una unidad de herencia. El ADN de un organismo

puede contener desde una docena de genes, como un virus, hasta decenas de miles, como los humanos. Se denomina cromosoma a una de las cadenas de ADN que se encuentra en el núcleo de las células. Los cromosomas son responsables de la transmisión de información genética. Cada gen es capaz de ocupar sólo una región en particular de un cromosoma (su "lugar" o "locus"). En cada determinado lugar pueden existir, en la población, formas alternativas del gen. A estas formas alternativas se les llaman alelos. Se llama Genoma a la colección total de genes (y por tanto, cromosomas) que posee un organismo. Se denomina individuo a un solo miembro de una población. Se denomina población a un grupo de individuos que pueden interactuar juntos, por ejemplo, para reproducirse. Se denomina fenotipo a los rasgos (observables) específicos de un individuo. Se denomina genotipo a la composición genética de un organismo (la información contenida en el genoma). Es decir, es lo que potencialmente puede llegar a ser un individuo. El genotipo da origen, tras el desarrollo fetal y posterior, al fenotipo del organismo.

Se denominan gametos a las células que llevan información genética de los padres con el propósito de efectuar reproducción sexual. Se denomina haploide a la célula que contiene un solo cromosoma o conjunto de cromosomas, cada uno de los cuales consta de una sola secuencia de genes, un ejemplo de esto es un gameto. A las células que contienen dos copias de cada cromosoma se denominan diploides. Las copias son homólogas, es decir, contienen los mismos genes en la misma secuencia. En muchas especies que se reproducen sexualmente, los genes en uno de los conjuntos de cromosomas de una célula diploide se heredan del gameto del padre, mientras que los genes del otro conjunto son del gameto de la madre. En la naturaleza, la mayoría de las especies capaces de reproducirse sexualmente son diploides. Durante la reproducción sexual ocurre la recombinación (o cruce), para el caso haploide se intercambian los genes entre los cromosomas (que a su vez son haploides). En el caso diploide, en cada padre se intercambian los genes entre cada par de cromosomas para formar un gameto, y posteriormente los gametos de los dos padres se aparean para formar un solo conjunto de cromosomas diploides. Se denomina reproducción a la creación de un nuevo individuo a partir de: a) Dos progenitores (sexual) y b) un progenitor (asexual)

Durante la mutación, se cambian nucleótidos individuales de padre a hijo. La mayoría de estos cambios se producen por errores de copiado. La aptitud de un individuo se define como la probabilidad de que éste viva para reproducirse (viabilidad), o como una función del número de descendientes que éste tiene (fertilidad). Se denomina ambiente a todo aquello que rodea a un organismo, puede ser "físico" (abiótico) o biótico. En ambos casos, el organismo ocupa un nicho que ejerce una influencia sobre su aptitud dentro del ambiente total. Un ambiente biótico puede presentar funciones de aptitud dependientes de la

frecuencia dentro de una población. En otras palabras, la aptitud del comportamiento de un organismo puede depender de cuántos más estén comportándose igual. A través de varias generaciones, los ambientes bióticos pueden fomentar la co-evolución, en la cual la aptitud se determina mediante la selección parcial de otras especies.

La selección es el proceso mediante el cual algunos individuos en una población son escogidos para reproducirse, típicamente con base en su aptitud. La selección dura se da cuando sólo los mejores individuos se mantienen para generar progenie futura. La selección blanda se da cuando se usan mecanismos basados en probabilidad para mantener como padres a individuos que tengan aptitudes relativamente bajas. Se llama pleiotropía al efecto en el cual un solo gen puede afectar simultáneamente a varios rasgos fenotípicos. Un ejemplo es un problema con la célula responsable de formar la hemoglobina, que al fallar afecta la circulación sanguínea, las funciones del hígado y las acciones capilares. Cuando una sola característica fenotípica de un individuo puede ser determinada mediante la interacción simultánea de varios genes se denomina poligenia. El color del cabello y de la piel son generalmente rasgos poligénicos.

Aunque no existe una definición universalmente aceptada de especie, se dice que es una colección de criaturas vivientes que tienen características similares, y que se pueden cruzar entre sí. Se denomina especiación al proceso mediante el cual aparece una especie. La causa más común de especiación es el aislamiento geográfico. Si una subpoblación de una cierta especie se separa geográficamente de la población principal durante un tiempo suficientemente largo, sus genes divergirán. Estas divergencias se deben a diferencias en la presión de selección en diferentes lugares, o al fenómeno conocido como desvío genético. Se llama desvío genético a los cambios en las frecuencias de genes/alelos en una población con el paso de muchas generaciones, como resultado del azar en vez de la selección.

El desvío genético ocurre más rápidamente en poblaciones pequeñas y su mayor peligro es que puede conducir a que algunos alelos se extingan, reduciendo en consecuencia la variabilidad de la población. En los ecosistemas naturales, hay muchas formas diferentes en las que los animales pueden sobrevivir (en los árboles, de la cacería, en la tierra, etc.) y cada estrategia de supervivencia es conocida como "nicho ecológico". Dos especies que ocupan nichos diferentes (por ejemplo, una que se alimenta de plantas y otra que se alimenta de insectos) pueden coexistir sin competir, de una manera estable. Sin embargo, si dos especies que ocupan el mismo nicho se llevan a la misma zona, habrá competencia, y a la larga, la especie más débil se extinguirá (localmente). Por lo tanto, la diversidad de las especies depende de que ocupen una diversidad de nichos (o de que estén separadas geográficamente). Se denomina migración a la transferencia de (los genes de) un individuo de una subpoblación a otra. Se dice que un gen es epistático cuando su presencia suprime el efecto de un gen que se

encuentra en otra posición. Los genes epistáticos son llamados algunas veces genes de inhibición por el efecto que producen sobre otros.

1.4.2 Conceptos de Computación Evolutiva.

Con el fin de comprender la aplicación del modelo natural de evolución a un sistema de cómputo que lo pueda simular, es práctico plantear las definiciones de conceptos que llevan el mismo nombre. Es casi imposible hacer un mapeo único entre los conceptos biológicos y los de computación evolutiva, por lo que se podrán encontrar elementos de computación evolutiva que no tienen un equivalente biológico y viceversa.

Se denomina cromosoma a una estructura de datos que contiene una cadena de parámetros de diseño o genes. Esta estructura de datos puede almacenarse, por ejemplo, como una cadena de bits o un arreglo de enteros(punto flotante). Se llama gen a una subsección de un cromosoma que, usualmente, codifica el valor de un solo parámetro. El término genotipo se refiere a la codificación de los parámetros que representan una solución del problema a resolverse; el término fenotipo está asociado a la decodificación del cromosoma, es decir, a los valores obtenidos al pasar de la representación, por ejemplo binaria a la usada por la función objetivo. Se llama individuo a un solo miembro de la población de soluciones potenciales a un problema. El alelo es cada valor posible que puede adquirir una cierta posición genética. Si se usa representación binaria, un alelo puede valer 0 ó 1. Cada individuo contiene un cromosoma (o de manera más general, un genoma) que representa una solución posible al problema a resolverse.

La aptitud es el valor que se asigna a cada individuo y que indica qué tan bueno es éste con respecto a las demás para la solución de un problema. El término paisaje de aptitud (*fitness landscape*) hace referencia a la hipersuperficie que se obtiene al aplicar la función de aptitud a cada punto del espacio de búsqueda. Generación es la respectiva iteración que da la medida de la aptitud y produce la creación de una nueva población por medio de operadores de reproducción(selección) y cruce. Una población puede subdividirse en grupos a los que se denomina subpoblaciones, normalmente sólo pueden cruzarse entre sí los individuos que pertenezcan a la misma subpoblación. En los esquemas con subpoblaciones, suele permitirse la migración de una subpoblación a otra, sobre todo en el contexto de algoritmos genéticos paralelos. Al evento de permitir el cruce sólo entre individuos de la misma subpoblación se le llama especiación; este concepto es una emulación del fenómeno natural que lleva el mismo nombre. La migración es la transferencia de los genes de un individuo de una subpoblación a otra. Existe un tipo de población usada en computación evolutiva, en la que cualquier individuo puede reproducirse con otro con una probabilidad que depende sólo de su aptitud, esto recibe el nombre de población panmítica. La mayor parte de los algoritmos evolutivos convencionales usan poblaciones panmíticas.

Debido a ruidos estocásticos, los algoritmos evolutivos tienden a converger a una única solución. Para evitar eso, y mantener la diversidad, existen técnicas que permiten crear distintos nichos para los individuos. Lo anterior es clave cuando se requiere trabajar con funciones de adaptación, las cuales podrían caracterizarse por ser de tipo multimodal y en la aplicación concerniente se requiriese localizar todos los picos que dicha función presenta. Como respuesta a este problema se acostumbra entonces formar subpoblaciones en forma análoga a lo que en la naturaleza corresponde a la ocupación de un terreno por diversas especies formando cada una un nicho; de lo anterior se puede entonces concluir que cada pico de una función multimodal se conoce en CE como nicho, y su uso tiene como consecuencia mantener la multiplicidad dentro de una población.

Un concepto que toma importancia a la hora de estudiar la efectividad de los algoritmos genéticos es la epistasis; ésta es la interacción entre los diferentes genes de un cromosoma y está referida a la medida en que la contribución de aptitud de un gen depende de los valores de los otros genes. Cuando un problema tiene poca epistasis (o ninguna), su solución es trivial, y por tanto un algoritmo escalando la colina(*hill climbing*) es suficiente para resolverlo. Este método de escalando la colina es un procedimiento de exploración heurística de grafos mediante el cual se genera un punto aleatorio dentro del espacio búsqueda, se evalúan los estados vecinos y según sea el mayor se toma dicho punto como referencia para generar otros vecinos y repetir este proceso hasta que se encuentre el óptimo local buscado, el cual se encuentra cuando el estado actual sea mayor que todos los estados vecinos.

Cuando un problema tiene una epistasis elevada, el problema será deceptivo, por lo que será muy difícil de resolver por un algoritmo evolutivo. Se llama bloque constructivo a un pequeño y compacto grupo de genes que han coevolucionado de tal forma que su introducción en cualquier cromosoma tiene una alta probabilidad de incrementar la aptitud de dicho cromosoma. La decepción es aquella condición donde la combinación de buenos bloques constructivos llevan a una reducción de aptitud, en vez de un incremento en su desempeño. Este fenómeno fue sugerido originalmente por Goldberg para explicar el mal desempeño del AG en algunos problemas.

La selección es el proceso mediante el cual algunos individuos en una población son escogidos para reproducirse(recombinarse), generalmente con base en la aptitud que cada uno posee. Existen diversas técnicas, las cuales se clasifican típicamente en: selección proporcional, selección mediante torneo y selección de estado uniforme. A su vez la selección proporcional según algunos autores se clasifica en: la ruleta, el sobrante estocástico, la universal estocástica y el muestreo determinístico, el cual a su vez tiene tres versiones: la selección de Boltzman, las jerarquías y el escalamiento sigma. Existe otra categorización que no es de tan común uso, pero que resulta interesante al menos mencionarla: la selección disruptiva, la competitiva y la de jerarquía no lineal. Ninguna técnica es

mejor que la otra pues cada una ha buscado resolver las falencias que presentan las otras en aspectos tales como: algoritmos ineficientes, convergencia prematura, requerir de parámetros extras entre otros.

El operador de recombinación es todo aquel mecanismo que tiene influencia en la forma en que se pasa la información genética de padres a hijos. Los operadores de recombinación caen en tres amplias categorías:

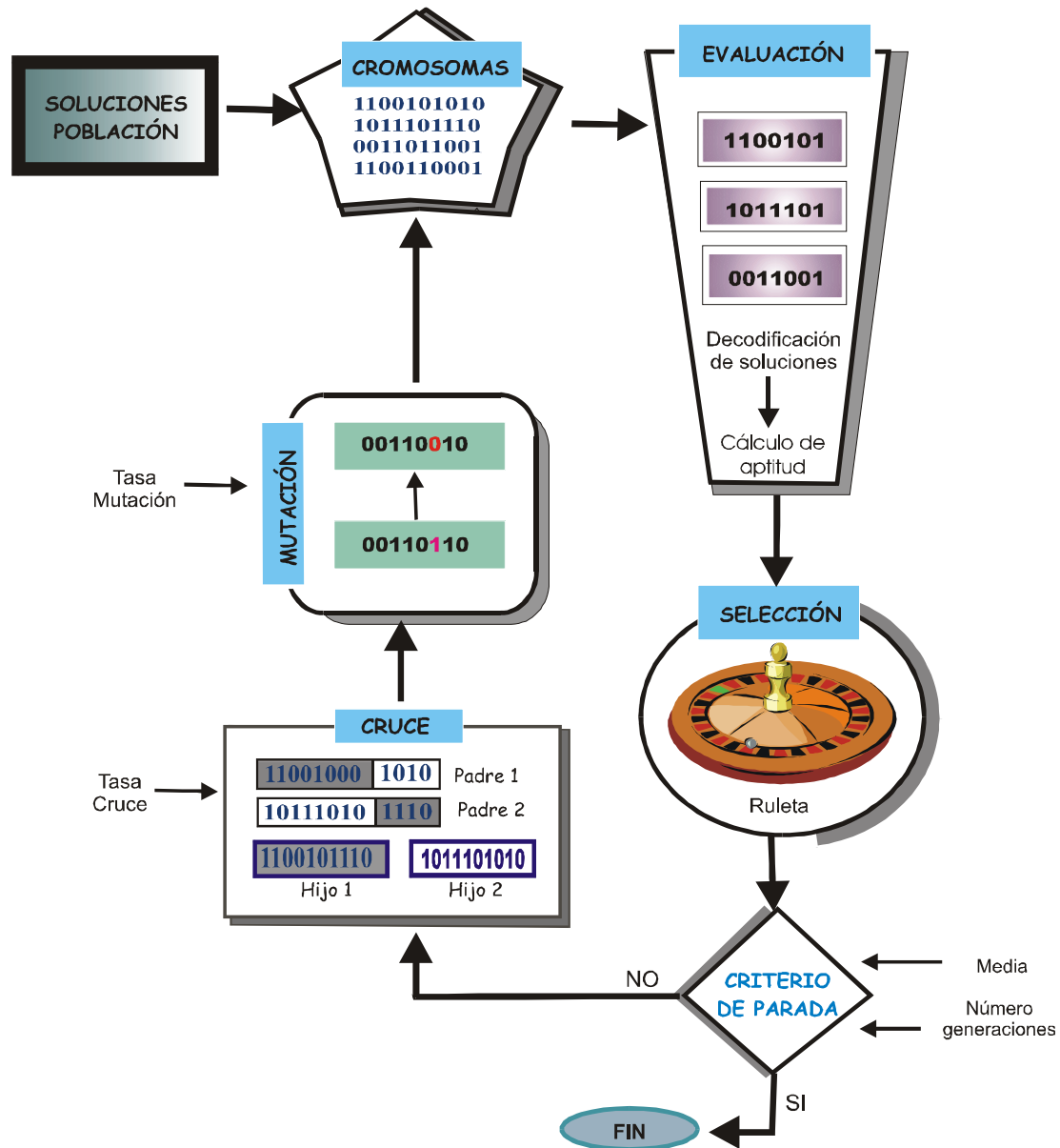
- ◆ Cruce.
- ◆ Mutación.
- ◆ Reordenamiento.

El cruce o recombinación es un operador que forma un nuevo cromosoma combinando partes de cada uno de sus cromosomas padres. Los métodos más comunes son el cruce en un punto, en dos puntos y el uniforme. La mutación es un operador que forma un nuevo cromosoma a través de alteraciones usualmente con una baja probabilidad en los valores de los genes de un solo cromosoma padre; comúnmente el método mas usado es el de la inversión de un bit, pero existen otros que se adecuan según el tipo de problema a resolverse y la codificación utilizada. Un operador de reordenamiento es aquél que cambia el orden de los genes de un cromosoma con la esperanza de juntar los genes que se encuentren relacionados, facilitando así la producción de bloques constructivos. La inversión es un ejemplo de un operador de reordenamiento en el que se invierte el orden de todos los genes comprendidos entre dos puntos seleccionados al azar en el cromosoma. En un AG, cuando una población no tiene requisito de la variedad, el cruce no será útil como operador de búsqueda, porque tendrá propensión a simplemente repetir a los padres. Es importante aclarar que en los AG los operadores de reproducción actúan sobre los genotipos y no sobre los fenotipos de los individuos. Otra característica interesante es el elitismo, el cual se entiende como el mecanismo utilizado en algunos algoritmos evolutivos para asegurar que los cromosomas de los miembros más aptos de una población se pasen a la siguiente generación sin ser alterados por ningún operador genético. Usar elitismo asegura que la aptitud máxima de la población nunca se reducirá de una generación a la siguiente. Sin embargo, no necesariamente mejora la posibilidad de localizar el óptimo global de una función.

Cuando se atraviesa un espacio de búsqueda, se denomina explotación al proceso de usar la información obtenida de los puntos visitados previamente para determinar qué lugares resultan más convenientes visitar a continuación. La exploración es el proceso de visitar completamente nuevas regiones del espacio de búsqueda, para ver si puede encontrar algo prometedor. La exploración involucra grandes saltos hacia lo desconocido. La explotación normalmente involucra movimientos finos. La explotación es buena para encontrar óptimos locales. La exploración es buena para evitar quedar atrapado en óptimos locales.

La figura 4 ilustra el diagrama de flujo de un AG simple o canónico como lo denominan algunos autores.

Figura 4. Algoritmo genético simple.



Fuente: Autor del proyecto.

En este proyecto se hará uso de lo que se conoce como AG simple o canónico, en el cual la selección se realiza mediante el método proporcional con la variante de la ruleta, el cruce se realizará en un punto y la mutación se hará mediante la

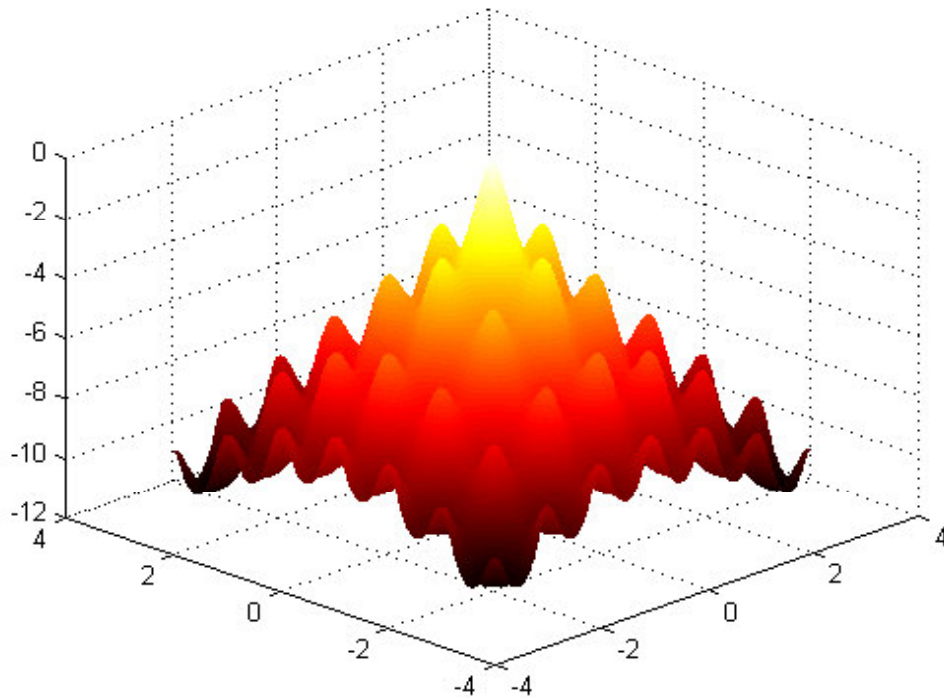
inversión; estos dos últimos operadores se trabajarán de forma probabilística (tasa de cruce-tasa de mutación).

1.5 EJEMPLO ILUSTRATIVO

Para fortalecer las ideas planteadas, es sano mediante un ejemplo ilustrar la metodología seguida por el AG para lograr así claridad con los distintos términos antes citados. Un ejemplo particular de optimización puede ser minimizar la función de Ackley, la cual se encuentra representada en forma escalada en la figura 5 para la región $x_1 \in [-3, 3]$ y $x_2 \in [-3, 3]$. La forma multimodal que presenta esta función propicia que fácilmente un método clásico pueda quedarse atrapado en un máximo (mínimo) local. La ecuación 1 ilustra la forma de dicha función.

$$y = e^{(0.5 \cos(2\pi x_1) + 0.5 \cos(2\pi x_2))} + 20 * e^{(-0.5 \sqrt{(0.1 x_1^2 + 0.1 x_2^2)})} - 20 - e^1 \quad (1)$$

Figura 5. Función de Ackley



Fuente: Autor del proyecto.

Para aplicar el AG simple lo primero que se debe realizar es la codificación. Para el presente caso se realizó de la siguiente forma:

- ◆ Codificación binaria con 20 bits.

- ♦ Representación en magnitud y signo, 19 bits para la magnitud y un bit para el signo.
- ♦ De la representación seleccionada en lo referente a la magnitud, 4 bits conforman la parte entera y los 15 bits restantes la parte decimal.

Con esta representación se puede trabajar con valores discretos en la región $x_1 \in [-7, 7]$ y $x_2 \in [-7, 7]$. En el siguiente ejemplo se ilustra un individuo A(cromosoma) con los 20 genes propios de la codificación elegida. La cadena de 1's y 0's es el genotipo y el significado como tal posterior a la decodificación de la cadena es el fenotipo.

$$A = 10100000010111110110 = -1 * (0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0 + 0 * 2^{-1} + 0 * 2^{-2} + 0 * 2^{-3} + 0 * 2^{-4} + 1 * 2^{-5} + 0 * 2^{-6} + 1 * 2^{-7} + 1 * 2^{-8} + 1 * 2^{-9} + 1 * 2^{-10} + 1 * 2^{-11} + 0 * 2^{-12} + 1 * 2^{-13} + 1 * 2^{-14} + 0 * 2^{-15}) = -4.046569824$$

Realizada la codificación respectiva, el siguiente paso a seguir es generar de forma aleatoria la población. El tamaño escogido de la población fue tomado a priori con un valor de 50, aunque este valor se cambió en diversos ejemplos que se corrieron. Es importante aclarar que por la metodología que se empleó se hizo necesario que la función sobre la cual se trabajó, que es la función de aptitud, deba ser siempre positiva en todo el intervalo de estudio; si la anterior condición no se cumple se debe realizar algún tipo de modificación, generalmente se recomienda sumarle una constante positiva; para el presente ejemplo la función tiene la característica de ser menor que cero para la región de búsqueda, por lo que se le sumó una constante positiva lo suficientemente grande, en este caso se utilizó 50.

El siguiente paso es la etapa de selección la cual opera según el criterio de la ruleta pesada. Esta metodología es la de mayor uso y consiste en calcular el valor esperado de cada individuo(h), el cual se computa dividiendo el valor de evaluación o aptitud de cada posible solución por la suma total de las evaluaciones parciales, como se muestra en la ecuación 2. Se recorre posteriormente toda la población sumando los valores esperados de los individuos hasta que dicha suma supere el valor de un número generado aleatoriamente que pertenezca al intervalo $[0, 1]$ (equivalente a hacer girar la ruleta), el individuo que haga que esta suma exceda el límite (número aleatorio), es el seleccionado. Este proceso se repite hasta que se escoja el número de individuos correspondiente al tamaño inicial de la población. El procedimiento es muy sencillo, como se observa se le asigna un peso a cada individuo según el desempeño de cada par; en este caso se habla de par, por que la función es de dos variables. De esta forma los individuos que sean mas aptos tendrán un mayor peso y por tanto mayor probabilidad de ser seleccionados, además el evento de elegir a un individuo con un valor bajo de aptitud se considera probable lo cual hace que la población

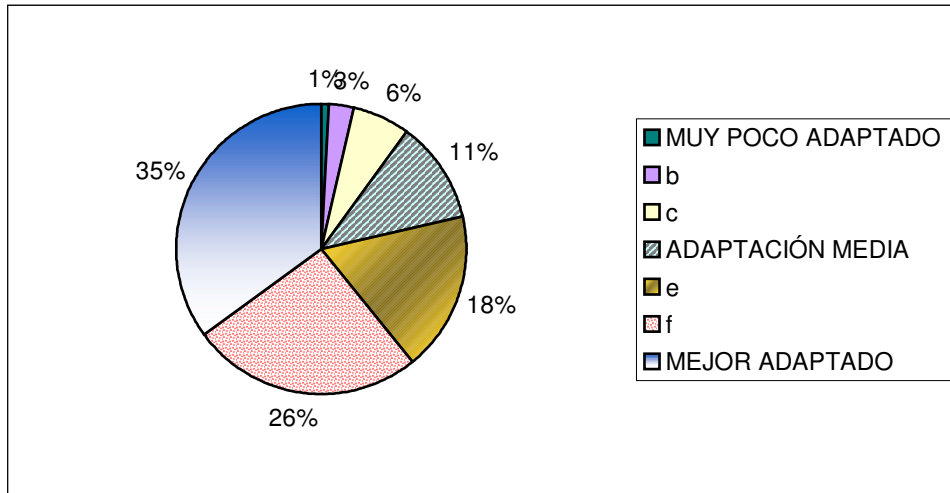
mantenga su heterogeneidad. La figura 6 ilustra la forma en que se distribuyen los pesos para un grupo de individuos.

$$\Pr(h)_i = \frac{aptitud(h_i)}{\sum_{j=1}^n aptitud(h_j)} \quad (2)$$

Para poder continuar con el proceso es importante que algunos parámetros vitales para el trabajo del AG se hayan definido, estos son a saber: la tasa de mutación y la tasa de cruce. Estos elementos se definen como entrada al AG desde su puesta en funcionamiento.

El significado de cada uno es respectivamente: el evento de realizar el cruce de dos individuos y el de mutar alguno de ellos. Para el presente ejemplo se escogieron: tasa de cruce del 80% y tasa de mutación del 4%. Estos valores mantienen la tendencia recomendada para hacer uso de un AG, como se citó anteriormente.

Figura 6. Ilustración del método de la ruleta pesada

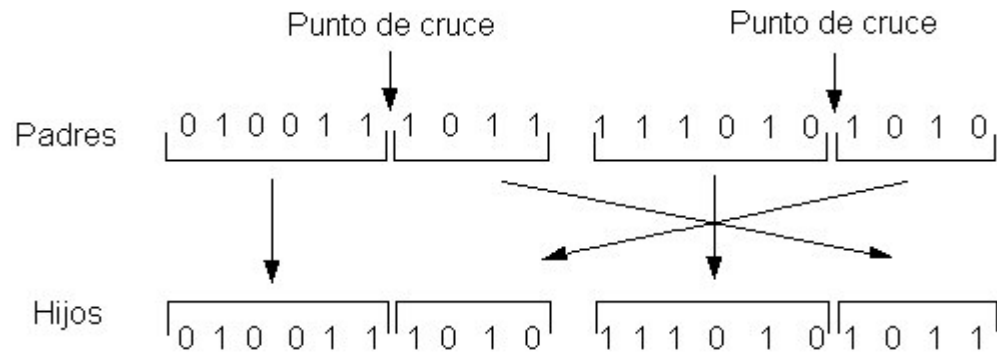


Fuente: Autor del proyecto.

Una vez seleccionados los candidatos al cruce, estos se guardan en alguna estructura de datos la cual se conoce generalmente como sitio de encuentro (*mating pool*). Para realizar el cruce (*crossover*) lo primero que se tiene que corroborar es la posibilidad de su ejecución o no. Para esto se genera un número aleatorio y se compara con la tasa de cruce, si este valor es superado el proceso se hará efectivo de lo contrario, los padres pasarán a conformar la nueva población. Cada par de individuos candidatos al cruce, se someten a este proceso de dos en dos según el orden en que hayan quedado. Dicho proceso se realiza mediante el método de un punto, como es indicado en la figura 7.

El sitio de cruce se selecciona mediante la generación de un número aleatorio que pertenezca al intervalo $[1, l]$, donde l es la longitud de la cadena, que para el presente ejemplo es 20. Por cada par de padres se producen dos hijos los cuales reemplazan a sus progenitores. Reconstruida la nueva población, esta se somete al operador de mutación.

Figura 7. Ilustración del cruce en un solo punto

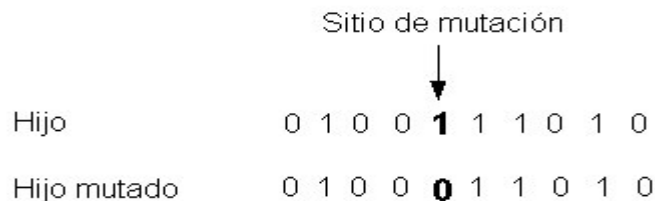


Fuente: Autor del proyecto.

El proceso consiste en generar un número aleatorio para luego revisar bit por bit y si el número aleatorio obtenido perteneciente al intervalo $[0,1]$, es mayor a la tasa de mutación se procede a complementar el respectivo bit, de lo contrario se deja idéntico y se procede hasta recorrer toda la cadena. Esto se hace con cada uno de los individuos, dicho proceso se expone en la figura 8.

Los anteriores pasos se repiten hasta la generación 15000, o la regla que se haya considerado para detener el proceso(criterio de parada). El dato anterior se consiguió después de realizar un buen conjunto de pruebas, las cuales consistieron en repetir la ejecución del programa observando el respectivo comportamiento de la evolución de las respuestas.

Figura 8. Ilustración del proceso de mutación.

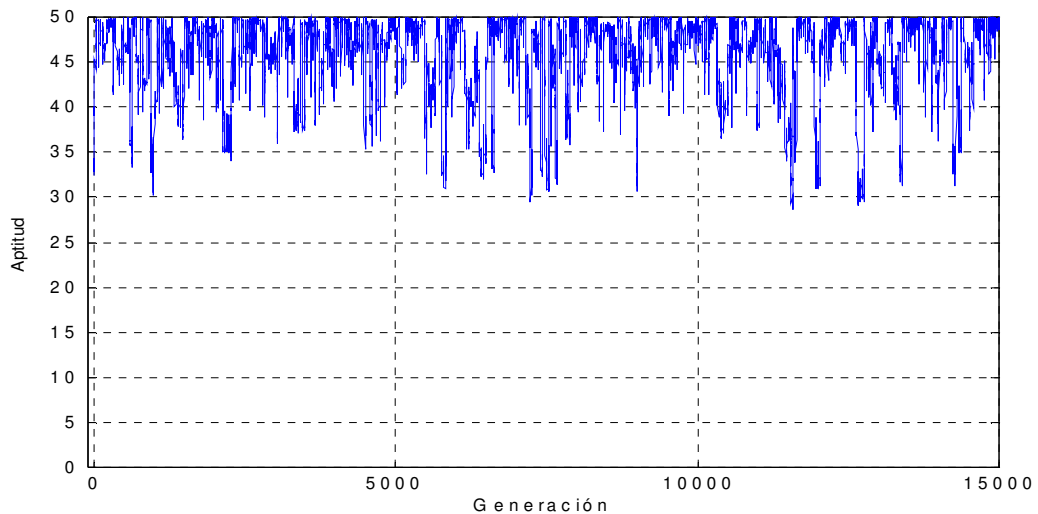


Fuente: Autor del proyecto

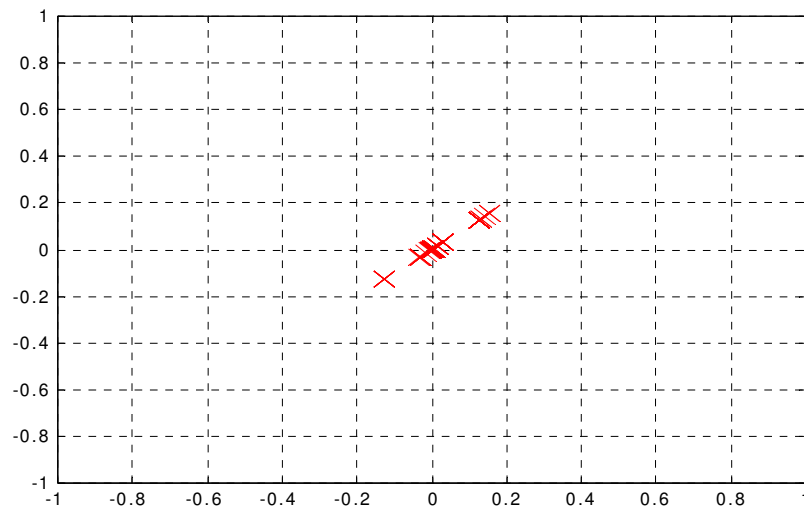
Realmente no se encontró la existencia de otra posible metodología en toda la bibliografía estudiada. En la figura 9 se muestra el desempeño del algoritmo

medido a través de la media en cada generación(7a) y en función de los valores de x_1 y x_2 , a los cuales llega el AG, después de correrlo 15000 generaciones. Es claro que la mayoría de respuestas oscilan alrededor del punto (0,0), el cual es el máximo global de esta función(7b). En el CD anexo se encuentra una carpeta denominada Ackley, la cual contiene los archivos en Matlab necesarios para revisar el proceso aquí estudiado. Un esquema bien detallado de los pasos seguidos en esta breve aplicación se encuentra en el apéndice A.

Figura 9. Resultados del desempeño del AG para la función de Ackley.
a) Media contra generación.



b) Ubicación de los puntos (x_1, x_2) solución.



Fuente: Autor del proyecto.

2. PRINCIPIOS MATEMÁTICOS BAJO LOS CUALES OPERAN LOS ALGORITMOS GENÉTICOS

2.1 INTRODUCCIÓN

La bibliografía consultada demuestra que no existe aún una teoría general aceptada la cual explique exactamente porqué los AG, con las propiedades que poseen ofrezcan resultados tan satisfactorios. Las investigaciones mas recientes hasta ahora se han concentrado en encontrar reglas empíricas para obtener de los AG el mejor desempeño posible. Sin embargo varias hipótesis han sido formuladas, las cuales pueden en forma parcial explicar el éxito de los AG. Dado esta tendencia se hace importante estudiarlas puesto que mediante estas se puede lograr una mejor implementación de los AG en las aplicaciones donde se usen⁷. Los AG cómo se describió en el capítulo anterior presentan una mecánica bastante sencilla desde el punto de vista computacional, esto lo demuestra el ejemplo propuesto, pero nada se mencionó acerca de la matemática que está detrás, lo cual precisa como lo han realizado diversos estudiosos de los AG, a efectuarse la siguiente pregunta: ¿Porqué funcionan los AG?. Para ello se requiere inicialmente hacer algunas definiciones básicas en lo referente a la estructura básica de lo que compone a un individuo, es decir a una solución candidata de cualquier problema en el que se pretenda aplicar esta disciplina de la IA. El término que mas comúnmente se usa es el de *esquema* o patrón de similitud; cómo su último nombre lo indica es un grupo de elementos que mantienen cierta similitud en posiciones muy precisas y los cuales aparecen en cada una de aquellas soluciones que más se acercan a la solución buscada. Los esquemas definen hiperplanos en el espacio de búsqueda. Otra forma interesante de interpretar lo que es un esquema está dada por el hecho de comprender al esquema como una región específica de todo el espacio de soluciones sobre el cual se esté trabajando. El alfabeto utilizado para la representación como ya se mencionó es generalmente el binario al cual se le agrega un elemento adicional, el carácter “x”, tan conocido en el estudio de los sistemas digitales como estado “no importa”(*don't care*); el significado que aquí toma es idéntico al contexto mencionado, su presencia indica que en aquella posición donde se encuentre no es de relevancia el valor que tome, sea “0” o “1”. Por ejemplo dados los siguientes individuos: {10100110, 11100000, 10100111, 10100011, 11000101}, se puede decir que existe el siguiente patrón de similitud: 1x100xxx, obviamente no es el único también podría decirse que xxx00xxx, también es un patrón de similitud aquí presente entre los muchos otros posibles. Se generaliza entonces que para un espacio de búsqueda de k^l , existen $(k+1)^l$ esquemas, donde l es la longitud de cada individuo, entendido como un número binario de 4 bits, se tiene que $k=2$, y $l=4$, entonces existirán $(2+1)^4=81$ esquemas posibles. Otra definición de

⁷ BEASLEY, Op. cit., p.7.

importancia es la referente al concepto de orden o especificidad de un esquema que se denota comúnmente como: $O(H)$, donde H se refiere a un esquema cualquiera de los $(k+1)^l$ posibles. Esta variable $O(H)$ se refiere al número de posiciones fijas diferentes a "x" en una cadena o esquema, es decir si $H=101x1xx1$, el orden es 5, por tanto $O(H)=5$. La siguiente definición es la referente a la distancia de un esquema, denotada por $d(H)$. Esta variable se refiere a la diferencia entre la última y primera posición fija de un esquema. Si se tiene una cadena $A=a_1a_2a_3a_4a_5a_6a_7a_8$ a la cual se le numeran cada posición como ahí se muestra entonces un esquema $H=101xx11x$, tiene distancia $d(H)=7-1=6$. Otro ejemplo aclara la idea si el esquema es $xxxxxxx0$, se tiene que su distancia sería $d(H)=8-8=0$. La forma mas adecuada que se tiene de entender por que los AG funcionan es mediante la Teoría de los Esquemas planteada por John Holland en 1975⁸. Este estudio se basa en observar los efectos de los distintos operadores, como selección, cruce y mutación, sobre determinado esquema perteneciente a una población y el cual se somete a diversos ciclos del AG o también denominados generaciones. En este capítulo solo se hará referencia a los operadores de un AG simple, los efectos de otros operadores, como inversión, reordenamiento, diferentes tipos de selección, cruce y mutación se dejan para estudios futuros.

2.2 EFECTO DE LA SELECCIÓN EN UN AG

El efecto de la selección sobre un determinado esquema es relativamente sencillo de entender. Sea $m(H,t)$ el número de representaciones de un esquema H en un instante t , o lo que es equivalente a una generación dada cualquiera que se da en el proceso del AG. Dado que el proceso de selección está basado en la probabilidad, pues una cadena es seleccionada para someterse al cruce según haya sido su valor de función de adaptación.

La probabilidad de una cadena A_i de ser seleccionada está dada por la ecuación 2 (ver capítulo 1). Se espera para la siguiente generación $t+1$ tener $m(H,t+1)$ representaciones del esquema H , es decir que para cada cadena la participación que puede tener en la selección está dada por:

$$e_i = \frac{f_i}{\bar{f}} = \frac{n^* f_i}{\sum_{j=1}^n f_j} \quad (3)$$

La participación que se da entonces en la siguiente generación para el esquema H se da por:

⁸ GOLDBERG, D. Op. cit., p.14.

$$m(H, t+1) = \frac{n * \sum_{i=1}^m f_i}{\sum_{j=1}^n f_j} \quad (4)$$

Definiendo la media $f(H)$ de los valores correspondientes a aquellas que contiene el esquema H se tiene:

$$f(H) = \frac{\sum_{i=1}^m f_i}{m(H, t)} \quad (5)$$

La expresión del número de esquemas H esperados, se puede obtener en función del número de esquemas de la generación anterior $m(H, t)$, como sigue:

$$m(H, t+1) = \frac{m(H, t) * f(H) * n}{\sum_{j=1}^n f_j} = \frac{m(H, t) * f(H)}{\bar{f}} \quad (6)$$

La ecuación anterior se puede interpretar como que aquellos esquemas que tengan una función de evaluación por encima de la media de toda la población, verán incrementado su número en la siguiente generación y visto desde otro punto de vista, esquemas por debajo de la media de toda la población tenderán a disminuir su participación o en el peor caso a desaparecer. El efecto al paso de las generaciones en el caso que permita el aumento de individuos que contengan el esquema H puede observarse al considerar $f(H)$ como un factor de la media poblacional como sigue:

$$f(H) = \bar{f} + c * \bar{f} \quad (7)$$

Lo cual produce que:

$$m(H, t+1) = m(H, t) * (1 + c) \quad (8)$$

La ecuación 8 en diferencias (t es una variable discreta), claramente tiene solución asumiendo que c es una constante de la siguiente forma:

$$m(H, t) = m(H, 0) * (1 + c)^t \quad (9)$$

El anterior resultado a pesar de la consideración de que c sea constante demuestra una tendencia exponencial (progresión geométrica) del número de individuos que contienen al esquema H en cada generación siguiente.

2.3 EFECTO DEL CRUCE EN LOS AG

El cruce es como se mencionó en el capítulo anterior, un intercambio de información de una forma estructurada aunque con un ingrediente aleatorio entre cadenas de individuos. La probabilidad que un esquema tiene de mantenerse o ser destruido, tiene una relación directa con la distancia del mismo esquema. El siguiente ejemplo muestra este fenómeno. Sea la cadena $A_1=10101110$ y los esquemas $H1=x01xxxx$ y $H2=xx1xxxx0$. Si por ejemplo el punto de cruce es 5 es claro que el esquema $H1$ se mantiene mientras que el esquema $H2$ se destruye. Al observar la distancia de cada uno de los esquemas mencionados se observa que $d(H1)=1$ mientras que $d(H2)=5$. Por lo anterior se puede definir la probabilidad de destrucción de un esquema como:

$$p_d = \frac{d(H)}{l-1} \quad (10)$$

Por las mismas propiedades de las variables aleatorias la probabilidad entonces de mantenerse está dada por:

$$p_s = 1 - p_d = 1 - \frac{d(H)}{l-1} \quad (11)$$

Ahora si se tiene en cuenta que para definir el lugar de cruce en una cadena se escoge un número aleatorio entre los $l-1$ sitios posibles y si se llama a esta probabilidad p_c de escoger un sitio determinado, entonces la probabilidad de que un esquema se mantenga nunca es menor a:

$$p_s \geq 1 - \frac{p_c * d(H)}{l-1} \quad (12)$$

Por tanto el efecto conjunto del cruce y la selección se pueden expresar afectando la expresión obtenida en una ecuación anterior por esta probabilidad como se muestra a continuación:

$$m(H, t+1) \geq \frac{m(H, t) * f(H)}{\bar{f}} \left[1 - \frac{p_c * d(H)}{l-1} \right] \quad (13)$$

Es claro entonces de esta ecuación que para que un esquema se mantenga en las siguientes generaciones debe conservar una media superior a la media de toda la población además de una distancia bastante corta.

2.4 EFECTO DE LA MUTACIÓN EN LOS AG

Como se conoce, la mutación es la alteración en forma aleatoria de un solo alelo en una cadena con una probabilidad p_m . Por tanto para que un alelo o bit dentro

de un esquema se mantenga inmodificable se requiere de una probabilidad $1-p_m$. Como cada mutación que se realice es estadísticamente independiente, entonces para que un esquema sobreviva en forma completa, deben sobrevivir cada una de las $O(H)$ posiciones que tiene y la probabilidad total que el esquema sobreviva en forma completa se da por el producto de $O(H)$ veces la probabilidad de que un alelo sobreviva. Así:

$$p_{so} = (1-p_m)(1-p_m)(1-p_m)\dots(1-p_m) = (1-p_m)^{O(H)} \quad (14)$$

Por la misma naturaleza de los AG se supone que el valor de la probabilidad de mutación debe ser muy inferior generalmente por debajo del 1%, por tanto:

$$p_{so} = (1-p_m)^{O(H)} \approx 1 - O(H) * p_m \quad (15)$$

Sumando entonces los efectos de la selección el cruce y la mutación se tiene que el número de individuos o cadenas que contienen un esquema en particular H , en cada generación siguiente está dado por:

$$m(H, t+1) \geq \left[\frac{m(H, t) * f(H)}{\bar{f}} \right] * \left[1 - \frac{p_c * d(H)}{l-1} \right] * [1 - O(H) * p_m] \quad (16)$$

Los resultados a los que conduce esta ecuación reciben el nombre de Teorema de los Esquemas o Teorema Fundamental de los AG⁹ el cual se enuncia de la siguiente forma: Esquemas de corta longitud y bajo orden tienden a proliferar o desaparecer en las generaciones siguientes de acuerdo con la aptitud o adaptación promedio que posean. A los esquemas que cumplen esta característica se les da el nombre de bloques constructivos.

2.5 ESQUEMAS PROCESADOS EN FORMA ÚTIL

Uno de los planteamientos sobre los cuales más se ha cuestionado a los AG es lo referente a cuantos esquemas realmente son procesados en forma útil cuando se corre un AG, pues de esta forma se puede plantear de cierta forma la efectividad de esta disciplina en la resolución de problemas de optimización. Los fundamentos subyacentes en la búsqueda genética son: en una representación binaria de longitud l , cada individuo de la población pertenece a 2^l esquemas y pertenece a todos los esquemas en los cuales aparece uno cualquiera de sus bits. Cada población de n individuos permite entonces estimar el grado de adaptación de entre 2^l y $n*2^l$ esquemas (el espacio se reduce, pues no siempre se trabaja con el total de individuos posibles que es 2^l , sino por el contrario se toman n individuos de los 2^l posibles, así resultan estos límites inferior y superior de los esquemas

⁹ GOLDBERG, D. Op: cit., p.33.

presentes). Se sabe que no todos ellos son procesados ya que, dependiendo de la longitud y el orden, unos tendrán una mayor probabilidad de supervivencia a los operadores de cruce y de mutación. Se ha estimado que en una generación de n estructuras o individuos se procesan del orden de $O(n^3)$ esquemas. Esta es una importante propiedad a la que Holland dio el nombre de paralelismo implícito. El paralelismo implícito es el que hace que un algoritmo genético que manipule una población de unos cuantos millares de cadenas, realmente esté tomando muestras de un número de regiones o esquemas enormemente mayor. Tal paralelismo implícito (en el sentido de procesamiento paralelo), proporciona al algoritmo genérico su ventaja principal sobre otros métodos de resolución de problemas. A pesar de la deducción planteada por John Holland acerca del tópico en cuestión, esta se le ha criticado en gran forma pues el hecho de tomar la población con un valor tan específico, hace que sea solo un caso particular como se afirma en varios artículos¹⁰. El planteamiento para demostrar entonces el orden $O(n^3)$, parte de considerar esquemas de longitud inferior a l_s , los cuales tienen una probabilidad mas grande de sobrevivir al cruce y a la mutación que una constante p_s . Si se toma como punto de partida un individuo de los n posibles en la población inicial, en el cual el número de esquemas que contiene sea de una longitud inferior a l_s y sobre el cual se dibuja una ventana imaginaria que se va desplazando posición por posición, se tienen entonces $l-(l_s-1)$ ventanas representando un grupo de esquemas, como se ilustra en la figura10:

Figura 10. Ilustración de desplazamiento de ventana en un esquema particular.

101010110101 → %%%%0110101 → 1%%%%110101 → 10%%%%10101 → 101%%%%0101 ...

Fuente: Autor del proyecto.

Para cada ventana se tienen entonces 2^{l_s-1} esquemas de longitud igual o inferior a l_s . En definitiva el número total de esquemas para una cadena cualquiera sería:

$$2^{(l_s-1)}(l-l_s+1) \quad (17)$$

Para una población de tamaño n sería:

$$n_s = n * 2^{(l_s-1)}(l-l_s+1) \quad (18)$$

Dado que n es un valor muy grande y da una sobreestimación de los esquemas que realmente se están procesando pues muchos de ellos de bajo orden se estarán duplicando, se considera entonces una población $n = 2^{(l_s/2)}$; esta última consideración es el talón de Aquiles de la deducción de Holland y que ha generado gran controversia.

Por tanto se tiene reemplazando que:

¹⁰ BERTONI, Alberto. Implicit Parallelism in Genetic Algorithms. En: Artículo IEEE; April 1993

$$n_s = n * 2^{\frac{(l_s-1)}{2}} * 2^{\frac{(l_s-1)}{2}} * (l - l_s + 1) = \frac{n^3 * (l - l_s + 1)}{2} \quad (19)$$

Como se observa $n_s = C * n^3$ de donde se concluye el resultado planteado inicialmente, que el número de esquemas procesados en forma efectiva tiene un límite inferior proporcional al cubo del tamaño de la población.

2.6 OTROS TÓPICOS DE INTERÉS EN EL ESTUDIO TEÓRICO DE LOS AG

Algunos temas abarcan el tiempo de los estudiosos del tema de los AG, entre los que se encuentra el problema de la mínima decepción, la hipótesis de los bloques constructivos y qué tipos de problemas pueden ser resueltos con esta herramienta. Los bloques constructivos son aquellos esquemas que componen cadenas de alto desempeño (mejor adaptación según la función de evaluación), tienen bajo orden y longitud corta y quienes se expanden de generación en generación en forma exponencial logrando obtener individuos cada vez mejor adaptados, es decir se acercan mas al óptimo; la hipótesis de los bloques constructivos se basan en esta última suposición la cual requiere por tanto de un estudio profundo para su respectiva confirmación. Los trabajos que hasta ahora se han realizado plantean una evidencia empírica limitada que no demuestran la teoría¹¹, sugieren únicamente que los AG se pueden aplicar a diferentes tipos de problemas que se encuentran normalmente.

Cualquier algoritmo de optimización requiere del uso de dos técnicas para encontrar un óptimo global: exploración para investigar nuevas y desconocidas áreas dentro el espacio de búsqueda y una de explotación que permita hacer el uso del conocimiento hasta ahora encontrado a través de los puntos ya visitados y que permitan entonces realizar la obtención de nuevos y mejores puntos. El problema radica en que estas dos actividades son contradictorias entre si y se asemejan a un problema clásico de la teoría de decisión estadística¹², conocido como el problema de la máquina tragamonedas de n brazos, en la cual el jugador tiene que decidir por cual brazo jugar pues se desconoce de antemano cual de ellos está asociado con el premio mayor. La búsqueda aleatoria es puramente del tipo de exploración, pero no de explotación como si ocurre con el método de escalando la colina el cual es puramente de explotación; combinar estas dos estrategias conducen al método mas efectivo, y es una de las razones según algunos seguidores de los AG que estos tiene para funcionar como lo hacen, pues mezclan la exploración mediante los operadores de la mutación y del cruce y la explotación mediante el mecanismo de selección, situación que no siempre ocurre con alguna

¹¹ GOLDBERG Op. cit., p.43.

¹² Ibid., p.43.

clase de problemas y se debe recurrir a buscar nuevas formas de operadores o averiguar el valor adecuado con el cual se aplique el operador seleccionado es decir un proceso de sintonización mediante el cual se puedan interpolar los parámetros de un AG como son la tasa de cruce, la tasa de mutación y el tamaño de la población entre otros.¹³

Los recientes estudios han acudido a las funciones de Walsh, para ayudar en el tema de demostrar la hipótesis de bloques constructivos y mostrar en cuales tipos de problemas los AG pueden o no ser aplicados. Esta transformada tiene características semejantes a la transformada de Fourier, y ha mostrado ser una herramienta valiosa en ciertos tipos de problemas, pero como no existe felicidad completa algunos problemas que requieren codificaciones por así decirlo especiales, han mostrado un resultado bastante pobre con este análisis. Sumado a lo anterior existen problemas en los cuales no funciona en forma adecuada un AG simple (selección, cruce, mutación) y conducen a casos conocidos como de decepción¹⁴, en los cuales el AG diverge de la solución óptima y se viola la hipótesis de los bloques constructivos, pues la función en estudio tiene óptimos aislados: los mejores puntos tienden a ser rodeados por los mas malos. La decepción se dice que es un caso de epistasis lo cual es, como ocurre en la naturaleza, el efecto inhibitor que tiene la presencia de un gen en suprimir el efecto de otro gen en otro locus. Para lograr que una codificación sea adecuada se deben cumplir las siguientes condiciones: genes relacionados deben estar lo mas juntos posibles dentro del cadena o individuo candidato y debe existir una baja interacción entre genes, lo que es equivalente a una baja epistasis la cual desafortunadamente siempre se presenta en funciones multimodales. El estudio sigue abierto y como se mencionó en un comienzo no existe aún una teoría generalizada que demuestre el por que del buen o mal funcionamiento de los AG en ciertos tipos de problemas, existen sin embargo algunos artículos que han abordado este tema con profundidad y que están en la bibliografía para su respectiva consulta.

¹³ CAVALCANTI, Marco Aurelio. Algoritmos genéticos: principios y aplicaciones. En: Artículo IEEE; p. 7.

¹⁴ BEASLEY, D. Op:Cit., p. 3.

3. EL PROBLEMA DEL PLANEAMIENTO DEL MOVIMIENTO

3.1 INTRODUCCIÓN

El problema del planeamiento de movimiento es muy sencillo de enunciar, aunque como se ha demostrado por los diferentes expertos en el tema es muy difícil de resolver. Un neófito en el tema fácilmente se le cruzan por la cabeza este tipo de ideas con tendencias hacia lo fútil en cuanto al tema en cuestión, y es natural pues si se mira al ser humano esta tarea resulta bastante elemental, bueno eso se puede pensar ahora de adulto quizá no cuando se era bebé. Cuando el ser humano realiza los movimientos que de todos es conocimiento, estos son el producto de muchos años de aprendizaje, lo cual pues demuestra una constante siempre presente y es que aquellas tareas que de cierta forma resultan triviales para el ser humano en cierta época de su vida, no lo son para las computadoras ya que son dominios de tipo interdisciplinario que terminan utilizando conocimiento referente a diversas áreas como la inteligencia artificial, geometría, geometría computacional, topología, modelado dinámico entre otras.

El planeamiento de movimiento es una tarea de gran importancia en la robótica. Esta se puede dividir en dos subtarear las cuales no están desligadas una de la otra: el planeamiento de camino(*path planning*) y el planeamiento de trayectoria(*trajectory planning*). Desde el punto de vista del español no hay diferencia entre las palabras camino y trayectoria, pero para aclaración se entiende como planeamiento de camino aquella tarea consistente en generar una ruta libre de colisiones en un ambiente lleno de obstáculos de cualquier naturaleza. El planeamiento de trayectoria tiene que ver ya con la planificación del movimiento en sí, velocidad, aceleración en cada rueda o articulación según sea el caso, para lograr seguir el camino planeado en la primera etapa. Se generan bajo este contexto diversas connotaciones en cuanto a clasificaciones se refiere. Una de ellas está dada por el denominado planeamiento de movimiento grueso y el planeamiento de movimiento fino. El planeamiento de movimiento grueso se refiere a problemas que involucran áreas en las cuales el espacio libre para el robot es mayor que el tamaño mismo de los objetos sumado al error posicional que provoca el robot en sí; lo anterior asegura que la inexactitud no provocará colisiones inesperadas mientras se realizan los movimientos generados por el planeador; el trabajo que se realiza en el presente proyecto clasifica en esta categoría. Para el caso de planeamiento de movimiento fino, este se ocupa del movimiento entre objetos en los cuales el espacio libre es tan estrecho que ocasiona que la precisión requerida para el movimiento exceda la precisión posicional misma del robot¹⁵. Cada situación merece consideraciones que son

¹⁵ HWANG, Yong. Gross Motion Planning. A Survey. En: Artículo ACM, Computing Surveys. Vol.24. 1992; p.220

propias de cada caso. Otro tipo de clasificación se da en el sentido que el planeamiento se haga en línea o fuera de ella, es decir con ambientes de trabajo que son variantes o invariantes en el tiempo, lo cual implica para el primer caso que cada vez que se detecte un cambio en el ambiente se haga un replanteo del camino propuesto inicialmente. Son casi obvias las dificultades que se derivan de esto y lo efectivo o no que pueden resultar ciertas metodologías. Cuando se considera el evento de la movilidad o inmovilidad de los obstáculos se habla de dos tipos de planeamiento: estático y dinámico.

En algunas situaciones los robots pueden mover los objetos y recibe el nombre de planeamiento de objetos móviles; esto recibe el nombre de planeamiento de movimiento múltiple; para el caso de robots que cambian su forma se habla de planeamiento graduable. Si existe algún tipo de restricciones, el planeamiento puede ser del tipo restringido o sin restricciones. Las restricciones que aquí se hablan hacen referencia a límites en la velocidad y aceleración máxima del robot, además de restricciones en las curvaturas de los caminos planeados para el robot. En el presente proyecto se trabajará sobre los dos problemas: planeamiento de camino y de trayectoria, además de considerar los obstáculos fijos y el robot como un polígono rígido desplazándose en un ambiente bidimensional de extensión $n \times n$.

Continuando con el proceso de contextualización que se planteó en los dos capítulos anteriores, en este tercero se pretende ubicar el problema sobre el cual se quiere trabajar, que es lo referente al planeamiento de movimiento, por lo cual se encontrarán algunas referencias al tema sin entrar en detalle pues no es el objeto. La presentación se hace basada en la diversa bibliografía consultada, la cual siempre ubica los problemas de la robótica y las soluciones desde el punto de vista de un manipulador, mas que de un robot móvil. Se presentan entonces los siguientes aspectos: unas definiciones básicas procurando ser lo mas estricto posibles con la notación comúnmente utilizada, plantear la complejidad del problema desde el punto de vista computacional, las diversas soluciones que se han dado al problema y por último detalles acerca de robots móviles.

3.2 DEFINICIONES BÁSICAS

A este punto es importante realizar algunas definiciones que son de frecuente uso en el campo de la robótica, haciendo la claridad que la mayoría de ellas harán referencia a robots móviles semejantes a un carro, también conocidos como plataformas móviles. Poco o nada se referirá a robots que se conocen como manipuladores o manipuladores móviles, aunque algunos conceptos pueden ser mas claros cuando se utiliza como referencia a estos tipos de robots. Sumado a lo anterior la gran mayoría de trabajos teóricos sustentados con sobresaliente rigidez matemática siempre se han hecho pensando en robots que cumplen las características de un manipulador, lo que obliga en ciertas situaciones a mencionar este tipo de objetos. La primer definición que se sugiere debiera darse

es la concerniente al término robot, sin recurrir mucho a la historia la palabra robot como aquí interesa sería un dispositivo mecánico -sea un brazo manipulador, una mano de múltiples dedos múltiples articulaciones, un vehículo con ruedas o patas, una plataforma de libre vuelo o en el caso mas extremo, una combinación de cualquiera de los mecanismos antes mencionados- el cual se encuentra equipado con actuadores y sensores bajo el control de un sistema de cómputo¹⁶. Este opera en un espacio de trabajo dentro del mundo real, el cual está poblado por objetos físicos y está sujeto a las leyes de la naturaleza.

El robot desempeña diversas tareas a través de la ejecución de movimientos en el espacio de trabajo. Dependiendo de la tarea a desarrollarse el robot puede llegar a considerarse como un punto, un polígono en el caso bidimensional o un poliedro en el caso tridimensional. La configuración de un robot es un conjunto de parámetros independientes que caracterizan la posición de cada punto en el objeto. El número de parámetros que especifican la configuración de un objeto, en este caso el robot, reciben el nombre de grados de libertad.

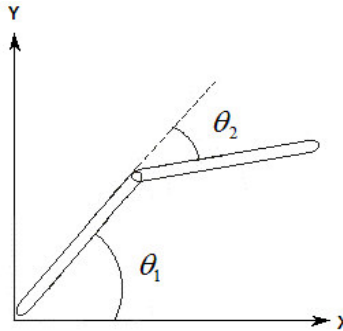
El conjunto de todas las configuraciones se denomina el espacio de configuraciones, comúnmente denotado en la literatura como *Cspace*. El espacio libre se considera a todas aquellas configuraciones para las cuales el robot no colisiona con los obstáculos, sea en el mundo real o en el mismo espacio de configuraciones. El camino para un robot es una curva en el espacio de configuraciones. La trayectoria es la misma curva a la que se refiere el camino, pero indexada con la variable del tiempo. Para el presente trabajo el robot se considerará como un polígono y el planeamiento se realizará desde la perspectiva del espacio bidimensional tal como este es.

Un objeto referenciado como robot, que en este caso es un manipulador con dos grados de libertad θ_1 y θ_2 , es mostrado en la figura 11. En esta situación se muestra una configuración en particular de las muchas posibles.

La figura 12a) es el movimiento libre de colisión para dicho manipulador, con dos eslabones ubicado en el mundo real, entre un punto de partida S (configuración inicial) y un punto de llegada G (configuración final) con varios obstáculos poligonales(en negro). En la figura 12b) se muestra el espacio de configuraciones del manipulador en su respectivo espacio de trabajo. Las figuras negras representan aquellas configuraciones las cuales provocarían una colisión con los obstáculos. El dibujo en color rojo corresponde al camino que produce los movimientos mostrados en la figura 12a).

¹⁶ LATOMBE, Jean. Robot Motion Planning. Boston: Kluwer Academic Publishers, 2001; p. 1.

Figura 11. Manipulador con dos grados de libertad.

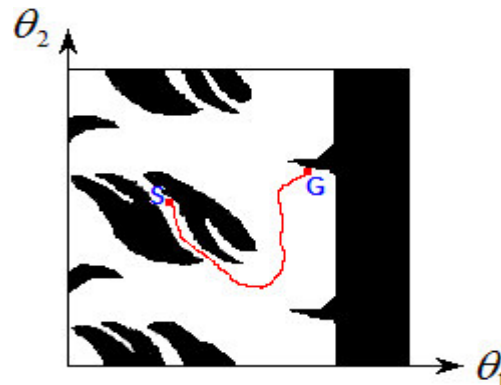
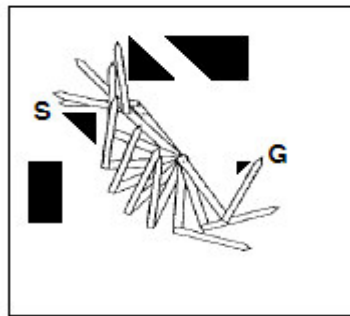


Fuente: Yong Hwang. Gross Motion Planning. A Survey. P.222.

Figura 12. Movimientos del manipulador con dos grados de libertad.

a). Movimientos libres de colisión en el mundo real

b) Espacio de configuraciones

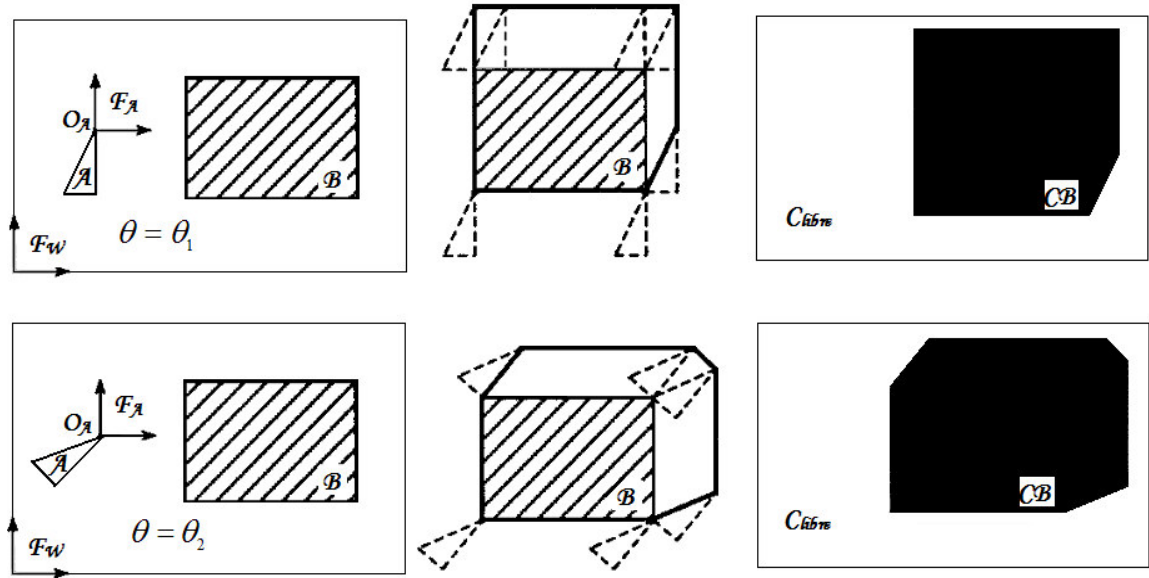


Fuente: Yong Hwang. Gross Motion Planning. A Survey. p.222.

Desde un punto de vista mas estricto matemáticamente las definiciones anteriores se pueden expresar como sigue. Al objeto en movimiento, robot, se le denota por A el cual se mueve en un espacio Euclidiano W que se denomina espacio de trabajo y se representa como R^N donde N puede tomar los valores $N=2, 3$. A los objetos rígidos distribuidos en W se les nota con B_i y reciben el nombre de obstáculos. Se asume que la geometría de A y B es conocida además de su respectiva localización inicial. Otra suposición que se acostumbra realizar es que A no posee ningún tipo de restricción cinemática con lo cual se dice que es un objeto de libre vuelo. El problema del planeamiento de camino se puede plantear entonces como sigue: dada una posición y orientación inicial además de un sitio de llegada deseado para A en W se desea generar un camino τ que especifique una secuencia continua de posiciones y orientaciones de A que eviten el contacto con los B_i obstáculos, arrancando desde la posición y orientación inicial, y

terminando en la posición y orientación de llegada deseada siempre y cuando exista el camino(solución).

Figura 13. El cuerpo A en movimiento en el espacio de trabajo y su respectivo espacio libre y espacio de obstáculos.



Fuente: Jean Latombe. Robot Motion Planning. P.90.

En cuanto al espacio de configuraciones se refiere la filosofía de este es representar al robot como un punto que se mueve en un espacio apropiado el cual es un mapeo de los obstáculos del mundo real. Para su definición se parte de suponer dos sistemas coordenados FW y FA asociados a cada elemento en cuestión, con la característica que FW es fijo mientras que FA es móvil y sigue al robot. Por definición cada punto a de A , por ser este un cuerpo rígido tiene una posición fija con respecto a FW , pero sus posiciones en W dependen de la posición y orientación de FA con respecto a FW . Para cada uno de los obstáculos ocurre situación similar, pues son considerados como cuerpos rígidos y fijos, por lo que cada punto de cualquier obstáculo B_i tiene una posición fija con respecto a FW . La configuración de un objeto arbitrario es una especificación de la posición de cada punto perteneciente a este objeto, relativo a un sistema de referencia fijo. Una configuración q de A es una especificación de la posición T y orientación Θ de FA con respecto a FW . El espacio de configuraciones de A es el espacio C de todas las configuraciones de A . El subconjunto de W ocupado por A en la configuración q es denotado por $A(q)$. En el mismo sentido el punto a sobre A , en la configuración q es denotado por $a(q)$ en W . Se podría describir la configuración por una lista de parámetros reales como podían ser las matrices de rotación y traslación de la representación Denavit-Hartenberg¹⁷, pero se puede considerar

¹⁷ FU, K. Robótica. Control, detección, visión e inteligencia. México: McGraw-Hill, 1990; p. 37.

esto como una representación redundante, por las mismas características que implicaría cumplir esta matriz. Otra posibilidad sería describir cada configuración con un grupo de parámetros de mínima cardinalidad como serían por ejemplo los ángulos de Euler, pero dado el problema de no poseer la característica de inyectiva o unívoca, complica aún mas la situación. Si cada configuración de un robot A tiene una lista de m parámetros independientes (dimensión de C), sería posible representar C como R^m , pero un estudio mas detallado de C conduce a que esta se entienda mas como si fuese localmente parecida a R^m , pero globalmente diferente. Lo anterior significa que C se descompone en una unión finita de p parches, donde cada uno se representa como R^m y matemáticamente recibe el nombre de carta. Si el robot sobre el cual se está estudiando solo realiza movimientos traslacionales, el espacio de configuraciones de A es una copia de R^N y es por lo tanto un espacio euclidiano. Los obstáculos B_i presentes en W se mapean sobre una región en C, la cual es representada por la siguiente expresión:

$$CB_i = \{q \in C / A(q) \cap B_i \neq \emptyset\} \quad (20)$$

la cual se denomina un C-obstáculo y la unión de todos los C-obstáculos es llamada C-región de obstáculos. La expresión (20) se lee así: La región procedente del obstáculo B_i en el espacio de configuraciones C es el conjunto de las diferentes configuraciones q pertenecientes a C tales que la intersección de la región de puntos correspondientes al robot $A(q)$ con los puntos correspondientes a todos los obstáculos B_i sean diferentes de vacío.

De esta misma definición aparece otra región la cual se define como sigue:

$$C_{libre} = C \setminus \bigcup_{i=1}^b CB_i = \left\{ q \in C / A(q) \cap \left(\bigcup_{i=1}^b B_i \right) = \emptyset \right\} \quad (21)$$

la cual recibe el nombre de espacio libre y se nota como C_{libre} , donde b es el número total de obstáculos. Cualquier configuración que pertenezca a C_{libre} se denomina como configuración libre. El desarrollo de esta teoría y los respectivos detalles, se encuentra en el libro: Robot Motion Planning de Jean Claude Latombe.

En la figura 13 el cuerpo rígido en forma triangular representa al robot que puede desplazarse libremente en el plano con una orientación fija. Su configuración es representada por el par $q=(x,y)$. El espacio de configuraciones es mostrado en el lado derecho de la figura 13 y la parte oscura es el espacio referente al mapeo del obstáculo que tiene forma cuadrada en el espacio de configuraciones. La notación mostrada en esta figura corresponde con la citada en regiones anteriores.

En cuanto al concepto de camino este, requiere del concepto de continuidad la cual obliga a definir una topología en C, y una forma clásica de hacerlo es especificar una función de distancia

$$d: C \times C \rightarrow R \quad (22)$$

la cual debe satisfacer la condición de:

$$d(q, q') = \max_{a \in A} \|a(q) - a(q')\| \quad (23)$$

en la cual $\|x-x'\|$ se refiere al concepto de distancia euclidiana entre dos pares de puntos x y x' pertenecientes a RN . Dadas estas definiciones se puede decir entonces que, un camino de A desde la configuración de partida $q_{inicial}$ hacia la configuración de llegada q_{final} es un mapa continuo $\tau: [0,1] \rightarrow C$ con $\tau(0)=q_{inicial}$ y $\tau(1)=q_{final}$.

3.3 COMPLEJIDAD DEL PROBLEMA

Es importante cuantificar el desempeño de los diversos métodos de planeamiento. Generalmente el tiempo que requieren dichos métodos para resolver el problema y la inherente complejidad computacional que les implica son los puntos de referencia para establecer que tan eficientes pueden ser. Este análisis es obligatorio pues permite sugerir nuevas estrategias cuando la proposición inicial conlleva a soluciones altamente costosas. Según la diversa literatura, el primero en establecer un límite inferior fue Reif¹⁵, en el trabajo que llamó “El problema del movimiento generalizado”, en el cual bajo las consideraciones de un espacio de configuraciones de dimensión arbitraria, el planeamiento de un camino libre para un robot hecho de un número arbitrario de cuerpos poliédricos conectados mediante articulaciones, en un ambiente de obstáculos finitos también poliédricos y entre dos configuraciones cualesquiera, es de tipo PSPACE-difícil. Un problema es de estas características cuando requiere un espacio polinomial de memoria para su solución.

El siguiente resultado que planteó un límite superior para el planeamiento de camino, expone que un camino libre en un espacio de configuraciones de cualquier dimensión m y cuyo espacio libre es un conjunto definido por n restricciones polinomiales de máximo grado d , puede ser computado por un algoritmo con tiempo de complejidad exponencial en m y polinomial en ambos n (complejidad geométrica) y d (complejidad algebraica). Este límite fue establecido en primera medida por Schwartz y posteriormente por Canny¹⁸. Estos resultados sugieren en forma muy fuerte que la complejidad del planeamiento de camino se incrementa exponencialmente con la dimensión del espacio de configuraciones, el número de restricciones polinomiales referentes al movimiento del robot y el grado

¹⁸ LATOMBE. Op. cit., p.34.

de estas restricciones. Para la reducción de esta complejidad se han realizado varios trabajos enfocados especialmente a reducir el tamaño del espacio de configuraciones por ejemplo mediante una proyección de este o reduciendo su dimensión, entre muchas otras estrategias que se encuentran en la literatura.

3.4 APROXIMACIONES BÁSICAS A LA SOLUCIÓN DEL PROBLEMA

Los tópicos que pueden considerarse de importancia crítica en cualquier problema de planeamiento de movimiento son: el espacio de configuraciones, la representación de los objetos -incluyendo el robot- y el ambiente de trabajo, la obtención de diversas soluciones que puedan conducir a una respuesta adecuada, un método de búsqueda y finalmente una estrategia de optimización. El espacio de configuraciones es importante porque es el espacio donde residen todas las soluciones; una gran cantidad de aproximaciones al problema lo usan, otras no, lo cual justifica en cierta medida la necesidad de presentar los temas citados en los numerales anteriores.

A continuación se hará una breve reseña de cómo operan diversas inventivas las cuales tienen como fin obtener soluciones en sentido general al problema de planeamiento de movimiento; además se presentarán algunas características que las destacan y las diferencias que presentan. Estas metodologías son:

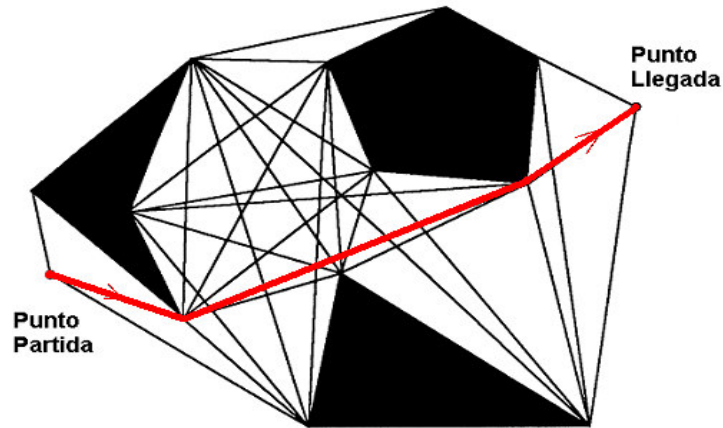
3.4.1 Mapas de Caminos.

El objetivo es aprehender la conectividad del espacio libre del robot en una red de curvas unidimensionales, que reciben comúnmente el nombre de mapas de caminos. Una vez construido este mapa se usa un conjunto de rutas que podrían denominarse estandarizadas, por el cual puede transitar el robot sin colisionar. Resuelto el problema de esta forma entonces lo único que queda es conectar el punto de partida con el de llegada y unir estos dos puntos para obtener un planeamiento de camino según cierto criterio. Existen varios métodos basados en esta idea como son:

1) Gráficos de visibilidad. Este método se aplica a espacios de configuraciones bidimensionales con obstáculos poligonales. Constituye un grafo con nodos correspondientes a los puntos iniciales y finales, así como los vértices de los obstáculos como se muestra en la figura 14.

Cada una de las líneas de enlace del grafo es un segmento rectilíneo que conecta a dos nodos y que no intercepta el interior de ningún obstáculo lo que constituye diversas alternativas de solución. Una vez establecidas las líneas de enlace, se procede a encontrar la ruta mas corta entre las dos configuraciones inicial y final.

Figura 14. Gráfica de visibilidad en el espacio de configuración bidimensional con obstáculos poligonales.



Fuente: Jean-Claude Latombe. Robot Motion Planning, p. 13

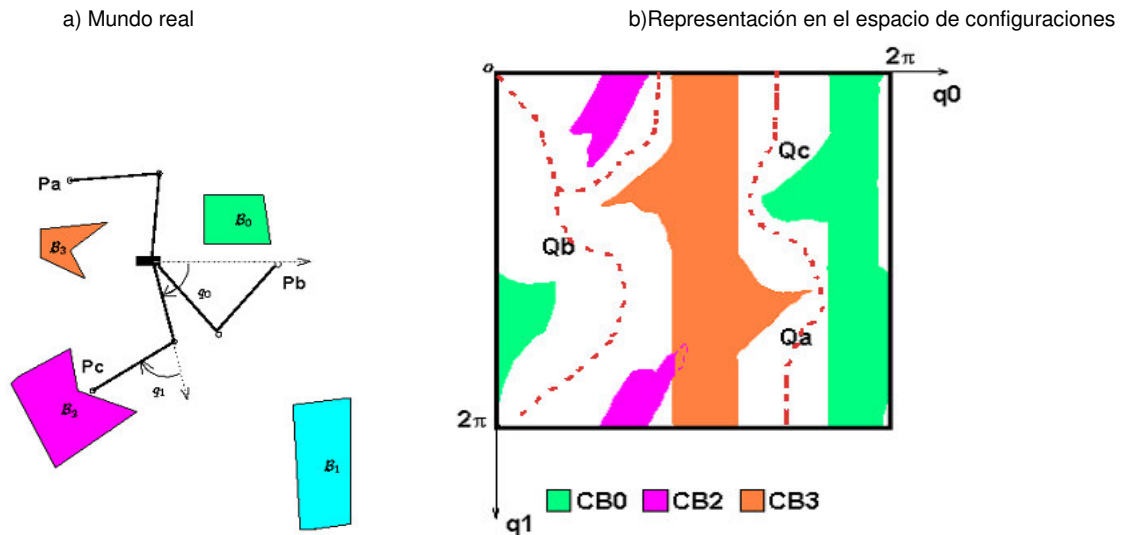
2) Retracción. Esta metodología consiste en definir una función continua del espacio libre del robot dentro de un subconjunto unidimensional de la misma, sujeta a la restricción de que esta función sea el mapa identidad. En topología esto se conoce como retracción a una función. En un espacio de configuración bidimensional, el espacio libre del robot se retracta típicamente dentro de lo que se conoce como diagrama de Voronoi. Este diagrama es el conjunto de todas las configuraciones libres cuya distancia hacia las regiones de obstáculos es mínima; lo anterior se consigue con al menos dos puntos en el límite del espacio del robot.

La ventaja de este método es que produce rutas libres que tienden a maximizar el espacio entre el robot y los obstáculos. Cuando el espacio de los obstáculos se modela como polígonos, este diagrama consta de segmentos rectilíneos y parabólicos, como se muestra en la figura15.

3) Vías libres. Este método se basa en la misma idea de retractar el espacio libre dentro de su diagrama de Voronoi, es decir mantener al robot lo mas lejos posible de los obstáculos. El método consiste en extraer figuras geométricas del espacio de trabajo conectándolas en una gráfica la cual recibe el nombre de vías libres, en la que posteriormente se buscará una ruta.

Su aplicación está restringida a los casos en que el robot se modela como un polígono que rota y se traslada entre obstáculos poligonales.

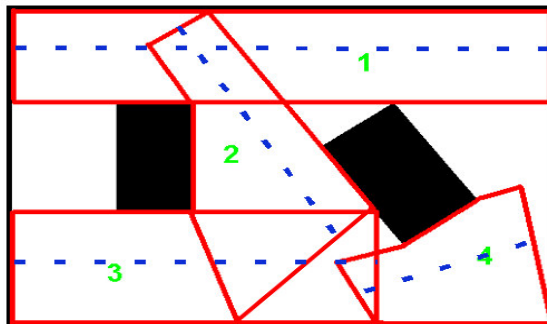
Figura 15. Brazo de dos grados de libertad ubicado entre obstáculos.



Fuente: <http://www-2.cs.cmu.edu/afs/cs/project/jair/pub/volume9/mazer98a-html/node2.html>

La vía libre es un cilindro generalizado que consiste en segmentos rectilíneos y el eje de dicho cilindro constituye lo que se denomina espina y es la ruta por donde puede transitar el robot, como se muestra en la figura 16.

Figura 16. Representación en vías libres, sobre un espacio de trabajo con obstáculos en forma de polígonos.



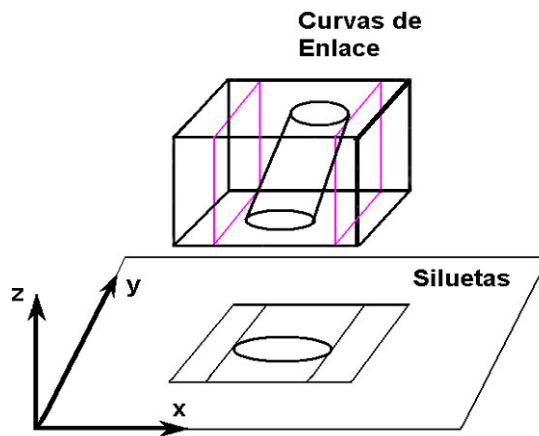
Fuente: Jean-Claude Latombe. Robot Motion Planning. P. 177.

4) Siluetas. Este método tiene la característica de resolver el problema de una forma exponencial referente a la dimensión del espacio de configuración.

Básicamente consiste en construir la silueta del espacio libre del robot cuando se va desde un punto en el infinito y añadiendo algunos segmentos curvos que enlacen los puntos críticos de la silueta a otros segmentos curvos de la silueta (ver figura 17).

La silueta y las curvas de enlace forman el mapa de caminos en el cual se buscará la ruta para el robot.

Figura 17. Método de las siluetas y la generación de una de ellas.



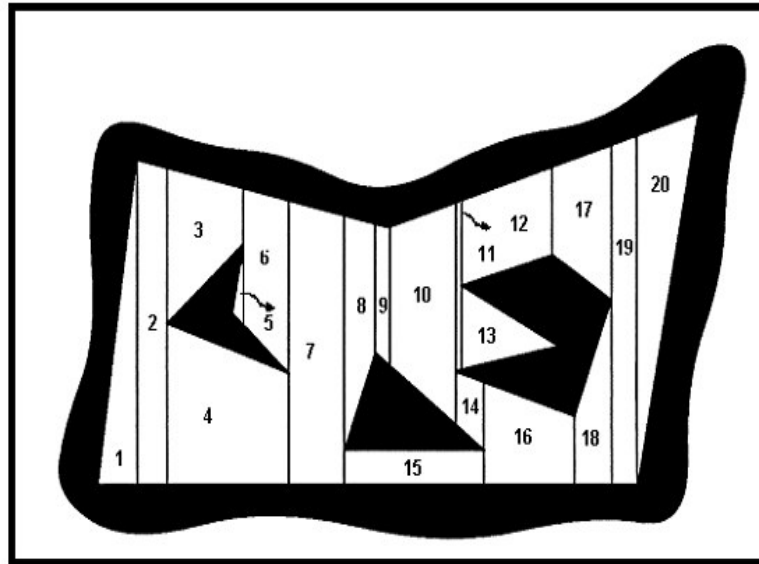
Fuente: Autor del proyecto

3.4.2 Descomposición En Celdas.

Estos métodos son los mas estudiados a la fecha y consisten en descomponer el espacio libre del robot en regiones simples, llamadas celdas, de una forma tal que generen fácilmente una ruta entre dos configuraciones cualesquiera. Paso seguido se construye un grafo no dirigido que representa la relación de adyacencia entre las celdas, para luego proceder a realizarse la búsqueda de una ruta en éste.

1) Exacta. Para este caso se descompone el espacio libre en celdas cuya unión es exactamente el espacio libre total del robot. La figura 18 muestra una descomposición exacta en un espacio de configuración bidimensional. El espacio libre se encuentra delimitado externamente por un polígono e internamente por tres polígonos; dicho espacio se descompone exactamente en celdas poligonales las cuales se construyen dibujando líneas verticales desde los vértices de los obstáculos.

Figura 18. Método exacto de descomposición en celdas



Fuente: Jean-Claude Latombe. Robot Motion Planning. P. 206.

2) Aproximada. Para este método se producen celdas de una forma predefinida cuya unión esta restringida únicamente al espacio libre (ver figura 19). El límite de una celda no caracteriza una discontinuidad de ningún tipo y no tiene ningún significado en particular. En pocas palabras este método no permite descomponer el espacio en forma exacta, por el contrario en una forma aproximada algo conservadora. De igual forma que en el método anterior se construye una gráfica de conectividad que representa la relación de adyacencia entre las celdas, para luego realizar la búsqueda de una ruta que una el punto de partida con el de llegada.

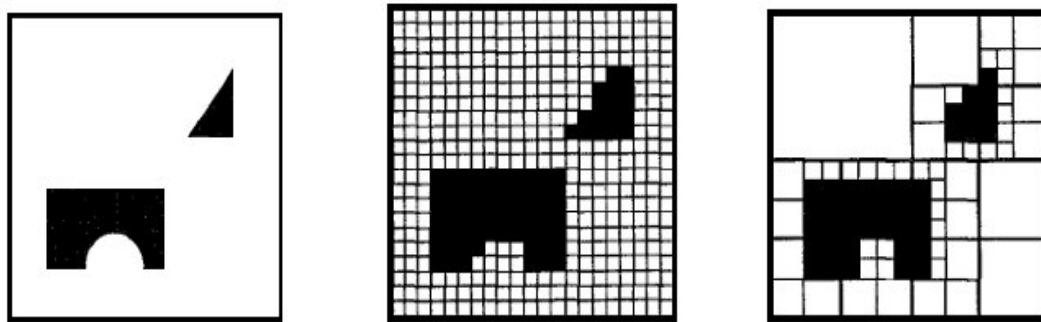
Dentro de esta misma metodología se puede acuñar una conocida como la de árboles cuaternarios, la cual consiste en dividir el espacio de trabajo en cuatro rectángulos, si el interior de un rectángulo se encuentra totalmente en el espacio libre no se divide mas, de lo contrario se procede a dividirlo en otros cuatro rectángulos y así sucesivamente hasta lograr cierta resolución predeterminada.

3.4.3 Campos de Potencial.

Una forma simple de atacar el problema del planeamiento es discretizar el espacio de configuraciones en una rejilla de tipo regular, en la cual se buscará una ruta libre (ver figura 20). Generalmente para tener éxito es necesario recurrir a alguna poderosa heurística la cual guíe la búsqueda; se han propuesto gran cantidad de ellas pero una de las de mayor éxito ha sido la de campos de potencial. La idea es

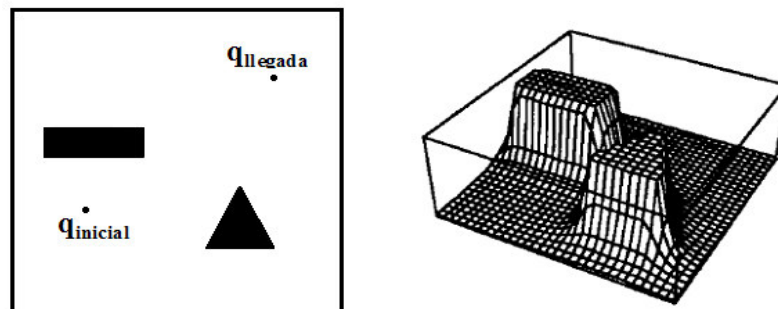
que el robot, el cual se representa como un punto en el espacio, es una partícula que se mueve bajo la influencia de un campo de potencial producido por la meta y los obstáculos. Generalmente la meta genera un potencial de atracción, mientras que los obstáculos un potencial de repulsión. El gradiente negativo del potencial total se trata como una fuerza artificial que se le aplica al robot. En cada configuración, la dirección de esta fuerza se considera como la mas promisoría para el robot. Comparado con otros métodos suele ser muy eficiente pero se enfrenta al problema de ser un método de optimización de descenso rápido lo que hace que fácilmente se quede atrapado en mínimos locales.

Figura 19. Tendencias en la representación de objetos.



Fuente: Hwang. Gross Motion Planning, A survey P. 231.

Figura 20. Representación de un espacio de trabajo mediante campos de potencial.



Fuente: Jean-Claude Latombe. Robot Motion Planning, p. 20.

3.4.4 Programación Matemática.

Parten de un conjunto de inecuaciones, las cuales representan el movimiento del robot y su interacción con los obstáculos. Generalmente se requiere aplicarles un algoritmo para optimizarlas y encontrar un camino lo mas adecuado posible el cual consiste en una curva que une la configuración de partida con la de llegada

minimizada con respecto a un escalar. Diversos métodos numéricos son de común uso, dada la complejidad que generan las diversas ecuaciones no lineales producidas.

3.5 TÉCNICAS DE BÚSQUEDA

Según sea el problema de planeamiento de movimiento que se tenga, se requiere de una estrategia que a partir de las soluciones obtenidas permita encontrar la solución. Si el problema es de tipo off-line, la mayoría de soluciones pueden ser consideradas y por tanto es posible encontrar el óptimo, en el caso que este exista. De las técnicas existentes no todas están en capacidad de encontrar la solución óptima, pero pueden encontrar en forma rápida una solución dentro de un conjunto particular de restricciones. Es común clasificar estas estrategias según diversas características entre las que se encuentran: ser completas(exactas y heurísticas) y el alcance que posean (globales y locales). Las exactas encuentran solución o en su defecto indican la ausencia de esta; tienen un alto costo computacional. Las heurísticas dan una solución en corto tiempo; pueden fallar, encontrar soluciones muy “pobres” o en problemas de un alto grado de complejidad encontrar solución.

En el caso de la característica de ser completos se menciona también desde dos puntos de vista: completitud en resolución y completitud probabilística. La referente a la resolución hace alusión al caso de tener que discretizar parámetros tales como dimensiones de los obstáculos, espacio de configuraciones entre otras. Un algoritmo se considera exacto en el límite de igual forma como lo discreto tiende a lo continuo.

En el caso de ser probabilístico lo completo se refiere al hecho de que la probabilidad de encontrar una respuesta se acerque a uno; esta característica es propia de aquellos algoritmos que usan la búsqueda aleatoria como son los AG y el recocido simulado, la desventaja de estos métodos es el gran costo que enfrentan en cuanto a tiempo se refiere. Lo global se refiere a la búsqueda en todo el espacio de trabajo y lo local a aquella búsqueda que se realiza en la vecindad del robot; es comúnmente usada para resolver el problema en ambientes con cierto grado de incertidumbre los cuales haciendo uso de un método global se encuentran con la aparición de obstáculos que en el modelo inicial no se contemplaron. Algunos ejemplos de técnicas de búsqueda pueden ser: búsqueda en profundidad, búsqueda en extensión, búsqueda del mejor primero, el algoritmo A*, la búsqueda bi-direccional y los métodos aleatorios (AG y recocido simulado). En el presente proyecto se hará uso de la descomposición en celdas y la aplicación de una estrategia heurística aleatoria y global mediante la aplicación de los AG.

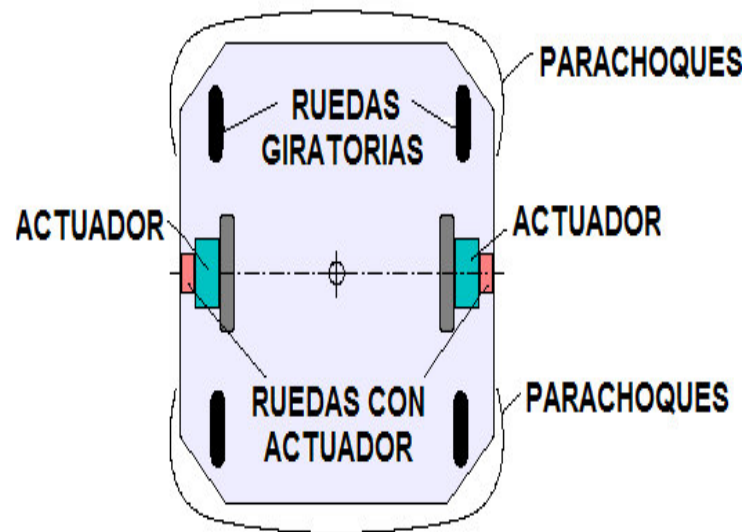
3.6 ROBOTS MÓVILES

Los robots móviles están provistos de patas, ruedas u orugas que los capacitan para desplazarse de acuerdo a su programación. Elaboran la información que reciben a través de sus propios sistemas de sensores y se emplean en determinado tipo de instalaciones industriales, sobre todo para el transporte de mercancías en cadenas de producción y almacenes. También se utilizan robots de este tipo para la investigación en lugares de difícil acceso o muy distantes, como es el caso de la exploración espacial y las investigaciones o rescates submarinos. Hay una gran variedad de modos de moverse sobre una superficie sólida; los vehículos de ruedas son, con mucho, los más populares por varias razones prácticas. Los robots con ruedas son relativamente más sencillos y más fáciles de construir además la carga que pueden transportar es un tanto mayor. Los robots basados en cadenas y en patas se pueden considerar generalmente más complicados y pesados que los robots de ruedas para una misma carga útil. A esto se puede añadir el que se pueden transformar vehículos de ruedas de radio-control para usarlos como bases de robots. La principal desventaja de las ruedas es su empleo en terrenos irregulares, en los que se comportan bastante mal; normalmente un vehículo con ruedas podrá sobrepasar un obstáculo que tenga una altura inferior al radio de sus ruedas, aunque existirán terrenos en los cuales esta condición es bastante crítica y el uso de las ruedas llegará a no ser práctico. Para robots que vayan a funcionar en un entorno natural las cadenas son una opción muy buena porque permiten al robot superar obstáculos relativamente mayores y son menos susceptibles que las ruedas de sufrir daños por el entorno, como piedras o arena. El principal inconveniente de las cadenas es su ineficiencia, puesto que se produce deslizamiento sobre el terreno al avanzar y al girar. Si la navegación se basa en el conocimiento del punto en que se encuentra el robot y el cálculo de posiciones futuras sin error, entonces las cadenas acumulan tal cantidad de error que hace inviable la navegación por este sistema. Potencialmente los robots con patas pueden superar con mayor facilidad que los otros los problemas de los terrenos irregulares.

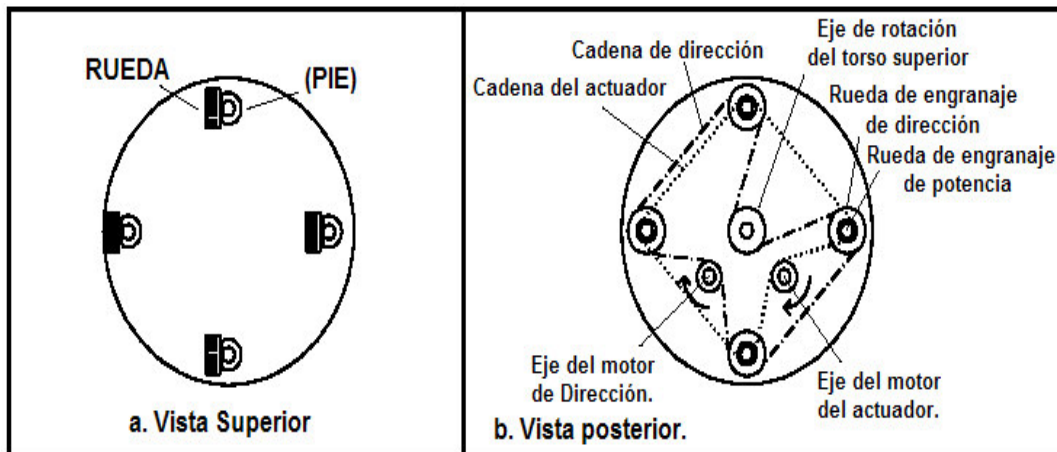
A pesar de que hay un gran interés en diseñar este tipo de robots, su construcción plantea numerosos retos. Estos retos se originan principalmente en el gran número de grados de libertad que requieren los sistemas con patas. Cada pata necesita como mínimo un par de motores lo que produce un mayor costo, así como una mayor complejidad y menor fiabilidad. Es más los algoritmos de control se vuelven más complicados por el gran número de movimientos a coordinar, los sistemas de patas son un área de investigación muy activo. Algunos tipos de sistemas usados en robots móviles se ilustran en la figura 21. Para el presente trabajo se tendrá como referencia el presentado en la figura 21c.

Figura 21. Tipos de actuadores para robots móviles.

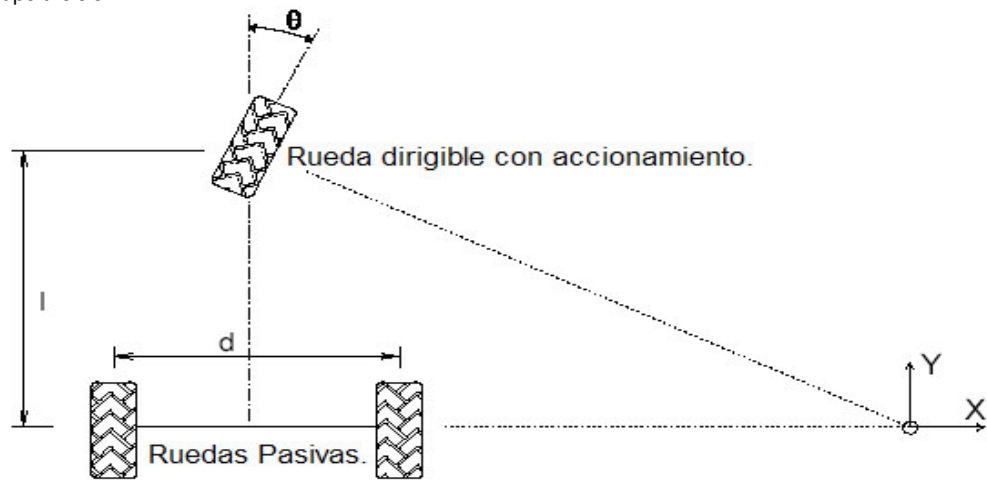
a) Actuador diferencial.



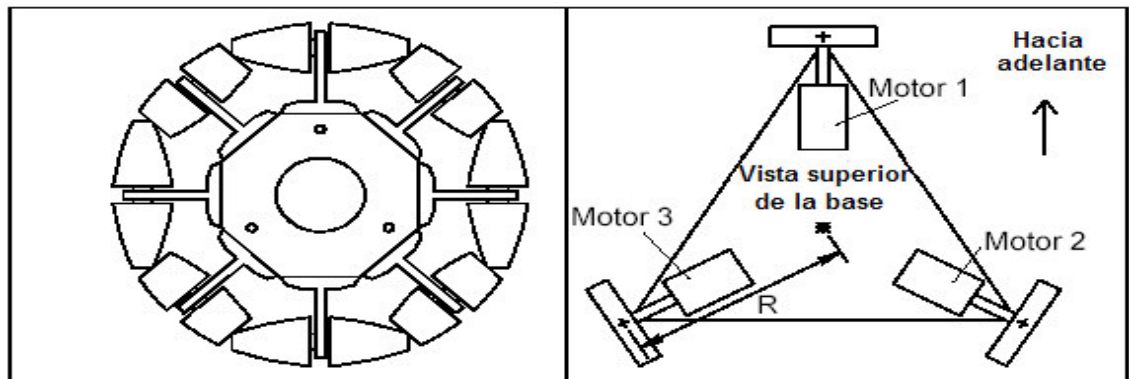
b) Actuador tipo sincro.



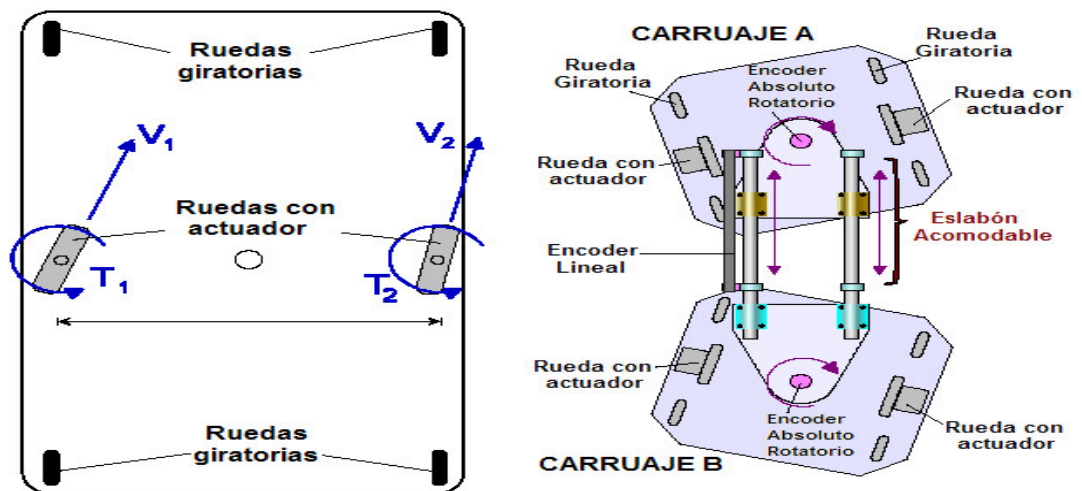
c) Actuador tipo triciclo.



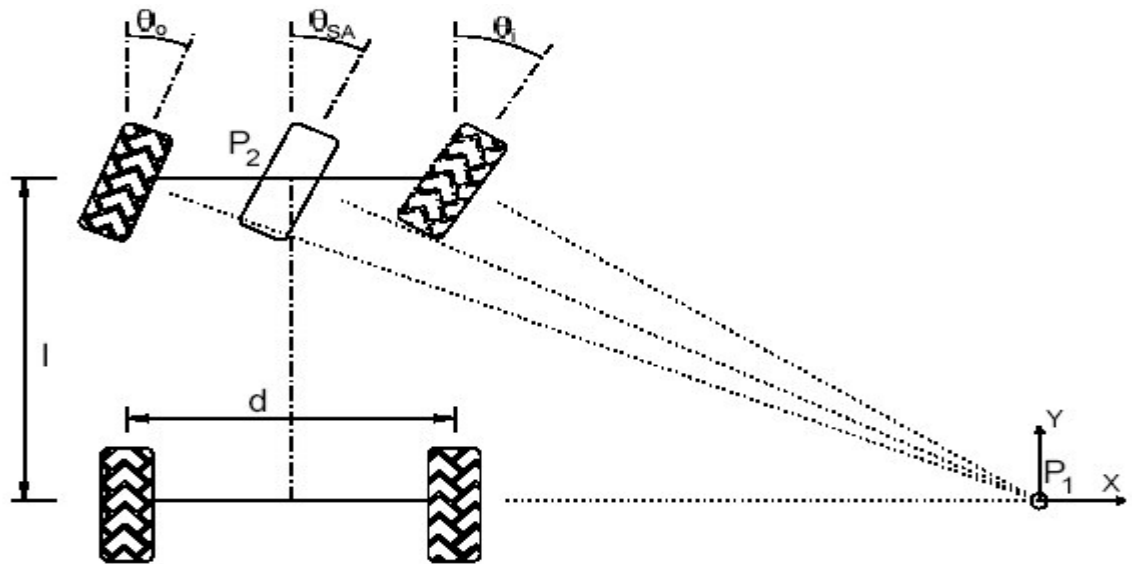
d) Actuador omnidireccional.



e) Vehículos de múltiples grados de libertad.



f) Actuador de dirección tipo Ackerman.



Fuente: BORESTEIN, J. "Where am I?". Sensors and methods for mobile robot positioning. The University of Michigan; p.19-28.

4. MODELOS DEL ROBOT Y DEL AMBIENTE DE TRABAJO

4.1 INTRODUCCIÓN

Como se citó en el capítulo anterior, existen diversidad de métodos para la obtención de caminos en robots móviles que generalmente se basan en hipótesis simplificadoras, tales como: ambiente conocido y estático, robots omnidireccionales con movimientos lentos y seguimiento de trayectorias casi perfecto entre otras. En definitiva gran cantidad de métodos se han planteado en el estado del arte para encontrar caminos libres de colisión que pueden ser optimizados en cuanto a distancia, especialmente en entornos que se modelan como polígonos y planteamientos tan particulares que el espacio libre se modela tratando de encontrar caminos que vayan por el centro del mismo¹⁹. Para facilitar el proceso de búsqueda se recurre a técnicas tales como la descomposición del espacio mediante celdas, utilización de restricciones de diversos niveles de resolución y búsqueda jerarquizada, que demuestran su enorme eficiencia en cuanto aplicaciones en tiempo real se refiere. La planificación de trayectoria entonces requiere realizarse en forma dinámica en la cual se considera la posición actual del robot y los puntos intermedios de paso definidos en la programación previa del camino. Para la definición de dicha trayectoria se requiere tener en cuenta las características cinemáticas del vehículo sea cual sea su disposición de ruedas y actuadores. Aparte de las características geométricas y cinemáticas es de gran importancia tener en cuenta los modelos dinámicos de comportamiento del vehículo manteniendo presente la interacción vehículo-terreno. Una vez realizada la planificación de trayectoria se procede a planear movimientos concretos y controlarlos para mantener al vehículo en la trayectoria proyectada. En la mayoría de casos el control automático con niveles finos aplicado a robots móviles resulta mas complejo que el de los manipuladores debido a la presencia de restricciones no holonómicas. Para un manipulador la experiencia ha demostrado que es relativamente fácil el cálculo y la medición de los respectivos pares y fuerzas que se ejercen sobre la estructura mecánica; en el caso de los robots móviles esta situación resulta bastante complicada²⁰. Aunque es costumbre el uso de métodos geométricos y modelos cinemáticos simplificados resulta de mayor conveniencia considerar el aspecto dinámico, especialmente en el manejo de velocidades altas. Sumado a este hecho se requiere disponer de medidas de la posición y orientación en intervalos de tiempo lo suficientemente cortos, para lo cual ha sido comúnmente utilizada la odometría, es decir la utilización de la información a través de los sensores situados en los ejes del movimiento típicamente codificadores ópticos. La otra tendencia ha sido el uso de sistemas de navegación inercial incluyendo giróscopos y acelerómetros; estos aspectos no son considerados en el presente trabajo, se considera ambiente conocido y estático,

¹⁹ OLLERO, Anibal. Robótica manipuladores y robots móviles. España: 2001.p.9.

²⁰ Ibid.,p.10.

información que se puede obtener a partir de una cámara. En contraparte los aspectos que se han pretendido estudiar son los referentes a un modelo cinemático y dinámico lo más completo posible con el fin de poder tener un conocimiento amplio del comportamiento respectivo para así poder desarrollar un planeamiento de trayectoria con características de robustez y soluciones óptimas, aunque por facilidad computacional se presenta un modelo simplificado el cual se estudia desde una perspectiva mas simple. Adicional a esto se presenta un breve estudio acerca del modelo propuesto del ambiente de trabajo utilizado en la solución del problema en estudio.

4.2 MODELO DEL ROBOT MÓVIL

4.2.1 Modelo cinemático.

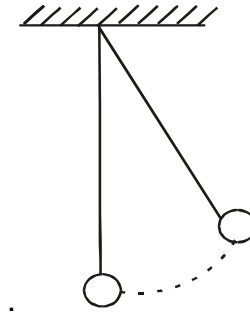
Para el estudio del modelo cinemático y dinámico del robot sobre el cual se quiere trabajar es importante recurrir al estudio de sistemas mecánicos que poseen restricciones no holonómicas, dichos sistemas mecánicos no holonómicos tienen diversas interpretaciones dependiendo del enfoque en el que se planteen, pero la forma más común de especificarlos es como aquellos sistemas en los que el movimiento está condicionado por restricciones no integrables, es decir restricciones sobre las velocidades del sistema que no surgen desde limitaciones únicamente propias de las configuraciones.

Una forma de decir lo que es un sistema no holonómico es planteando lo que es un sistema holonómico. Estos sistemas están sujetos a restricciones que limitan sus posibles configuraciones como es planteado por Bloch²¹; la palabra holonómico u holónomo proviene de palabras griegas que significan integral, refiriéndose al hecho de que tales restricciones son dadas como limitaciones en la velocidad, las cuales pueden ser integradas y re-expresadas como restricciones en las variables de las configuraciones; desde una óptica matemática un sistema holonómico cumple las condiciones de Frobenius. Un ejemplo bastante sencillo es el caso de la prohibición a la cual está sometido la longitud de un péndulo simple (figura 22). Para el caso de un sistema no holonómico este posee restricciones no integrables en sus velocidades, un ejemplo de ello es el caso de un disco o una bola que rueda.

En pocas palabras un sistema es holonómico cuando las restricciones que presenta no dependen de sus velocidades en el caso no holonómico ocurre todo lo contrario. Una interpretación adicional que se da comúnmente, pero mas desde el punto de vista del control es que un sistema no holonómico, es aquel que no es controlable.

²¹ BLOCH, A.M. Nonholonomic Mechanics and Control. New York: Springer, 2003. p.11.

Figura 22. Péndulo simple como ejemplo de un sistema holonómico.

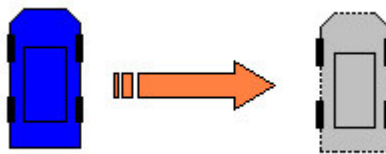


$$x = l \sin(\theta) \quad y = l \cos(\theta)$$

Fuente: Autor del proyecto

Otra interpretación de lo que es un sistema holonómico es entendiéndolo como aquel en el cual entre las posibles posiciones que puede tomar todas son concebibles no solo mediante movimientos continuos sino también como movimientos posibles; para el caso de un sistema no holonómico este restringe los tipos de movimiento pero no la posición. Esto quiere decir que un sistema no holonómico para llegar a una determinada posición puede no hacerlo mediante movimientos simples como por ejemplo el caso de un carro (figura 23) en el cual, para pasar a una posición transversal, no lo puede realizar en movimientos sencillos.

Figura 23. Ejemplo de un sistema no holonómico.



Fuente: Autor del proyecto.

La presencia de este tipo de consideraciones hace imperativo introducirse al estudio de los sistemas mecánicos en los cuales resultan conceptos importantes a resaltar como es el de coordenadas generalizadas y grados de libertad. Las coordenadas generalizadas se pueden definir exactamente como un grupo de cantidades escalares las cuales especifican completamente cualquier configuración (posición) del sistema; se notan generalmente mediante un vector como se muestra a continuación:

$$q = (q_1, q_2, q_3, \dots, q_n) \quad (24)$$

Al número de cantidades requeridas para la especificación de todas las configuraciones permisibles de un sistema respecto a una referencia, se les llama grados de libertad. Se establece entonces que si existen m restricciones cinemáticas independientes en un sistema con n coordenadas generalizadas, el número de grados de libertad dof (*degree of freedom*) está dado por:

$$dof = n - m \quad (25)$$

La notación comúnmente aceptada para las restricciones holonómicas y no holonómicas son respectivamente:

$$\begin{aligned} G(q, t) &= a(q) = 0 \\ G(q, \dot{q}, t) &= a(q)\dot{q} = 0 \end{aligned} \quad (26)$$

Para la definición adecuada del modelo de un robot se hace necesario realizar ciertas consideraciones que a simple vista tienen una connotación bastante simple, pero en definitiva, juegan un papel importante al realizar ciertas simplificaciones. Entre las consideraciones mas comunes aparecen²²:

- ◆ El robot se mueve sobre una superficie plana, con lo cual el contacto con ella se da a través de un solo punto.
- ◆ Los ejes del vehículo son perpendiculares al suelo.
- ◆ Rodadura pura.
- ◆ Las ruedas del robot tienen forma de discos rígidos.
- ◆ El robot no posee partes flexibles.
- ◆ En algunas situaciones, se considera masa concentrada, por tanto los momentos de inercia de ruedas y demás accesorios se desprecian.
- ◆ Limitaciones en cuanto al par producido.
- ◆ Caminos consistentes en formas geométricas simples, tales como líneas rectas y arcos circulares

Para el presente estudio se pretende trabajar con un robot tipo triciclo clásico como el que aparece en la figura 24 sobre el cual pueden tenerse al menos dos consideraciones como son: la rueda delantera sirve para tracción y direccionamiento y el eje trasero posee dos ruedas pasivas; el otro caso consiste en que la rueda delantera (rueda loca) sea pasiva y las ruedas traseras sean las encargadas de la tracción y orientación.

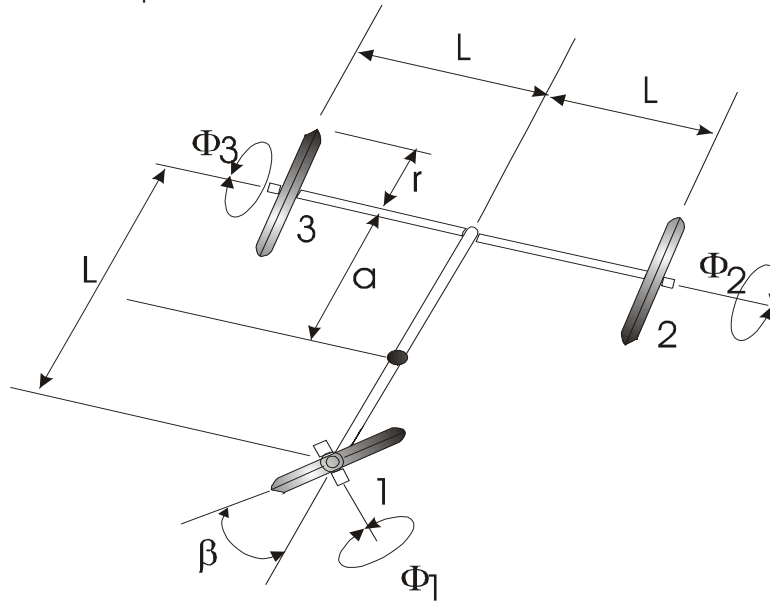
²² SHILLER, Zvi. Dynamic Motion Planning of Autonomous Vehicles. En: IEEE Vol.7, No.2 (April 1991); p..241.

La posición global del robot se dará mediante los valores (x, y, θ) . El proceso a seguir parte de caracterizar en forma general a una rueda.

El centro B de la rueda está conectado al trole por un vástago rígido de longitud d , el cual puede girar alrededor de un eje vertical, como lo ilustra la figura 25. La posición de este vástago con respecto al trole está dado por dos constantes: l y α : el ángulo de rotación de la rueda se indicará por β y la orientación del plano de la rueda con respecto al vástago se denotará por γ .

El radio de cada rueda será R y el ángulo de giro alrededor de su eje horizontal recibirá el nombre de ϕ . En general la posición de cualquier rueda respecto al sistema se dará por el siguiente grupo de valores: $\{R, l, d, \alpha, \gamma\}$ y su movimiento por dos variables angulares: $\{\phi(t), \beta(t)\}$, respectivamente rotación y orientación.

Figura 24. Modelo del robot móvil tipo triciclo.



Fuente: Autor del proyecto.

En definitiva para el caso del triciclo se tienen los siguientes parámetros con sus valores:

RUEDAX={	R	l	d	α	β	γ	ϕ	(27)
RUEDA1={	R	$l=L$	d	270°	β	90°	ϕ_1	
RUEDA2={	R	L	0	0°	0°	0°	ϕ_2	
RUEDA3={	R	L	0	180°	0°	0°	ϕ_3	

Para determinar el comportamiento del sistema se procede a determinar la velocidad de cada rueda en el punto de contacto con la superficie de donde se extraen las respectivas componentes de la velocidad en el plano de la rueda y en uno ortogonal a este. Las dos ecuaciones son respectivamente²³.

$$\begin{aligned} V_{Tx} &= A_x \dot{x} + B_x \dot{y} + [(d_x \cos(\gamma_x) + l_x \cos(\beta_x + \gamma_x))] \dot{\theta} + d_x \cos(\gamma_x) \dot{\beta}_x + R_x \dot{\phi}_x \\ A_x &= [-\sin(\alpha_x + \beta_x + \gamma_x) \cos(\theta) - \cos(\alpha_x + \beta_x + \gamma_x) \sin(\theta)] \\ B_x &= [-\sin(\alpha_x + \beta_x + \gamma_x) \sin(\theta) + \cos(\alpha_x + \beta_x + \gamma_x) \cos(\theta)] \end{aligned}$$

Reemplazando en las ecuaciones (28) los valores respectivos de cada uno de los parámetros y expresándolas en forma matricial se obtienen las expresiones que se muestran a continuación. Para que se cumplan las condiciones de pura rodadura y no deslizamiento es restricción necesaria que cada una de estas velocidades sea cero, razón por la cual las ecuaciones aparecen igualadas a cero.

57

$$\begin{bmatrix} vtrueda1 \\ vtrueda2 \\ vtrueda3 \end{bmatrix} = \begin{bmatrix} -\sin(\beta) & \cos(\beta) & -L*\sin(\beta) \\ 0 & 1 & L \\ 0 & -1 & L \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} vnrueda1 \\ vnrueda2 \\ vnrueda3 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & \sin(\beta) & d + L*\cos(\beta) \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \dot{\beta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

El movimiento del robot queda completamente descrito mediante el siguiente vector de coordenadas generalizadas.

$$q(t) = (x, y, \theta, \beta, \phi_1, \phi_2, \phi_3) \quad (29)$$

Por la definición anterior las restricciones se puede entonces expresar según la ecuación 29 como aquí se muestra.

$$\begin{aligned} [vtruedas] &= J_1 * R * \dot{P} + J_2 \dot{\phi} = \vec{0} \\ [vnruedas] &= C_1 * R * \dot{P} + C_2 \dot{\beta} = \vec{0} \end{aligned} \quad (30)$$

Donde:

$$\begin{aligned} J_1 &= \begin{bmatrix} -\sin(\beta) & \cos(\beta) & -L*\sin(\beta) \\ 0 & 1 & L \\ 0 & -1 & L \end{bmatrix} \quad J_2 = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \\ R &= \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad C_1 = \begin{bmatrix} \cos(\beta) & \sin(\beta) & d + L*\cos(\beta) \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad C_2 = \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} \quad \dot{P} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \\ \dot{\phi} &= \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} \end{aligned}$$

Para obtener los grados de libertad se procede a calcular el rango de la matriz de restricciones, la cual mediante algunas manipulaciones se encuentra que es igual a cinco y por tanto el valor de grados de libertad es de dos. Esto también es consecuencia que las restricciones respectivas a las ruedas dos y tres sean equivalentes.

$$\dot{x} * \cos(\theta) + \dot{y} * \text{sen}(\theta) = 0 \quad (31)$$

De la ecuación 31 se tiene entonces que para cualquier grupo de valores de (x, y, θ) , que cumplan la respectiva restricción, existe entonces un solo valor de β y uno de ϕ el cual satisfaga las otras restricciones. Estos valores son entonces:

$$\dot{\beta} = -C_2^{-1} * C_1 * R * \dot{P} \quad (32)$$

$$\dot{\phi} = -J_2^{-1} * J_1 * R * \dot{P}$$

De lo anterior los modelos cinemáticos directos e inversos son respectivamente

$$\dot{P} = -R^{-1} * J_1^{-1} * J_2 * \dot{\phi}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = - \begin{bmatrix} \cos(\theta) & -\text{sen}(\theta) & 0 \\ \text{sen}(\theta) & \cos(\theta) & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{-1}{\text{sen}(\beta)} & \frac{\cos(\beta) - \text{sen}(\beta)}{2 * \text{sen}(\beta)} & \frac{-\cos(\beta) - \text{sen}(\beta)}{2 * \text{sen}(\beta)} \\ 0 & \frac{1}{2} & \frac{-1}{2} \\ 0 & \frac{1}{2 * L} & \frac{1}{2 * L} \end{bmatrix} * J_2 * \dot{\phi}$$

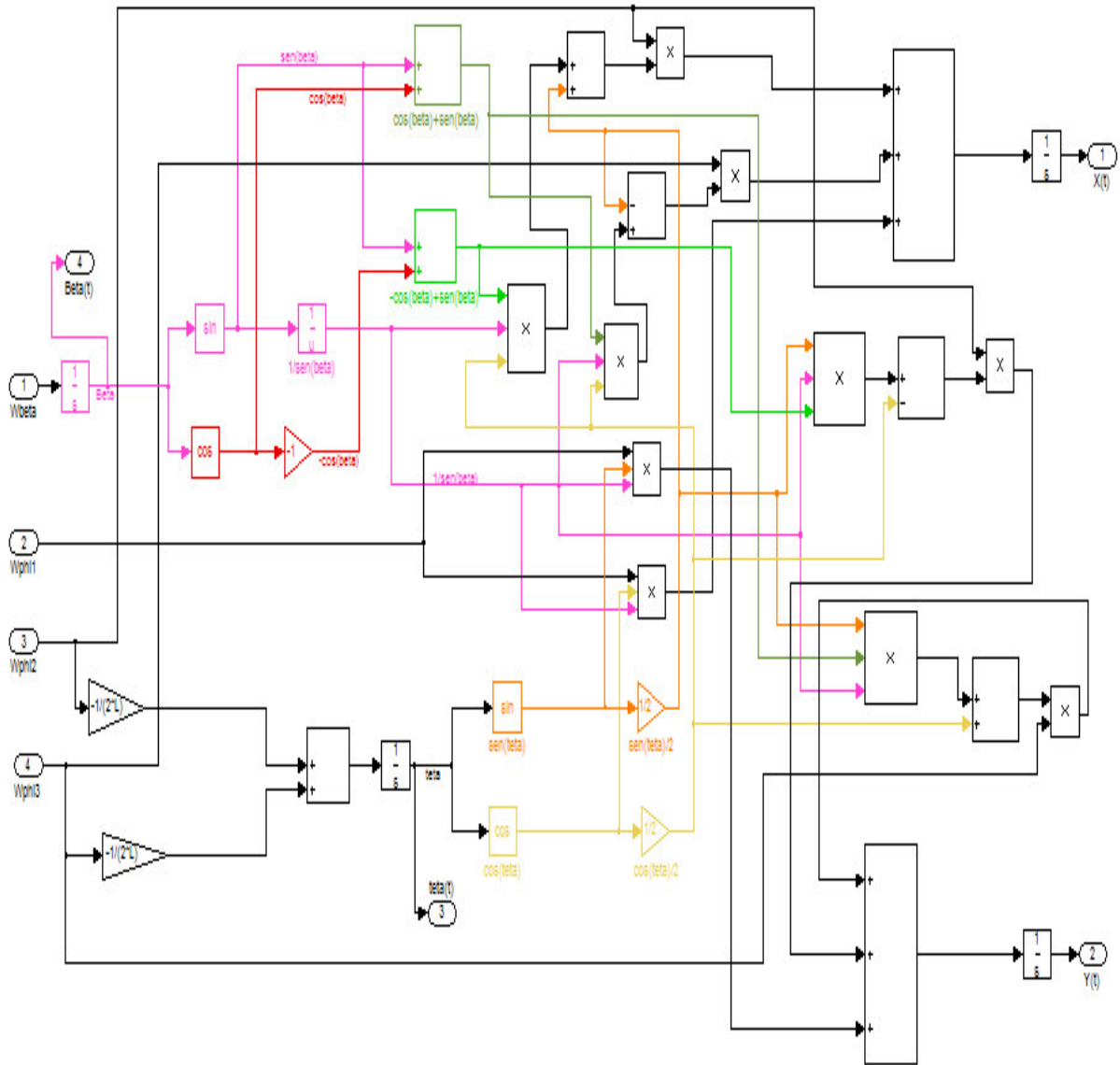
$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \frac{\text{sen}(\beta)}{R} & \frac{-\cos(\beta)}{R} & \frac{L * \text{sen}(\beta)}{R} \\ 0 & \frac{-1}{R} & \frac{-L}{R} \\ 0 & \frac{1}{R} & \frac{-L}{R} \\ \frac{-\cos(\beta)}{d} & \frac{-\text{sen}(\beta)}{d} & \frac{-(d + L * \cos(\beta))}{d} \end{bmatrix} \begin{bmatrix} \dot{x} * \cos(\theta) + \dot{y} * \text{sen}(\theta) \\ -\dot{x} * \text{sen}(\theta) + \dot{y} * \cos(\theta) \\ \dot{\theta} \end{bmatrix} \quad (33)$$

Los respectivos modelos simulados con Simulink de Matlab se muestran en la figura 26.

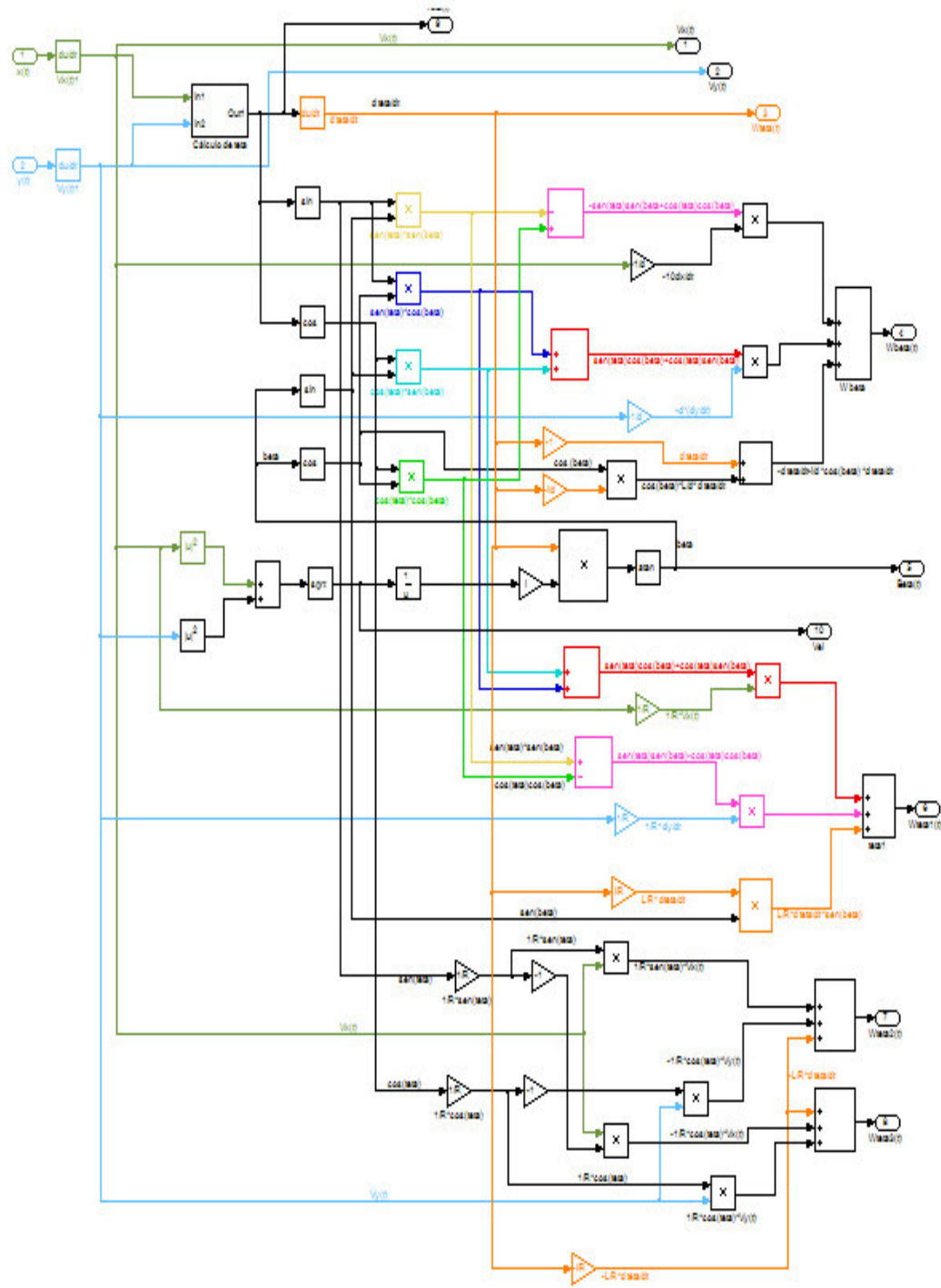
Los archivos que contienen estos modelos se encuentran en el CD anexo en la carpeta modelos y llevan los nombres respectivamente de “triciclodirecto1.mdl” y “tricicloinverso1.mdl”.

Figura 26. Modelo cinemático del robot teniendo en cuenta el efecto de las tres ruedas.

a) Modelo cinemático directo.



b) Modelo cinemático inverso.



Fuente: Autor del proyecto.

4.2.2 Modelo dinámico.

En este estudio se busca presentar el modelo dinámico el cual es clave para aspectos referentes al control y algún proceso de optimización como el que se desarrolla en el presente trabajo, solo con la diferencia que la función a optimizar sea la energía (en la actual exposición es la distancia). Un aspecto el cual es de igual forma clave y que no se consideró en este proyecto son los detalles propios de este análisis y que se hace preponderante a la hora de construir un robot móvil. Como es obvio el análisis previo se basó únicamente en el aspecto geométrico y nada se citó en cuanto a las fuerzas generalizadas (fuerzas en sí o pares) que podían producir las velocidades y respectivas aceleraciones. En muchos sistemas el control se ha realizado mediante el modelo cinemático²⁴, estos resultados se consideran válidos siempre y cuando los movimientos del robot se supongan “suaves”. Es además importante involucrar en este estudio los conceptos relacionados con las distribuciones de masas y las inercias y al hablar de robots móviles se hace importante considerar no solo la dinámica del chasis del robot si no también la interacción de este con el terreno.

Para el análisis del modelo dinámico es clave el estudio del Lagrangiano el cual permite estimar la energía consumida por todo el sistema. Como se citó en el ítem anterior las coordenadas generalizadas fueron $q(t) = (x, y, \theta, \phi_1, \phi_2, \phi_3)$, por lo que el Lagrangiano se escribe como se muestra a continuación:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{P}} \right) - \frac{\partial T}{\partial P} = R^T * (J_1^T * \lambda + C_{10}^T * \mu + C_{11}^T \nu) \quad (34)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}} \right) - \frac{\partial T}{\partial \beta} = C_{22}^T \mu + B_1 * u_1$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\phi}} \right) - \frac{\partial T}{\partial \phi} = J_2^T \lambda + B_2 * u_2$$

$$\begin{aligned} C_{10} &= [\cos(\beta) \quad \sin(\beta) \quad d + L * \cos(\beta)] \\ C_{11} &= [1 \quad 0 \quad 0] \\ C_{22} &= [\dot{\beta}] \end{aligned} \quad (35)$$

Donde las variables $\lambda = [\lambda_1, \lambda_2, \lambda_3]$, μ y ν son los cinco multiplicadores de Lagrange asociados con las restricciones cinemáticas independientes. Bu es el conjunto de fuerzas externas generalizadas (torques) aplicadas al sistema; estas (siete) ecuaciones asociadas a las restricciones cinemáticas describen el comportamiento dinámico del sistema.

²⁴ OLLERO, Op. cit., p.116.

Es conveniente diferenciar dos posibles casos según se especifique la posición del actuador, una de ellas considera que los motores están en las ruedas dos y tres, con B_1 nulo y B_2 se da por:

$$B_1 = 0$$

$$B_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

El otro caso, que es el considerado en el presente proyecto, los dos motores están implementados en la rueda1, por lo que B_2 es nulo y B_1 se da por:

$$B_1 = 1$$

$$B_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Para el cálculo de la energía cinética ésta se da por la ecuación (36)

$$T = \frac{1}{2} \dot{q}^T \begin{bmatrix} R^T * M * R & R^T * V & 0 \\ V^T * R & I_\beta & 0 \\ 0 & 0 & I_\phi \end{bmatrix} \dot{q}; \quad q = \begin{bmatrix} x \\ y \\ \theta \\ \beta \\ \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \quad (36)$$

Donde M es la matriz de masas definida como se muestra a continuación:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

$$M_{11} = M_{22} = Mt + \sum_{i=1}^3 m_i \quad M_{33} = Mt * (e_1^2 + e_2^2) + I_0 + \sum_{i=1}^3 m_i * (d_i^2 + l_i^2 + 2 * d_i * l_i * \cos(\beta_i))$$

$$\begin{aligned}
M_{13} &= M_{31} - Mt * e_2 - \sum_{i=1}^3 m_i * d_i * \text{sen}(\alpha_i + \beta_i) - \sum_{i=1}^3 m_i * l_i * \text{sen}(\alpha_i) \\
M_{23} &= M_{32} - Mt * e_1 - \sum_{i=1}^3 m_i * d_i * \cos(\alpha_i + \beta_i) - \sum_{i=1}^3 m_i * l_i * \cos(\alpha_i) \\
M_{12} &= M_{21} = 0
\end{aligned}$$

El valor de V es el siguiente:

$$V = \begin{bmatrix} -m_1 * d_1 * \text{sen}(\alpha_1 + \beta_1) \\ m_1 * d_1 * \cos(\alpha_1 + \beta_1) \\ m_1 * d_1^2 + m_1 * d_1 * l_1 * \cos(\beta_1) + I_{p1} \end{bmatrix} \quad (37)$$

$$I_\beta = m_1 * d_1^2 + I_{p1}$$

$$I_\phi = \begin{bmatrix} I_{r1} & 0 & 0 \\ 0 & I_{r2} & 0 \\ 0 & 0 & I_{r3} \end{bmatrix}$$

M_t = Masa total del trole.

m_i = masa de cada rueda.

e_1, e_2 = coordenadas del centro de masa del trole.

I_o = momento de inercia del trole alrededor del eje vertical que pasa a través de su centro de masa.

I_{pi} = momento de inercia de cada rueda alrededor del eje vertical que pasa a través de su centro.

I_{ri} = momento de inercia de cada rueda alrededor de su eje horizontal.

Esta expresión general de la energía es útil para considerar un proceso de optimización de caminos a partir un criterio de energía mínima, que no es considerada en este proyecto y se deja para estudios futuros.

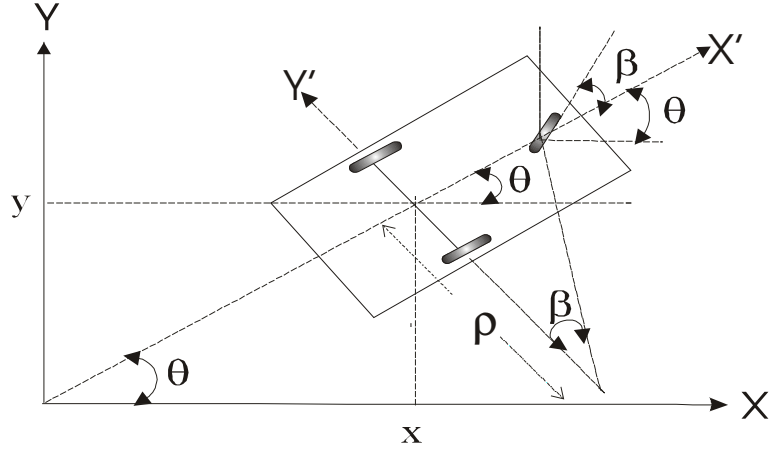
4.2.3 Modelo Simplificado.

Suponiendo que la rueda delantera sea la encargada de la tracción y la dirección la cual tiene una velocidad lineal v y una velocidad de orientación $\dot{\beta}^{25}$, además de una posición y orientación global (x, y, θ) como muestra la figura 27, es posible desarrollar un modelo con un grupo de ecuaciones mas sencillas las cuales son:

²⁵ ZHAO, Yilin. Kinematics, Dynamics and control of wheeled mobile robots. En: Artículo IEEE, (May 1992).

$$\begin{aligned}
\dot{x} &= -v \cos(\beta) \cos(\theta) \\
\dot{y} &= v \cos(\beta) \sin(\theta) \\
\dot{\theta} &= \frac{\sin(\beta)}{L} v
\end{aligned} \tag{38}$$

Figura 27. Ubicación general del chasis del robot móvil.



Fuente: Autor del proyecto.

Estas ecuaciones representan el modelo cinemático directo y que expresadas en forma matricial permiten obtener el modelo cinemático inverso.

La velocidad lineal puede expresarse como:

$$v = \dot{\phi} * R \tag{39}$$

Donde R es el radio de la rueda principal y $\dot{\phi}$ su velocidad angular.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos(\beta) * \cos(\theta) & 0 \\ \cos(\beta) * \sin(\theta) & 0 \\ \frac{\sin(\beta)}{L} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \dot{\beta} \end{bmatrix}$$

Para este caso dado que la matriz no es cuadrada se puede realizar su cálculo mediante la seudo inversa, la cual se puede obtener por la ecuación 40:

$$\{J^T * J\}^{-1} * J^T \tag{40}$$

Con lo cual se llega al modelo cinemático inverso:

$$\begin{bmatrix} v \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\frac{L^2 \cos(\theta) \cos(\beta)}{L^2 \cos^2(\beta) + \sin^2(\beta)} & \frac{L^2 \sin(\theta) \cos(\beta)}{L^2 \cos^2(\beta) + \sin^2(\beta)} & \frac{L \sin(\beta)}{L^2 \cos^2(\beta) + \sin^2(\beta)} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta} \end{bmatrix}$$

Los respectivos modelos simulados con Simulink de Matlab se muestran en la figura 28.

En la escritura de este modelo fue necesario partir de un valor de β a partir de la siguiente ecuación:

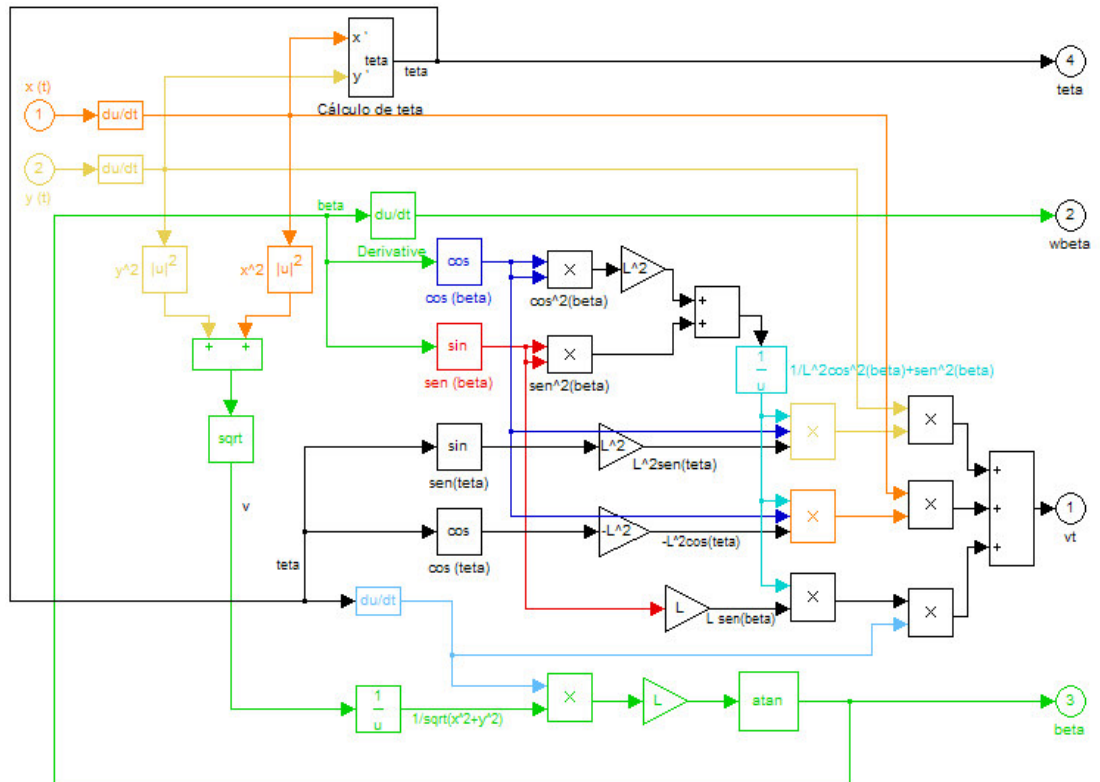
$$\beta = \tan^{-1} \left(\frac{L \dot{\theta}}{v'} \right) \quad (41)$$

$$v' = \sqrt{(\dot{x}^2 + \dot{y}^2)}$$

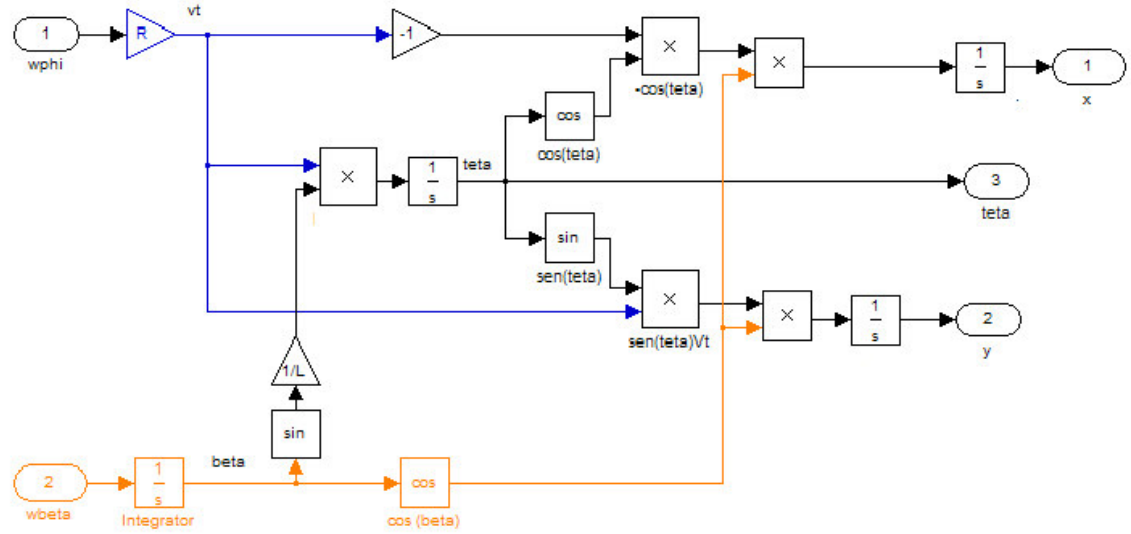
Los archivos que contienen estos modelos se encuentran en el CD anexo en la carpeta modelos y llevan los nombres respectivamente de “tricciodirecto.mdl” y “triccioinverso.mdl”.

Figura 28. Modelo cinemático del robot móvil, sin tener en cuenta los efectos de las dos ruedas traseras.

a) Modelo cinemático inverso.



b) Modelo cinemático directo.

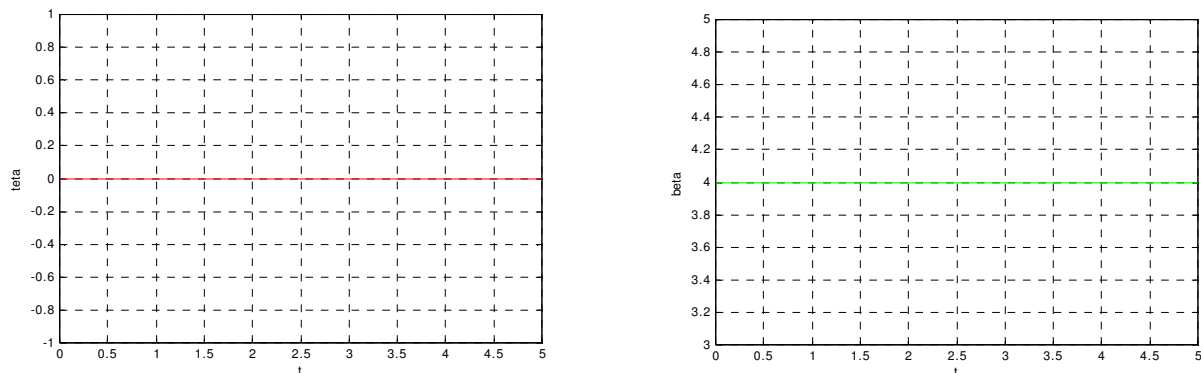


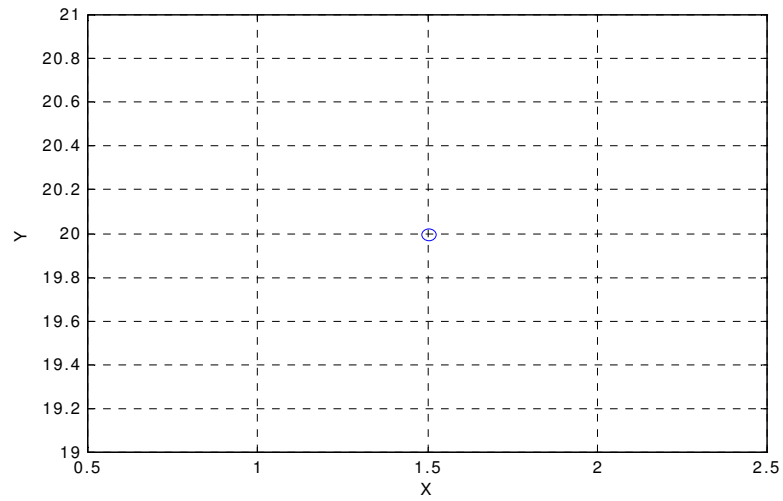
Fuente: Autor del proyecto.

4.2.4 Comprobación del modelo simplificado.

Una forma sencilla de valorar el comportamiento del modelo utilizado en el planeamiento de trayectoria es colocarle como entrada diversos valores de velocidad angular en la rueda principal (permite orientación) y de velocidad angular en la rueda que permite el desplazamiento; lo anterior partiendo de unos valores específicos de condiciones iniciales. Las figura 29 y 30 muestran este efecto. Inicialmente se da un valor cero para la velocidad lineal y se mantiene constante la velocidad angular con lo cual el robot se mantiene estático en su punto de partida con las coordenadas iniciales. Si ahora se intercambian estos valores, el desplazamiento del robot en el plano x-y dependerá de las condiciones iniciales de la orientación. En primera instancia se tomó un valor inicial de cero para β y en el segundo se tomó un valor inicial para β de $\pi/4$.

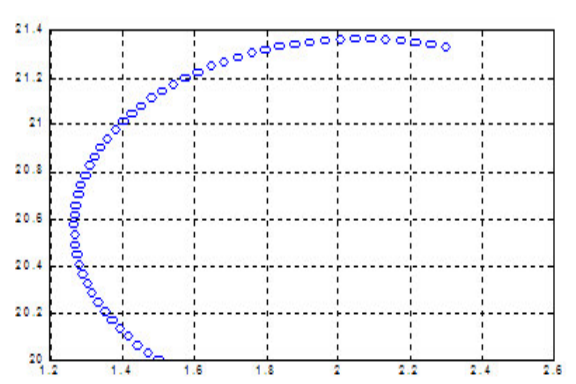
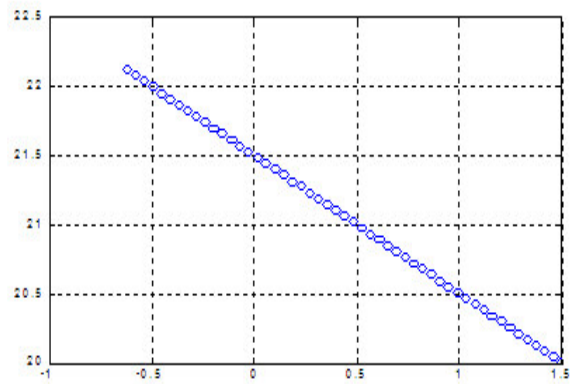
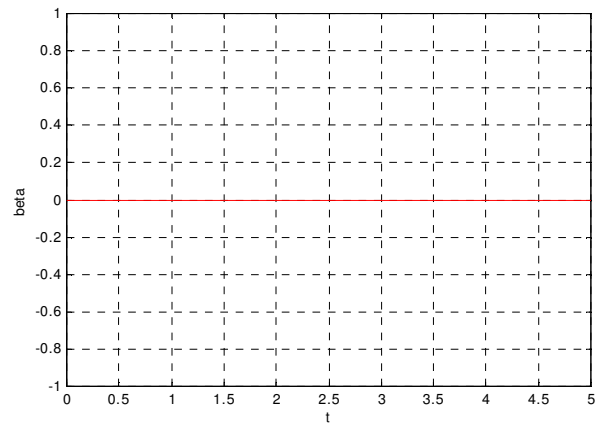
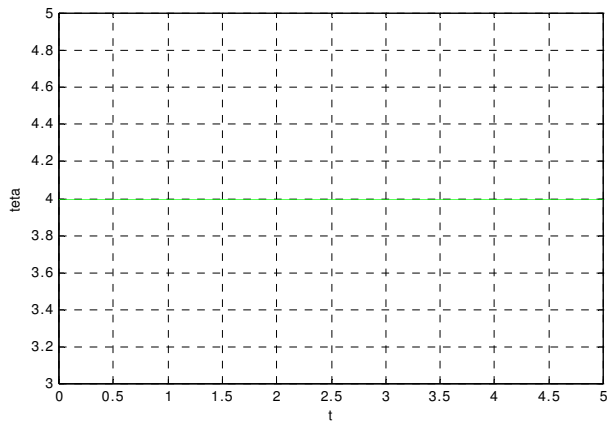
Figura 29. Perfiles de velocidad angular y su respectivo desplazamiento con beta inicial de cero





Fuente: Autor del proyecto.

Figura 30. Perfiles de velocidad y su respectivo desplazamiento con beta inicial de cero y $\pi/4$.

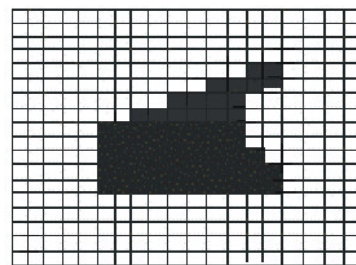


Fuente: Autor del proyecto.

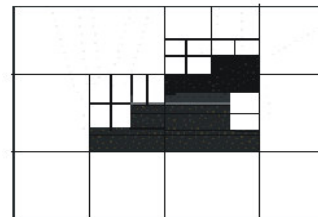
4.3 MODELO DEL ESPACIO DE TRABAJO

La representación de los objetos que ocupan el espacio de trabajo es clave a la hora de realizar un adecuado planeamiento de camino para un robot móvil. Una manera de obtener esta información es mediante una cámara; en planeadores en línea este conocimiento se adquiere a través de sensores a medida que el robot se mueve en el medio. Para el presente trabajo se utilizará la estrategia de representar los objetos(robot y obstáculos) mediante un arreglo de rejillas, dicho espacio de trabajo es subdividido en una cuadrícula en la cual cada celda tiene el mismo tamaño. Una celda se marcará si está ocupada en total o parte por un objeto. Esta aproximación conduce de cierta forma a ver los elementos como si se constituyeran por celdas más pequeñas. La exactitud depende de la resolución, lo cual a su vez requerirá también de una gran cantidad de memoria. Los resultados muestran que esta estrategia trabaja bien en ambientes con objetos irregulares²⁶. Esta no es la única estrategia. También se utilizan los árboles de celdas, la representación poligonal y la geometría de sólidos constructiva. La figura 31 ilustra esas representaciones.

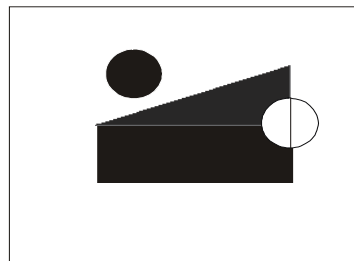
Figura 31. Representación de objetos.



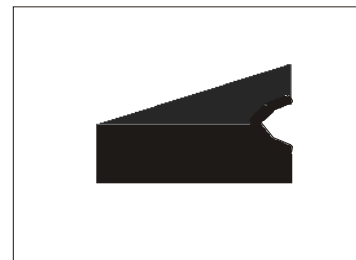
Arreglo de rejilla



Arbol de celdas



Geometría sólida constructiva



Aproximación del polígono

Fuente: Autor del proyecto.

²⁶ Gill Mark and Zomaya Albert. Obstacle avoidance in multi-robot systems. Experiments in Parallel Genetic Algorithms. Singapore: World Scientific, 1998. p.41-45.

Este método es semejante al propuesto por diversos artículos, en cuanto a planeamiento de camino se refiere, y es conocido como ya se citó en descomposición de celdas en el cual un camino es encontrado desde la celda que se toma como punto de partida, recorriendo una celda y luego la adyacente y así sucesivamente hasta encontrar la meta.

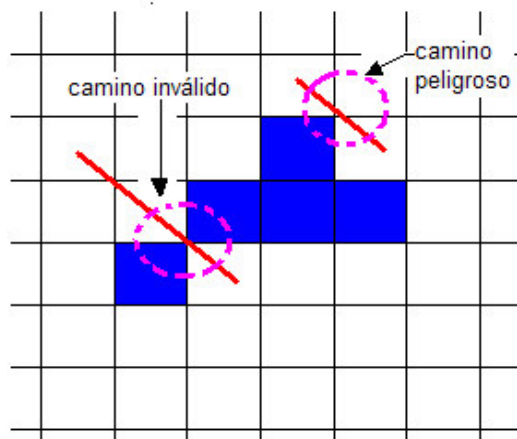
En este trabajo se tendrá en cuenta que el ambiente es estático, que se conocen de antemano los obstáculos y el tamaño del robot se considera a lo sumo de una celda. El representar los objetos de la forma proyectada en el presente trabajo tiene una desventaja y es que se pueden de cierta forma generar caminos inválidos cuando se encuentren obstáculos adyacentes en forma diagonal como se muestra en la figura 32a) además de generar colisiones con los bordes de una celda que representa parte del obstáculo.

En el trabajo de pregrado de Julio Becerra y Freddy Rojas -dirigido por el autor- se planteó una solución como se muestra en la figura 32b) la cual se deja como referencia para futuras mejoras.

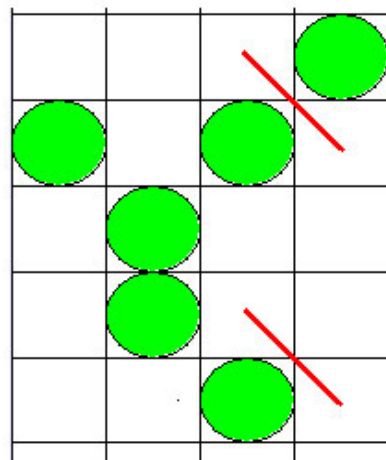
La figura 33 representa la forma en que se desarrolló la herramienta software y como se modeló el espacio de trabajo y en ella se pueden ver algunos obstáculos ejemplo.

Figura 32. Representaciones de obstáculos en el espacio de trabajo y trayectorias inválidas.

a) Representación actual

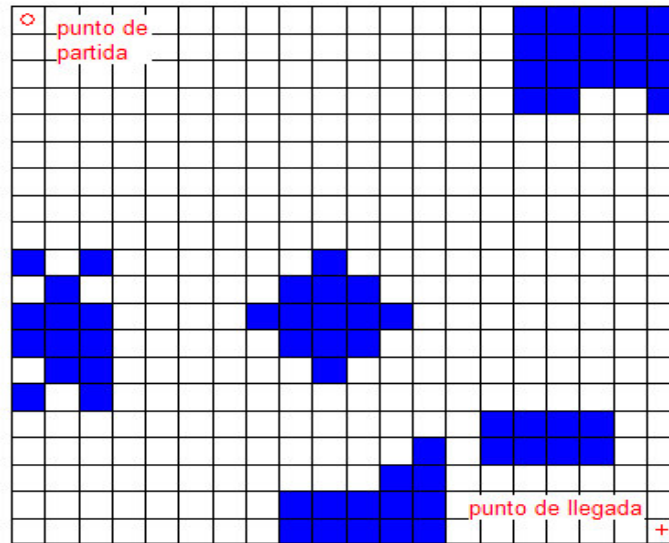


b) Representación mejorada.



Fuente: Autor del proyecto.

Figura 33. Ejemplo de la representación de espacio de trabajo con algunos obstáculos.



Fuente: Autor del proyecto.

5. METODOLOGÍA PARA APLICACIÓN DE LOS AG EN LA SOLUCIÓN AL PROBLEMA DEL PLANEAMIENTO DE MOVIMIENTO

5.1 INTRODUCCIÓN

En anteriores capítulos se plantearon los AG como una herramienta adecuada para la búsqueda de soluciones a problemas de diversa índole, entre ellos se encuentra el problema de planeamiento de movimiento para robots móviles. En el presente capítulo se describe una metodología que busca resolver dicha problemática. Los resultados del capítulo cuatro sirven de soporte para el desarrollo de la metodología en cuanto a planeamiento de trayectoria se refiere. El hecho de ser un trabajo pionero en el tema dentro de la escuela de Ingenieras Eléctrica, Electrónica y Telecomunicaciones, justifica lo extenso y a veces un tanto especulativa que puede parecer la discusión en cuanto a los tópicos básicos, pero es en este apartado donde se materializa toda esa discusión teórica.

A continuación se plantean las dos metodologías: la de camino y la de trayectoria; cada una se profundiza en su propia naturaleza destacando sus fortalezas y posibles debilidades, paseándose por tres aspectos claves para el AG: la codificación escogida, la función de evaluación y el valor de los parámetros (tasa de cruce, tamaño de población y demás). También se muestran aspectos referentes al software desarrollado para el presente trabajo.

5.2 PLANEAMIENTO DE CAMINO

El método que se propone consiste fundamentalmente en establecer un grupo de candidatos a solución, los cuales serán caminos que unirán el punto de partida y la meta.

El primer paso que se realizó consistió en codificar. Por el tipo de problema a resolver se escogió el referente a cadenas binarias. Cada camino se puede clasificar en dos tipos posibles, monótono en x y monótono en y; se ha demostrado según literatura consultada que estos caminos monótonos permiten expresar de una forma poderosa tanto recorridos sencillos como complicados²⁷.

5.2.1 Codificación del problema.

Para establecer la codificación más adecuada es importante recordar las sugerencias que resultan como consecuencia de la hipótesis de los bloques

²⁷ KAZUO, Sugihara and SMITH, John. Genetic Algorithm for adaptive planning of path and trajectory of a mobile robot in 2d terrains. p-1.

constructivos²⁸. Entre estas sugerencias se encuentran: seleccionar una codificación de forma tal que los esquemas cortos de bajo orden que son importantes para el problema que está en estudio estén relativamente poco relacionados a esquemas que están en otras posiciones fijas. Además se debe seleccionar el alfabeto mas pequeño posible, que permita una expresión conveniente al problema en estudio; el poder cumplir esas condiciones y el definir una función de evaluación adecuada es en verdad un arte, que solo se aprende con el trabajo continuo con los AG. Un aspecto que no se menciona con frecuencia en esta discusión, pero que es de enorme trascendencia en la aplicación de los AG al problema del planteamiento de movimiento es lo referente a la longitud de cada cadena candidata a solución; por la naturaleza del problema se puede encontrar sin dificultad que muchas cadenas solución presentan entre sí una característica común: el constituirse de longitud variable. Existen varios artículos que han planteado esta idea y la han trabajado con éxito, pero se sabe de antemano la complejidad computacional que implica realizar estos tipos de codificaciones para aplicar los AG en la solución al problema en cuestión²⁹. En contraparte, el hecho de tomar la estrategia de caminos monótonos garantiza que se puedan tener cadenas de longitud fija que representen en forma clara cada posible recorrido. Un camino se dice monótono en x cuando al trazarse una recta perpendicular a este eje, dicha recta solo corta al camino en un punto; algo semejante se aplica al camino monótono en y .

Para el presente trabajo se utilizó una codificación en la cual cada camino se representó mediante una secuencia de $n-1$ grupos (pares) de bits para dirección y distancia, con la que cada grupo corresponde al avance en una columna en el caso de monotonía en x (avance en una fila para monotonía en y), con n como el número de columnas y número de filas. Lo anterior concuerda con lo que se mencionó en el capítulo cuatro: el ambiente se modelará mediante una rejilla de $n*n$ celdas de igual tamaño. La figura 34 ilustra esta situación. El valor de n es definido por el usuario y en cierta forma el valor que tenga tiene una relación directa mas con la resolución del camino que con otra cosa. En el CD anexo dentro de la carpeta *camínag* aparece la función *rectan()* la cual se encarga de dicha labor además de permitir la inserción o eliminación de obstáculos.

En cuanto a la cantidad de bits necesarios para representar la longitud de un desplazamiento entre celdas este se puede deducir fácilmente a partir de:

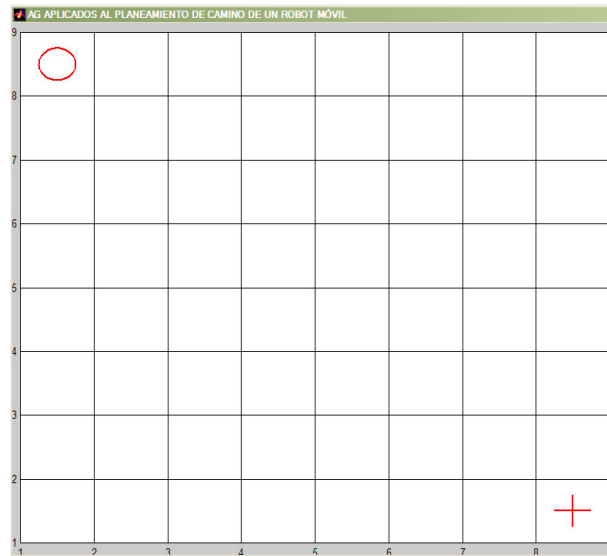
$$\#bits = \frac{\ln(n)}{\ln 2} \quad (42)$$

²⁸ GOLDBERG, Op. cit., p. 80.

²⁹ GERKE Michael. Genetic Path Planning for mobile robots. En :IEEE, American control conference. San Diego: (June 1999); p.2424-2429.

Donde el resultado se redondea al entero mas cercano. Si n es potencia de dos se trabaja con $n+1$. Existen algunos casos especiales en los cuales la trayectoria se sale del espacio de trabajo generando un camino inválido; para estas situaciones se tomó la decisión de sesgar el resultado de forma tal que se considere que el camino está dentro del espacio de trabajo y la cadena sea considerada como válida.

Figura 34. Ilustración de rejilla de 8x8 celdas con su respectivo punto de partida y llegada.



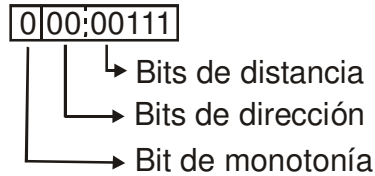
Fuente: Autor del proyecto.

Con lo anterior se logra tener mayor cantidad de individuos valederos y se da la posibilidad de que diversas cadenas puedan llegar a representar el mismo camino, en los demás individuos dicha representación se puede considerar única.

Para desplazamientos a lo largo de columnas o filas el movimiento que se hace siempre involucra el paso hacia la celda adyacente, mediante un avance diagonal siempre y cuando el camino no se salga del espacio de trabajo; si esto llega a ocurrir se eligió entre las muchas posibilidades un desplazamiento en forma de L hasta los límites del área de trabajo.

Para distinguir si un camino es monótono en x o y se utiliza un bit, **bit de monotonía**, (ver figura 35) al comienzo de cada cadena. Este bit es "1" para indicar monotonía en y y "0" para indicar monotonía en x .

Figura 35. Cromosoma básico constituyente de un individuo para $n=8$ (elemento par-distancia).



Fuente: Autor del proyecto.

Las consideraciones generales que se definieron asumiendo monotonía en x son:

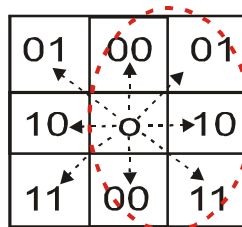
- ◆ El robot siempre se mueve de izquierda a derecha.
- ◆ Los desplazamientos largos son hacia arriba -número negativo- y hacia abajo -número positivo-, pero siempre a lo largo de la columna.
- ◆ El punto de partida del robot siempre estará en la parte superior izquierda y el de llegada en el otro extremo.

Se definen entonces los valores en bits para el primer elemento del grupo, **el de dirección**, que como se muestra en la figura 35 tiene asociado dos bits para cada una de las direcciones posibles, por ejemplo para un desplazamiento en dirección diagonal superior el grupo de bits que se asume sería “01” (ver figura 36).

Para el otro elemento del grupo, **el de distancia**, se tiene que su tamaño es función de n y se utiliza una representación tipo magnitud y signo, en la cual el bit mas significativo indica que si es negativo el desplazamiento es hacia arriba y positivo hacia abajo.

En los casos que el elemento de dirección tome los valores “01”, “10” y “11” este componente (longitud) del par no es de interés, aunque se genera para homogeneizar el tamaño de la cadena.

Figura 36. Interpretación del par de dirección para monotonía en x .



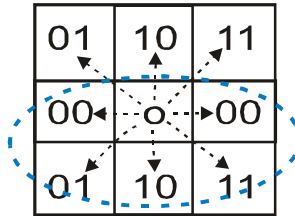
Fuente: Autor del proyecto.

En cuanto a la monotonía en y , la codificación es semejante; la figura 37 ilustra su interpretación. De dicha figura se deduce que el robot siempre se mueve hacia

abajo, y los movimientos largos los hace a través de las filas. Para aclarar este proceso se ilustran en la figura 38 dos ejemplos de trayectorias monótonas en x y y respectivamente. La longitud de cada cadena se da entonces de acuerdo a la ecuación 43, con la cual queda definido el genotipo de cada individuo.

$$longcadena = 1 + (n - 1) * \left(3 + \text{round} \left(\frac{\ln(n)}{\ln 2} \right) \right) \quad (43)$$

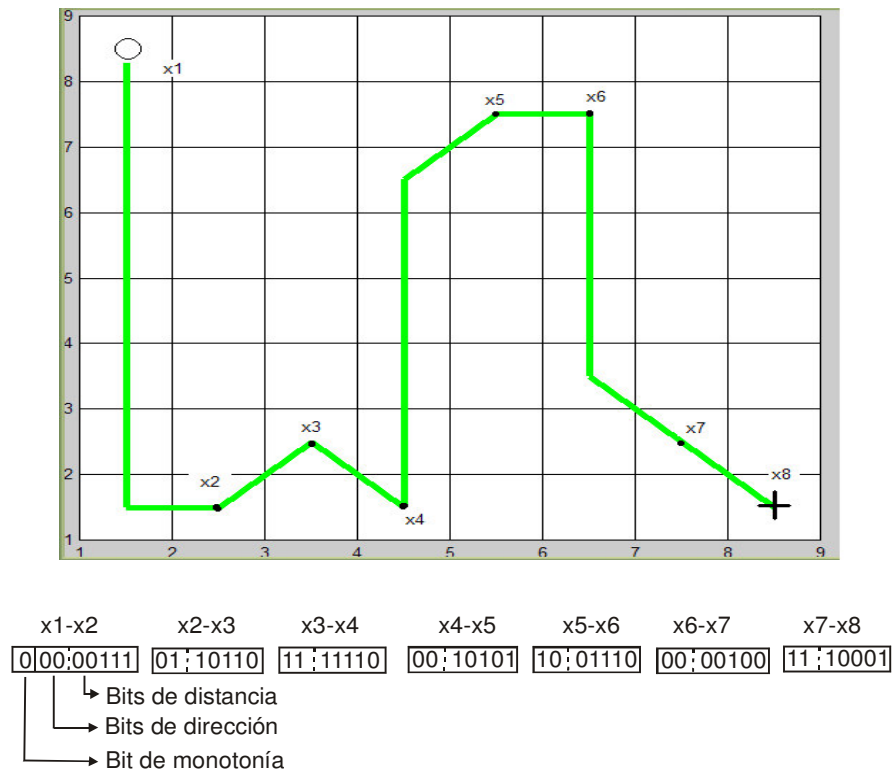
Figura 37. Interpretación del par dirección para monotonía en y.



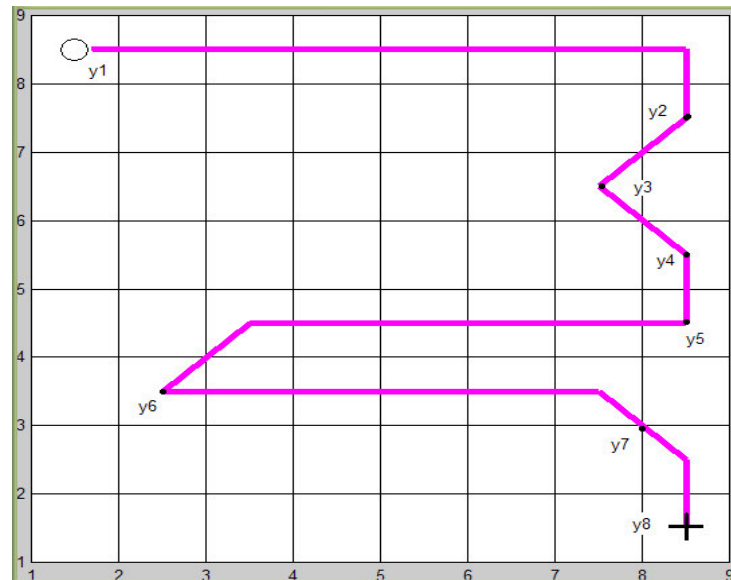
Fuente: Autor del proyecto.

Figura 38. Ilustración de caminos con su respectiva codificación.

a) Monotonía en X



b) Monotonía en y.



y1-y2	y2-y3	y3-y4	y4-y5	y5-y6	y6-y7	y7-y8
1 00:00111	01:10110	11:11110	10:11111	00:10101	00:00101	10:10001
└─ Bits de distancia └─ Bits de dirección └─ Bit de monotonía						

Fuente: Autor del proyecto.

El código que realiza la función de decodificación de una cadena lleva por nombre *estudiocamino()*, y es uno de los de mayor complejidad y consumo en tiempo de cálculo.

5.2.2 La función de evaluación.

En un AG simple, la función de evaluación(adaptación), también conocida en algunos contextos como función objetivo debe cumplir unas características mínimas entre las cuales se encuentra, ser una función siempre positiva³⁰ y además entenderse siempre involucrada en un proceso de maximización. Si estas dos condiciones anteriores no se cumplen se debe entonces establecer una normalización y una especie de mapeo desde la función original, la función objetivo, hacia la función de adaptación. Esta función permite obtener el fenotipo de un individuo candidato a solución, lo cual hace referencia en cierta medida al desempeño que este presenta en un ambiente en particular que para el presente caso representa el no colisionar y tener una distancia mínima.

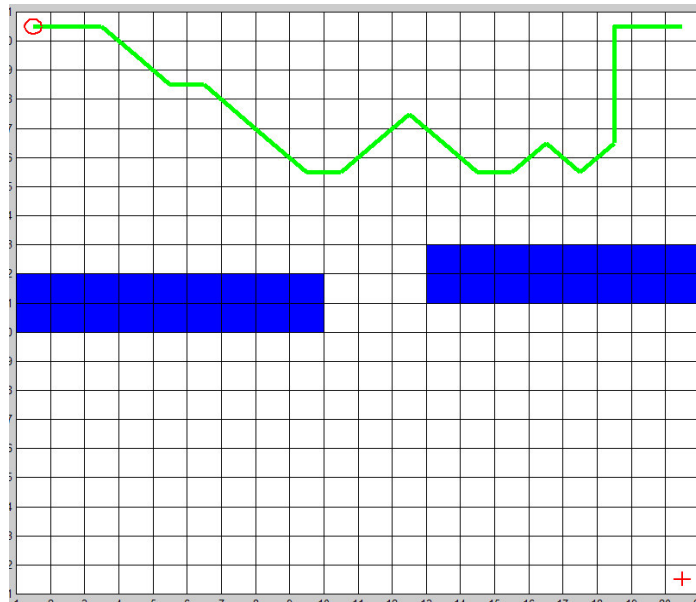
³⁰ GOLDBERG, Op. cit., p-75.

El proceso seguido para establecer la función de adaptación, que en cierta forma fue de tipo exploratorio, se dividió en dos partes. Como primera medida se partió de una fórmula propuesta en los artículos estudiados, la cual tiene la siguiente forma:

$$f(x) = \begin{cases} 1 & \text{Sí intercepta al menos un obstáculo} \\ 4 * n^2 - longitudcamino & \text{No intercepta ningún obstáculo} \end{cases} \quad (44)$$

Esta función se puso a prueba y aunque en su naturaleza parece contener los elementos básicos para catalogar a un camino de “bueno” o “malo” según la variable a optimizar distancia(tiempo), presentó cierta problemática ya que para el caso tomado como tipo -el de la figura 39-, el análisis después de correrlo cierto número de veces dio como resultado que para diversos caminos que están muy lejos del óptimo (conocido en este caso), la media de los individuos generados generación tras generación tendían hacia un comportamiento anómalo, en el cual la distancia es mas corta pero no se consigue nunca llegar a la meta. La figura 39 ilustra esta situación.

Figura 39. Comportamiento anómalo donde la distancia es menor al óptimo.



Fuente: Autor del proyecto.

Por lo anterior se dio la necesidad de incluir un parámetro adicional el cual castigue este comportamiento. La función que se propuso se muestra entonces a continuación:

$$f(x) = \begin{cases} 1 & \text{Sí intercepta al menos un} \\ & \text{obstáculo.} \\ 4 * n^2 - longitudcamino - dx * n^2 & \text{No intercepta ningún} \\ & \text{obstáculo.} \end{cases} \quad (45)$$

Donde dx es la distancia desde el punto de meta hasta donde llega el camino candidato a solución. El comportamiento posterior a esta modificación fue bastante aceptable como se verá en el siguiente capítulo.

5.2.3 Parámetros del AG.

Para el AG los tres operadores que se utilizaron en el presente trabajo fueron: la selección, el cruce y la mutación. La utilidad de otros operadores como la inversión, la especiación y el elitismo entre otros no se estudió, pero se espera que las ideas aquí planteadas sean el primer paso hacia trabajos de investigación en las áreas propuestas. Los parámetros de entrada que requiere un AG posterior a la escogencia de los operadores son los siguientes: tamaño de población, cantidad de individuos por generación, tasa de cruce y tasa de mutación. La tendencia general, presentada en la mayoría de artículos, recomienda que la tasa de cruce, es decir, la posibilidad que un par de individuos puedan ser cruzados, está generalmente del orden del 60% en adelante. El operador de cruce utilizado fue el de un solo punto, aunque este punto se varió de dos formas: la primera referente a realizar el cruce en cualquier parte y la otra a realizarlo únicamente en puntos donde se cambia de cromosoma. Para la tasa de mutación el valor que se recomienda debe ser inferior al 10%. Tal vez la justificación de esto sea la misma adaptación del modelo natural, donde la mutación ocurre con muy poca frecuencia y el cruce todo lo contrario. El operador de mutación se trabajó a lo largo de toda la cadena.

Para el proceso de selección se utilizó el método de la ruleta, también conocida como selección proporcional a la adaptación. El hecho de utilizar estos operadores se puede justificar desde la óptica de que en que la mayoría de artículos consultados se tienda a presentarlos bajo el concepto del AG simple; los resultados con los mismos no fue del todo satisfactoria, lo que llevó a la posibilidad de explorar el uso de otras propuestas de operadores que pudieran mejorar el desempeño; entre estas se observó que la inclusión de un individuo “bueno” en forma artificial, provocaba que en la siguiente generación donde este participaba se destruyera y se convirtiera en un individuo de peores características que las iniciales, por lo que se introdujo una forma de elitismo bastante primaria que dista un tanto de lo planteado en la literatura en cuanto a este operador se refiere. La forma consistió en revisar generación por generación cual era el mejor individuo obtenido y mantenerlo como referencia para compararlo en las siguientes generaciones.

En cuanto al número de individuos se propone utilizar un valor proporcional a n , el cual sea siempre superior a $2n$. Dentro de la información consultada poco se habla de este aspecto y los resultados propuestos por algunos estudiosos del tema solo se aplican a ciertas funciones de evaluación en forma muy cerrada, las cuales comparadas con las propuestas en el presente trabajo no tienen mucha aplicabilidad. El criterio de parada para la presente aplicación se realizó con base en el número de generaciones, aunque es difícil establecer un número específico para este valor se recomienda que sea también múltiplo de n , de todas formas no se recomienda darle un valor grande (se recomienda inferior a 60, aunque parezca arbitrario con este se obtuvieron buenos resultados, claro para poblaciones iguales o superiores a cien individuos) pues el tiempo de cálculo se hace muy grande y en muchos casos el equipo donde se este corriendo la aplicación se puede tornar inestable. Es mejor realizar el proceso de a 60 generaciones hasta que se obtenga el resultado esperado. De todas formas se observó una alta homogeneización de la población después de correrse cierto número de generaciones, lo que puede conducir también a tener en cuenta como criterio de parada la media que se vaya presentando a lo largo de la simulación, aspecto que no se consideró. En cuanto a la función encargada de albergar el AG esta se denomina *geneticocamino()*; las funciones que se encargan de evaluar generación a generación cual es el mejor individuo y mantenerlo también como reseña para considerar el desempeño del AG son: *mejor()* y *obtienemejor()*; todas estas funciones se encuentran en el CD anexo dentro de la carpeta *caminag*.

5.3 PLANEAMIENTO DE TRAYECTORIA

Para esta tarea se planteó la siguiente metodología. Partiendo del resultado obtenido en el caso anterior, el del planeamiento de camino se procedió entonces a proponer una metodología que sea capaz de realizar una búsqueda en el espacio de las velocidades teniendo como referencia el modelo cinemático que se describió en el capítulo 4. Este modelo tiene como entrada dos variables: la velocidad angular de la rueda principal y la velocidad lineal del robot, las cuales pueden tener algunas restricciones según se exceda o no los valores reales que pueda llegar a tener un actuador. Para el presente trabajo se consideró que no se excediera de una velocidad de 1 [m/s], el cual no deja de ser un valor un tanto arbitrario. Para la generación de las posibles soluciones se proyectó un procedimiento muy similar al anterior realizada con el planeamiento de camino.

5.3.1 Codificación del problema.

Cada posible solución será representada por un alfabeto de 3 elementos: 1,0 y -1. La intención de realizar esto radica en que a partir de aceleraciones constantes, se puede generar un perfil de velocidad que sirva como entrada al modelo del sistema³¹ y que en definitiva pueda realizar el seguimiento del camino planeado. Cada individuo entonces se compondrá de dos genes uno para la aceleración

³¹ GALLARDO, Domingo. Generación de trayectorias robustas mediante computación evolutiva. En: IEE., p.1.

angular y otro para la aceleración lineal, codificando de esta forma el movimiento de una celda a otra dentro del espacio de trabajo. Por lo tanto la longitud total de cada individuo será de $n-1$ cromosomas donde n es el número de celdas en los cuales se dividió el espacio de trabajo para el planeamiento de camino que se tiene como referencia. La figura 40 ilustra esta representación.

Figura 40. Representación de la codificación del problema.

$$\langle 01 \rangle, \langle -11 \rangle, \langle -10 \rangle, \langle 00 \rangle, \dots$$

Fuente: Autor del proyecto.

Un ejemplo un poco más concreto de esta codificación se muestra a continuación partiendo de $n=20$. En este caso existen 19 pares que representan la aceleración lineal y angular respectivamente; también se muestran los perfiles de velocidad resultantes para un intervalo de tiempo de 250[S].

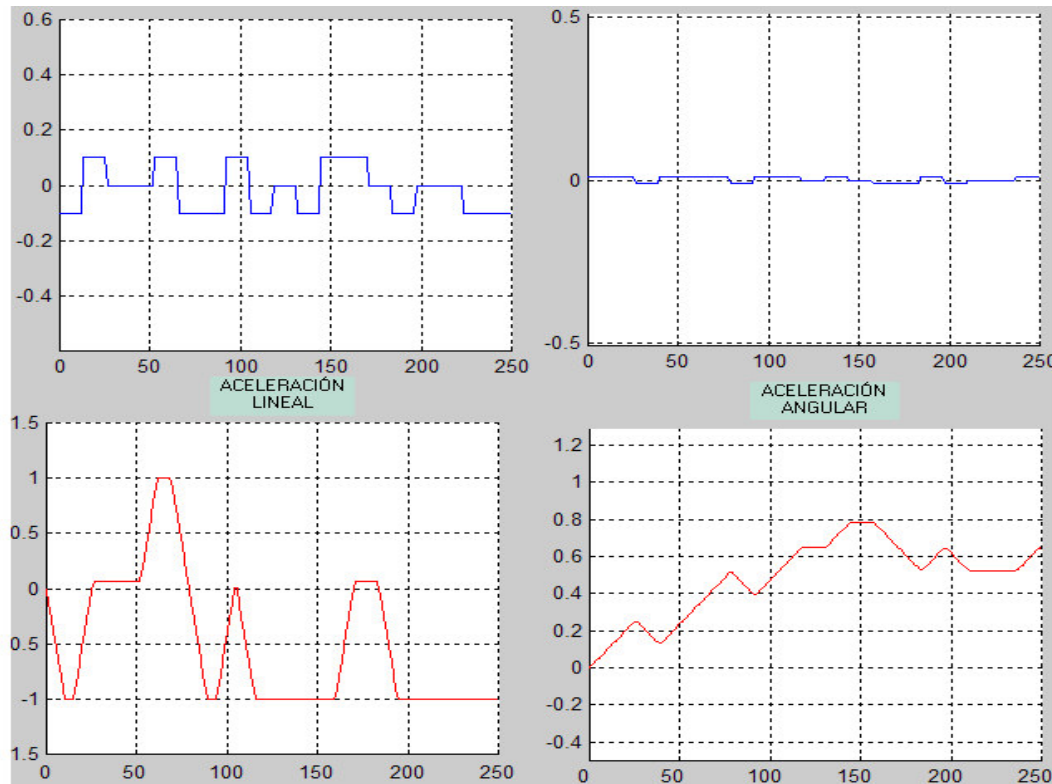
$$a = \langle -11110-10111-11-1-111-1100-11101-10-1-110-100-10-11 \rangle$$

La interpretación que se le da a esta cadena es dependiente de la variable tiempo, la cual se estableció mediante el esbozo de un delta de tiempo el cual se maneja de forma indirecta a partir de la misma simulación realizada con la herramienta *Simulink*. Es importante aclarar que el tiempo para moverse de una celda a otra se da entonces por el cociente del tiempo de simulación sobre $n-1$.

La interpretación que se le da a cada par de individuos es la siguiente: “0” para aceleración nula, “1” para aceleración constante diferente de cero y “-1” desaceleración constante diferente de cero. Los valores de aceleración se consideraron en el intervalo $[-a_{\max}, a_{\max}]$ y $[-w_{\max}, w_{\max}]$, donde estos valores se tomaron de forma intuitiva para las simulaciones realizadas, pero que están disponibles para que el usuario los modifique. En cuanto a la velocidad se refiere también se impuso un límite, con el fin de hacer el sistema lo mas parecido a un sistema real. El código donde se alojan estas tareas tiene la característica de no ser del tipo función, por el contrario están mas como *script*, ya que para la respuesta del modelo en *Simulink* ante determinada entrada, el grupo de parámetros debe estar en lo que se denomina el *workspace* de *Matlab* y esto no se logra mediante una función. El *script* donde se encuentra el primer conjunto básico de tareas que permite generar los individuos y mostrar la función de evaluación se denomina *generalvel1()*. Este *script* llama a otros como es *comportamodelo*, el cual realiza la tarea de decodificar cada individuo y evaluar su comportamiento dando entrada de estos datos al modelo de *Simulink* que se denominó en el capítulo 4 como *triciclodirecto*. Otro script de interés hace referencia al que se denomina *geneticotrayecto*, en el cual está implementado el AG para el planeamiento de trayectoria. Todos estos archivos se encuentran el CD anexo dentro de la carpeta *dinamizag*.

Un ejemplo adicional de un individuo en particular se muestra en la figura 41, donde se esboza lo referente a los perfiles de aceleración angular y aceleración lineal, seguido de las velocidades producidas, a lo largo de un determinado intervalo de tiempo. En este caso se tomó un valor de $t=250$ [S]. También se muestra el resultado arrojado a través de la aplicación de estas dos variables al modelo del robot y el respectivo desplazamiento en el espacio de trabajo.

Figura 41. Representación de los resultados de la codificación.



Fuente: Autor del proyecto.

5.3.2 La función de evaluación.

Teniendo en cuenta los mismos delineamientos trazados en el caso de planeamiento de camino, se tiene entonces que la función de evaluación debe significar un problema de maximización en el cual entre a jugar un parámetro que indique que tan parecido sea el camino que se obtiene con el deseado, al darle entrada al modelo de *simulink*. Para saber que tan semejante es un camino con respecto a lo que se desea se planteó inicialmente el hecho de calcular el error al desplazarse entre celda y celda y magnificarlo elevándolo al cuadrado (error medio cuadrático), para luego irlo sumando y restarlo a una constante algo grande la cual es función de n . Con esta metodología existe una dificultad que consiste en cuales puntos tomar para la referencia y como asociarlos a los que componen el camino que se quiere evaluar, si para cada uno el distanciamiento entre dato y

dato es diferente. Lo anterior se vio reflejado en los resultados que se fueron produciendo con esta función de evaluación.

El resultado que se obtuvo en definitiva fue bastante negativo y se optó por otra metodología. El proceso propuesto consiste en utilizar la correlación de los puntos pertenecientes al camino de referencia con respecto a los que se obtienen a partir de la codificación propuesta.

El valor de este coeficiente de correlación estará en el rango de $[-1,1]$, queriendo decir obviamente que 1 es para un camino idéntico y -1 para uno que esté bien alejado de lo que se desea. Esta consideración se estandarizó al intervalo $[0,2]$, con lo cual el que tenga una correlación de 1 su error será nulo, mientras que el que tenga un valor de -1 su error se cuantificará de una forma bastante elevada. Los resultados para esta propuesta mejoraron en gran forma con respecto a la que se propuso inicialmente.

$$f(x) = 5n^3 - error * n^3 \quad (46)$$

Donde 5 es una constante algo arbitraria. Los resultados mediante la anterior función de evaluación tuvieron una mejora adicional agregándole una consideración, la cual tiene en definitiva la cualidad de reducir el espacio de búsqueda, mediante el “castigo” a aquellas trayectorias que dan como resultado caminos que cruzan la recta $x=1$ y avanzan hacia $-\infty$ o la recta $y=n+1$ y avanzan hacia ∞ ; a estas trayectorias se les asignó una función de evaluación de valor 1.

Sumado a lo anterior se castigó también a aquellos individuos que en algo se parecen al camino deseado, pero a medida que pasa el tiempo y cruzan el espacio de trabajo se alejan del punto de llegada, a estos se les castigó reduciéndoles su valor de adaptación como se muestra a continuación.

$$f(x) = \begin{cases} 1 & \text{Sí cruza la recta } x=1 \text{ o la recta } y=n+1. \\ 5n^3 - error * n^3 - dx * n^2 & \text{En otro caso} \end{cases} \quad (47)$$

Donde dx es la distancia del punto por el cual se cruza el espacio de trabajo y el punto de llegada.

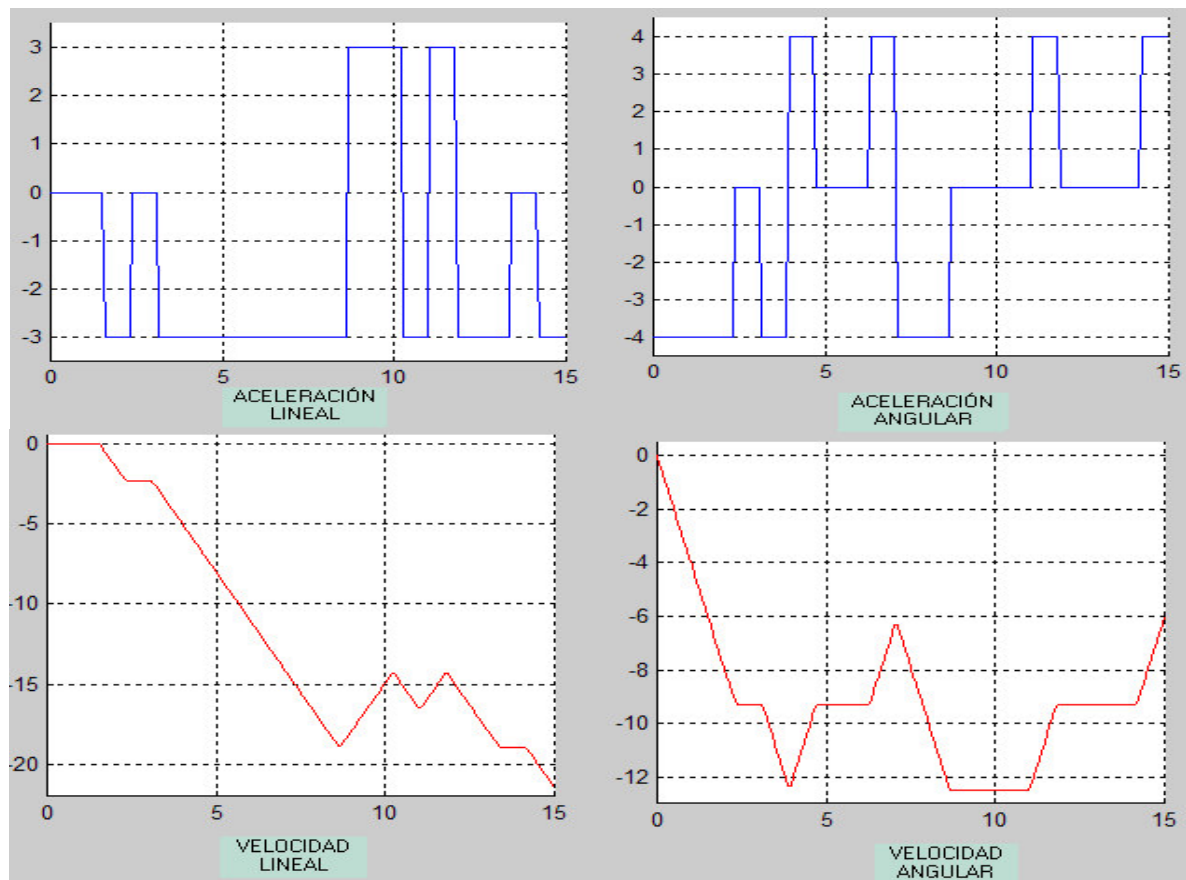
5.3.3 Parámetros del AG.

La metodología en la construcción del AG para este proceso se dio de igual forma que la diseñada para el planeamiento de camino, con una tasa de cruce superior a la tasa de mutación. El criterio de parada que se eligió es el número de generaciones motivado por los resultados obtenidos, en los cuales la tendencia del algoritmo no es del todo convergente hacia el mejor individuo, obviamente existe un acercamiento en la vecindad del camino de referencia, pero no se encuentra la

ruta esperada. Igualmente es válido el hecho de que para poblaciones grandes se debe buscar un punto medio entre tiempo de cálculo y efectividad, pues para poblaciones medianas el tiempo de convergencia aumenta considerablemente, y si se tienen grandes poblaciones por generación pues la convergencia se demora en gran medida y el tiempo de ejecución del algoritmo se hace bastante elevado.

En cuanto a la mutación es importante aclarar cómo se realizó ya que por la codificación planteada no solo basta con complementar un bit, por lo que una vez establecido si se efectúa o no este operador, se procede a generar un número aleatorio el cual dependiendo del valor que tenga actualmente el bit contenido en la cadena (individuo) que se está mutando se procede a cambiar por las otras dos posibilidades; esta elección se hace observando la magnitud del número aleatorio, si es mayor o menor a 0.5. La figura 42 representa en forma completa el resultado de un individuo con los perfiles de aceleración y velocidad, además de la representación en el espacio x-y de tamaño $n \times n$ de la trayectoria que generan.

Figura 42. Perfiles para una determinada codificación en particular.





Fuente: Autor del proyecto.

5.4 HERRAMIENTA SOFTWARE PARA LA APLICACIÓN DE LOS AG EN EL PLANEAMIENTO DE CAMINO

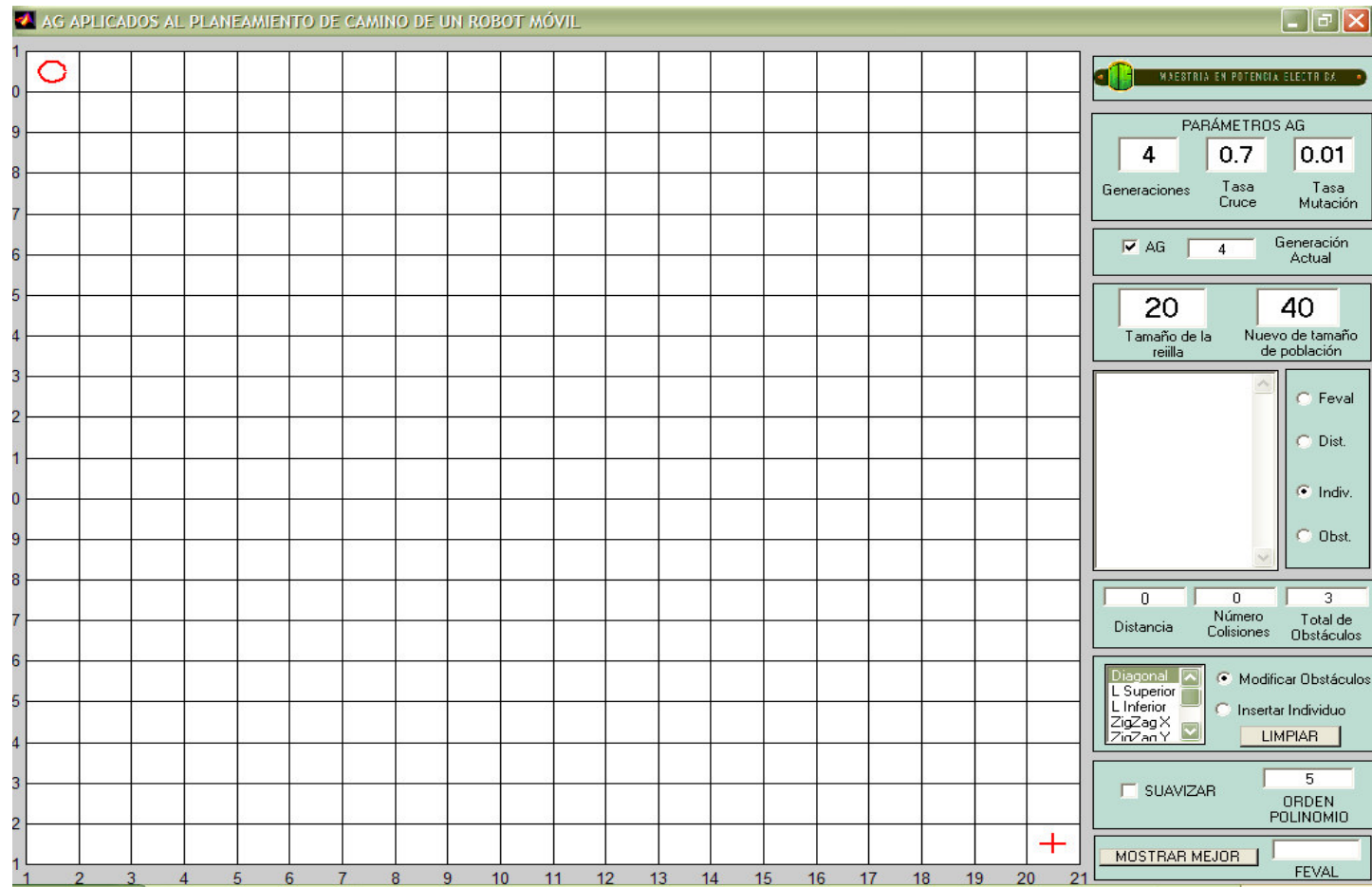
Para el desarrollo de la presente aplicación se utilizó el software Matlab 5.3 y con este la herramienta “GUI”, útil para la implementación de interfaces gráficas con el usuario. La herramienta que se desarrolló se muestra en la figura 43.

El manejo de esta herramienta es de por sí bastante intuitivo, pues los objetos que allí se incorporaron permiten interactuar en forma clara y concisa de acuerdo a un plan previo; existen acciones que no se pueden realizar sin haber ejecutado otras con anterioridad; esto se resalta mediante el despliegue de unas ventanas las cuales muestran un mensaje de error o advertencia indicando la falla que se puede presentar.

Las variables que tiene el usuario para observar el comportamiento son: distancia total recorrida, función de evaluación (función objetivo), número de obstáculos interceptados y el individuo con su codificación respectiva.

El usuario puede variar a su acomodo parámetros propios de un AG simple como son: número de generaciones, tamaño de la población, tasa de cruce y tasa de mutación. La herramienta además da la posibilidad de mostrar el mejor individuo que se obtenga a lo largo de la ejecución del AG durante el número de generaciones que se quiera. Sumado a lo anterior se presenta una opción para que el usuario pueda agregar individuos a su antojo decidiendo entre trayectorias monótonas en x o monótonas en y ; y otro grupo de trayectorias como son la diagonal, una L monótona en x o en y ; y un zigzag monótona en x o en y , respectivamente.

Figura 43. Herramienta para planeamiento de camino desarrollada con "GUI" de Matlab. 5.3.



Fuente: Autor del proyecto

Otra característica adicional que se agregó fue la referente a un suavizado de la trayectoria que se obtiene mediante la aplicación de un algoritmo propio de Matlab que es útil para esto.

La información que esta aplicación entrega está referida al camino y los puntos mínimos con los cuales se puede representar este trazado.

5.5 HERRAMIENTA SOFTWARE PARA LA APLICACIÓN DE LOS AG EN EL PLANEAMIENTO DE TRAYECTORIA

De igual forma como se citó en el ítem anterior el software utilizado para el desarrollo de esta aplicación fue Matlab 5.3 y la herramienta que este posee para el desarrollo de interfaces gráficas con el usuario (GUI). La figura 44 muestra la presentación de esta herramienta.

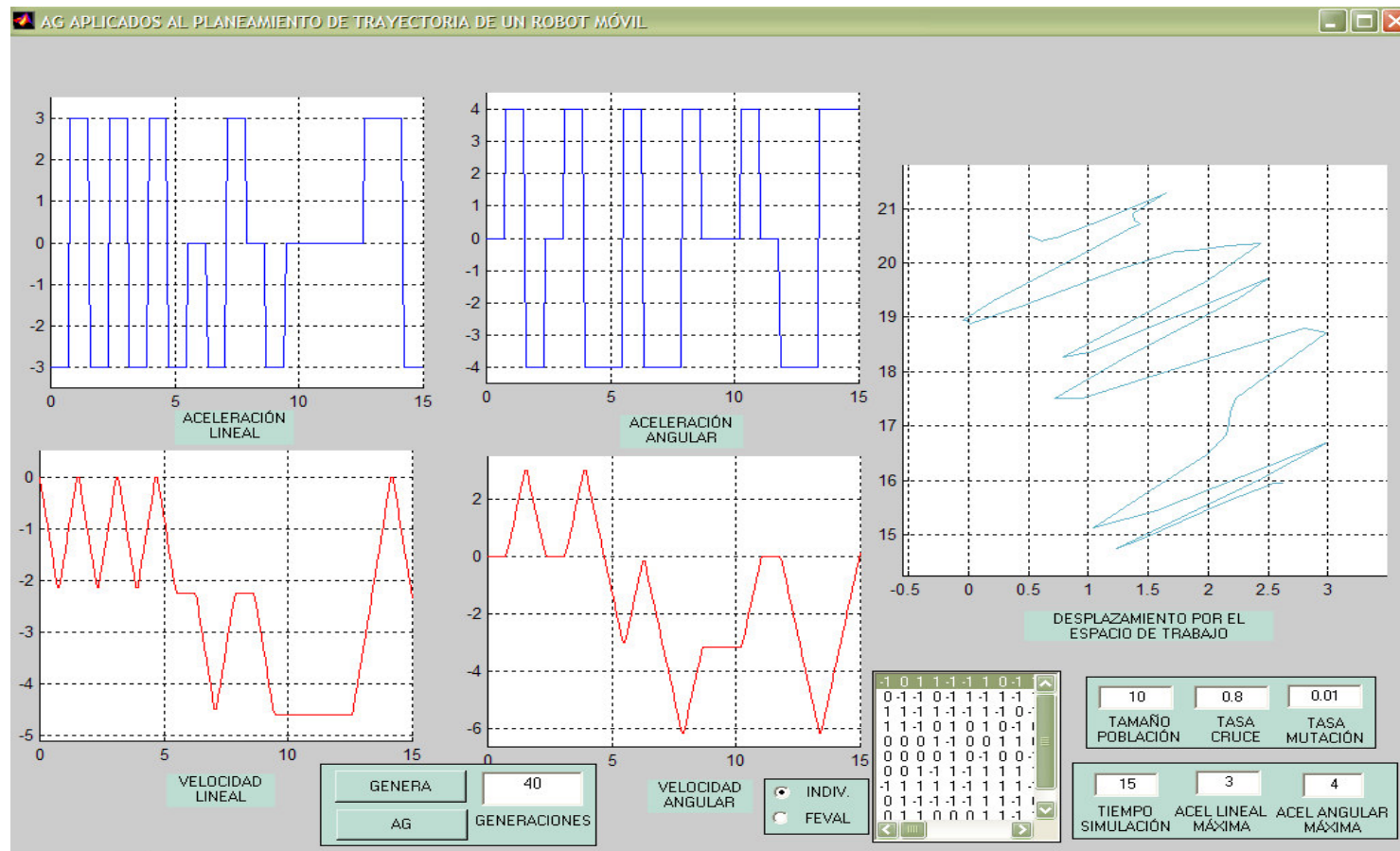
Como se puede observar tiene gran similitud con la aplicación anterior en cuanto a los parámetros con los que el usuario puede interactuar, con mayor relevancia lo referente a los parámetros del AG; lo que se requirió adicionalmente fue poder variar el valor de la aceleración máxima lineal, aceleración máxima angular y tiempo de simulación.

La información que esta aplicación entrega está referida a los perfiles de aceleración (angular y lineal) y velocidad (lineal y angular), los cuales entran al modelo en Simulink el cual a su vez entrega el desplazamiento en un espacio bidimensional (x,y) de tamaño $n*n$.

Para evitar posibles errores cuando se quiera correr la aplicación en cualquier PC es recomendable revisar el archivo denominado ayuda, con el cual se pueden hacer unos ajustes que son necesarios para el perfecto funcionamiento de todas las funciones y *scripts* aquí referenciados.

El trabajo consiste generalmente en cambiar un camino donde se guardan unos datos y de donde se cargan otros; estos caminos cambian de PC a PC, por lo que es importante considerar este aspecto.

Figura 44. Herramienta del Planeamiento de trayectoria desarrollada con "GUI" de Matlab. 5.3



Fuente: Autor del proyecto.

6. ANÁLISIS DE LOS RESULTADOS OBTENIDOS

6.1 INTRODUCCIÓN

Para evaluar el desempeño de los AG aplicados al problema en cuestión se procedió a plantear unos casos tipo que contienen formas de obstáculos en los cuales el óptimo se puede deducir en forma intuitiva; con lo anterior se quiso comprobar la eficacia de la metodología propuesta. Los parámetros del AG a saber: tamaño de población, número de generaciones, tasa de cruce, tasa de mutación y criterio de parada, se establecieron de una forma a priori según algunas recomendaciones encontradas en la bibliografía consultada, las cuales fueron objeto de estudio y comprobación como se muestra mas adelante. Como la codificación propuesta para la solución de los dos problemas es de cierta forma particular, poca información se tiene a la mano como referencia lo cual condujo a plantear una metodología de tipo exploratorio, para observar el comportamiento del AG cambiando en forma dinámica los distintos parámetros de este. A continuación se presentan los resultados y el análisis de las dos herramientas computacionales, con un grupo de ejemplos y algunos datos tomados para poder a partir de ellos elaborar conclusiones y proponer mejoras.

6.2 ESTUDIO DE LOS RESULTADOS OBTENIDOS EN CUANTO A PLANEAMIENTO DE CAMINO

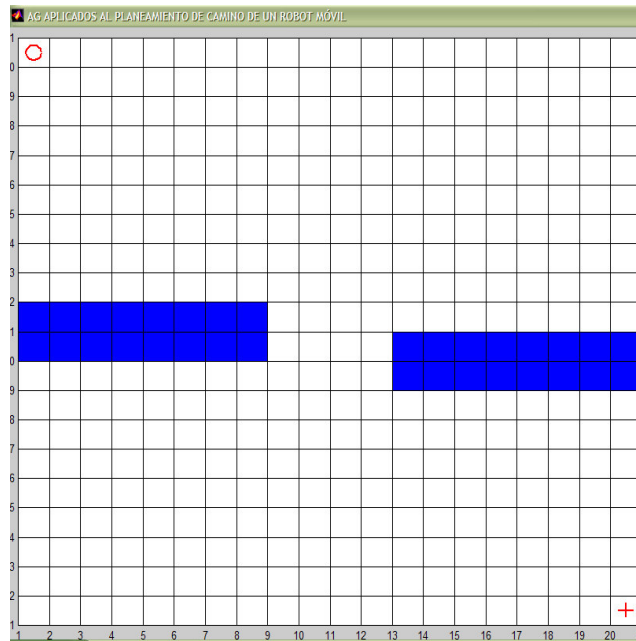
Uno de los casos tipos con los que se trabajó es el que se muestra en la figura 45. Es de inmediato evidente que el óptimo para este trazado de obstáculos es la diagonal, entre el punto de partida y el punto de llegada.

Tomando como referencia, un número de generaciones 20, 40, 60, 80 y 100, se comprobó el algoritmo generación por generación, observando cómo se obtenía una solución cercana a la deseable; este proceso se realizó al menos cincuenta veces y algunos de estos resultados se agregan en un archivo de Excel dentro del CD anexo.

De igual forma para comprobar el efecto de otros parámetros en el AG se fue cambiando el valor de la tasa de cruce y de la tasa de mutación, de forma tal que se pudieran estudiar todas las posibilidades y su respectivo efecto.

El comportamiento o tendencia del mismo AG se puede medir por la media poblacional, la cual predice de cierta forma la homogeneidad de la población a medida que transcurre cada generación. En la figura 46 se muestran los resultados para mejor individuo y en la figura 47 están los resultados para la media poblacional. Estas gráficas solo muestran los valores para tamaños de población de 10 y 20 individuos, lo anterior con el fin de evitar congestionar el texto con tanta información.

Figura 45. Caso tipo de obstáculos.



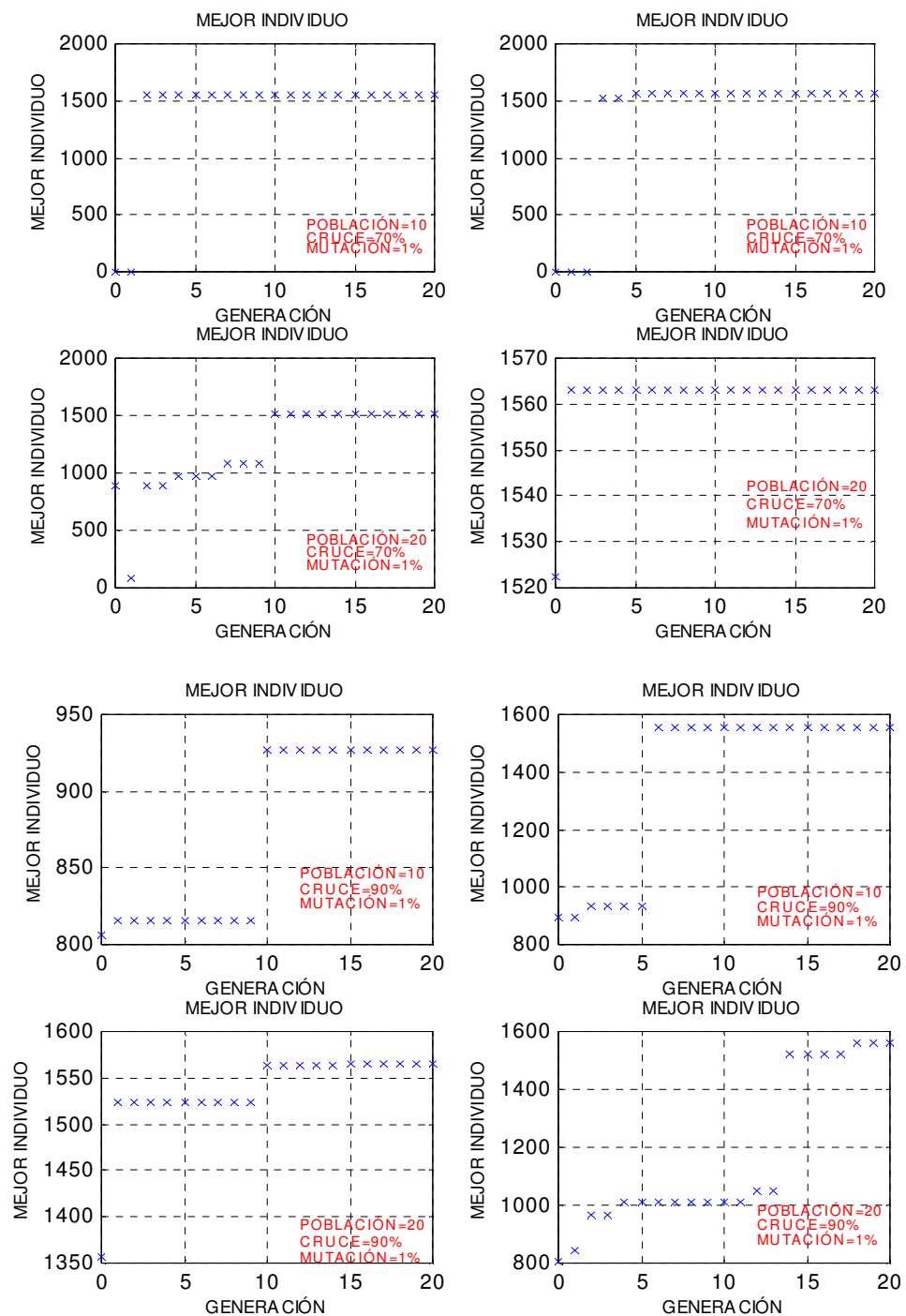
Fuente: Autor del proyecto.

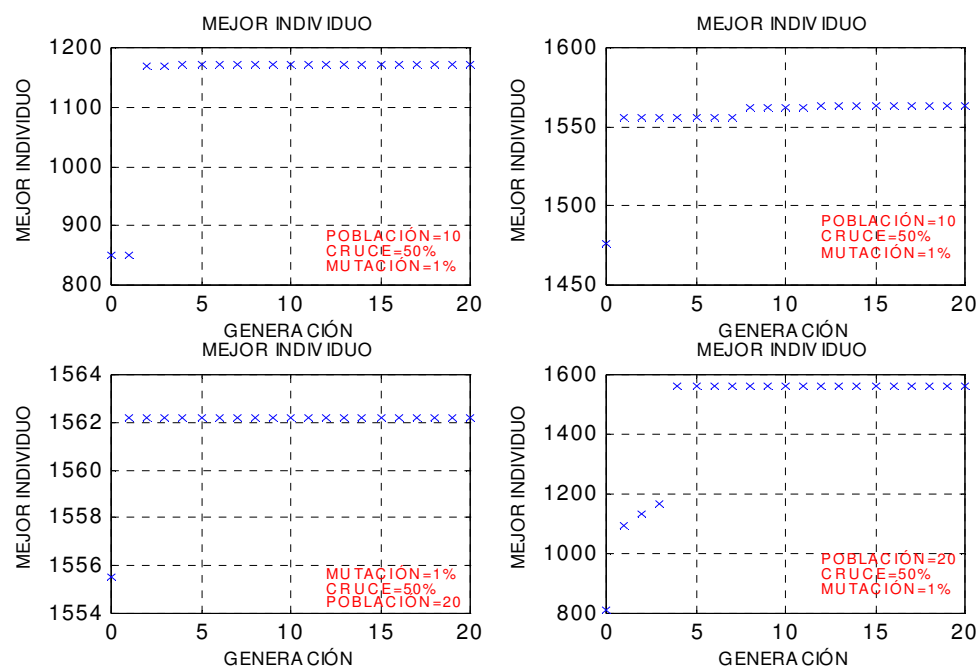
Para estas gráficas el eje de las ordenadas mantiene el valor de la función de evaluación, la cual para el óptimo es de 1573.13, con lo cual se puede tener una referencia para su comparación y análisis.

De los resultados obtenidos se puede concluir que una tasa de cruce muy alta (90%) no mejora el proceso, por el contrario tiende a hacer mas lenta la convergencia comparado con los resultados obtenidos para un valor de tasa de cruce medio (70%). Para una tasa de cruce baja (50%) el resultado no es tan crítico como el anterior aunque no es tan bueno como para el de 70%, por tanto lo mejor es tomar un valor de tasa de cruce del 70 al 80%, para tener un punto medio entre efectividad y eficiencia.

Un aspecto interesante es que en cuanto al cruce realizado en cualquier parte de la cadena perteneciente a un individuo comparado con el cruce realizado por cromosomas (pares dirección-distancia) este último tiende a ser mas eficiente en el sentido de lo rápido que se puede llegar a una respuesta aceptable; estos resultados no se muestran aunque los aquí presentes en la figura 46 y siguientes, están realizados con este último esquema mencionado.

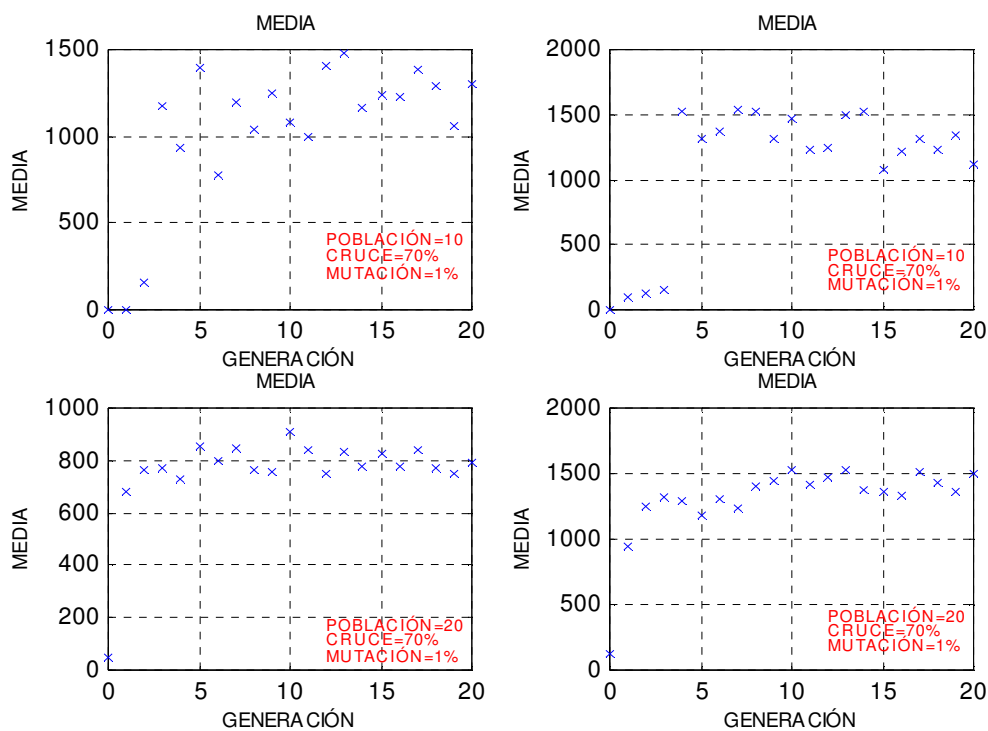
Figura 46. Resultados del mejor individuo variando el tamaño de población entre 10 y 20 individuos, además de mantener la tasa de mutación y variar la tasa de cruce.

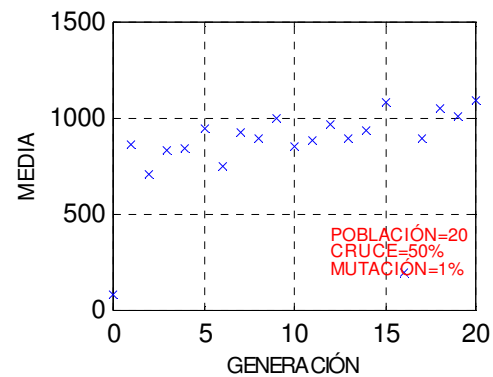
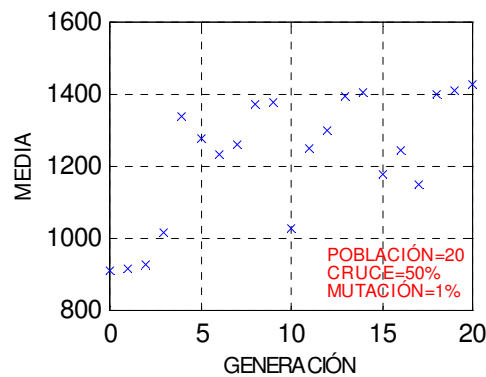
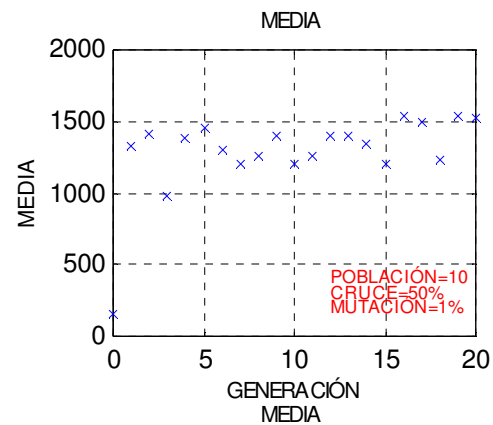
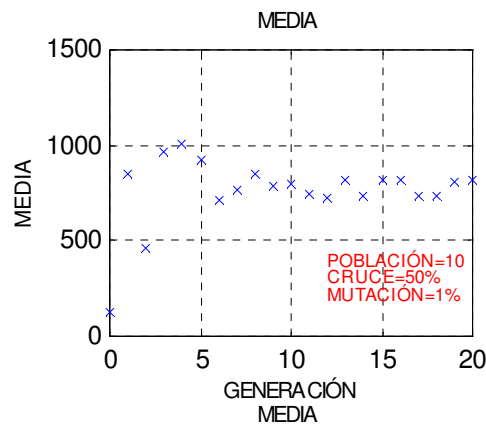
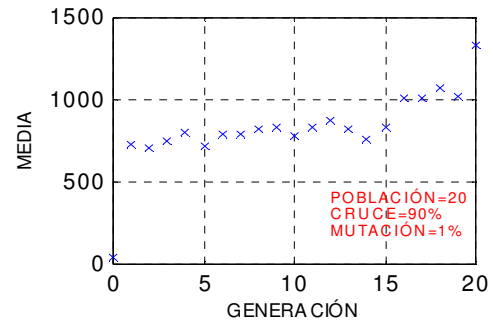
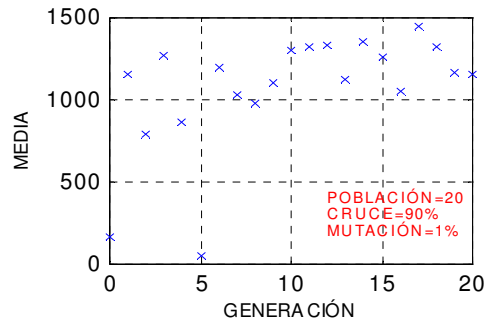
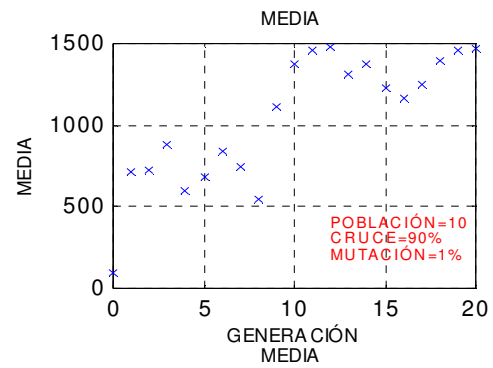
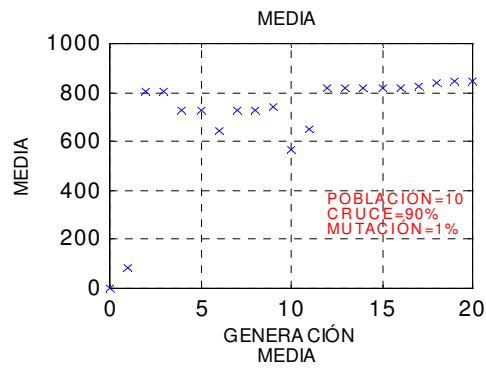




Fuente: Autor del proyecto.

Figura 47. Resultados de la media poblacional variando el tamaño de población entre 10 y 20 individuos, además de mantener la tasa de mutación y variar la tasa de cruce.

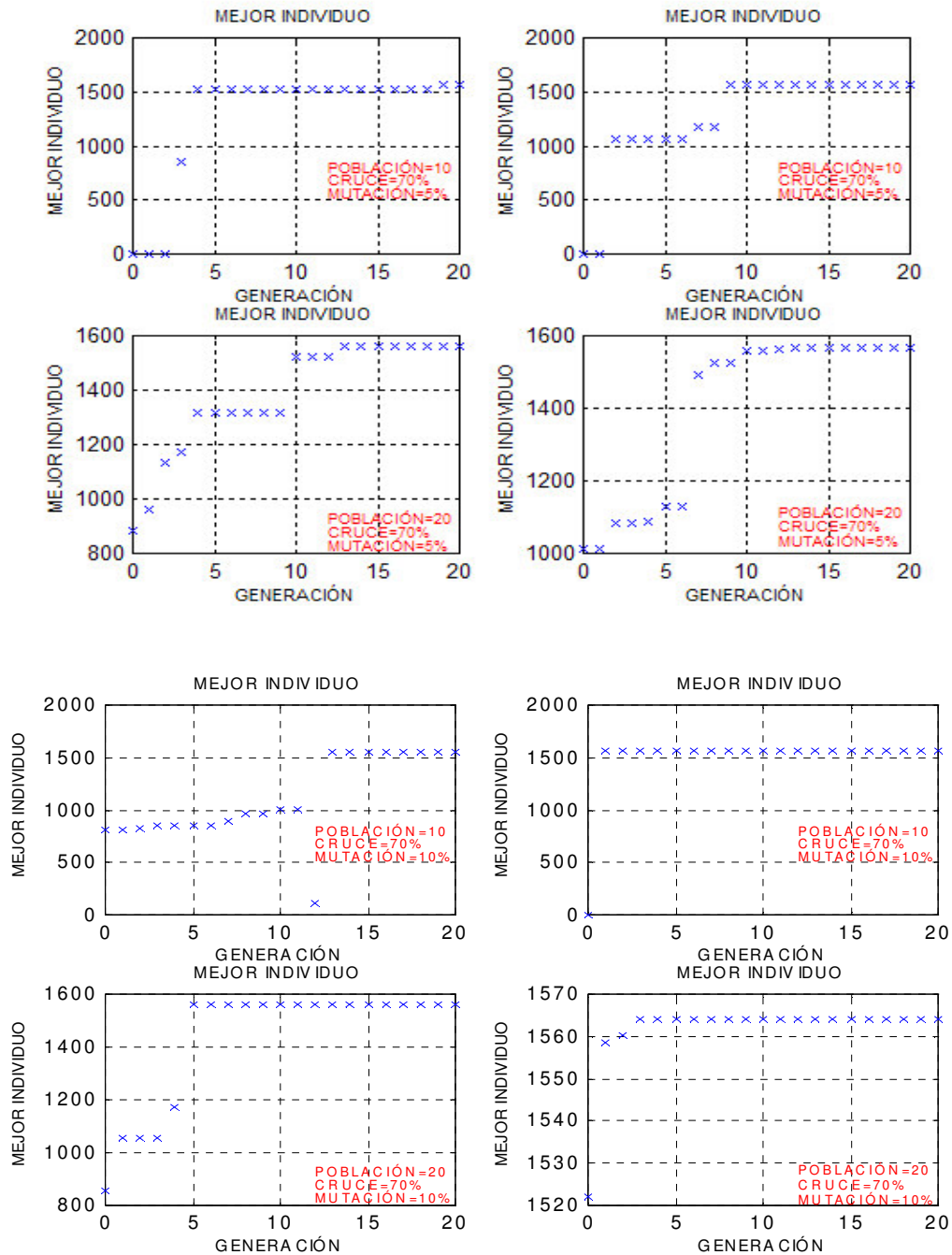




Fuente: Autor del proyecto.

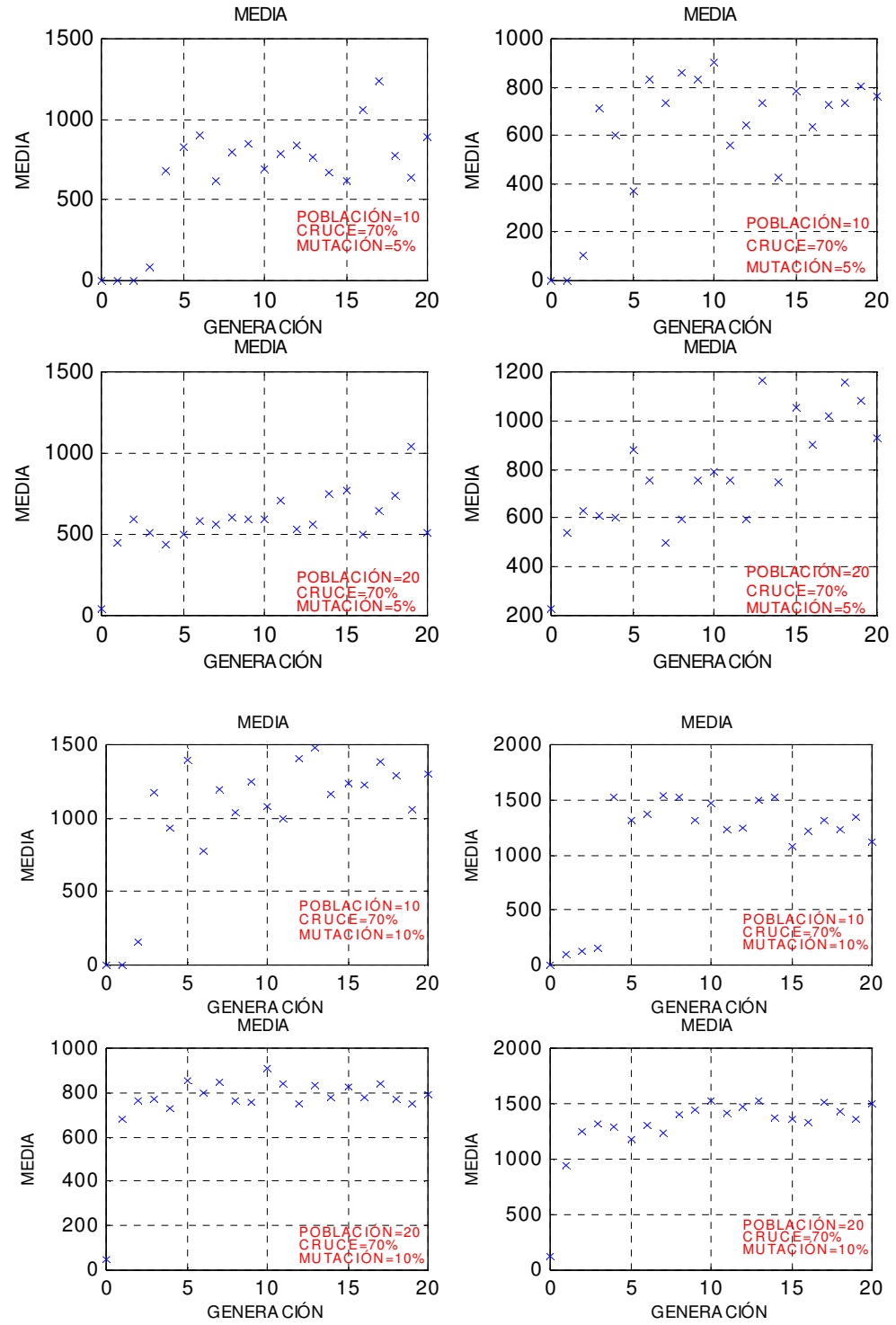
En cuanto a la tasa de mutación los resultados del mejor individuo se muestran en la figura 48; estos se hicieron para diferentes tamaños de población y con tasa de cruce del 70%. La figura 49 muestra los resultados de la media.

Figura 48. Resultados del mejor individuo variando el tamaño de población y la mutación.



Fuente: Autor del proyecto.

Figura 49. Resultados de la media variando el tamaño de población y la tasa de mutación.



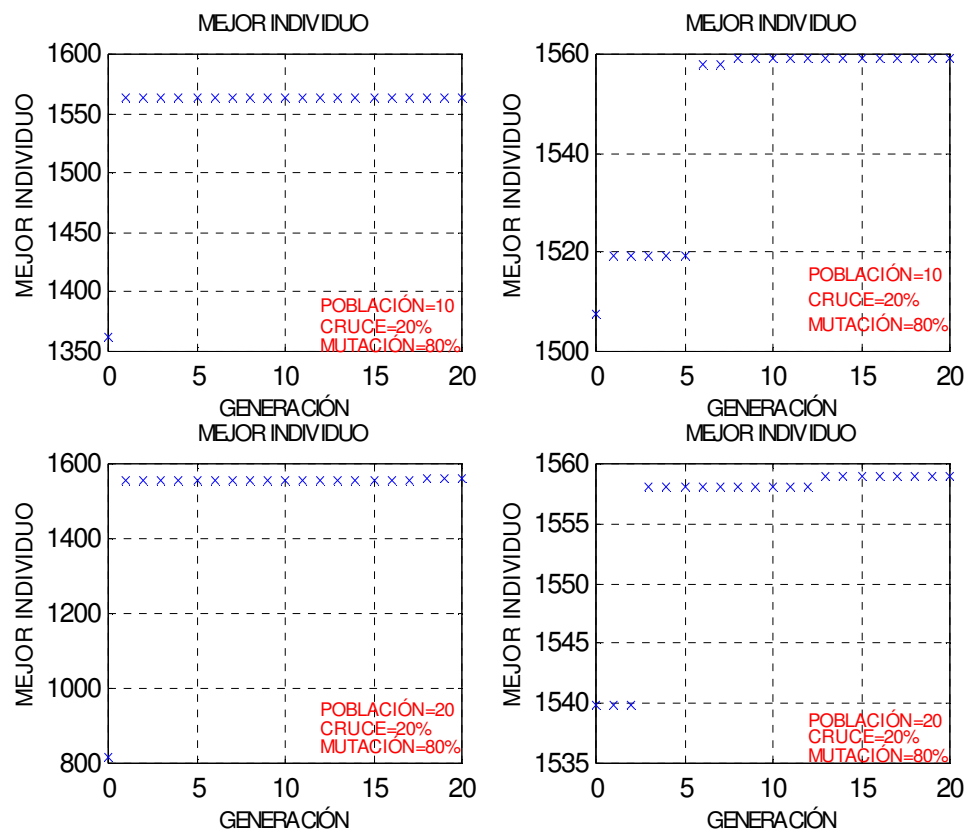
Fuente: Autor del proyecto.

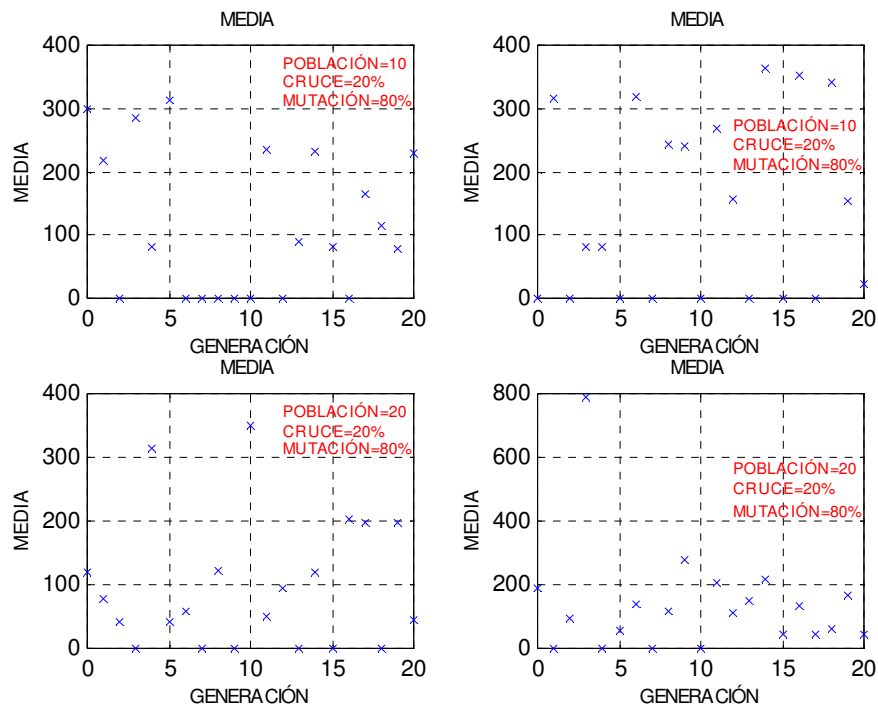
De las anteriores gráficas se puede concluir que para una tasa de mutación de valor medio (5%), la media de la población tiene una tendencia lejos del óptimo y en el tercer caso una tasa de mutación de valor alto (10%) conduce a una media con un valor aún mas bajo mostrando su lejanía del óptimo, lo cual demuestra en forma efectiva la tendencia que se presenta en la mayoría de artículos referentes a AG, en los cuales se afirma que este parámetro debe tener un valor bastante bajo, generalmente por debajo del 1%.

Otra forma sencilla de demostrar el buen comportamiento que puede presentar un AG cuando se utilizan valores para la tasa de cruce altos y para la tasa de mutación bajos, es realizando una simulación de un caso discorde.

En la figura 50 Se muestra los resultados que se obtuvieron tomando una tasa de cruce del 20% y tasa de mutación del 80%.

Figura 50. Resultados del mejor individuo y la media con tasa de cruce 20% y mutación 80%



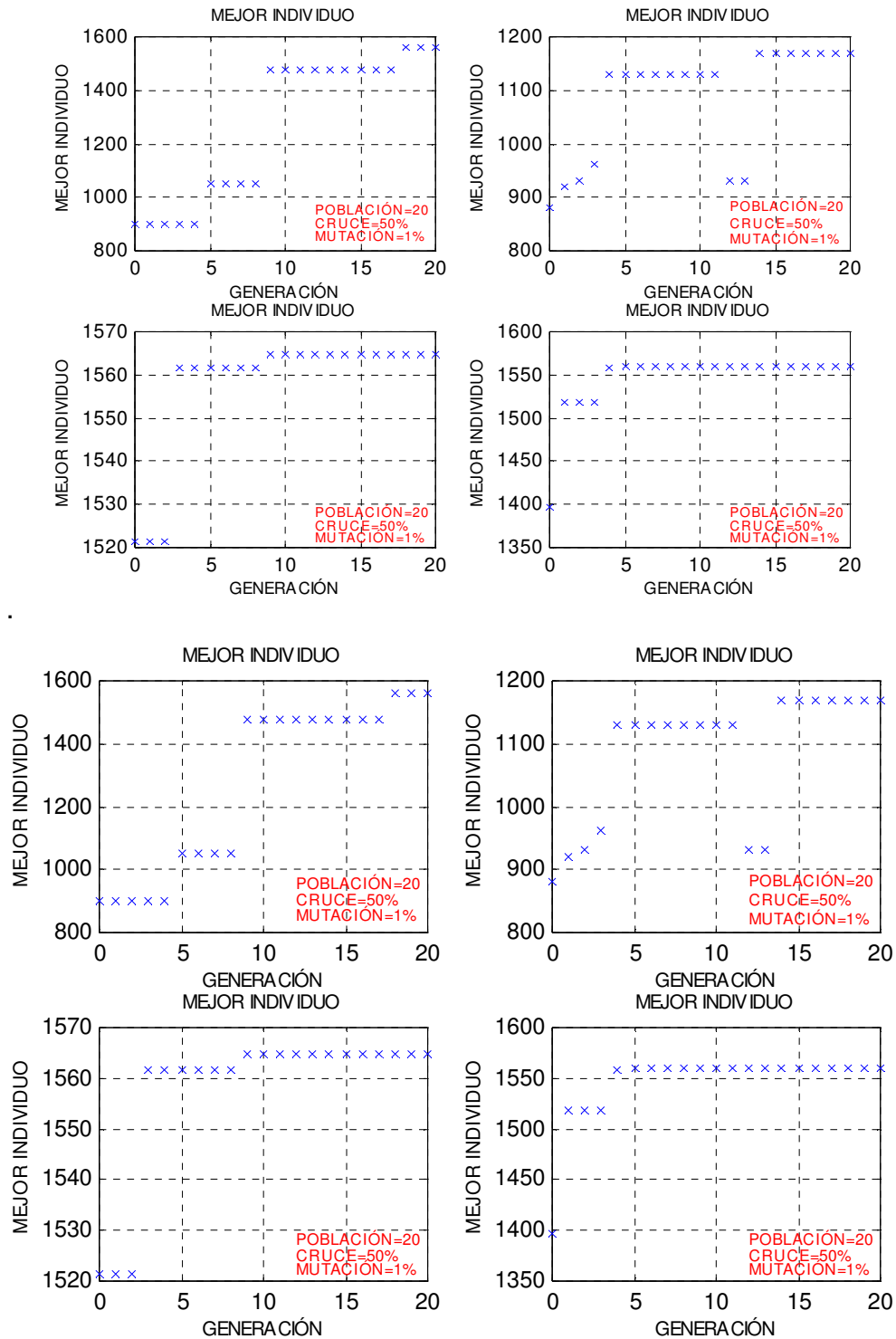


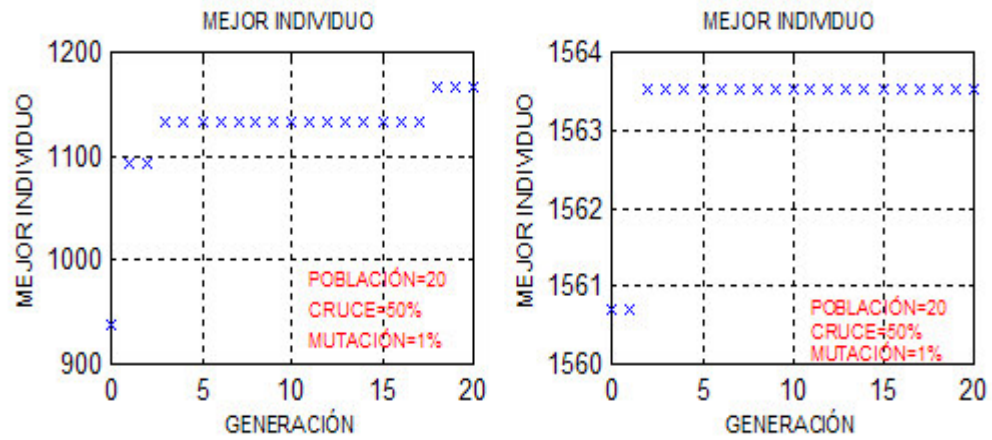
Fuente: Autor del proyecto.

De estas figuras se puede deducir que la tendencia en cuanto a la media se refiere es bastante pobre pues los valores siempre están por debajo de los resultados obtenidos en los casos anteriores (valores por debajo de 1500). En cuanto al mejor individuo el algoritmo encuentra una respuesta aceptable la cual no es mas que el producto de la aleatoriedad implícita en el proceso que otra cosa; esto también demuestra la bondad de la estrategia de estar manteniendo el mejor individuo a lo largo de todo el proceso. Con el fin de comprobar que la tendencia de ciertos parámetros del AG se mantienen se procedió a correr estos casos varias veces para poder determinar si se mantenía este comportamiento. Para este proceso se escogió el caso de tamaño de población de 20, tasa de mutación del 1% y tasa de cruce variable para 50, 70 y 90% corriendo cada caso diez veces.

En la figura 51 se muestra el caso para tasa de cruce del 50%, con lo cual se confirman los resultados obtenidos de primera mano; en cuanto a los otros casos el resultado es muy semejante, pero no se ilustran en el texto. La figura 52 muestra algunos resultados en cuanto al espacio de trabajo y el obstáculo tipo propuesto, donde nuevamente es claro que el algoritmo responde, acercándose el óptimo deseado(diagonal).Y a través de la figura 53 se pueden observar las respuestas del algoritmo de acuerdo a las diversas formas de obstáculos.

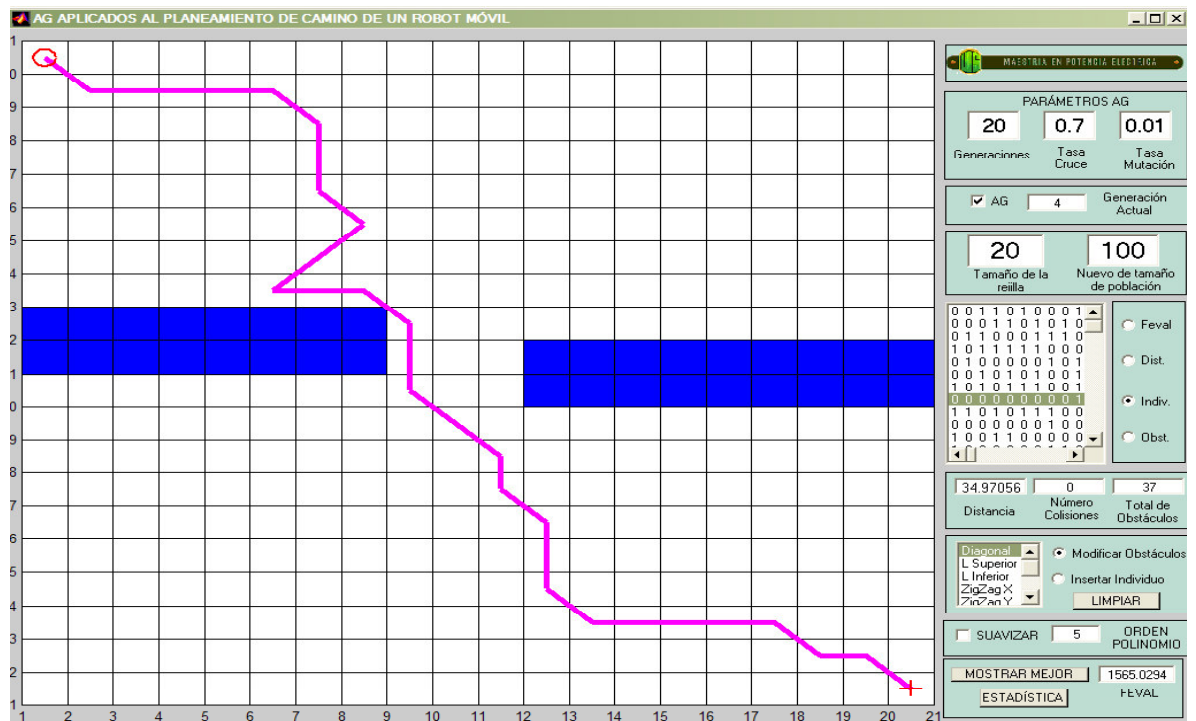
Figura 51. Resultados del mejor individuo y la media con tasa de cruce 50% y mutación 1%.





Fuente: Autor del proyecto.

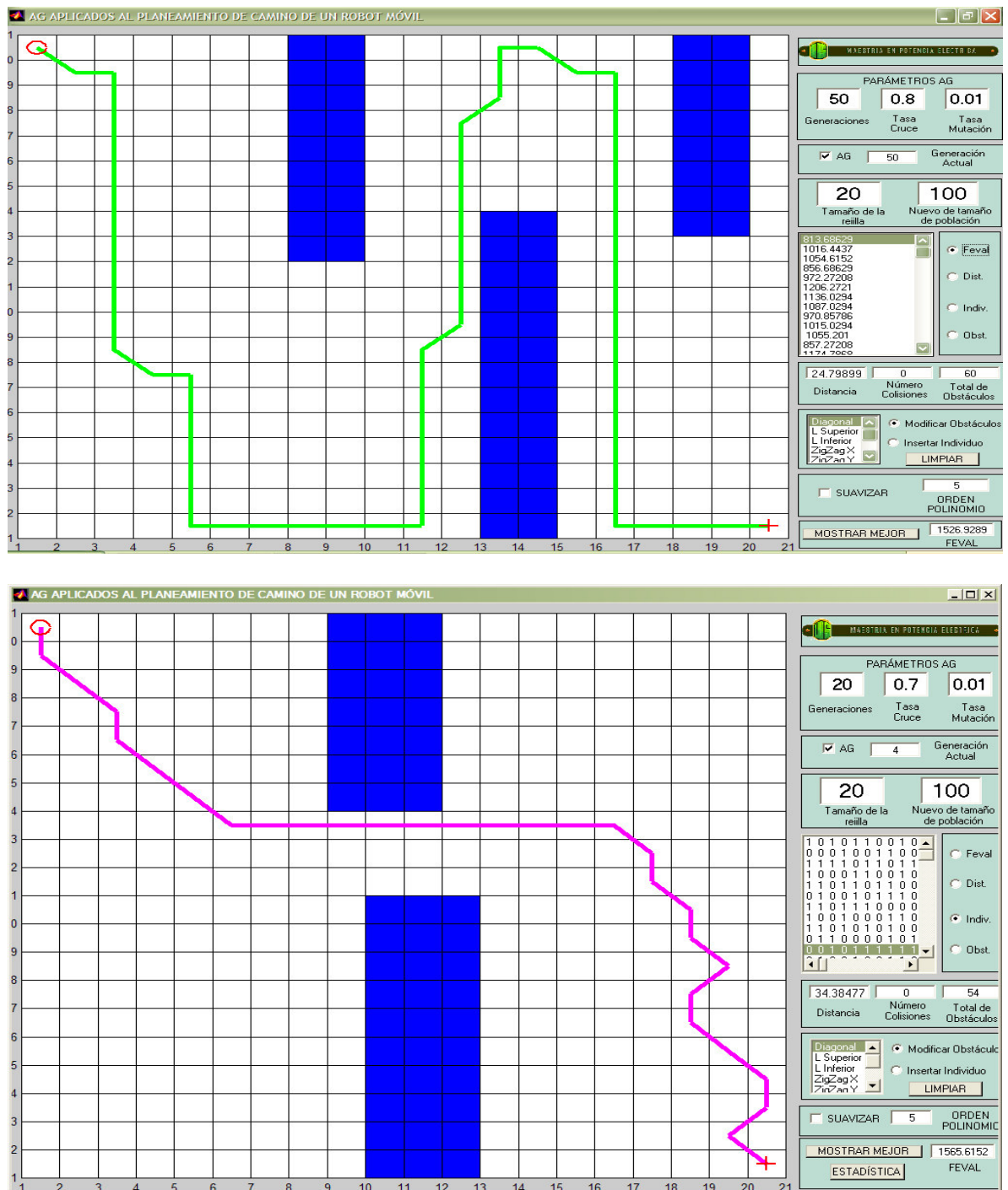
Figura 52. Respuestas del algoritmo a obstáculos tipo.





100

Figura 53. Respuestas del algoritmo a diversas formas de obstáculos.

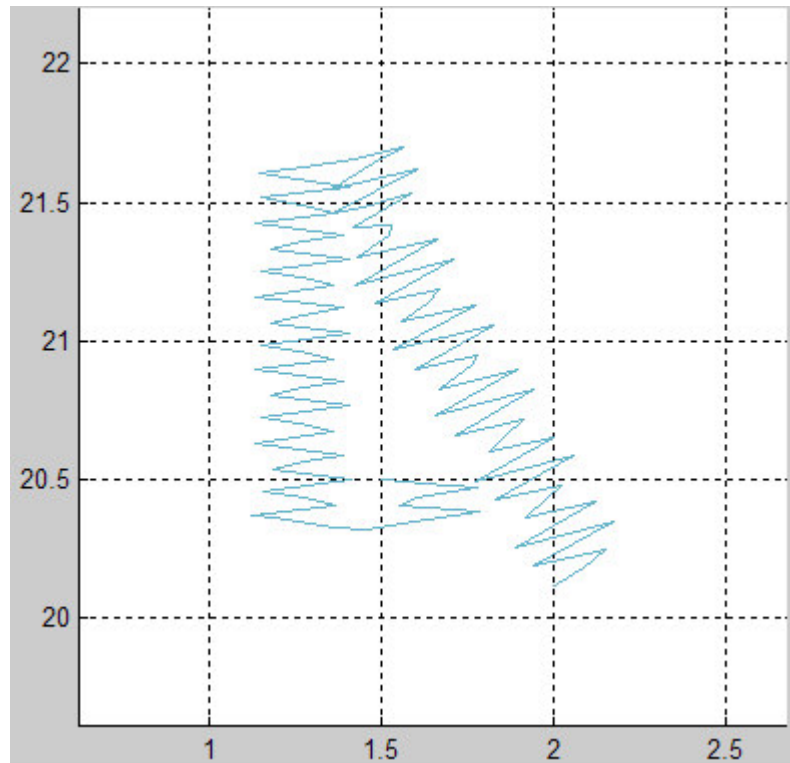


Fuente: Autor del proyecto.

6.3 ESTUDIO DE LOS RESULTADOS OBTENIDOS EN CUANTO A PLANEAMIENTO DE TRAYECTORIA

Para el estudio de esta metodología se realizó un proceso muy similar al planteado en el aspecto referente al planeamiento de camino. El objetivo de esta etapa es generar los perfiles de velocidad y aceleración, que permitan obtener una respuesta adecuada semejante al camino que se obtuvo como resultado final en el planeamiento que lleva el mismo nombre. El análisis partió de considerar unos valores de aceleración lineal y angular un tanto arbitrarios $3[\text{m/s}^2]$ y $4[\text{rad/s}^2]$ con lo cual se observó un comportamiento bastante zigzagueante como se observa en la siguiente figura 54. Se encuentra que para valores bajos de aceleraciones, como es el caso de $0.1[\text{m/s}^2]$ y $0.01[\text{rad/s}^2]$, el comportamiento es mas “suave”. El camino tomado como referencia fue el de una diagonal que cruza el espacio de trabajo, el cual se escogió por su simplicidad y porque a la vez permite dar continuidad al estudio dado en el planeamiento de camino; también se corrieron otros casos que se mencionan mas adelante.

Figura 54. Comportamiento zigzagueante para valores de aceleraciones altos.



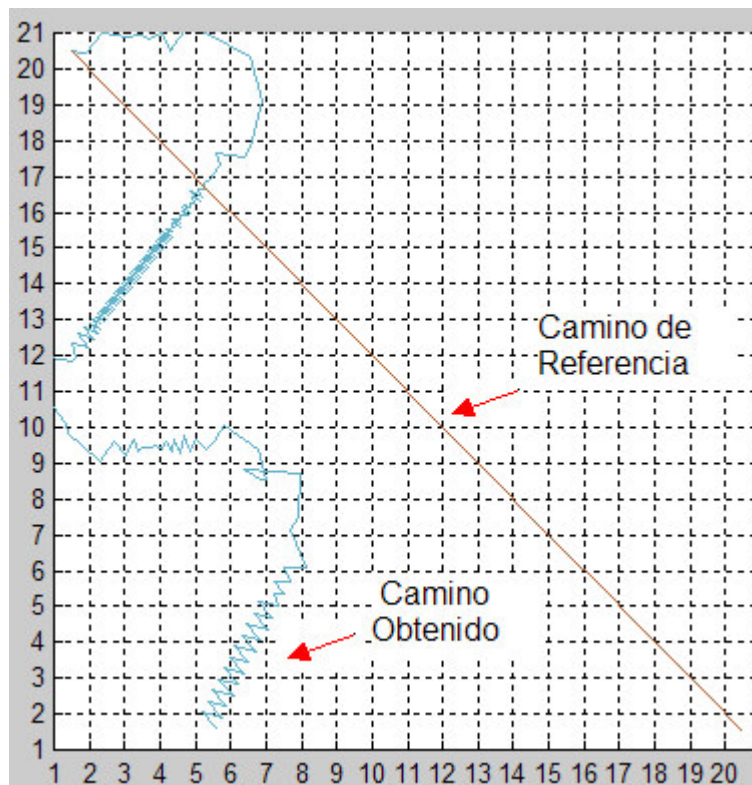
Fuente: Autor del proyecto.

La figura 55 ilustra el resultado de un individuo que se aproxima al camino deseado; es claro el comportamiento oscilatorio suave que mantiene, el cual se da

por mantener valores bajos de aceleración, además de mantener las restricciones referentes a no superar velocidades máximas.

El análisis que se realizó en esta sección se hizo mas observando, que el camino fuese parecido al que se tiene como referencia y no tanto al análisis estadístico de resultados, aunque a continuación se muestran algunos datos tomando como referencia el caso tipo(diagonal), el cual se simuló para 500 generaciones con tasa de cruce del 80%, tasa de mutación de 1% y en la figura 56 con 100 individuos por generación y en la figura 57 con 50 individuos por generación.

Figura 55. Ejemplo de un camino obtenido mediante la herramienta para planeamiento de trayectoria.



Fuente: Autor del proyecto.

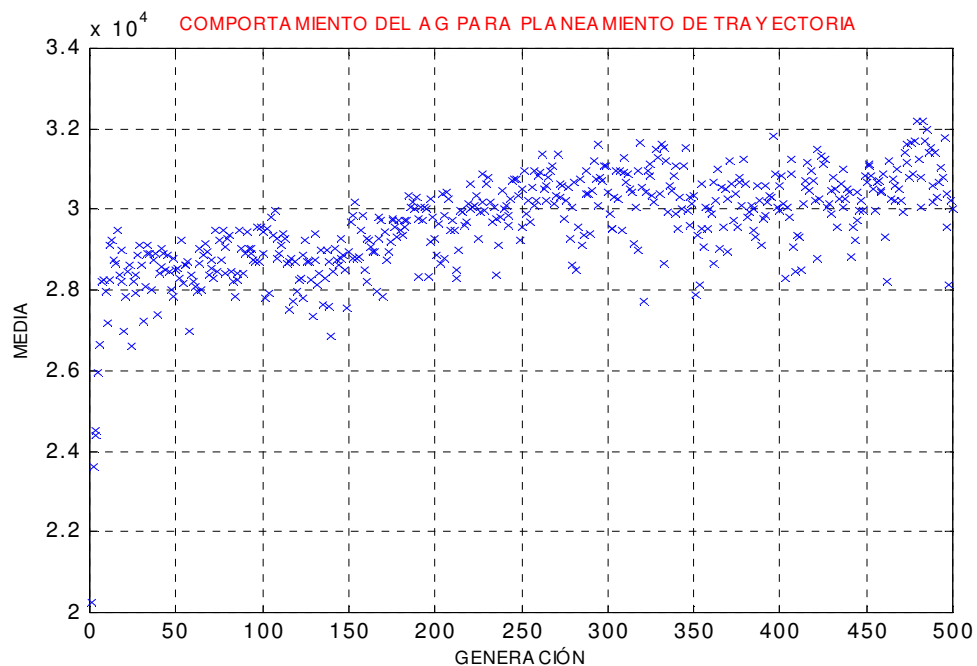
Es claro de estas gráficas que a medida que se aumenta el número de generaciones, la media se hace mas constante y por tanto la población es mas homogénea, aunque es mas uniforme para una población grande, en este caso 100. La figura 58 se muestran algunos resultados, para el caso en cuestión.

Para caminos con cierto grado de dificultad la aproximación no es tan acertada, lo cual deja de presente que la metodología puede tener ciertas falencias. La figura 59 muestra algunos resultados para estos caminos. La razón de esta lejanía con

respecto a lo que se desea podría entenderse un poco analizando lo que ocurre en el caso del movimiento de una celda a otra. Estos detalles se muestran en la figura 60 donde se hace visible, que para la unidad básica de desplazamiento dada, es poco probable que se pueda seguir el camino de referencia si se asumen unas orientaciones y posiciones iniciales.

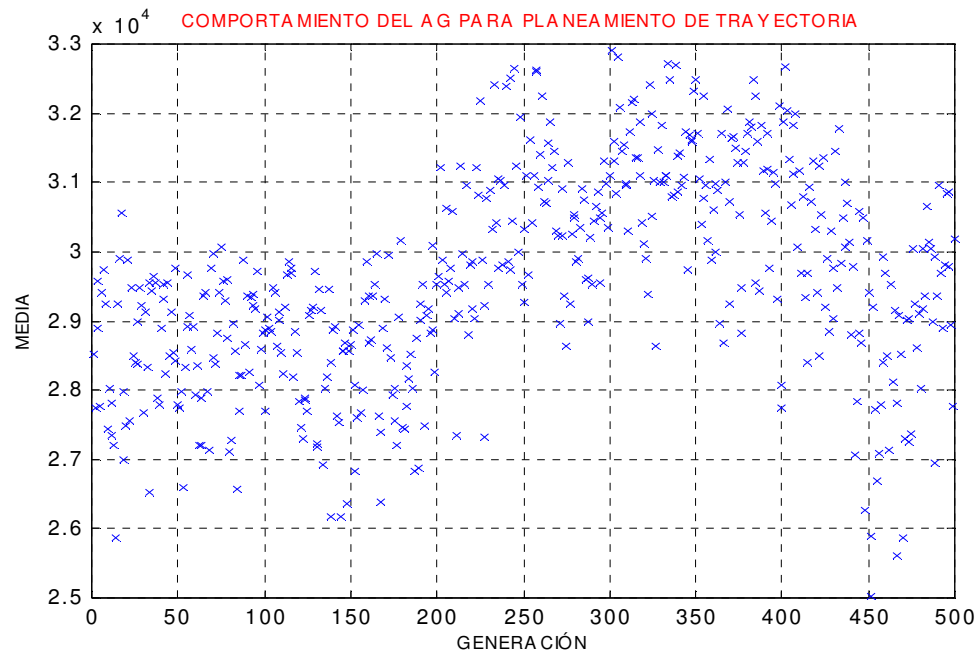
Para el caso de la figura 60 la orientación inicial fue de $\beta_{etaini}=13.1108$ y $t_{etaini}=1.7259$, y se muestran los diversos resultados para todas las posibles situaciones de codificación, en particular si lo que se desea es por ejemplo ir de una celda a otra mediante una diagonal. De estas gráficas se puede concluir fácilmente que es imposible seguir el camino de una forma bastante fiel a lo que se quiere partiendo de cualquier par de condicionales iniciales a un nivel bastante simple, pues de igual forma será si se quisiera seguir un camino que involucre el desplazamiento entre varias celdas, donde obviamente el error se irá aumentando a medida que se avance en el camino. Los resultados obtenidos no son del todo lo que se desea que fueran, pero se espera que en trabajos futuros se puede establecer una codificación mas adecuada que conduzca a mejores resultados, además de una función de evaluación que permita una mejor discriminación entre los individuos más aptos y los que tengan una baja aptitud, o proponer otra metodología que implique el uso de los AG como herramienta alternativa de solución.

Figura 56. Grafica que relaciona la media para 500 generaciones con tamaño de población de 100.



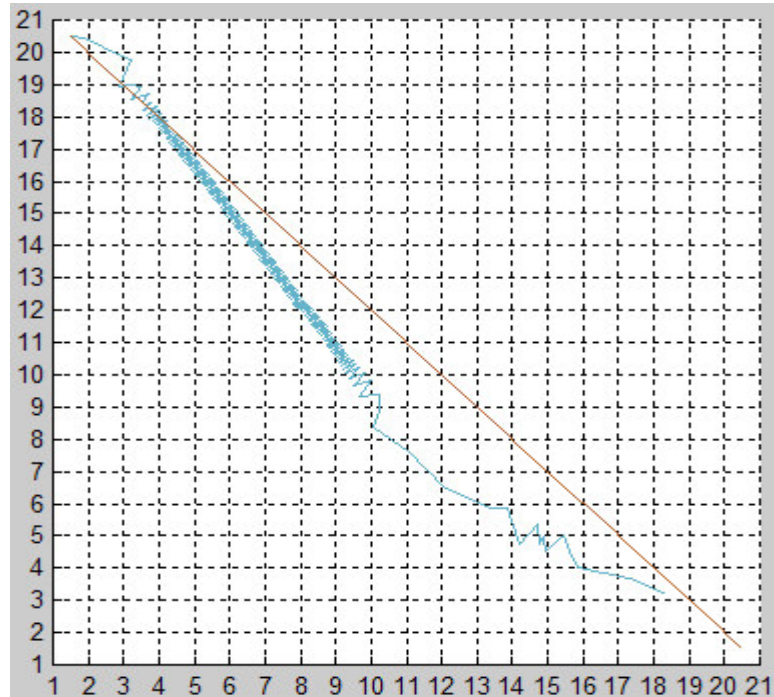
Fuente: Autor del proyecto.

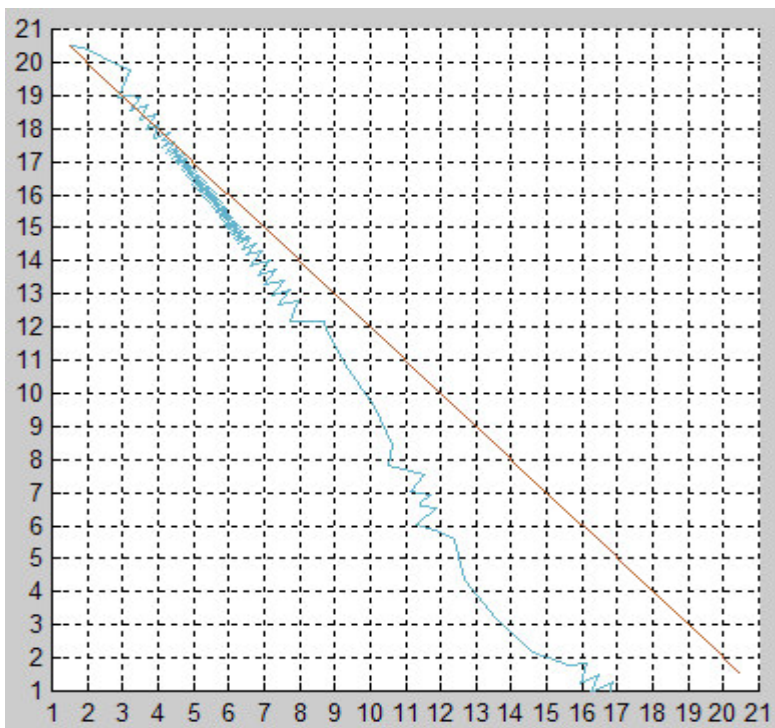
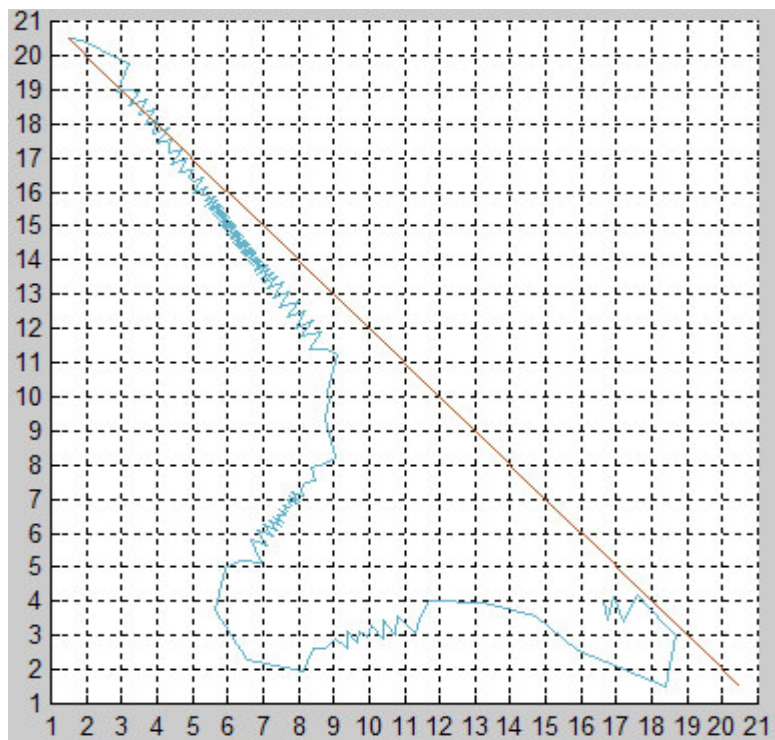
Figura 57. Grafica que relaciona la media para 500 generaciones con tamaño de población de 50.

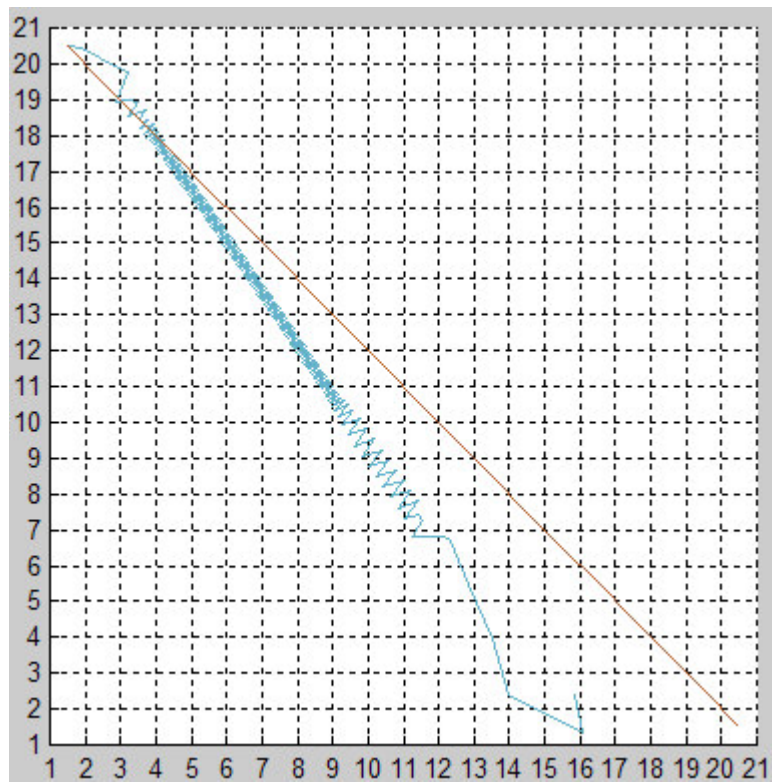


Fuente: Autor del proyecto.

Figura 58. Resultados para el camino de referencia (diagonal).

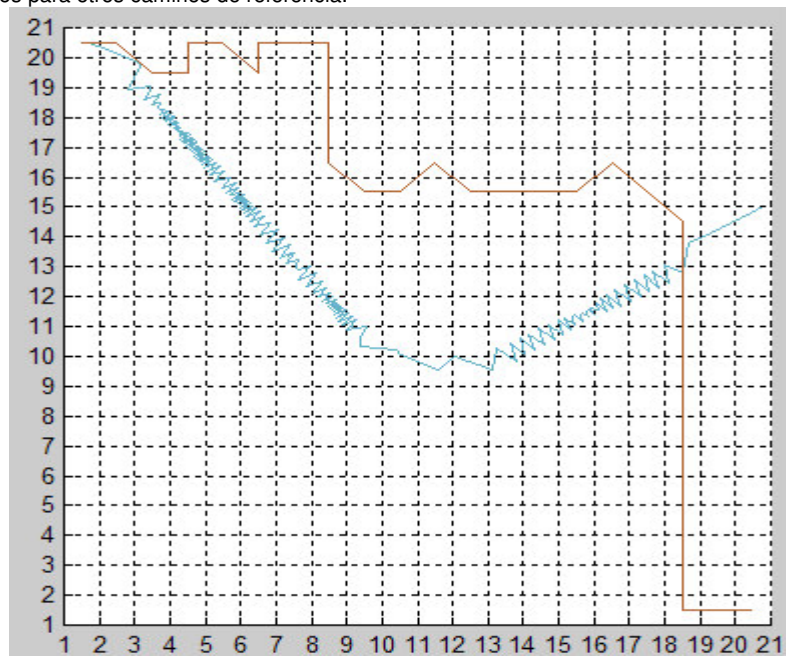


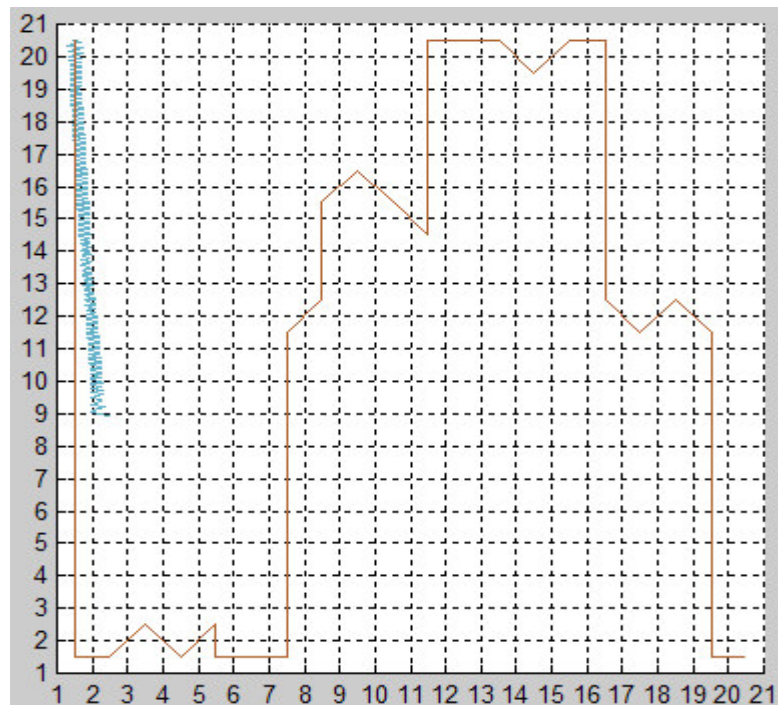




Fuente: Autor del proyecto.

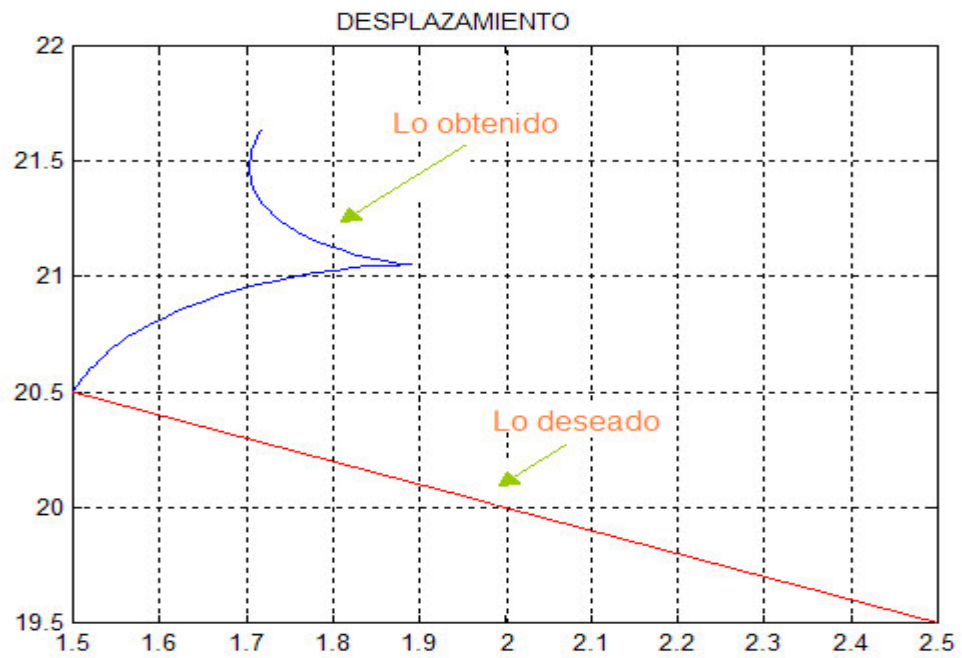
Figura 59. Resultados para otros caminos de referencia.

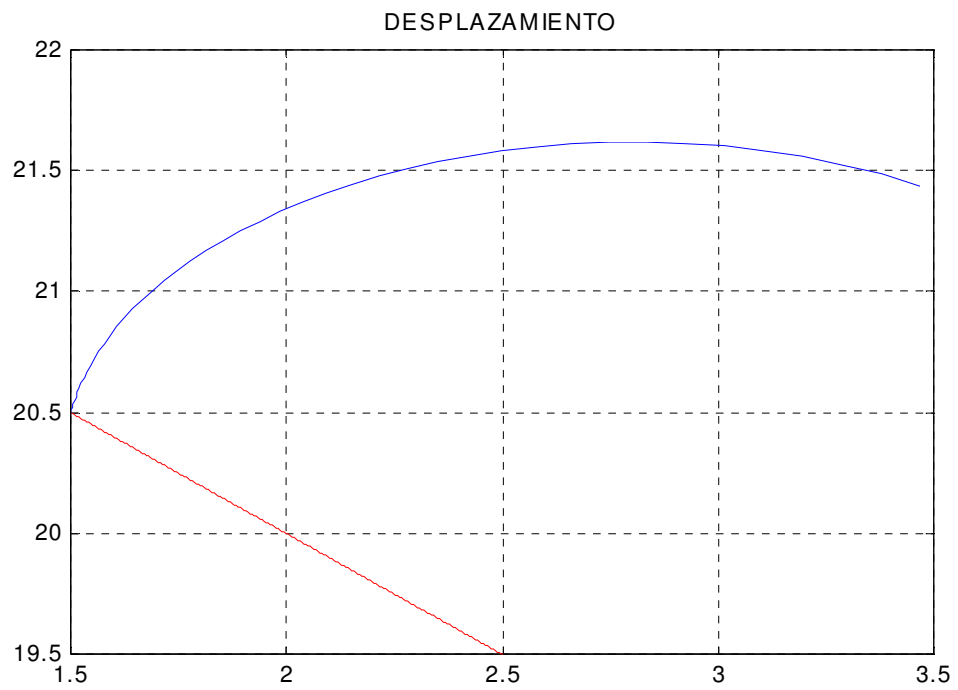
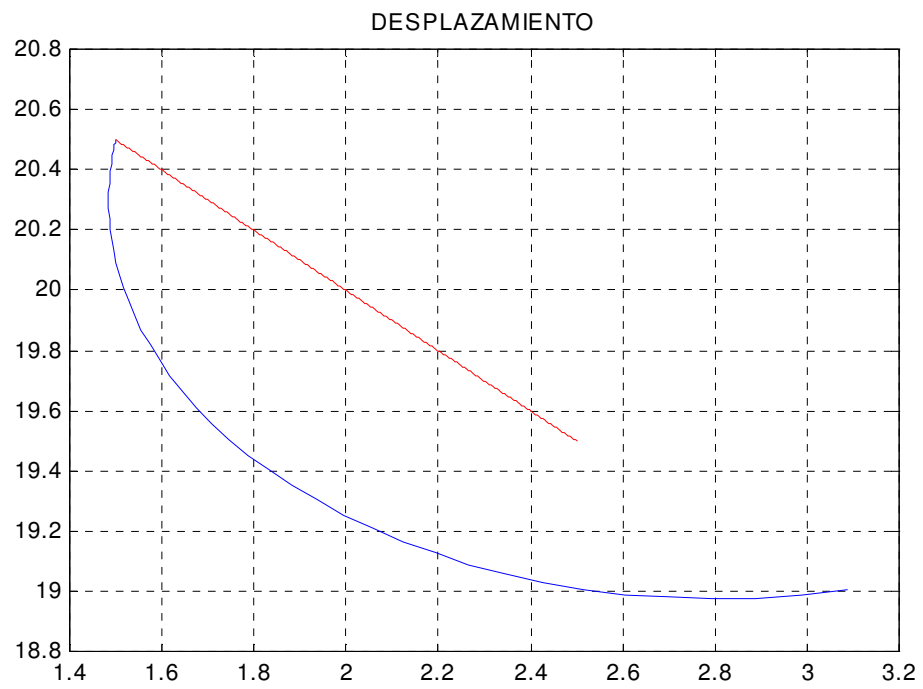


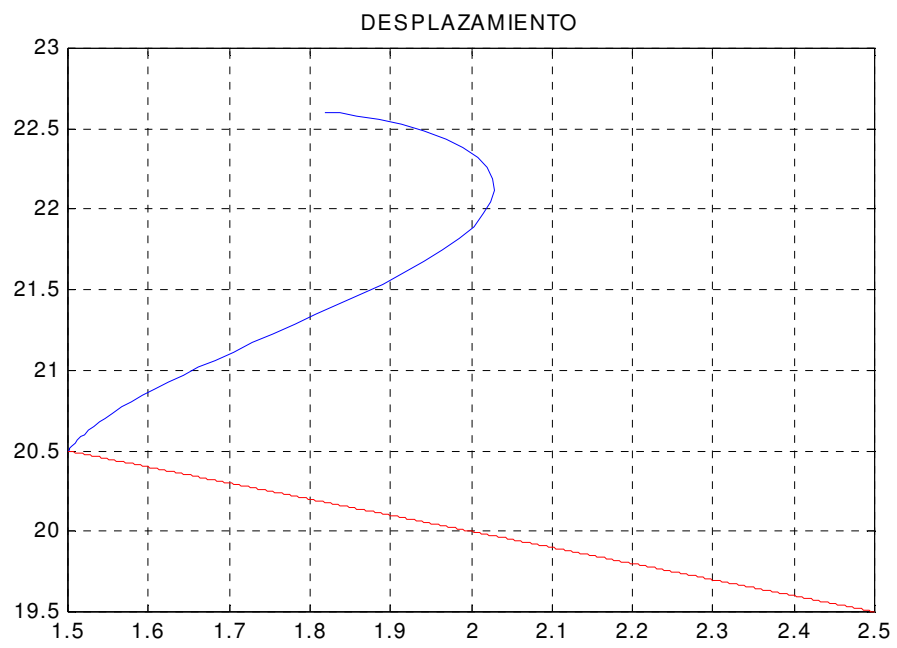
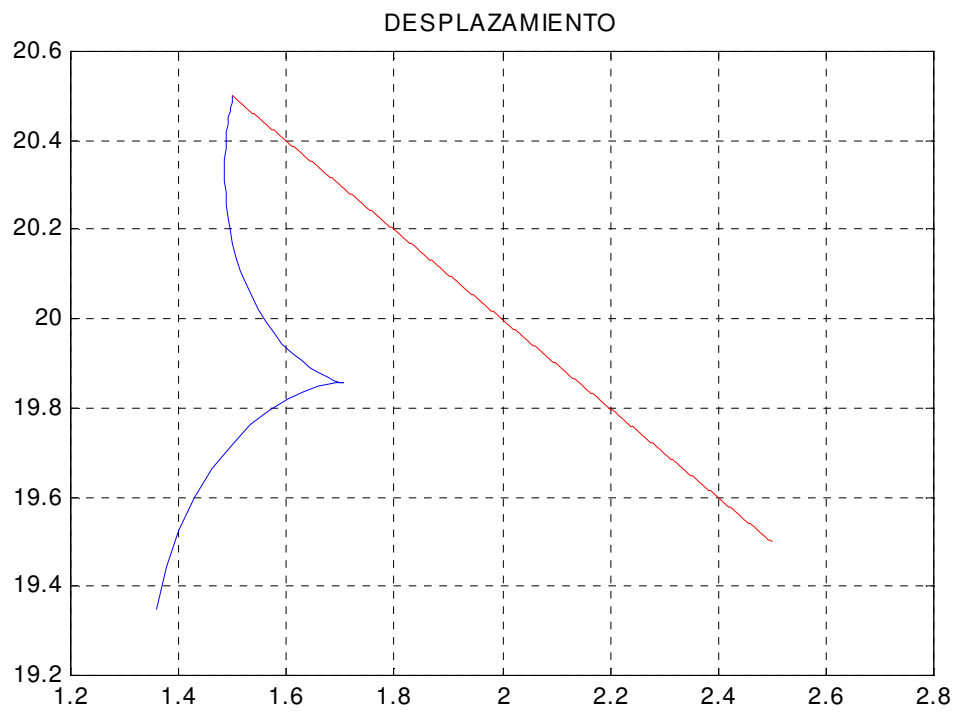


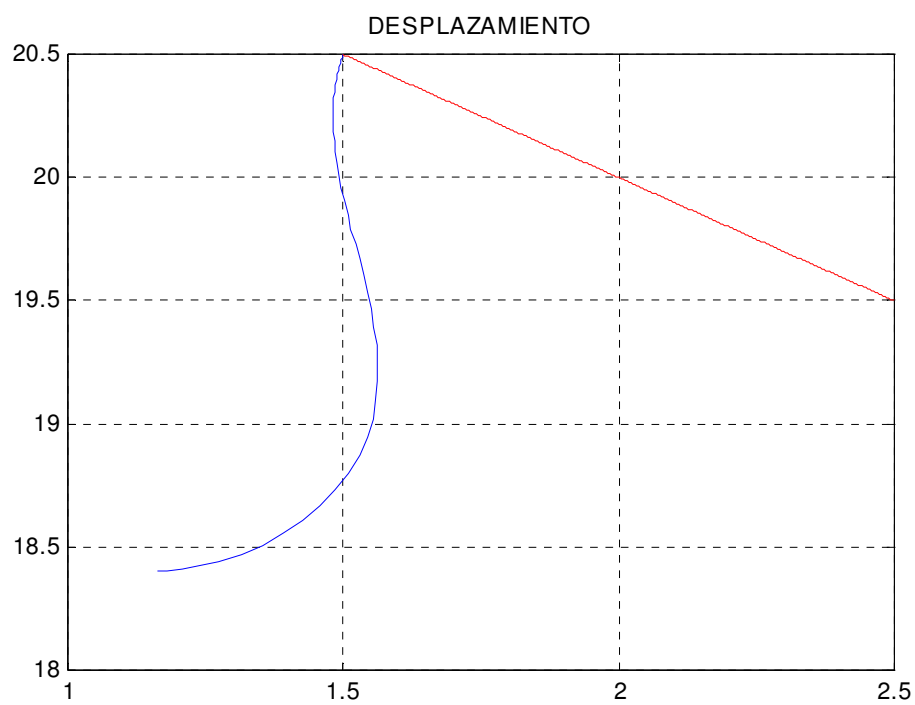
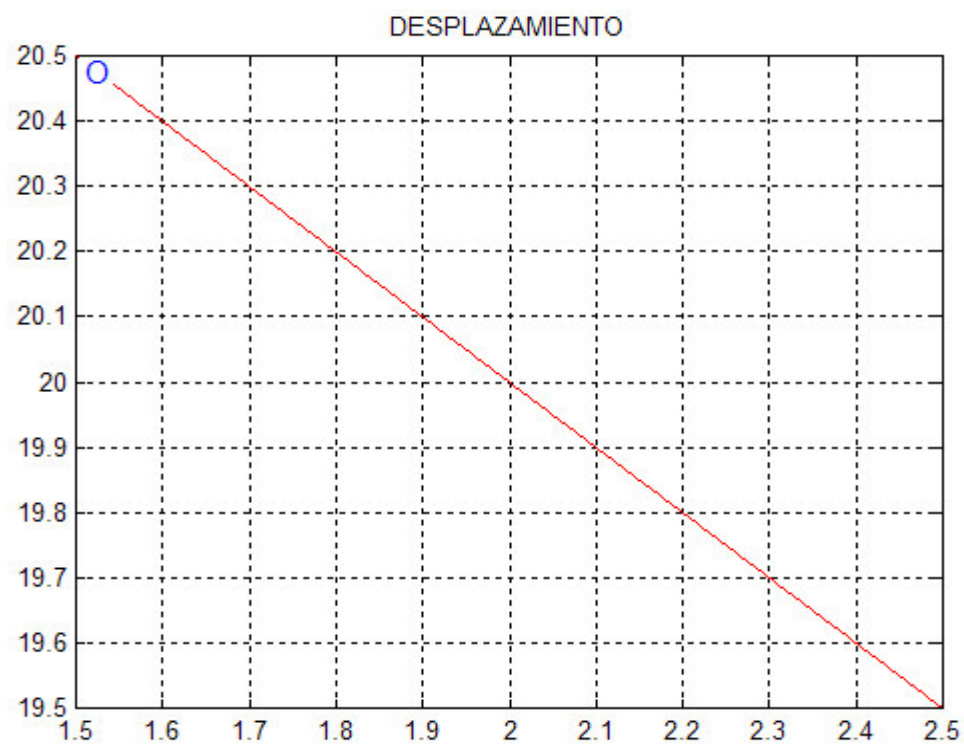
Fuente: Autor del proyecto.

Figura 60. Resultados para desplazamientos básicos de una celda a otra.









Autor del proyecto.

Fuente:

7. CONCLUSIONES

Posterior a todo el trabajo detallado y comentado anteriormente, se puede obtener un grupo interesante de conclusiones y recomendaciones como son: se desarrolló una herramienta computacional que mediante simulación obtiene una solución al planeamiento de movimiento de un robot móvil utilizando los algoritmos genéticos como estrategia de búsqueda. Para lo anterior se propuso una codificación con base en rutas monótonas en x y y , que permiten expresar en una forma aceptable los diversos caminos que pueden darse en un espacio bidimensional con obstáculos fijos.

De entre diversidad de robots móviles que existen, se realizó la propuesta de un modelo de robot tipo triciclo con características no holonómicas, el cual responde en forma adecuada a los perfiles de velocidad que se le impusieron. En cuanto al espacio de trabajo se propuso un modelo basado en la descomposición del espacio de trabajo en celdas, la cual permite acercarse en buena forma a lo que puede ser el mundo real y las diversas formas de obstáculos que pueden encontrarse en este.

Los AG, como método de búsqueda presentan una respuesta satisfactoria ante una variedad no compleja de obstáculos, aunque para los casos que se corrieron, teniendo en cuenta que el número de generaciones fue bastante pequeño, no se obtuvo el óptimo esperado. En caso de obstáculos de tipo complejo se sugiere realizar un análisis a nivel local, con lo cual los resultados esperados se pueden perfeccionar. Los parámetros de mayor influencia en la sintonización del algoritmo genético simple son los referentes a la tasa de cruce y a la tasa de mutación. Se comprobó además que una tasa de cruce aunque debe tener un valor alto comparado con la mutación, no debe ser superior al 80%. Algo similar ocurre para la tasa de mutación. El AG tiene una buena respuesta para valores del orden del 1%. Para lo anterior se pudo de igual forma comprobar la dificultad que existe a la hora de sintonizar un AG, y lo mas adecuado podría ser tener una dinámica en el comportamiento de los parámetros del AG. Los operadores utilizados para las diversas etapas del AG (selección, cruce y mutación) fueron el cruce en un solo punto, la selección mediante el método de la ruleta pesada y la mutación de tipo aleatorio a lo largo de toda la cadena. Un aspecto adicional que se pudo comprobar es la importancia de la función de evaluación, que al igual que la fase de codificación, son etapas críticas y obtener la que mejor resultados produzca es todo un arte.

Dada la respuesta del AG, se propone para estudios posteriores establecer diversos métodos de cruce y en cierta forma incluir el operador de elitismo tal como se encuentra referenciado en la mayoría de los artículos que se ocupan de

los AG. Para el presente trabajo se propuso una versión de elitismo que logró mejorar los resultados que se habían obtenido hasta ese momento, manteniendo el mejor individuo que se fuera consiguiendo a lo largo de toda la simulación.

Se propuso una metodología para el seguimiento de una trayectoria en la cual se aplican los AG en la solución de este problema realizando la búsqueda en el espacio de velocidades; en forma desafortunada los resultados no fueron los mejores para esta propuesta, aunque si se muestra una tendencia que en algo se acerca a lo planeado, y por tanto si se reduce el problema a realizar un planeamiento de tipo local los resultados obtenidos podrían mejorar. Sumado a esto se espera que en estudios posteriores se cuestione la codificación y función de evaluación aquí presentadas, las cuales pueden ser el origen de este comportamiento algo deceptivo, es decir que no obtiene solución, para el AG propuesto.

El modelo dinámico propuesto para el robot tipo triciclo ofrece la posibilidad de ahondar en el estudio de este tipo de sistemas, aunque para el presente trabajo solo se trabajó con los resultados referentes al modelo cinemático. En caso de querer realizar una optimización a partir de energía se dejaron las ecuaciones para este caso, y se espera que sean de análisis posteriores.

BIBLIOGRAFÍA

ANDRÉA, B and BASTIN, G. Modelling and control of non holonomic wheeled mobile robots. En: IEEE Conf. International Robotics and automation. Sacramento. (April 1991); p.1130-1134.

ARAI, Hirohiko. Human interface for maneuvering nonholonomic systems. En: IEEE. Conf. International Robotics and automation. Seoul. (May 21-26. 2001); p.1870-1871.

ARENAS JIMÉNEZ; Alexander y SERRANO ARANGO, Diego. Aplicación de los algoritmos genéticos en la optimización de una bomba de pistones axiales de desplazamiento fijo. Vol. 1. Bucaramanga, 2000, p.166-212. Trabajo de Grado (Ingeniero Mecánico). UIS. Facultad de Ingenierías Fisicomecánicas.

ÁVILA, Freddy. Propuesta de una Red de Intercambio de calor para Toppin usando Algoritmos Genéticos. Bucaramanga, 2001. Trabajo de Grado (Ingeniero Químico). UIS. Facultad de Ciencias Fisicoquímicas.

BEASLEY, David; BULL, David and MARTIN, Ralph. An overview of genetic algorithms: part. 1: fundamentals. Vol. 15. UCISA. 1993; p.1-15.

_____. An overview of genetic algorithms: part. 2, research topics. Vol.15, UCISA. 1993; p.1-15.

BECERRA Julio y ROJAS Freddy. Optimización de trayectorias para un robot móvil utilizando algoritmos genéticos. Bucaramanga, 2003. 120 p. Trabajo de Grado (Ingeniero Electrónico). UDI. Facultad de Ingeniería Electrónica.

BERTONI, Alberto and DORIGO, Marco. Implicit Parallelism in genetic algorithms. Appeared in Artificial Intelligence (61) 2. (April 1993); p.307-314.

BESSIÈRE, Pierre and AHUACTZIN, Juan. The “Ariadne`s clew “ algorithm: global planning with local methods. En: IEEE. Int. Conf. On intelligent robots and systems Vol.2. Yokohama. (July 1993); p.1373-1380..

_____, _____. Using genetic algorithms for robot motion planning. En: IEEE. Int. Conf. On intelligent robots and systems. (July 1993); p.1-6.

BLOCH, A. M. Nonholonomic mechanics and control. New York: Springer, 2003. p.1-30.

BORESTEIN, J. "Where am I?". Sensors and methods for mobile robot positioning. The University of Michigan. p.19-28.

C, Alex and VERMEER, Pamela. The weighted Voronoi diagram and its applications in least-risk motion planning. En: IEEE. 1989.

CARNAHAN, Joseph and SIMHA, Rahul. Nature's algorithms. En: IEEE. (April May. 2001); p.21-24.

CAUSSE, Oliver and CROWLEY, Jim. Navigation with constraints for an autonomous mobile robot. LIFIA, CNRS. p.1899-1905.

CAVALCANTI PACHECO, Marco Aurelio. Algoritmos genéticos: principios e aplicaciones. Pontificia Univeridade Católica do Rio de Janeiro. p.1-7.

CHOSSET, Howie and BURDICK, Joel. Sensor-based: Exploration: The Hierarchical generalized Voronoi graph. The international Journal of robotics research. Vol.19 No.2. (Feb 2000); p.96-124.

CHUANG; Jen. Potencial- based modeling of two dimensional workspace using several source distributions. En: Proc. IEEE. Int. Conf. Multisensor fusion and integration for intelligent systems. Las Vegas: (Oct. 2-5.1994); p.98-102.

DULEBA, Ignacy and SASIADEK, Jerzy. Energy-efficient Newton- based nonholonomic motion planning. En: American control conf. Arlington, V.A. (June 25-27. 2001); p.1859-1863.

FORREST, Stephanie and MITCHELL, Melanie. ¿What makes a problem hard for a genetic algorithm?. Some anomalous results and their explanation. Machine Learning:1993, p.285-319.

FOUX, Guy and BRUCKSTEIN, Alfred. Two dimensional robot navigation among unknown stationary polygonal obstacles. IEEE, robotics and automation. Vol.9, No.1. February. 1993. P.96-101.

GALLARDO, Domingo, COLOMINA, Otto FLÓREZ, Francisco y RIZO, Ramón. Generación de trayectorias robustas mediante computación evolutiva. Universidad de Alicante. p.1-8.

GASS. Saul. Linear Programming. Methods and applications. p.305

GERKE, Michael. Genetic path planning for mobile robots. En: IEEE, American control conference. San Diego: (June 1999); p.2424-2429.

GILL, Mark and ZOMAYA, Albert. Obstacle avoidance in multi-robot systems. Experiments in parallel genetic algorithms. Singapore: World Scientific. 1998; p.59.

GOLDBERG, David. Genetic algorithms in search, optimization, and machine learning. Boston: Addison Wesley, 2000. 412p.

GOLDBERG, David *et al.* Rapid accurate optimization of difficult problems using fast messy genetic algorithms. En: IlliGAL Report No. 9304, (February 1993); p.1-14.

HALPERIN, Dan-KAVRAKI; Lydia and LATOMBE, Jean. Robotics. p.1-18.

HWANG, Yong and AHUJA; Narendra. A potential field approach to path planning. En: IEEE, transactions on robotics and automation, vol. 8, No. 1. (Feb 1992).

_____. Gross motion planning- A survey. En: ACM computing Surveys. Vol. 24. No. 3 , (Sept 1992); p.219-234.

HOCAOGLU, Cem. Planning multi-paths using speciation in genetic algorithms. Electronics Agile Manufacturing Research Institute, and Electrical, Computer, and Systems Engineering Department Rensselaer Polytechnic Institute. Troy, New York. IEEE 1996.

INAGAKI, Jun; HASEYAMA, Miki and KITAJIMA, Hideo. A genetic algorithm for determining multiple routes and its applications. En: IEEE, 1999; p.VI-137-VI-139.

IÑIGO, Rafael M and ALLEY, Daniel. Path planning algorithms for mobile robots. En: IEEE, 1990; p.1032-1036.

LAUMOND, Jean Paul. Robot motion planning and control. Centre National de la Recherche Scientifique. LAAS report 97438. Springer,1998; p.6.

LATOMBE, Jean. Robot motion planning. Boston: Kluwer academic Publishers. 1991; 587p.

LEE, T. H and LEUNG, F. H. F. A practical fuzzy logic controller for the path tracking of wheeled mobile robots. En: IEEE. 27th annual conf. of industrial electronic society 2001. p.574-576.

LÓPEZ GARCÍA, Javier y PINZÓN DÍAZ, Ricaurte Iván. La inteligencia artificial en el diseño mecánico: aplicación de los algoritmos genéticos en la optimización del diseño mecánico. Volumen 2. Bucaramanga. 1998. p.368-407. Trabajo de Grado (Ingeniero Mecánico). UIS. Facultad de Ingenierías Fisicomecánicas.

MEGHERBI, Dalila and WOLOVICH, William. Modeling and automatic real-time motion control of wheeled mobile robots among moving obstacles theory and applications. IEEE, 32nd conference on decision and control. San Antonio, Texas (December 1993); p.2676-383.

MICHALEWICZ, Zbigniew. Evolutionary computation: practical issues. En: IEEE. 1996; p.30-37.

NOBORIO, Hiroshi; YAMAMOTO, Iku and KOMAKI, Toshihiro. Sensor-based path-planning algorithm for a nonholonomic mobile robots. En: IEEE, conf. international intelligent robots and systems. 2000. p.917-924.

OLLERO BATURONE, Aníbal. Robótica. Manipuladores y robots móviles. Barcelona: Alfaomega, 2001; p.97-125.

OSTROWSKI, James P. Computing reduced equations for robotic systems with constraints and symmetries. En: IEEE transactions on robotics and automation. Vol.15, No.1. (Feb 1999); p.111-117.

PARK, Ming gyu; JEON, Jae and LEE, Min. Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing. En: IEEE. 2001; p.1530-1535.

SAMPEI, Mitsuji and ISHIKAWA, Masato. Control strategies for mechanical systems with various constraints. Control of non-holonomic system. En: IEEE. 1999. p.158.

SIERRA BUENO, Daniel. Planeación de trayectorias para un robot en una celda de manufactura. Bucaramanga. 2001. p.34-37. Trabajo de Grado (Ingeniero. Electricista- Electrónico). UIS. Facultad de Ingenierías Fisicomecánicas.

SHAN LIN, Hoi and MICHALEWICZ, Zbigniew. Evolutionary algorithm for path planning in mobile robots environment. En: IEEE. 1994. p.211-214.

SHARIR, Micha. Algorithmic motion planning. En: CRC press LCC, 1997. p.734-749.

SHILLER, Zvi and GWO, Yu. Dynamic motion planning of autonomous vehicles. En: IEEE, robotic and automation. Vol.7 No.2. (April 1991); p.241-249.

SIK SMI, Jun and HWAN KIM, Jong. Robust control of Non-holonomic wheeled mobile robot based on evolutionary programming for optimal motion. En: IEEE, Conf. On Industrial Technology, China. 199;. p.245-249.

SUGIHARA, Kazuo and SMITH, Jhon. GA Toolkit on the web. University of Hawaii at Manoa. p.1-7.

_____, Genetic algorithms for adaptative motion planning of an autonomous mobile robot. En: IEEE. 1997. p.138-142.

_____, Genetic algorithms for adaptative planning of path and trajectory of a mobile robot in 2D terrains. ICS. (May 26, 1997); p.1-14.

VÉLEZ, Oswaldo. Aplicación de los Algoritmos Genéticos en Problemas de Optimización Combinatoria. Bucaramanga. 1999. Trabajo de Grado (Ingeniero de Sistemas). UIS. Facultad Ingenierías Fisicomecánicas

WADA, Masayoshi and MORI, Shunji. Modeling and control of a new type of omnidirectional holonomic vehicle. En: IEEE. 1996. p. 265-270.

WATANABE, Keigo. Control of an omnidirectional mobile robot. En IEEE. 2th conf. international Wknowledge based intelligent electronic systems. 21-23 april. 1998. P. 51-60.

XIAO, Jing and MICHALEWICZ, Zbigniew. Adaptative evolutionary planner/navigator for mobile robots. En: EEE, evolutionary computation. Vol1. No.1. (April, 1997); p.18-27.

ZALZALA, A. M. and FLEMING, P. Genetic algorithms in engineering systems. London: Bookcraft. 1997; 254p.

ZHAO, Yilin and BeMENT, Spencer. Kinematics, dynamics and control of wheeled mobile robots. En: IEEE, International Conference robotics and automation. Nice, France. (May. 1992); p. 91-96.

ANEXO A

DETALLES DE LA IMPLEMENTACIÓN DEL AG PARA MAXIMIZAR LA FUNCIÓN DE ACKLEY.

Se especificarán aquí los pasos correspondientes a la obtención del máximo de la función de Ackley mediante el AG simple; se espera que esto sirva como referencia para entender un poco los pasos que deben seguirse a la hora de hacer uso del AG implementado en el presente trabajo. Se muestran algunos resultados con el fin de comprobar el proceso.

a. CODIFICACIÓN. Este proceso se realiza mediante la escogencia de una estrategia para representar la posibles soluciones. Para el presente caso se decidió tomar 20 bits: 19 para la magnitud y uno para el signo. Esta escogencia es algo arbitraria, pero con ella se ilustra claramente el proceso a seguir. El otro aspecto es que como el intervalo de trabajo será la región comprendida entre $x_1 \in [-3, 3]$ y $x_2 \in [-3, 3]$, por lo que con 4 bits en la parte entera, se puede claramente expresar los valores enteros que pertenecen a este intervalo.

Para facilitar el proceso se hará una explicación para el caso que se tengan 6 individuos como población. Para generar la población inicial se implementó en Matlab el siguiente código:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:cantidadindividuos,
    for j=1:20;
        x=rand;
        if x<=0.5
            x=0;
        else
            x=1;
        end
        poblacion(i,j)=x;
        poblacion(i,j+1)=i;
    end
end
%Producción de individuos para la variable x
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Como se observa se hace uso de la función generar número aleatorio *rand()* con la cual, se producen los números aleatorios los cuales están uniformemente distribuidos en el intervalo [0.1]. Se utilizó el criterio que si es inferior a 0.5 se toma como “0” y “1” para el caso contrario. Con este paso se obtiene el genotipo.

	i	INDIVIDUOS (Genotipo)																SIGNIFICADO (Fenotipo)					
X	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0	1	0	0,878234863		
	2	0	1	1	1	0	0	0	1	0	1	0	0	1	1	0	0	1	0	0	14,16235352		
	3	1	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	1	1	0	-2,751373291		
	4	0	0	1	0	1	1	0	1	0	0	1	0	0	0	0	1	1	1	0	1	5,64151001	
	5	0	1	0	1	1	0	1	1	0	1	0	1	1	0	1	0	1	0	0	0	11,41918945	
	6	1	1	0	0	1	1	1	1	0	1	1	1	1	1	0	1	0	1	0	0	-9,936157227	
Y	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	-0,470214844	
	2	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	1	0	1	0	1	-12,42153931	
	3	1	1	1	0	0	0	0	1	1	1	1	1	1	1	0	1	0	1	1	1	0	-12,24749756
	4	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	1	0	0	1	1	0	-15,20233154
	5	0	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	1	1	1	1	15,92208862
	6	1	1	1	0	1	1	0	1	0	1	0	0	1	0	1	0	1	1	1	1	1	-13,66159058

Para el proceso de decodificación, lo que se conoce como fenotipo, se implementó el siguiente código:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x=0;
z=0;
for j=1:cantidadindividuos,
    l=3;
    for k=2:5,
        x=poblacion(j,k)*2^(l)+x;
        l=l-1;
    end
    for k=6:20,
        z=poblacion(j,k)*2^(-k+5)+z;
    end
    x=x+z;
    y=poblacion(j,1);
    if y==1
        x=x*-1;
    end
    poblacion1(j,1)=x;
    poblacion1(j,2)=poblacion(j,k+1);
    x=0;
    z=0;
end
x=0;
z=0;
for j=1:cantidadindividuos,
    l=3;
    for k=2:5,
        x=poblaciony(j,k)*2^(l)+x;
        l=l-1;
    end
    for k=6:20,
        z=poblaciony(j,k)*2^(-k+5)+z;

```

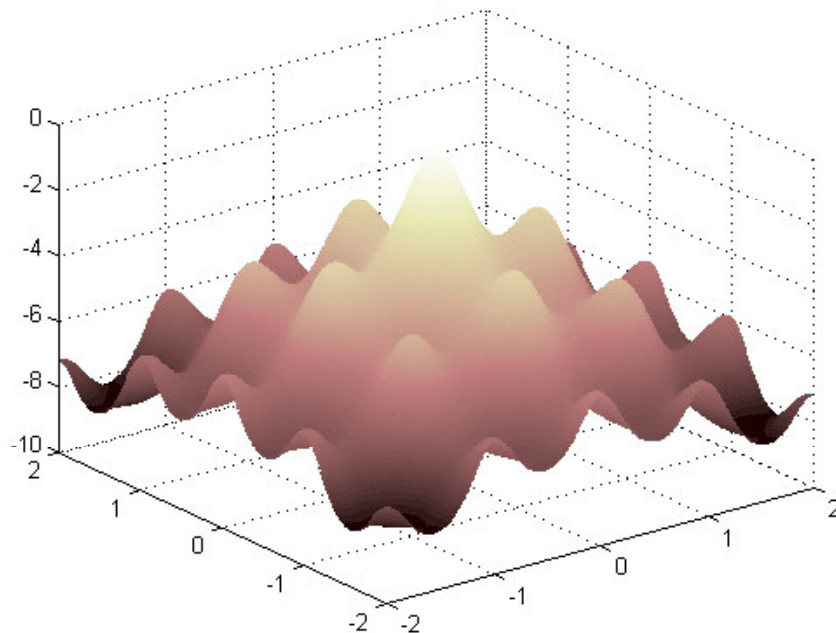
```

end
x=x+z;
y=poblaciony(j,1);
if y==1
    x=x*-1;
end
poblaciony(j,1)=x;
poblaciony(j,2)=poblaciony(j,k+1);
x=0;
z=0;
end
x1=poblacion1(:,1);
x2=poblacion1(:,1);
%Decodificación de los individuos para las variables x y y.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

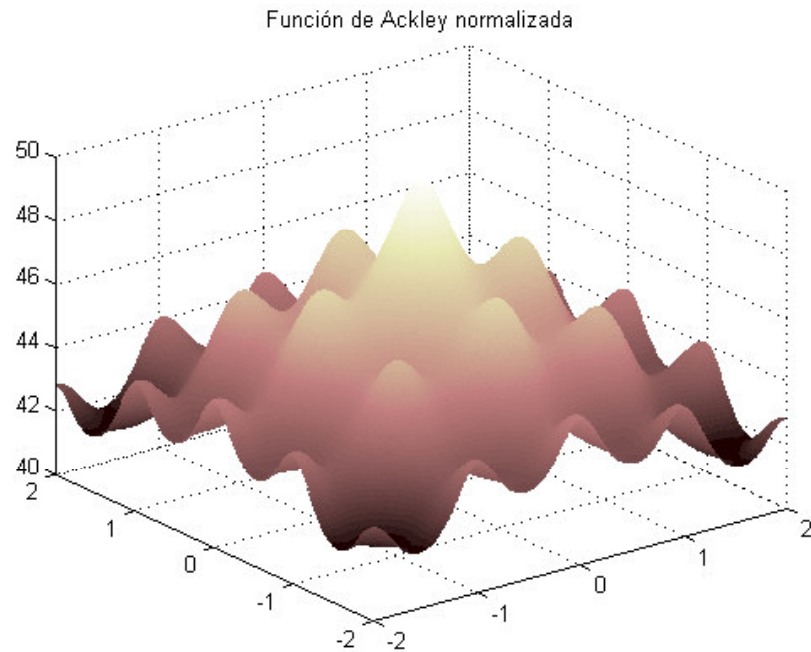
```

b. FUNCIÓN DE EVALUACIÓN: Para el presente caso, obtener la función de evaluación es muy sencillo. Basta con normalizar la función a maximizar, en este caso la función de Ackley, la cual como se observa en la siguiente figura es negativa para toda la región de trabajo, por lo que se hace necesario sumarle una constante que haga que sea siempre positiva. Para el caso se escogió 50.

Función de Ackley sin normalizar



$$y = e^{(0.5 \cdot \cos(2\pi x_1) + 0.5 \cdot \cos(2\pi x_2))} + 20 \cdot e^{(-0.5 \sqrt{(0.1x_1^2 + 0.1x_2^2)})} - 20 - e^1$$



$$y = 50 + e^{(0.5 \cdot \cos(2\pi x_1) + 0.5 \cdot \cos(2\pi x_2))} + 20 \cdot e^{(-0.5 \sqrt{(0.1x_1^2 + 0.1x_2^2)})} - 20 - e^1$$

i	FUNCIÓN DE EVALUACIÓN	PROBABILIDAD DE SER SELECCIONADO	# INDIVIDUOS ESPERADOS
1	45,24461859	0,233025922	1,398155533
2	29,1353195	0,150057287	0,900343719
3	31,04223373	0,159878575	0,959271449
4	29,66760114	0,152798727	0,916792361
5	29,188847	0,150332972	0,901997833
6	29,8826912	0,153906517	0,923439104

Total	194,1613112
Media	32,36021853
Máximo	45,24461859

d. SELECCIÓN: Este proceso se realizó mediante la ruleta pesada, la cual consiste en simular el proceso de una ruleta, la cual se distribuye proporcionalmente según sea la “bondad” de cada uno de los individuos, para luego hacer girar la rueda(generar un número aleatorio) y donde esta se detenga corresponderá al individuo seleccionado. El código empleado para esta labor se muestra a continuación:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:cantidadindividuos,
    sumafitness=evaluacion(i)+sumafitness;
end
pesoelemento=ones(cantidadindividuos,1);
elementopesosuma=ones(cantidadindividuos,1);
nuevoelementol=ones(cantidadindividuos,2);
nuevoelementoly=ones(cantidadindividuos,2);
for i=1:cantidadindividuos,
    pepo=evaluacion(i,1);
    pesoelemento(i,1)=pepo/sumafitness;
    if i==1
        elementopesosuma(i)=pesoelemento(i);
    else
        elementopesosuma(i)=pesoelemento(i)+elementopesosuma(i-1);
    end
end
for i=1:cantidadindividuos,
    y=rand;
    contador=0;
    for j=1:cantidadindividuos,
        if(elementopesosuma(j)>=y) break;
    end
    end
    nuevoelementol(i,1)=poblacionl(j,1);
    nuevoelementol(i,2)=poblacionl(j,2);
    nuevoelementoly(i,1)=poblacionly(j,1);
    nuevoelementoly(i,2)=poblacionly(j,2);
end
%Fase de reproducción(selección)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

	i	INDIVIDUOS SELECCIONADOS																SIGNIFICADO				
X	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	1	0	0,878234863		
	4	0	0	1	0	1	1	0	1	0	0	1	0	0	0	0	1	1	0	1	5,64151001	
	6	1	1	0	0	1	1	1	1	0	1	1	1	1	1	0	1	0	1	0	-9,936157227	
	4	0	0	1	0	1	1	0	1	0	0	1	0	0	0	0	1	1	1	0	1	5,64151001
	5	0	1	0	1	1	0	1	1	0	1	0	1	1	0	1	0	1	0	0	0	11,41918945
	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	1	0	1	0	0,878234863
Y	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	-0,470214844
	4	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	-15,20233154
	6	1	1	1	0	1	1	0	1	0	1	0	0	1	0	1	0	1	1	1	1	-13,66159058
	4	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	-15,20233154
	5	0	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	1	1	1	15,92208862
	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	-0,470214844

e. **CRUCE:** Este proceso se realizó mediante la técnica del cruce en un solo punto. La metodología consiste en tomar el primer par de posibles padres y generar un número aleatorio el cual si es menor que la tasa de cruce, el intercambio de información se hace efectivo generando dos hijos. En caso contrario se mantienen los padres para la siguiente generación. Para determinar el sitio de cruce se genera aleatoriamente un número entre 1 y la longitud de la cadena. Para el presente ejemplo los tres cruces se hicieron efectivos con las posiciones que allí mismo se mencionan. El código empleado para esta labor se muestra a continuación.

POSIBLES	16
POSICIONES	13
DE CRUCE	4

		INDIVIDUOS DESPUÉS DEL CRUCE																SIGNIFICADO
X	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0,878326416
	3	0	0	1	0	1	1	0	1	0	0	1	0	0	0	1	1	5,641418457
	5	0	0	1	0	1	1	0	1	0	0	1	0	0	1	0	1	5,643188477
	4	1	1	0	0	1	1	1	1	0	1	1	1	1	0	0	1	-9,93447876
	5	1	1	0	0	1	1	1	1	0	1	1	1	1	0	0	1	-9,93447876
	6	1	1	0	0	1	1	1	1	0	1	1	1	1	1	0	1	-9,936157227
Y	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	-0,470397949
	3	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	-15,20214844
	5	1	1	1	1	1	0	0	1	1	0	0	1	1	0	1	1	-15,20065308
	4	1	1	1	0	1	1	0	1	0	1	0	0	1	1	1	0	-13,66326904
	5	1	1	1	0	1	1	0	1	0	1	0	0	1	1	1	0	-13,66326904
	6	1	1	1	0	1	1	0	1	0	1	0	0	1	0	1	1	-13,66159058

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
papa1l=0;
papa2l=0;
papa1y=0;
papa2y=0;
papa1=0;
papa2=0;
hijo1l=ones(1,20);
hijo2l=ones(1,20);
papa1y=0;
papa2y=0;
hijo1y=ones(1,20);
hijo2y=ones(1,20);
for i=1:cantidadindividuos/2,
    sitiocruce(i)=ceil(rand*20);
end
for i=1:cantidadindividuos/2,

```

```

y=rand;
if (y<=ratacruce)
    papa1=nuevoelemento1(i,2);
    papa2=nuevoelemento1(i+1,2);
    papa11=poblacion(papa1,1:20);
    papa22=poblacion(papa2,1:20);
    hijo11=[papa11(1:sitiocruce(i)) papa22(sitiocruce(i)+1:20)];
    hijo22=[papa22(1:sitiocruce(i)) papa11(sitiocruce(i)+1:20)];
    papaly=nuevoelementoly(i,2);
    papa2y=nuevoelementoly(i+1,2);
    papa11y=poblaciony(papaly,1:20);
    papa22y=poblaciony(papa2y,1:20);
    hijo11y=[papa11y(1:sitiocruce(i)) papa22y(sitiocruce(i)+1:20)];
    hijo22y=[papa22y(1:sitiocruce(i)) papa11y(sitiocruce(i)+1:20)];
    if i==1
        poblacion(i,1:20)=hijo11;
        poblacion(i,21)=i;
        poblacion(i+1,1:20)=hijo22;
        poblacion(i+1,21)=i+1;
        poblaciony(i,1:20)=hijo11y;
        poblaciony(i,21)=i;
        poblaciony(i+1,1:20)=hijo22y;
        poblaciony(i+1,21)=i+1;
    else
        poblacion(2*i-1,1:20)=hijo11;
        poblacion(i,21)=2*i-1;
        poblacion(2*i,1:20)=hijo22;
        poblacion(2*i,21)=2*i;
        poblaciony(2*i-1,1:20)=hijo11y;
        poblaciony(i,21)=2*i-1;
        poblaciony(2*i,1:20)=hijo22y;
        poblaciony(2*i,21)=2*i;
    end
else
    if i==1
        poblacion(i,1:20)=papa11;
        poblacion(i,21)=i;
        poblacion(i+1,1:20)=papa22;
        poblacion(i+1,21)=i+1;
        poblaciony(i,1:20)=papa11y;
        poblaciony(i,21)=i;
        poblaciony(i+1,1:20)=papa22y;
        poblaciony(i+1,21)=i+1;
    else
        poblacion(2*i-1,1:20)=papa11;
        poblacion(2*i-1,21)=2*i-1;
        poblacion(2*i,1:20)=papa22;
        poblacion(2*i,21)=2*i;
        poblaciony(2*i-1,1:20)=papa11y;
        poblaciony(2*i-1,21)=2*i-1;
        poblaciony(2*i,1:20)=papa22y;
        poblaciony(2*i,21)=2*i;
    end
end
end

```

```

end
%Fase de cruce
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

f. MUTACIÓN: Este proceso se realizó revisando toda la cadena y generando un número aleatorio, el cual si es menor que la tasa de mutación ésta se hace efectiva, invirtiendo el bit respectivo. El código empleado para esta labor se muestra a continuación:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:cantidadindividuos,
    for j=1:20,
        y=rand;
        if(y<=ratamuta)
            poblacion(i,j)=1-poblacion(i,j);
            poblaciony(i,j)=1-poblaciony(i,j);
        end
    end
end
%Fase de mutación
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```