

Particle Swarm Optimizatin On Global Path Planning with Hermite and Bezier Splines

Arnaldo Gaspar and Joe Cabezas

Departamento de Informática

Universidad Técnica Federico Santa María

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_{inicial}) a otra región objetivo (p_{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, tiempo, energía, 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 para un entorno dinámico. Un entorno estático es aquel que no cambia su configuración durante el transcurso del tiempo.

Para el caso estático, el punto inicial, el punto de llegada, las regiones a evitar y las regiones a visitar son conocidas antes que el plan sea construido y se asume que las 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 *Dinamic 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, en cambio para *Dinamic Path Planning* el tiempo es crítico.

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 cuadrada con celdas de tamaño fijo y constituye el dominio de cada variable.

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 representada 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.

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 como Splines de Hermite es muy probable que las rutas generadas escapen del cuadro del mapa al no satisfacer la restricción de ajustarse al polígono de espacio mínimo (convex hull).

Para efectos de optimización, se consideran las colisiones como restricciones blandas y se penalizan en la función objetivo dado que su consideración dura podría no generar soluciones factibles iniciales.

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 utilizada en este artículo se consideran solamente las restricciones más generales del GPP y en especial, para efectos de 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.

En términos generales se espera que el algoritmo pueda 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

además el hecho de haber penalizaciones en rutas muy cortas de solución las cuales no representen una solución adecuada al problema.

La representación del problema planteado en forma matemática se puede expresar para cualquier dimensión, 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 $|O| = m$ obstáculos se define como

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

Definiendo 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 . El espacio ocupado por o_i será

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

Por lo tanto, el espacio libre corresponde a

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

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

$$p = \{s \in C_{\text{libre}} : s_1, \dots, s_l \text{ tal que } s_1 = A, s_l = B\} \quad (4)$$

Con el fin de hacer un estudio en una escala aplicable, se considera un modelo más simple, cuyas consideraciones son:

- Una matriz que describa el espacio de operación en dos dimensiones, segun las coordenadas de la matriz:
 - Si en una cierta coordenada existe un obstáculo, en la matriz se escribe un 1
 - Si está libre se escribe 0
- El agente tiene un tamaño tal que en cualquier instante esta en sólo una casilla, es decir, tiene tamaño 1×1
- Al agente no se le permiten movimientos en diagonal
- Ciertas consideraciones físicas no serán parte de este problema, no existe aceleración, velocidad, ni fuerzas que prohiban el movimiento libre por el espacio de operación

1. Para almacenar la informacion básica:

- Una matriz C , que describe el espacio de operación tal que:
 - $C \in \mathbb{R}^2$

$$C_{ij} = \begin{cases} 1 & \text{si hay obstáculo en la posición } i, j \\ 0 & \text{si no} \end{cases} \quad (5)$$

- Un vector p , de largo variable l que almacena la lista de casillas visitadas en orden donde:

- p_0 es la posición de inicio
 - p_l es la posición final
 - p_i es adyacente con p_{i+1}
- Este vector contendrá la solución final

2. La función objetivo:

$$\text{Min} : F(p) = \text{lenght}(p) + Pn(p) \quad (6)$$

donde:

- p es el vector de ruta obtenido
- $\text{lenght}(p)$ 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 funcion 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:

$$Pn(p) = NC(1 + \text{lenght}(p)^\alpha) \quad (7)$$

Con:

- NC = número de colisiones

El razonamiento detrás de esta función de penalización es como sigue:

Primero 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.

Segundo, además penaliza de manera proporcional a la cantidad de colisiones y potencialmente 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 mas corta entre el punto de inicio $X = C_{i,j}$ y el punto de meta $X' = C_{k,l}$ mide: $\text{lenght}(p) = |i - k| + |j - l|$, 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 mas largas pero que no tienen colisiones.

3. Sujeto a las siguientes restricciones:

- Los vectores de ruta p deben tener en su primer elemento la casilla de inicio, y como último elemento la casilla de meta
- El elemento del vector p_i es adyacente con p_{i+1}

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 planos [?].

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 Descomposition*) [?], 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 concen-

tricos 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 inválidas 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

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.

1. El mapa está representado como una matriz con coeficientes binarios

$$Map = \begin{bmatrix} (1,1) & \cdots & (1,m) \\ \vdots & \ddots & \vdots \\ (n,1) & \cdots & (n,m) \end{bmatrix}$$

2. Una partícula corresponde a una solución del problema, es decir una ruta r de longitud l

$$r = \{(p_{1x}, p_{1y}), \dots, (p_{lx}, p_{ly})\}, \quad |r| = l$$

3. Todas las combinaciones de rutas posibles dentro del mapa representan el dominio de cada ruta
4. La longitud de k la velocidad es fija, y cada punto se calcula con k incrementos de $\delta = \lfloor l/k \rfloor$ sobre la ruta p .

$$v = \{(p_{\delta x}, p_{\delta y}), \dots, (p_{k\delta x}, p_{k\delta y})\}, \quad |v| = k$$

Representación

1. Cada ruta p será construida pseudo aleatoriamente
2. La velocidad de la partícula en el inicio corresponderá a sus pivotes
3. Para evaluar la velocidad entre las partículas se actualizan los pivotes de las velocidades en vez de la posición.
4. La actualización de la posición corresponde a la generación de nuevas rutas que pasen por los pivotes antes calculados.

Función objetivo

$$\text{mín } F(r) = \text{lenght}(r) + C(1 + \text{lenght}(r)^\alpha) \quad (8)$$

1. $\text{lenght}(r)$ es el largo de la ruta
2. C es el número de colisiones
3. α es el nivel de penalización por rutas cortas con colisiones

Posición y velocidad de la partícula

$$r = \{p_1, \dots, p_l\} \quad (9)$$

$$v = \{v_1, \dots, v_k\} \quad (10)$$

1. La ruta r se vá generando pseudo aleatoriamente
2. Por ende, todas las rutas tienen largos distintos.
3. Sin embargo, la longitud del vector velocidad es constante dado como parámetro
4. En este caso difiere respecto a la velocidad respecto a la del PSO original.

Actualización de velocidad y posición de la partícula

$$v_{i+1} = \omega v_i + \rho_g \phi_g(g - x_i) + \rho_p \phi_p(p_i - x_i) \quad (11)$$

$$r_{i+1} = r_i + v_i \quad (12)$$

Los parámetros son:

1. ω que pondera el efecto de la velocidad actual
2. ϕ_g parámetro para favorecer la explotación
3. ϕ_p parámetro para favorecer la exploración
4. $\rho_g, \rho_p \sim U(0, 1)$
5. g la mejor posición global
6. p_i la mejor posición de la partícula i
7. r_i posición de la partícula i (la ruta i).

Actualización de velocidad y posición de la partícula

$$v_{i+1} = \omega v_i + \rho_g \phi_g(v(g) - v(r_i)) + \rho_p \phi_p(v(p_i) - v(r_i)) \quad (13)$$

$$r_{i+1} = \text{Construir ruta a partir de } v^{t+1} \quad (14)$$

Se calculan las diferencias entre los pivotes (velocidades)

1. $v(g) - v(r_i)$, la velocidad se calcula como
2. $v(p_i) - v(r_i)$

Donde la velocidad corresponde a un punto en el plano con componentes x, y

$$v_x = \omega v_{ix} + \rho_g \phi_g(v(g_x) - v(r_x)) + \rho_p \phi_p(v(p_x) - v(r_x)) \quad (15)$$

$$v_y = \omega v_{iy} + \rho_g \phi_g(v(g_y) - v(r_y)) + \rho_p \phi_p(v(p_y) - v(r_y)) \quad (16)$$

Inicialización del enjambre

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

Result: Mejor partícula s_i .

```
for i = 1 to S do
    r_i = Inicializar posicion aleatoriamente;
    p_i ← r_i;
    v_i = Inicializar velocidad aleatoriamente;
    if f(p_i) < f(g) then
        g ← p_i
```

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 ~ U(0, 1);
        v_i ← ωv_i + Φ_p r_p (p_i - r_i) + Φ_g r_g (g - r_i);
        x_i ← x_i + v_i;
        if f(r_i) < f(p_i) then
            p_i ← r_i;
            if f(p_i) < f(g) then
                g ← p_i;
            end
        end
    end
    i = i + 1;
end
return g ;
```