

# Optimizacin por Enjambre de Partculas Para Planificacin Global de Rutas basado en Interpolacin Polinomial

Arnaldo Gaspar and Joe Cabezas

*Departamento de Informtica*

*Universidad Tcnica Federico Santa Marfa*

*Santiago, Chile*

*agaspar@alumnos.inf.utfsm.cl and jcabezas@alumnos.inf.utfsm.cl*

## I. DESCRIPCIÓN DEL PROBLEMA

El principal objetivo de Motion Planning es encontrar una o más rutas para que un agente en particular logre llegar desde una región inicial ( $p_{\text{inicial}}$ ) a otra región objetivo ( $p_{\text{objetivo}}$ ) bajo restricciones como obstáculos, condiciones de movimiento, balance y cualquier indeseable presente en la trayectoria.

En el contexto de investigación de la Inteligencia Artificial (IA), Motion Planning corresponde en su expresión más frecuente a un problema de optimización que puede incluir criterios como distancia a recorrer, tiempo utilizado durante el trayecto, energía consumida conforme a las condiciones del entorno, etc. Sus aplicaciones se pueden efectuar en la robótica, la manufactura, el ensamblado, el transporte y servicios entre otros.

El método general es la resolución para un entorno estático, y posteriormente crear un plan de movimiento para un entorno dinámico. Un entorno estático es aquel en el que no cambia la configuración durante el transcurso del tiempo.

Para el caso estático, la región inicial, la región de llegada, las regiones a evitar y las regiones a visitar son conocidas antes que el plan sea construido y se asume que tales condiciones no cambiarán.

El planemantien en entornos estáticos recibe el nombre de *Global Path Planning* (GPP).

Para el caso dinámico, algunas de las regiones a evitar no se conocen y su planificación no puede ser realizada de antemano. Este problema recibe el nombre de *Dinamyc Path Planning* (DPP).

Aunque ambos casos en general son similares, el contexto en el que se desenvuelven es diferente y por ende sus soluciones también.

En *Global Path Planning* (GPP) no es muy importante minimizar el tiempo requerido para formular el plan de movimiento porque es realizado a priori, en cambio para *Dinamyc Path Planning* el tiempo es crítico pues el agente puede agotar sus recursos durante la exploración del entorno.

En un entorno estático, el problema de Global Path Planning puede ser expresado como un problema de encontrar al menos un camino libre de colisiones de tamaño mínimo.

El entorno puede ser descrito como un mapa, donde un camino entre dos posiciones es aproximado como una secuencia de posiciones adyacentes.

Una instancia del problema está dada por un mapa de  $m \times n$ , con sectores de tamaño fijo marcados como obstáculos.

La representación más sencilla es la de una malla cuadrículada con celdas de tamaño fijo y constituye el dominio de cada variable de decisión.

El principal objetivo del GPP es encontrar rutas factibles que se encuentren en las zonas libres de obstáculos, por lo tanto la representación del entorno determina la estructura de la ruta.

La forma más simple de representar rutas dada una instancia descrita por un mapa en forma matriz es a través de una secuencia de posiciones adyacentes que vayan desde el punto inicial hasta el objetivo.

Cada ruta generada constituye una variable de decisión y el dominio corresponde a los puntos o áreas del mapa cualesquiera para constituir la ruta. Esta forma considera la generación de soluciones iniciales infactibles las cuales van mejorando durante el transcurso de la ejecución del algoritmo que implementa una metaheurística para este caso.

La principal restricción es que la ruta generada esté confinada a los márgenes de la instancia. Esta restricción puede parecer un tanto burda, sin embargo, al utilizar técnicas de interpolación a tramos (Splines) es muy probable que las rutas generadas sobrepasen el cuadro del mapa al no satisfacer la restricción de ajuste al polígono de espacio mínimo (convex hull). Para tales razones también se considera interpolación y aproximación por tramos que cumplan dicha propiedad.

Desde la perspectiva de la optimización, se consideran las colisiones como restricciones que se penalizan en la función objetivo dado que su consideración dura podría no generar soluciones iniciales de acuerdo a las diferentes instancias que puedan ser estudiadas.

También existen restricciones desde la perspectiva del contexto en el que se encuentre; por ejemplo en la robótica se consideran restricciones como la energía utilizada, la inercia, etc.

Para efectos prácticos e investigativos de la técnica utiliza-

da en este artículo se consideran solamente las restricciones más generales del GPP y en especial, para esta investigación, no se consideran las regiones a visitar.

Se espera encontrar rutas las cuales no pasen por obstáculos y que tengan la menor longitud posible. Otra característica de la solución a evaluar puede ser la suavidad de la ruta generada. Muchas representaciones consideran la inicialización de las partículas o soluciones iniciales como rutas generadas pseudo aleatoriamente de forma constructiva (agregación de puntos consecutivamente), lo cual no garantiza suavidad aunque existan intenciones de suavización post generación de rutas factibles. Otra forma de abarcar el problema es considerando la interpolación paramétrica entre pares de puntos garantizando suavidad de movimiento para el agente.

En términos generales el algoritmo debe dar resolución a problemas que puedan tender a converger prematuramente, por esa razón, además de penalizar la cantidad de colisiones y el largo de la ruta, se considera el hecho de que existan rutas muy cortas de solución las cuales no representen una solución adecuada al problema, situación que también sufre castigos reflejados en la función objetivo.

La representación del problema planteado en forma matemática se puede expresar para cualquier dimensión según el espacio de operación, sin embargo en el presente estudio se considera la resolución en el plano, siendo  $C \in \mathbb{R}^2$  el espacio de operación, donde el conjunto de  $O$  de  $|O| = m$  obstáculos  $o_i$  se define como

$$O = \{o_i : o_i \in C, \forall i = 1 : m\} \quad (1)$$

Donde se define a  $a$  como la región inicial en  $C$  y  $b$  como la región objetivo. El espacio determinado por la aplicación  $R : \mathbb{R} \rightarrow \mathbb{R}^n$  donde  $q \rightarrow R(q)$  corresponde a la región ocupada por el agente en el punto  $q$  en  $C$ . Además, el espacio ocupado por  $\hat{o}_i$  será el considerado para penalizar en la función objetivo, el cual representa a las colisiones que vaya teniendo el agente durante el transcurso de la ruta  $r \subseteq C$ , donde

$$\hat{o}_i = \{q \in r : R(q) \cap o_i \neq \emptyset, \forall i = 1 : m\} \quad (2)$$

Por lo tanto, el espacio libre corresponde al conjunto

$$C_{\text{libre}} = C - \bigcup_{\forall i} \hat{o}_i \quad (3)$$

Donde una trayectoria válida sería aquella ruta  $r \subseteq C_{\text{libre}}$  de largo  $|r| = l$  tal que

$$r = \{q \in C_{\text{libre}} : q_0, \dots, q_\ell \text{ tal que } q_0 = a, q_\ell = b\} \quad (4)$$

Con el fin de hacer un estudio en una escala aplicable, se considera un modelo más simple, cuyas consideraciones son las de un espacio de operación en  $\mathbb{N}^2$  como una matriz  $C \in \text{Mat}(m \times n, \{0, 1\})$  que describe el espacio de operación

según las coordenadas. Si en una cierta coordenada  $i, j$  existe un obstáculo, entonces  $C_{ij} = 1$ , de lo contrario  $C_{ij} = 0$ . Además, se considera la aplicación  $R(q) = q$ , donde  $q$  representa un elemento en la matriz descrita. Respecto a las restricciones, ciertas consideraciones físicas no serán parte de este problema, no existe aceleración, velocidad, ni fuerzas que prohíban el movimiento libre por el espacio de operación. La ruta  $r$  está representada por un vector de largo variable  $\ell$  que almacena la lista de casillas visitadas sin repetición donde  $q_0$  es la región inicial,  $q_\ell$  la región final y  $q_i$  la región adyacente a  $q_{i+1}$  representadas por celdas de la cuadrícula o matriz.

El largo de la ruta  $r$  está dado por el valor  $\ell$ . La función objetivo por lo tanto queda expresada como

$$\text{Min} : F(\ell) = \ell + Pn(\ell) \quad (5)$$

$\ell$  es el largo de la ruta, que en este caso, como la distancia entre lugares es constante entre lugares adyacentes y vale  $d_{\min} = 1$ , esta función es equivalente a la cantidad de elementos del vector  $p$ .

$Pn(p)$  es una función de penalización que vale 0 si es una ruta sin colisiones, es decir, no pasa por ningún lugar marcado como obstáculo, en caso contrario, esta función se define:

$$Pn(\ell) = NC(1 + \ell^\alpha) \quad (6)$$

$NC$  = número de colisiones

El razonamiento detrás de esta función de penalización garantiza que se penalice por cada casilla marcada como obstáculo que intenta pasar, contándola dos veces, una vez al contar el largo del vector, y una segunda vez si la casilla es un obstáculo. Además penaliza de manera proporcional a la cantidad de colisiones y de manera potencial al largo de la ruta, por lo que si  $\alpha > 0$  se puede penalizar de manera más fuerte a las rutas cortas, debido a que la ruta más corta entre el punto de inicio  $a = C_{i,j}$  y el punto de meta  $b = C_{k,\ell}$  mide:  $\ell = |i - k| + |j - \ell|$ , la cual es una ruta directa la cual probablemente pase por más de algún obstáculo y al ser muy corta, puede quedar como candidata en vez de privilegiar a aquellas que son más largas pero que no tienen colisiones.

## II. ESTADO DEL ARTE

El problema clásico en el Path Planning se describe como:

dado un cuerpo rígido tridimensional y un conocido conjunto de obstáculos, la tarea de encontrar una ruta libre de colisiones desde una configuración inicial a una objetivo. Adicionalmente, esta tarea debe ser completada en una cantidad razonable de tiempo.

Este problema es conocido como el problema del movedor de pianos [?].

Existen paradigmas clásicos para resolver Motion Planning como lo son los métodos de campo potencial (potential field methods), métodos de mapas de ruta como los métodos de grafos de visibilidad y probabilísticos y los métodos de grillas [?] y [?] .

Los métodos de campo potencial primeramente descritos en [?], son métodos usados para evitar colisiones en base a sensores de proximidad. La idea es caracterizar al robot y los obstáculos como cargas positivas y al objetivo como carga negativa. La navegación ocurre a medida de las interacciones de repulsión entre el agente y los obstáculos y la atracción del agente con el objetivo. La interacción total es  $U(q) = U_{attr}(q) + U_{repul}(q)$  y cada paso es tomado sobre el gradiente negativo del potencial. El robot termina el movimiento cuando descubre que el gradiente es cero.

Los métodos basados en mapas de rutas son aquellos que consideran como mapa a una estructura de datos usada para planear rutas subsecuentes más rápidamente. La estructura de datos intenta capturar la conectividad y características de la configuración del espacio del agente. Usando el mapa, un planeador (planner) puede encontrar rutas entre dos configuraciones donde lo primero es encontrar una ruta libre de colisiones desde una de las configuraciones al mapa de rutas y así desde el mapa de rutas a la configuración de destino.

Dentro de los métodos basados en mapas de rutas están los mapas de visibilidad, primeramente estudiados por NJ Nilsson en 1969, los cuales consideran los nodos de los mapas como vértices de un polígono y dos nodos del grafo de visibilidad comparten un vértice si sus vértices correspondientes son vistos por el otro. La línea de visión puede estar dada por un sensor como la lectura de un sonar.

También, dentro de los métodos basados en mapas, están los mapas de rutas probabilísticos, que se suelen ocupar cuando el número de grados de libertad del agente son moderadamente grandes y las aproximaciones para la planificación exacta se vuelven muy difíciles de calcular o imposible. Para esta clase de problemas se hace necesario recurrir a la utilización de heurísticas para encontrar soluciones las que básicamente consisten en chequear si la configuración del robot  $q$  está dentro del área libre de obstáculos.

El modelo común en la investigación de Motion Planning ha sido desacoplar los problemas generales de robótica relacionados con sensores, etc. resolviendo primero Global Path Planning y luego encontrar una trayectoria que pueda satisfacer las restricciones dinámicas y seguir ese camino [?]

También se sugiere que si se está preocupado por la inercia del robot, causando con ello un camino cinemático planificado, se puede también simplemente ejecutar el camino muy lentamente, para minimizar los efectos de la dinámica. Aunque estas suposiciones son razonables en ciertas ocasiones, no pueden siempre ser justificadas. Incluso puede darse el caso de que el camino conseguido sea completamente no factible debido al entorno del objeto y

sus límites de fuerzas y torques [?]

Path Planning Problem puede ser dividido en dos grandes categorías. La primera es crear un camino planeado a seguir dentro de un entorno estático. La segunda es crear un plan para hacer un recorrido en un entorno dinámico [?].

En un primer caso, se tiene, un punto de comienzo, un punto de meta (final o de llegada) y un set de regiones a evitar, todos conocidos incluso antes de que comience a crearse un plan para crear el recorrido, mas aún, se asume que estas condiciones no cambiarán, con ello, se refiere a este caso como Global Path Planning [?].

En el segundo caso, parte de las regiones a evitar, o en su totalidad, son desconocidas, por ejemplo, situaciones de peligro solo se conocen cuando se está en ese sector y dentro del rango de los sensores, esto modela de manera más fiel los casos de la realidad, con ello, se hará referencia a este caso como Dynamic Path Planning

Ambas aproximaciones pueden parecer similares, pero de hecho, son muy diferentes. Por ejemplo en Global Path Planning, no se tiene mayor preocupación sobre el tiempo que demora en formular una solución, pues se puede determinar antes de que se comience a navegar, mientras que en el caso dinámico la consideración del tiempo es más crítica, pues en cada momento se está actualizando el plan a seguir, según cambien las condiciones. De aquí se hace una distinción entre una formulación en tiempo real y otra no en tiempo real [?]

Otra gran diferencia entre ambos esquemas es la capacidad de decisión, mientras que en el caso global es posible analizar si un camino es seguro, en el caso dinámico puede darse el caso en el que no se puede concluir si un camino formulado es seguro o no. [?]

Global Path Planning ha sido resuelto para dominios de robot y aeronaves a través de una técnica llamada: "Descomposición de Collin" (*Collin's Decomposition*) [?], el cual consiste en dividir el entorno en muchas regiones más pequeñas y formulando un camino seguro e individual para cada uno de ellos, luego cuando las regiones estén completadas, se agrupan nuevamente estos sub-caminos para formar una solución global, y en aquellos casos en que se requiere pasar por diferentes regiones, esta técnica puede ser realizada varias veces.[?]

Ciertamente cuando se requiere evitar zonas peligrosas, se pueden usar zonas potenciales la que consiste en asignar valores numéricos que permitan medir el nivel de seguridad, a aquellas regiones peligrosas y establecer círculos concéntricos a ellos con valores intermedios, de esta forma, los robots pueden evitarlas y tomar decisiones como esquivarlas o regresar por donde vinieron. En especial este último recurso no es una opción para muchos problemas de Dynamic Path Planning como por ejemplo aeronaves que no tienen otra opción, por limitaciones físicas, de regresar rápidamente o "dar la vuelta".

Para solucionar ello, Goldman propone una solución llamada *Subgoal Avoidance*, el que consiste en establecer submetas previas antes de dirigirse a la meta global, y con ellas evitar zonas riesgosas. Se representan las zonas de peligro como un espacio de dos dimensiones y se traza una línea perpendicular entre el camino original y el punto de riesgo y se mide la distancia  $d$  entre la recta del camino original y el punto de riesgo, luego se crea una circunferencia de radio  $2d$ , la intersección entre la circunferencia y el bisector perpendicular es un nuevo punto que será una submeta por la cual se debe pasar para evitar el peligro.[?]

Desde la perspectiva de las heurísticas de resolución también existen aproximaciones utilizando simulated annealing, algoritmos genéticos, algoritmos basados en colonias de hormigas, en enjambre de partículas y en sistemas inmunes y combinaciones entre ellos.

Se proponen también estrategias de planificación de rutas como el *simulated annealing dissipative ant system* [?] cuyo objetivo, en contraste con los sistemas de hormigas tradicionales, es utilizar simulated annealing para aumentar la capacidad global de búsqueda (exploración) del algoritmo donde solo las mejores rutas en cada iteración son actualizadas para incrementar la rapidez de aprendizaje del método. A esto se le introduce la entropía en el sistema para salir de los mínimos locales.

Dentro de los algoritmos genéticos hay investigaciones recientes utilizando algoritmos genéticos elitistas aplicados al Global Path Planning [?] los cuales consisten básicamente en dos algoritmos genéticos elitistas paralelos para mantener una mayor diversidad y así inhibir una convergencia prematura en comparación con los algoritmos genéticos convencionales. La forma de generar soluciones iniciales factibles que sean suaves son a través de técnicas de interpolación como la B-Spline cúbica [?] las cuales son continuas.

Dentro de los algoritmos basados en enjambre de partículas se proponen algoritmos especializados que garantizan convergencia para la optimización [?]. La forma de representar el problema es configurar un mapa el cual conecta el nodo inicial con el nodo final donde cada nodo en la trayectoria es codificado como una partícula. Entonces, una región particular “activa” para partículas es mapeada de acuerdo a la ubicación de los obstáculos. La población inicial de partículas se genera dentro de la región activa para buscar la ruta óptima. En el proceso de búsqueda, los coeficientes de aceleración y la inercia del algoritmo de optimización basado en partículas es auto adaptativo ajustado y las partículas invalidas son remplazadas por un óptimo local o global en áreas adyacentes.

El método de malla para resolver Global Path Planning muy usado en robots móviles. Para este método se han implementado algoritmos como los basados en las direcciones principales para mejorar el desempeño de esta aproximación reduciendo el número de nodos en la búsqueda [?].

Los algoritmos basados en colonias de hormigas son muy utilizados para resolver Global Path Planning, donde hay soluciones muy diversas como algoritmos de cobertura completa los cuales integran coberturas de sub áreas locales con Global Path Planning [?] que utilizan descomposición celular donde los agentes cubren una sub área a través de un movimiento de vaivén donde la distancia entre cada sub área es refinada, la cual incluye conectividad, distancia restante y número de obstáculos entre sub áreas. A partir de aquello se calcula una matriz de distancias entre sub áreas del entorno la cual es utilizada para Global Path Planning, de esta manera se utilizan los algoritmos basados en colonias de hormigas con dicha matriz para obtener la secuencia de optimización de sub áreas. También se proponen mejoras para superar los defectos de la precocidad y el tiempo requerido para la construcción de la población inicial en el algoritmo tradicional al aumentar la colonia de hormigas para Global Path Planning. En [?] se propone un algoritmo mejorado para aumentar la colonia de hormigas donde las operaciones de cruce y mutación del algoritmo genético (GA) se utilizan en aumentar la optimización de colonia de hormigas, y la función de probabilidad heurístico se agrega al proceso de la construcción inicial de la población.

Al respecto también existen soluciones basadas en algoritmos inmunes donde el algoritmo tiene capacidades de adaptación en base a la inmunidad del sistema inmunológico y para alcanzar el objeto de destino de manera segura y con cumplir con éxito con su tarea a través de camino óptimo. Los algoritmos inmunes tienen resultados similares a los algoritmos genéticos sin embargo tienen mejor desempeño cuando el entorno es más complejo [?] .

Considerando estas dos aproximaciones surgen modelos que consideran algoritmos basados en colonias de hormigas y redes inmunes [?] donde el mecanismo de estimulación y supresión entre antígeno y anticuerpo es usado para encontrar la ruta, la cual resuelve el modelamiento complejo de los sistemas basados en hormigas, lo cual mejora la eficiencia en encontrar la ruta óptima. El algoritmo basado en hormigas es usado para buscar en la red de anticuerpos, lo cual aumenta el efecto en la planificación de ruta. Los autores concluyen que estos algoritmos están caracterizados por tener una rápida convergencia y la planeación de rutas cortas, el cual resuelve planificación en ambientes complejos [?].

### III. DESCRIPCIÓN DEL ALGORITMO

En esta investigación, se usó el algoritmo de “Optimización por Enjambre de Partículas” (Particle Swarm Optimization)[?], el cual consiste en buscar una solución óptima a un problema dentro de un espacio de búsqueda grande, usando un enjambre de partículas, las que individualmente representan una solución distinta según en la posición en que se encuentran.

Éstas partículas tienen la capacidad de moverse por este espacio de búsqueda, y hacerlo con una cierta velocidad que pueden variar según su entorno. Además estas partículas tienen una memoria que les permite saber siempre cual ha sido su mejor ubicación dentro del espacio de búsqueda.

Luego, las partículas se comunican entre ellas para informar la calidad de su posición para conocer cuál tiene la mejor posición, con esta información las demás partículas deben decidir si moverse en dirección de la mejor partícula, dirigirse hacia su mejor posición individual según su memoria, o moverse aleatoriamente a otra posición.

La decisión o probabilidad de acercarse a la mejor de todas las partículas o no, depende de ciertos parámetros configurables del algoritmo los cuales, en este documento se presentan de la siguiente manera:

$\phi_p$  = nivel de importancia para moverse hacia la mejor posición individual de la partícula, según su memoria.  
 $\phi_g$  = nivel de importancia para moverse hacia la mejor partícula dentro del enjambre.  
 $\omega$  = nivel de importancia para moverse hacia una posición aleatoria.

### III-A. Representación del enjambre

El enjambre de partículas corresponde al conjunto de  $S$  partículas, donde partícula es una eventual solución y corresponde a una ruta a la que se le calcula una velocidad de movimiento respecto a su mejor posición histórica y respecto a la mejor posición global determinada la inicialización y posteriormente determinado durante la iteración. El objetivo del Global Path Planning PSO es que cada partícula tome movimientos relativos al mejor del grupo y a su propia información de mejor posición.

El enjambre además de la lista de partículas, tiene en memoria cuál ha sido la mejor partícula de todo el proceso de búsqueda, con ello, su posición (la solución) y su velocidad.

### III-B. Representación de la partícula y de las soluciones

Una partícula corresponde a una eventual solución del problema, es decir una partícula  $p$  es una ruta que describe el camino a seguir desde un nodo de inicio  $p_s$  al nodo de término  $p_g$  dentro de un espacio bidimensional  $C$  de tamaño  $m \times n$ . Así, las rutas se definen como:

$$p = \{p_0, p_1, \dots, p_n : p_i \in C, p_0 = p_s, p_n = p_g, n \leq mn\} \quad (7)$$

Luego, una ruta, debido a la naturaleza del problema tiene una longitud variable ya que distintas rutas pueden ser de distinta longitud.

Esto crea el problema de formular una representación óptima para PSO ya que se necesita introducir el concepto de velocidad, y con ello, el concepto de dos soluciones se puedan “acercar” para de cierta manera “parecerse” a la cual se está acercando.

Para explicar mejor el problema de la representación, se supone el siguiente ejemplo:

Se tienen 2 rutas del mismo largo  $n = 4$ , cada elemento de la partícula representa una casilla del mapa a recorrer, además ambas deben tener el mismo elemento al principio y al final ya que representan el inicio y el final del recorrido.

$$r_1 = \{1, 2, 4, 5\} \quad (8)$$

$$r_2 = \{1, 7, 8, 5\} \quad (9)$$

Si la partícula representada por la ruta  $r_2$  reporta un camino de mejor calidad que  $r_1$ , una forma para hacer que la partícula  $r_1$  se acerque para parecerse a  $r_2$  y con ello intentar mejorar su calidad, se puede modificar cada elemento de  $r_1$  para que tome el valor, como ejemplo:

$$r_{1i} = \left\lceil \frac{(r_{1i} + r_{2i})}{2} \right\rceil, \forall i = 1 : 4 \quad (10)$$

Es decir, el promedio entre los valores individuales (aproximado hacia arriba para mantener concordancia con el espacio discreto), con esto las rutas finalmente quedan como:

$$r'_1 = \{1, 5, 6, 5\} \quad (11)$$

$$r_2 = \{1, 7, 8, 5\} \quad (12)$$

Así, la partícula  $r_1$  está mas cerca de  $r_2$ .

El real problema nace debido a la naturaleza del problema, no se puede suponer que las rutas sean todas de la misma longitud, algunas serán más largas que otras y serán mejores ya que evitan mas obstáculos.

Para solucionar esta problemática se introduce el concepto de “puntos de apoyo” o “pivotes”, para poder redefinir el concepto de partícula.

Usando pivotes, se crean submetas por las cuales las rutas deben pasar obligatoriamente antes de seguir a la siguiente submeta (siguiente pivote), una vez visitados todos los pivotes, se debe buscar una ultima subruta que comience en el ultimo pivote y el punto final deseado.

Gracias a los pivotes la creación de rutas completas es simplificada a la búsqueda de rutas mas cortas y fáciles de resolver cuya característica más importante es que una partícula  $p$  puede ser definida como una secuencia de pivotes  $P_v$  de longitud constante  $n$ :

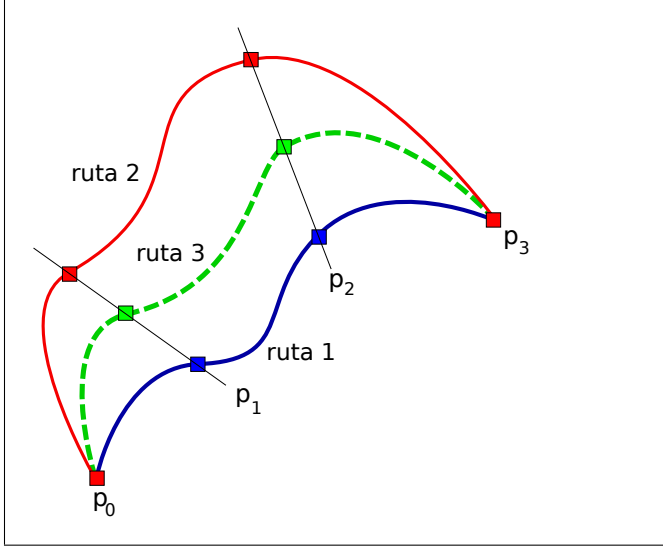


Figura 1: Acercamiento de partículas usando pivotes

$$p = \{p_0, p_1, \dots, p_n : p_i \in P_v, p_0 = p_s, p_n = p_g, n = \#P_v\} \quad (13)$$

Con esta representación, las partículas efectivamente pueden moverse dentro de una dimensión fija, y acercarse para poder imitar a otras partículas con mejores características.

Por ejemplo, una partícula  $p_1$  representando una ruta mas corta que una partícula  $p_2$ , la partícula  $p_2$  puede acercar sus pivotes a los de la partícula  $p_1$  para poder así, mejorar su calidad al parecerse a  $p_1$ , generando una partícula  $p_3$  representada como una ruta de largo intermedio entre  $p_1$  y  $p_2$ , como se muestra en la Figura ??.

### III-C. Generacion de rutas mediante pivotes

de longitud fija  $k$  de puntos  $q \in \mathbb{N}^2$  con su velocidad  $v$  de igual longitud  $k$  corresponde a la configuración  $s = (x, p, v)$  donde  $x$  es la posición instantanea,  $p$  la mejor posición histórica y  $v$  la velocidad de la partícula.

$$x = \{a, q_2, \dots, q_{k-1}, b\} \quad (14)$$

$$v = \{v_0, \dots, v_k\} \quad (15)$$

En este caso  $x$  y  $p$  obviamente tienen los mismos puntos al comienzo y al final dado que representan una solución candidata donde los puntos de origen y final representan a las regiones inicial y objetivo respectivamente.

La posición  $x$  de la partícula  $s \in S$  corresponde a un conjunto de puntos que interpolados o aproximados entre sí constituyen la ruta  $r$  mediante diferentes métodos de interpolación o aproximación. Para esta investigación se realizaron implementaciones de los métodos de interpolación paramétrica Cubic Hermité Splines y Cubic Bezier Splines, dado los puntos sugeridos por cada partícula.

El método general de interpolación corresponde a aproximar un conjunto de puntos (o pares de puntos) mediante un polinomio de grado no muy alto (para evitar grandes oscilaciones), donde en el caso de la resolución del Global Path Planning corresponde a utilizar la representación paramétrica

$$P(t) = at^3 + bt^2 + ct + d \quad (16)$$

donde  $P(t)$  corresponde a una función vectorial de la forma  $P(t) = x(t)\hat{i} + y(t)\hat{j}$ , tal que

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \quad (17)$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y \quad (18)$$

La forma general de describir la parametrización mediante Splines entre pares o conjuntos de puntos subsiguientes es de la forma

$$\begin{bmatrix} x(t) & y(t) \end{bmatrix} = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} M G \quad (19)$$

Donde  $M$  es una matriz de  $4 \times 4$  deducida del método en específico y  $G$  una matriz de  $4 \times 2$  con puntos y/o gradientes (según el método). El resultado de la ecuación anterior genera una matriz de  $2 \times 1$  donde cada columna representa a la parametrización de  $x$  e  $y$ . La forma más sencilla de hallar las matrices  $M$  y  $G$  son las de generar los polinomios dentro del intervalo  $t \in [0, 1]$  por razones prácticas de planteamiento.

Se realizaron experimentos con B Splines pero se descartaron porque no aproxima desde la región inicial ni a la región objetivo aunque pueden ser utilizadas otras técnicas para satisfacer tal condición pero introducen complejidad adicional para realizar comparaciones de los métodos.

La representación de la ruta está dada entonces por el conjunto de polinomios interpoladores o aproximadores sobre los pares sucesivos de los  $k$  puntos  $q$  definidos como posición en la partícula  $s$ . Las formas

**Cubic Bezier Splines:** Esta aproximación es utilizada para mantener la propiedad de cobertura de area mínima (convex hull) de los polinomios generados por cada par de puntos asegurando que las rutas generadas satisfagan la restricción de preservar la frontera de entorno de operación. Cada polinomio  $i$  generado entre los puntos  $i$  e  $i + 1$  se genera a partir de la expresión

$$\begin{bmatrix} x_i(t) & y_i(t) \end{bmatrix} = \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}^T \underbrace{\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{M_{\text{bezier}}} \underbrace{\begin{bmatrix} q_{i-1} \\ q_i \\ q_{i+1} \\ q_{i+2} \end{bmatrix}}_{G_{\text{bezier}}} \quad (20)$$

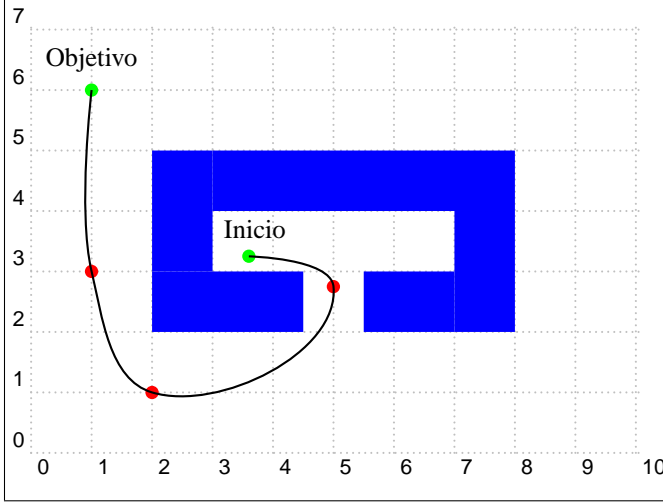


Figura 2: Ejemplo de resolución usando interpolación Hermite con tres pivotes (círculos en rojo)

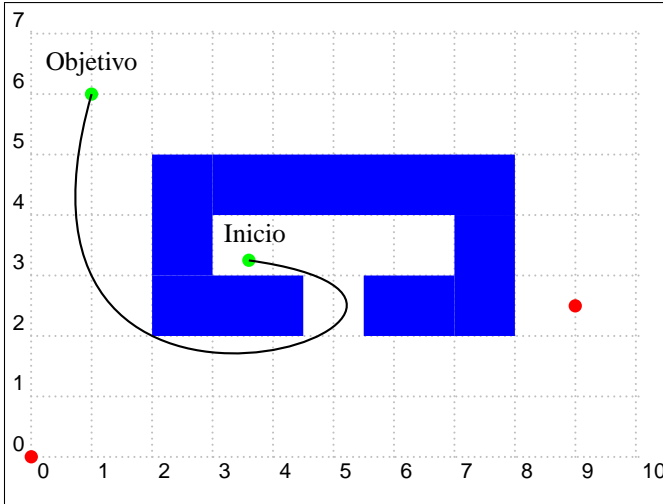


Figura 3: Ejemplo de resolución usando aproximación Bezier con dos pivotes (círculos en rojo)

*Cubic Hermite Splines:* Esta interpolación consta además de una variante en la representación de la partícula  $s$  la cual considera, además de los puntos en el entorno de operación, la gradiente de cada punto ( $\nabla q$ ) para determinar la forma de la ruta generada, por tanto  $x_h = (x, \nabla x)$ . Cada polinomio  $i$  generado entre los puntos  $i$  e  $i + 1$  se genera a partir de la expresión:

$$[x_i(t), y_i(t)] = \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}^T \underbrace{\begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{M_{\text{hermite}}} \underbrace{\begin{bmatrix} q_i \\ q_{i+1} \\ \nabla q_i \\ \nabla q_{i+1} \end{bmatrix}}_{G_{\text{hermite}}} \quad (21)$$

Todas las combinaciones de rutas posibles dentro del mapa representan el dominio de cada ruta ya que estas son

generadas a partir de puntos y gradientes descritas por la partícula.

La longitud de la ruta está dada por

$$\ell = \sum_{i=1}^{k-1} \int_0^1 \sqrt{[x'_i(t)]^2 + [y'_i(t)]^2} dt \quad (22)$$

Cada posición  $x$  y velocidad  $v$  es generada pseudo aleatoriamente dentro del encuadre del entorno de operación y son entendidas como pivotes o puntos de control (según el método utilizado).

La actualización de la posición corresponde a la generación de nuevas rutas que pasen por los pivotes o puntos de control modificados respecto a la información que se tenga en el enjambre. La función objetivo

$$\min F(\ell) = \ell + NC(1 + \ell^\alpha) \quad (23)$$

donde  $C$  es el número de colisiones y  $\alpha$  es el nivel de penalización por rutas cortas con colisiones.

Bajo esta representación todas las rutas tienen largos distintos pero las longitudes de las posiciones y velocidades permanecen constantes lo que significa gran comodidad para operar durante la ejecución del algoritmo

La actualización de la velocidad de la partícula  $s$  está dada por

$$v_{i+1} = \omega v_i + \rho_g \phi_g(g - x_i) + \rho_p \phi_p(p_i - x_i) \quad (24)$$

$$x_{i+1} = x_i + v_i \quad (25)$$

donde los parámetros mencionados son:  $\omega$  que pondera el efecto de la velocidad actual,  $\phi_g$  parámetro para favorecer la explotación,  $\phi_p$  parámetro para favorecer la exploración,  $\rho_g, \rho_p \sim U(0, 1)$ ,  $g$  la mejor posición global,  $p_i$  la mejor posición de la partícula  $i$ ,  $x_i$  posición de la partícula  $i$ .

Para realizar las operaciones se consideró una definición especial para restar, sumar y multiplicar posiciones y velocidades las que se realizan punto a punto, lo que produce incrementos o decrementos en los puntos dentro del entorno de operación para realizar la exploración y explotación necesaria para el algoritmo.

El algoritmo PSO consta de una fase de inicialización, luego una fase de iteración para terminar con la finalización Inicialización del enjambre

**Data:**  $S, \omega, \Phi_g, \Phi_p, g \leftarrow \text{MAX\_INT}$

**Result:** Mejor partícula  $s_i$ .

**for**  $i = 1$  **to**  $S$  **do**

$x_i =$  Inicializar posición aleatoriamente;

$p_i \leftarrow x_i$ ;

$v_i =$  Inicializar velocidad aleatoriamente;

**if**  $f(p_i) < f(g)$  **then**

$g \leftarrow p_i$

**end**

**end**

**Algorithm 1:** Inicialización PSO

Iteración del enjambre

**Data:**  $S, \omega, \Phi_g, \Phi_p, I$

**Result:** Mejor partícula  $g$ .

```

i = 0 while i < I do
  for i = 1 to S do
     $r_p, r_g \sim U(0, 1);$ 
     $v_i \leftarrow \omega v_i + \Phi_p r_p (p_i - x_i) + \Phi_g r_g (g - x_i);$ 
     $x_i \leftarrow x_i + v_i;$ 
    if  $f(r_i) < f(p_i)$  then
       $p_i \leftarrow x_i;$ 
      if  $f(p_i) < f(g)$  then
         $g \leftarrow p_i;$ 
      end
    end
  end
  i = i + 1;
end
return g ;

```

**Algorithm 2:** Iteración PSO

#### IV. EXPERIMENTOS

El objetivo principal de esta experimentación es el de evaluar la utilización de parámetros adecuados para cada una de las instancias a resolver de acuerdo a las técnicas de generación de las rutas descritas en la sección anterior. Se espera, una vez realizada la experimentación, obtener una perspectiva del efecto de los parámetros adicionales al problema, como lo son la cantidad de pivotes, el número de partículas, y la penalización de soluciones cortas infactibles, que son propias de este trabajo en particular y de la representación escogida, y además establecer hipótesis respecto al uso de estos como efectos en la función objetivo y en la resolución del problema. Los parámetros originales del PSO son ajustados respecto a criterios basados en la experimentación previa a la formulación de este experimento.

También es importante medir la calidad (en relación a la tendencia central) de las soluciones encontradas para posteriormente hacer un breve análisis de éstas o evaluar la convergencia según las instancias y sus formas de resolverlas.

La estrategia entonces es la de realizar experimentos asociados a diferentes combinaciones de parámetros y así observar el comportamiento de la función objetivo en función de estos.

Una vez analizado el comportamiento de los parámetros respecto a la función objetivo, es importante estudiar el comportamiento particular de aquellos parámetros seleccionados por cada instancia, y utilizar un conjunto de realizaciones con parámetros conocidos (del paso anterior) que dan buena respuesta a cada instancia (conocimiento a priori) y luego realizar un análisis de hipótesis de los parámetros apreciados.

Una forma de analizar los datos provenientes de combinaciones de distintos parámetros es el de estudiar el

comportamiento de los datos desde la perspectiva de un modelo, donde los parámetros del problema son las variables explicativas (o características) y medidas como el fitness o el tiempo de ejecución son la variable de salida.

La hipótesis estructural entonces en una primera instancia es la del modelo de regresión múltiple, que en términos simples, propone el valor de la función objetivo para cada una de las instancias  $y$  como

$$y_j = \beta_0 + \sum_{i=1}^k \beta_i x_{ij} + u_j, \quad j = 1 \dots n \quad (26)$$

donde los supuestos básicos son de lo anterior corresponde a

- $y_j$  y  $u_j$  como variables dependientes o explicadas
- $x_i$  variable explicativa estocástica
- $\beta_0, \beta_i$  el parámetro que pondera el efecto de la variable explicativa

y además se considera que  $u_{ij} \sim N(0, \sigma)$  (error aleatorio Gaussiano) donde el valor esperado es

$$E[y_j / x_1, x_2, \dots, x_k] = \beta_0 + \sum_{i=1}^k \beta_i x_{ji} \quad (27)$$

Este modelo de regresión pretende develar efecto de las variables explicativas más importantes y representa las restantes mediante una variable aleatoria. en términos generales se propone una explicación de la función objetivo del tipo

$$y = f(x_1, x_2, x_3, \dots, x_k) + \underbrace{g(x_1, x_2, x_3, \dots, x_k)}_u \quad (28)$$

El supuesto fuerte acá es que la función  $f$  admite una representación lineal del tipo

$$f = \sum_{j=0}^k \beta_j x_j, \quad \beta_0 = 1 \quad (29)$$

por lo tanto, la explicación del valor de la función objetivo corresponde a la función  $F$

$$y = F(|S|, \alpha, \omega, \phi_g, \phi_p, q, k) \quad (30)$$

La idea de estudiar el fitness explicado a través de los parámetros puede ser considerada porque la relación entre ambos no es fuertemente causal, dado que el algoritmo inicializa pseudo-aleatoriamente a las partículas y existen incertidumbres asociadas al valor final de la función objetivo, lo que permite realizar análisis estadístico para estudiar tal situación.

Es importante señalar que por bajo que sea el valor de la función objetivo, no se puede concluir que el enjambre de partículas ha resuelto satisfactoriamente el problema, por lo tanto esta medición otorga un atisbo sobre parámetros adecuados, dado que la función objetivo admite soluciones infactibles. La razón por la cual se estudia el valor de la función objetivo con soluciones infactibles es la de poder



medir el efecto de los parámetros al variar mas los datos que en situaciones de soluciones factibles, las que por lo demás requieren de una mayor cantidad de iteraciones para inicializar el enjambre y para encontrar una buena solución.

Sin embargo, pueden haber legítimas sospechas de que la relación entre los parámetros y la función objetivo no sea del tipo lineal, por lo tanto, en tales casos convendría más utilizar modelos empíricos de redes neuronales artificiales (ANN), pero, para este trabajo se considera el modelo de regresión multiple lineal pues es una primera aproximación a comprender el efecto de los parámetros en el fitness y los modelos empíricos ANN no otorgan una visualización directa de estos efectos.

Con un modelo estadístico asociado al problema es posible realizar simulaciones y así ampliar la perspectiva desde el punto de vista de las ejecuciones, dado que la obtención de un conjunto variado de soluciones constituye consumir largos períodos de tiempo, sin embargo, es importante señalar que el objetivo del análisis no es el de determinar o proponer un modelo de regresión u otra clase de ajuste, sino el de estudiar los efectos de los parámetros en el valor de función objetivo, que no establece estrictamente si el problema ha sido resuelto o nó, por lo tanto, el modelo en ningún caso representa la resolución efectiva problema que estamos estudiando en esencia.

Una vez estudiados los parámetros, se utilizan en realizaciones para comprobar la hipótesis de que dichos parámetros resuelven efectivamente el problema midiendo un parámetro como el valor medio  $\mu$  de la resolución efectiva del problema.

Se plantea la hipótesis nula, representada como  $H_o$  a la afirmación a contrastar, en contraposición de la hipótesis alternativa, representada por  $H_a$ .

El proceso de contrastar otorga información acerca de decidir si acaso se tienen evidencias suficientes para rechazar la hipótesis nula o alternativa. Para realizar el contraste de hipótesis se plantea una distribución teórica para explicar la variable estudiada que permite determinar si los datos obtenidos son estadísticamente distintos en forma significativa respecto a los valores teóricos.

- $H_o$ : Utilizando el conjunto de parámetros  $K$  se resuelve el problema con  $\mu = \mu_t$
- $H_a$ : Utilizando el conjunto de parámetros  $K$  se resuelve el problema con  $\mu \neq \mu_t$

#### IV-A. Organización de los Experimentos

Se realiza por cada instancia una serie de experimentos probando conjuntos con diferentes parámetros, midiendo la calidad de las soluciones y observando el tiempo de ejecución del algoritmo. Se observa la convergencia del valor de la función objetivo durante las iteraciones, lo que permite ver a posteriori la rapidez promedio de la convergencia.

Los experimentos constan de dos fases secuenciales, las que permiten atisbar los parámetros a utilizar en el contraste

de hipótesis posterior y así concluir respecto al uso de estos. Las fases son

- Fase 1 Realización del algoritmo con conjuntos de parámetros por cada una de las instancias utilizando ambas técnicas de interpolación (Hermité y Bezier)
- Fase 2 Realización del algoritmo con parámetros (heurísticas) fijos por cada una de las instancias.

La realización del algoritmo se programa mediante un “script” que ejecuta secuencialmente el programa generando resultados sobre el valor de la mejor partícula en términos de la función objetivo en cada iteración y los tiempos asociados.

*Ejemplo de ejecución del algoritmo:* El algoritmo especifica todos los parámetros antes mencionados y además especificar el modo de interpolación y la instancia.

```
./psomp.bin -iteraciones 50 -particulas 500
-alpha 2 -omega 2 -phip 20 -phig50
-map rooms`easychico.dat
-pivotes 8 -mode bezier
```

El programa puede ser descargado gratuitamente de internet y ejecutado con las mismas instancias con las que se probó para este caso utilizando el siguiente repositorio.

[git@github.com:arnaldog/Motion-Planning.git](https://github.com:arnaldog/Motion-Planning.git)

*IV-A1. Fase 1: Realización del algoritmo con combinaciones de parámetros:* Los parámetros a utilizar inicialmente corresponden a parámetros probados anteriormente que dan cuenta de resultados exitosos en intentos puntuales.

El procedimiento de la experimentación de este algoritmo consiste en considerar el siguiente escenario.

- Instancias: Bugtrap, T, Rooms, Complex
- $|S| = \{50; 250\}$
- $\alpha = \{2; 2,5; 3\}$
- $\omega = \{0,7; 1,5\}$
- $\phi_g = \{0,6; 1; 1,4\}$
- $\phi_p = \{0,6; 1; 1,4\}$
- $q = \{1; 2; 4; 6\}$
- Metodos de interpolación: Hermite y Bezier
- Iteraciones:  $\{5; 10; 30\}$
- Ejecuciones repetidas: 5 para Bezier y 1 para Hermite
- Bases hermite:  $\{1; 10; 20\}$

En total son 3456 ejecuciones para el PSO utilizando Bezier y 7588 usando Hermite Cada configuracion se realiza 3 veces, lo que significa un total de 220368 ejecuciones del algoritmo.

Una vez obtenido los resultados de este procedimiento se da inicio a la Fase 2, que consiste en la realización del algoritmo con parámetros deducidos de esta fase.

*IV-A2. Fase 2: Realización del algoritmo con parámetros fijos:* Se realiza el algoritmo para cada instancia con buenos parametros encontrados durante el proceso anterior, y parametros sesgados dada experimentacion manual.

Los parámetros que varían son específicamente los de el tamaño del enjambre (cantidad de partículas), el valor de la penalización  $\alpha$  y la cantidad de pivotes ya que el resto de los parámetros, dada la experimentación de la Fase 1, no manifiestan efectos importantes en la función objetivo al tener cada uno un rango muy pequeño en sus valores.

#### IV-B. Resultados

De las realizaciones anteriores se presentan los resultados resumidos en gráficos y tablas conforme a la especificación del experimento para cada fase.

*IV-B1. Resultados de la Fase 1: realización con combinaciones de parámetros:* Del total de las ejecuciones para esta fase, se obtuvieron los valores de la función de la mejor partícula en la inicialización y finalización, y también el tiempo total de ejecución.

- Gráfico de convergencia para cada método
- Tablas con los resúmenes de los datos para el valor de la función objetivo
- Tabla con los resúmenes para el tiempo de ejecución usando bezier
- Gráfico de distribución según parametros propios del algoritmo para cada instancia usando bezier
- Gráfico de distribución según parametros propios del algoritmo para cada instancia usando hermite

Dado el alto volumen de datos generados para involucrar una tabla descriptiva, se presentan los datos en sus respectivos histogramas para visualizar la distribución de los valores de la función objetivo respecto a los parámetros de interés.

*Gráficos de convergencia del algoritmo:* Se presentan los gráficos que describen el promedio y las desviaciones estándar de la función objetivo durante cada iteración del algoritmo para cada una de las instancias en las figuras ?? y ??.

La información otorgada en los gráficos corresponde al valor promedio de la convergencia por iteración, y se lee como el fitness promedio obtenido para la instancia en particular versus el número de iteración donde se consideran todos los datos obtenidos de las ejecuciones.

La convergencia del algoritmo usando el método de aproximación Bezier muestra mas variabilidad debido a que se realizaron las ejecuciones cinco veces por cada una de las configuraciones de los parámetros en total, vale decir que cada procedimiento se generó en cinco oportunidades.

La ejecución considerando el método de interpolación de Hermite Splines se realizó una sola vez ya que utiliza un parámetro adicional como el valor máximo de la gradiente a utilizar para la inicialización de las partículas.

Pese a aquello, la convergencia del algoritmo es muy rápida, teniendo una forma de exponencial en su valor promedio lo que permite suponer que la fase de inicialización de algoritmo es muy importante pues las partículas van aprendiendo de la mejor muy rápidamente y se estabilizan en iteraciones tempranas. En todos los casos el algoritmo

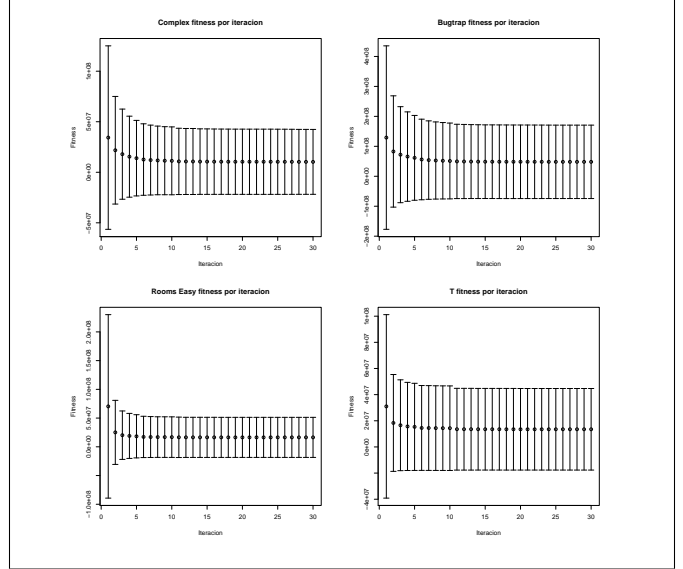


Figura 4: Convergencia de la función objetivo para las ejecuciones utilizando el método Bezier

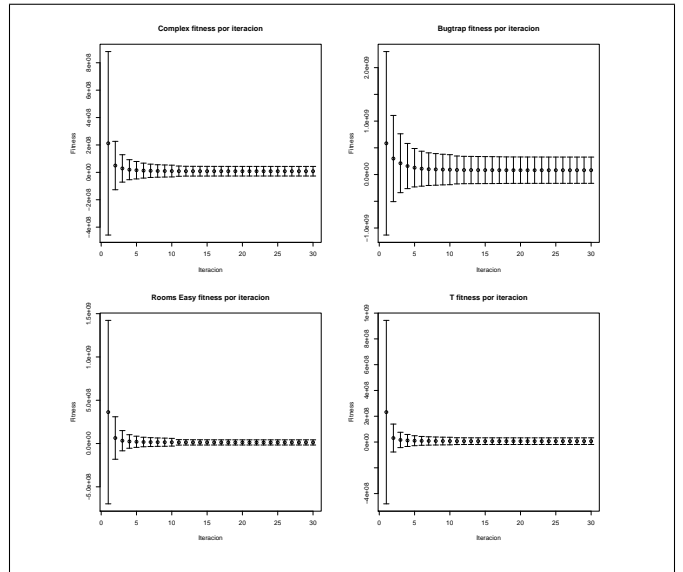


Figura 5: Convergencia de la función objetivo para las ejecuciones utilizando el método Hermite

converge muy prematuramente en la iteración número 10, que da cuenta de lo sensible que puede ser a los parámetros de entrada.

*Resumen de datos de tiempo de ejecución y valor de la función objetivo:* A continuación se presentan los resúmenes de los resultados y los histogramas para cada una de las instancias, considerando el total de los experimentos. Se separan en resúmenes para los experimentos hechos para bezier y hermité.

Los resultados obtenidos mediante la utilización del algo-

ritmo mediante el método Bezier para el valor de la función objetivo son los siguientes.

Instancia	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Bugtrap	101	303	205100	12330000	3170000	250400000
Complex	101	202	1377000	55370000	25570000	751100000
Rooms	202	275700	2320000	17730000	7991000	361600000
T	101	81810	410200	15050000	6393000	139100000

Cuadro I: Tabla con los resúmenes para el valor de la función objetivo usando método Bezier

Instancia	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Bugtrap	0.080	0.420	0.910	1.614	1.790	9.880
Complex	0.070	0.420	0.910	1.529	1.790	9.840
Rooms	0.080	0.420	0.830	1.362	1.580	7.840
T	0.080	0.410	0.860	1.453	1.700	10.060

Cuadro II: Tabla con los resúmenes para el tiempo de respuesta usando método Bezier

Instancia	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Bugtrap	101	303	205100	12330000	3170000	250400000
Complex	101	202	1377000	55370000	25570000	751100000
Rooms	202	275700	2320000	17730000	7991000	361600000
T	101	81810	410200	15050000	6393000	139100000

Cuadro III: Tabla con los resúmenes para el valor de la función objetivo usando método Hermite

Instancia	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Bugtrap	0.080	0.420	0.910	1.614	1.790	9.880
Complex	0.070	0.420	0.910	1.529	1.790	9.840
Rooms	0.080	0.420	0.830	1.362	1.580	7.840
T	0.080	0.410	0.860	1.453	1.700	10.060

Cuadro IV: Tabla con los resúmenes para el tiempo de respuesta usando método Hermite

*Resúmenes de histograma para los resultados:* En las figuras ?? y ?? se presenta la distribución de los valores de la función objetivo por cada parámetro particular del problema (cantidad de pivotes, valores de  $\alpha$  y tamaño del enjambre). En esta oportunidad no se muestran los gráficos de densidad respecto a los demás parámetros porque al estos variar muy poco en la realización del experimento no manifiestan diferencias significativas respecto a la distribución del valor de la función objetivo.

Al respecto, a través de los gráficos de densidad se visualiza la influencia de los parámetros sobre la distribución del fitness en cada instancia y con esta información se construye la tabla ??.

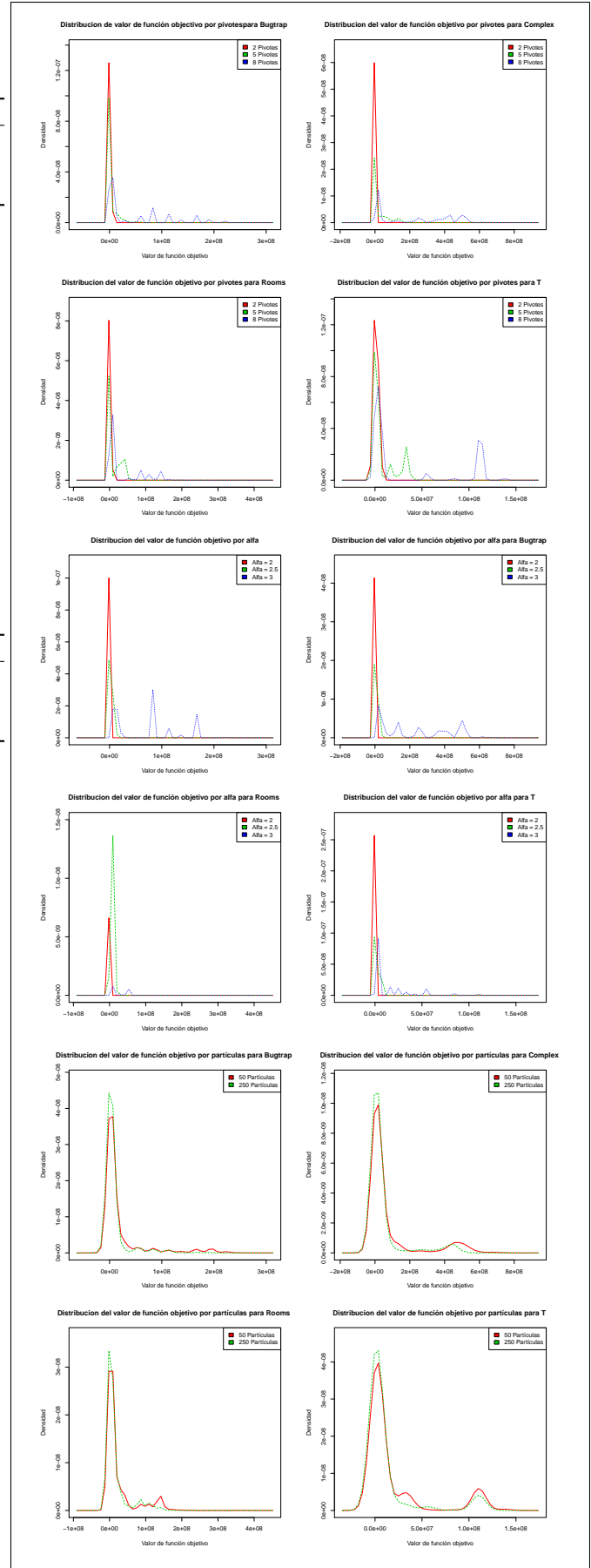


Figura 6: Grafico distribuciones para cada instancia usando interpolación Bezier

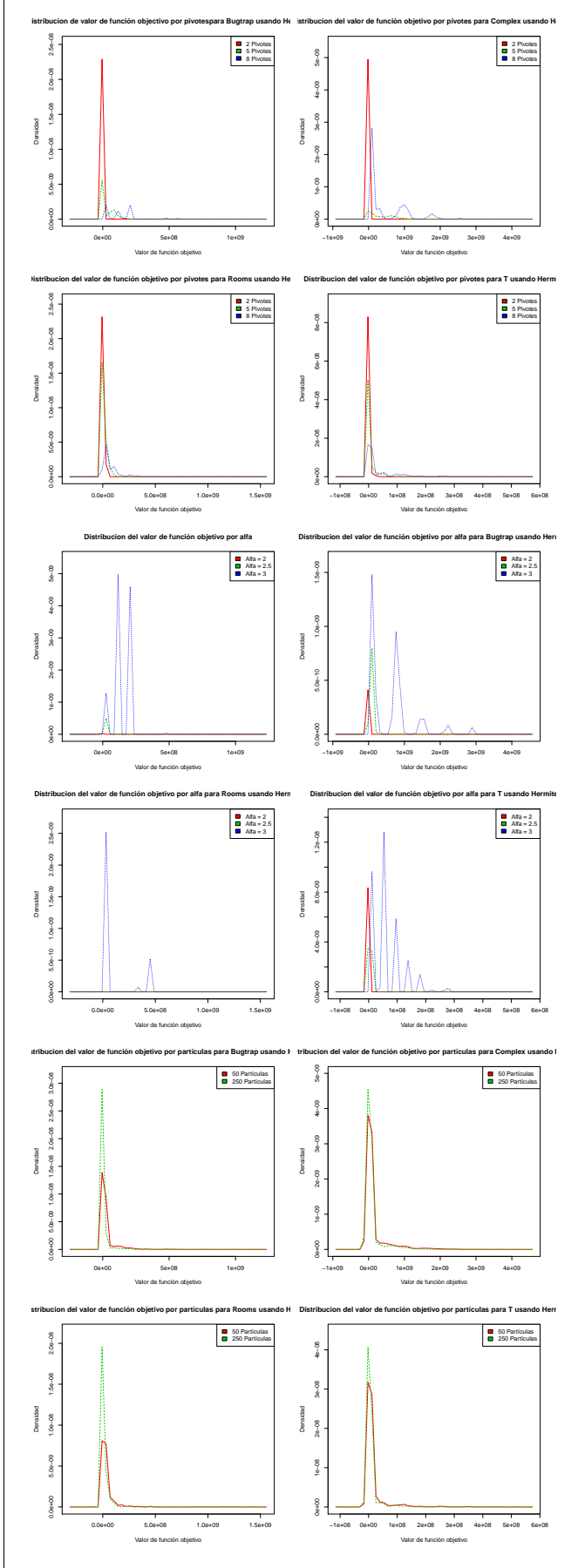


Figura 7: Grafico distribuciones para cada instancia usando interpolación Hermite

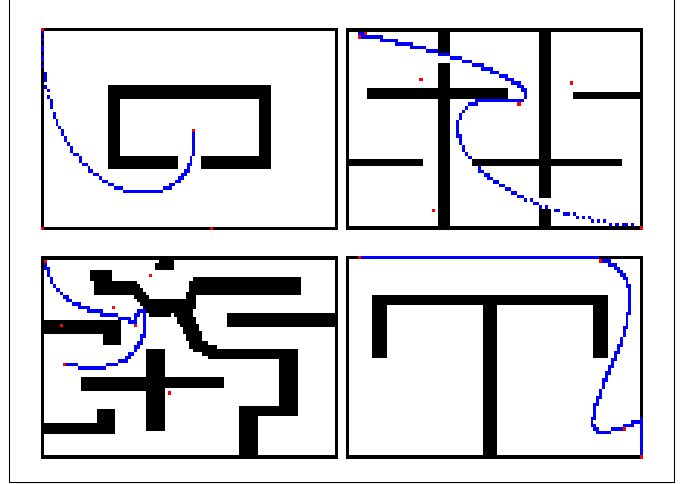


Figura 8: Ejemplos reales de resolución con ambas técnicas (hermité y bezier) de las diferentes instancias.

Instancia	$ S $	$\alpha$	$\omega$	$\phi_p$	$\phi_q$	$q$	Met.	It	Ej.	B
Bugtrap	250	2	1.5	1.5	1.5	2	B	10 0	100	10
T	250	2	1.5	1.5	1.5	2	B	10 0	100	10
Rooms	250	2.5	1.5	0.6	1.5	5	B	100	100	10
Complex	250	2	1.5	1.5	1.5	5	B	100	100	10
Bugtrap	250	3	1.5	1.5	1.5	2	H	100	100	10
T	250	3	1.5	1.5	0.6	2	H	100	100	10
Rooms	250	3	1.5	0.6	0.6	5	H	100	100	10
Complex	250	3	1.5	0.6	1.5	5	H	100	100	10

Cuadro V: Parámetros utilizados en la segunda fase de la experimentación

*IV-B2. Resultados del algoritmo con parámetros estimados de la fase 1:* Del procedimiento anterior se proceden a obtener los parámetros puntuales para esta fase. Por ejemplo la mayor cantidad de valores bajos (en cifras significativas) de la función objetivo para “Rooms” utilizando el método de aproximación Bezier, corresponde a la utilización de dos pivotes, valor de alpha de 2.5 y 250 partículas en el enjambre. El cuadro ?? muestra los parámetros sugeridos por el análisis realizado en la fase 1, por considerarlos más influyentes y sobre ellos se generan las nuevas ejecuciones para revisar el comportamiento de estos.

#### IV-C. Análisis

El análisis corresponde en su primera instancia a visualizar la importancia de los parámetros en general respecto al valor de la función objetivo.

*IV-C1. Analisis de datos para la Fase 1:* Para la fase 1 se analizan los parámetros utilizando el modelo de regresión múltiple por ser el más sencillo. No se realizan test de hipótesis de sus coeficientes ni mayor análisis (anova, andeva) pues lo que interesa es tener una perspectiva solamente y no la construcción de un modelo explicativo.

*Modelos de regresión:* A raíz de los datos obtenidos se generan los siguientes modelos de regresión para cada uno

de los métodos de interpolación.

*Coefficientes de la regresión lineal múltiple:* Se presenta la tabla con los coeficientes de la regresión lineal múltiple que significa la ponderación por unidad de los parámetros para el valor de la función objetivo.

p	Bugtrap	Complex	Rooms	T
B0	-9.580e+06	-5.266e+07	-2.856e+07	-1.765e+07
1	-1.225e+04	-1.857e+03	-1.929e+03	6.232e+03
2	4.477e+06	1.542e+07	1.175e+07	7.198e+06
3	4.195e+06	1.657e+07	4.008e+05	-8.947e+05
4	-2.693e+06	-6.604e+06	2.327e+04	-3.012e+05
5	-2.470e+06	-2.760e+06	-1.223e+06	-1.727e+06
6	4.752e+05	2.123e+06	8.519e+05	8.150e+05
7	3.112e-01	3.852e-01	1.917e-01	4.026e-01
R	0.7163	0.8697	0.8733	0.853
R2	0.7158	0.8695	0.8731	0.8528

Cuadro VI: Coeficientes estimados de la regresión Lineal Múltiple para cada Problema utilizando Bezier Splines.

p	Bugtrap	Complex	Rooms	T
B0	-6.723e+06	-1.187e+08	3.265e+14	-4.149e+06
1	-4.659e+04	-5.599e+04	1.994e+11	-1.990e+04
2	8.022e+06	5.119e+07	7.479e+13	6.404e+06
3	3.658e+06	3.846e+07	-1.570e+15	-5.522e+06
4	-5.294e+06	-1.108e+07	1.115e+15	6.736e+05
5	-6.130e+06	-4.485e+07	2.321e+13	-3.300e+06
6	8.154e+05	6.222e+06	1.128e+13	4.343e+05
7	3.988e-02	1.449e-01	2.815e+00	2.595e-02
R	0.3565	0.743	0.7373	0.3945
R2	0.3549	0.7427	0.7366	0.393

Cuadro VII: Coeficientes estimados de la regresión Lineal Múltiple para cada Problema utilizando Hermite Splines.

De los resultados se desprenden datos interesantes

*IV-C2. Resultados para la Fase 2:* A continuación se establece el análisis para el segundo caso.

*Test de hipótesis:* Se realiza un test de hipótesis para cada una de las resoluciones y bla bla baa

#### IV-D. Resultados Generales

Tabla con los resultados generales del experimento

Instancia	Calidad	Calidad Promedio	Tiempo	Entregable
T Chico	32	12	3124	230

Cuadro VIII: Resultados PSO utilizados según instancia

## V. CONCLUSIONES

Se concluye que

- Se debe elegir un número apropiado de partícula
- El algoritmo converge muy rápido
- Es muy fácil de implementar
- Con Bezier mejora el rendimiento

- Con Hermite cuesta el doble de tiempo procesar (por los pivotes)
- El parametro más influyente es:
- Las técnicas de interpolacion son muy efectivas
- La implementación es independiente del tamaño de la grilla, solo juega en contra el rendimiento
- Otras.