

Planificación de caminos: Un enfoque basado en Diagramas de Voronoi y Algoritmos Genéticos

Facundo Benavides

MINA - Instituto de Computación
Facultad de Ingeniería - Universidad de la República
Montevideo - Uruguay

40 JAIIO / XII ASAI, 2011

Agenda

- 1 Motivación
 - Antecedentes
 - Introducción
 - El problema de planificar un camino
 - Trabajos previos
- 2 Nuestra propuesta
 - Implementación
 - Resultados

Agenda

- 1 Motivación
 - Antecedentes
 - Introducción
 - El problema de planificar un camino
 - Trabajos previos
- 2 Nuestra propuesta
 - Implementación
 - Resultados

Teleoperación

El hombre tiene el control.

- Solución para un momento histórico.
- Luego ...
 - Dependencia del humano.
 - Fatiga cognitiva.
 - Retardos de comunicación.
 - Caída del enlace.
 - Otros.

Teleoperación

El hombre tiene el control.

- Solución para un momento histórico.
- Luego ...
 - Dependencia del humano.
 - Fatiga cognitiva.
 - Retardos de comunicación.
 - Caída del enlace.
 - Otros.

Agenda

- 1 Motivación
 - Antecedentes
 - **Introducción**
 - El problema de planificar un camino
 - Trabajos previos
- 2 Nuestra propuesta
 - Implementación
 - Resultados

Autonomía e Inteligencia

¿Pueden las máquinas tomar el control?

- Autonomía

- “Habilidad de un agente robótico de dominar la realización de un conjunto de tareas en un entorno dado, sin la necesidad de recurrir al diseñador humano para lograr cumplir sus objetivos”. [1]
- Dependiendo de la naturaleza del entorno el agente puede o no necesitar “inteligencia”.

- Inteligencia

- “Capacidad que permite a una entidad o agente funcionar adecuadamente en un entorno dado”. [1]
- Un alto grado de inteligencia permitiría al agente adaptarse a un mayor espectro de entornos.
- Restricciones físicas limitan el poder de adaptación.

Autonomía e Inteligencia

¿Pueden las máquinas tomar el control?

- Autonomía

- “Habilidad de un agente robótico de dominar la realización de un conjunto de tareas en un entorno dado, sin la necesidad de recurrir al diseñador humano para lograr cumplir sus objetivos”. [1]
- Dependiendo de la naturaleza del entorno el agente puede o no necesitar “inteligencia”.

- Inteligencia

- “Capacidad que permite a una entidad o agente funcionar adecuadamente en un entorno dado”. [1]
- Un alto grado de inteligencia permitiría al agente adaptarse a un mayor espectro de entornos.
- Restricciones físicas limitan el poder de adaptación.

Autonomía e Inteligencia

¿Pueden las máquinas tomar el control?

- Autonomía

- “Habilidad de un agente robótico de dominar la realización de un conjunto de tareas en un entorno dado, sin la necesidad de recurrir al diseñador humano para lograr cumplir sus objetivos”. [1]
- Dependiendo de la naturaleza del entorno el agente puede o no necesitar “inteligencia”.

- Inteligencia

- “Capacidad que permite a una entidad o agente funcionar adecuadamente en un entorno dado”. [1]
- Un alto grado de inteligencia permitiría al agente adaptarse a un mayor espectro de entornos.
- Restricciones físicas limitan el poder de adaptación.

Autonomía e Inteligencia

¿Pueden las máquinas tomar el control?

- Autonomía

- “Habilidad de un agente robótico de dominar la realización de un conjunto de tareas en un entorno dado, sin la necesidad de recurrir al diseñador humano para lograr cumplir sus objetivos”. [1]
- Dependiendo de la naturaleza del entorno el agente puede o no necesitar “inteligencia”.

- Inteligencia

- “Capacidad que permite a una entidad o agente funcionar adecuadamente en un entorno dado”. [1]
- Un alto grado de inteligencia permitiría al agente adaptarse a un mayor espectro de entornos.
- Restricciones físicas limitan el poder de adaptación.

Autonomía e Inteligencia

¿Pueden las máquinas tomar el control?

- Autonomía
 - “Habilidad de un agente robótico de dominar la realización de un conjunto de tareas en un entorno dado, sin la necesidad de recurrir al diseñador humano para lograr cumplir sus objetivos”. [1]
 - Dependiendo de la naturaleza del entorno el agente puede o no necesitar “inteligencia”.
- Inteligencia
 - “Capacidad que permite a una entidad o agente funcionar adecuadamente en un entorno dado”. [1]
 - Un alto grado de inteligencia permitiría al agente adaptarse a un mayor espectro de entornos.
 - Restricciones físicas limitan el poder de adaptación.

Entornos

El robot como agente situado.

Los entornos pueden clasificarse según las siguientes dimensiones o perspectivas [2]:

- Total o parcialmente observables.
- Deterministas o estocásticos.
- Secuenciales o episódicos.
- Dinámicos o estáticos.
- Discretos o continuos.
- Mono o multi agente.

Planificación del movimiento

Cómo moverse en un mundo dinámico y desconocido.

Lo mínimo que se puede esperar de un robot autónomo es que tenga la habilidad de planificar sus propios movimientos. [3]

- ¿Cómo llegar a un destino?
 - Navegación.
- ¿Dónde estoy?
 - Localización.
- ¿Dónde he estado?
 - Construcción de mapas.

Planificación del movimiento

Cómo moverse en un mundo dinámico y desconocido.

Lo mínimo que se puede esperar de un robot autónomo es que tenga la habilidad de planificar sus propios movimientos. [3]

- ¿Cómo llegar a un destino?
 - Navegación.
- ¿Dónde estoy?
 - Localización.
- ¿Dónde he estado?
 - Construcción de mapas.

Planificación del movimiento

Cómo moverse en un mundo dinámico y desconocido.

Lo mínimo que se puede esperar de un robot autónomo es que tenga la habilidad de planificar sus propios movimientos. [3]

- ¿Cómo llegar a un destino?
 - Navegación.
- ¿Dónde estoy?
 - Localización.
- ¿Dónde he estado?
 - Construcción de mapas.

Agenda

- 1 Motivación
 - Antecedentes
 - Introducción
 - **El problema de planificar un camino**
 - Trabajos previos
- 2 Nuestra propuesta
 - Implementación
 - Resultados

Enfoques

Métodos de resolución [3]

- Roadmap
 - Consiste en capturar la conectividad del espacio libre en una red de caminos.
- Cell Decomposition
 - Posiblemente el método más estudiado.
 - Consiste en descomponer el espacio libre en celdas. Asume que la navegación intra-celda es fácil de resolver y construye un grafo para representar la conectividad inter-celdas.
- Potential Field
 - Consiste en discretizar todo el espacio de configuración utilizando un grilla regular de pequeñas celdas.
 - Luego mover el robot asumiendo que es una partícula sometida en cada celda a diferentes campos de potencial.

Enfoques

Métodos de resolución [3]

- Roadmap
 - Consiste en capturar la conectividad del espacio libre en una red de caminos.
- Cell Decomposition
 - Posiblemente el método más estudiado.
 - Consiste en descomponer el espacio libre en celdas. Asume que la navegación intra-celda es fácil de resolver y construye un grafo para representar la conectividad inter-celdas.
- Potential Field
 - Consiste en discretizar todo el espacio de configuración utilizando un grilla regular de pequeñas celdas.
 - Luego mover el robot asumiendo que es una partícula sometida en cada celda a diferentes campos de potencial.

Enfoques

Métodos de resolución [3]

- Roadmap
 - Consiste en capturar la conectividad del espacio libre en una red de caminos.
- Cell Decomposition
 - Posiblemente el método más estudiado.
 - Consiste en descomponer el espacio libre en celdas. Asume que la navegación intra-celda es fácil de resolver y construye un grafo para representar la conectividad inter-celdas.
- Potential Field
 - Consiste en discretizar todo el espacio de configuración utilizando un grilla regular de pequeñas celdas.
 - Luego mover el robot asumiendo que es una partícula sometida en cada celda a diferentes campos de potencial.

Planificación de caminos

Tres partes de un solo problema.

- Path planning.
 - Resolver un camino libre de obstáculos entre el estado inicial en el que se encuentra el robot y un estado destino bajo ciertos criterios de evaluación.
 - Los más usados son:
 - Distancia, Seguridad, Consumo energético, Suavidad.
- Trajectory planning.
 - Introduce en la consideración del problema la variable Tiempo.
 - Control de movimientos de bajo nivel.
 - Resuelve el problema del seguimiento de las trayectorias determinadas previamente en los pasos anteriores.

Planificación de caminos

Tres partes de un solo problema.

- Path planning.
 - Resolver un camino libre de obstáculos entre el estado inicial en el que se encuentra el robot y un estado destino bajo ciertos criterios de evaluación.
 - Los más usados son:
 - Distancia, Seguridad, Consumo energético, Suavidad.
- Trajectory planning.
 - Introduce en la consideración del problema la variable Tiempo.
 - Control de movimientos de bajo nivel.
 - Resuelve el problema del seguimiento de las trayectorias determinadas previamente en los pasos anteriores.

Agenda

- 1 Motivación
 - Antecedentes
 - Introducción
 - El problema de planificar un camino
 - Trabajos previos
- 2 Nuestra propuesta
 - Implementación
 - Resultados

Propuestas anteriores

- Caminos de visibilidad: Construcción de mapas de trayectorias a partir de Diagramas de Voronoi aplicando ciertos criterios de optimización. [4]
- Path Planning basado en AG: A partir de una representación del entorno denominada Digital Potential Fields, aplicar un AG para mejorar el largo, seguridad y suavidad de los caminos a recorrer. [5]

Propuestas anteriores

- Caminos de visibilidad: Construcción de mapas de trayectorias a partir de Diagramas de Voronoi aplicando ciertos criterios de optimización. [4]
- Path Planning basado en AG: A partir de una representación del entorno denominada Digital Potential Fields, aplicar un AG para mejorar el largo, seguridad y suavidad de los caminos a recorrer. [5]

Diagramas de Voronoi I

- Una de las principales construcciones dentro de la Geometría Computacional y una de las técnicas más usuales en la construcción de mapas de trayectorias.
- Sea S un conjunto de entidades geométricas en \mathcal{R}^d y una distancia métrica $\|\cdot\|$, el diagrama de Voronoi correspondiente a S es la partición de \mathcal{R}^d en la máxima cantidad de regiones, tales que los puntos pertenecientes a cada región de Voronoi cumplen la condición de estar más cerca de una sola entidad de S que de ninguna otra. [6]
- En robótica móvil han sido aplicados considerando, usualmente, el espacio correspondiente a \mathcal{R}^2 , la distancia Euclideana entre puntos y el principio del vecino más próximo.

Diagramas de Voronoi I

- Una de las principales construcciones dentro de la Geometría Computacional y una de las técnicas más usuales en la construcción de mapas de trayectorias.
- Sea S un conjunto de entidades geométricas en \mathcal{R}^d y una distancia métrica $\|\cdot\|$, el diagrama de Voronoi correspondiente a S es la partición de \mathcal{R}^d en la máxima cantidad de regiones, tales que los puntos pertenecientes a cada región de Voronoi cumplen la condición de estar más cerca de una sola entidad de S que de ninguna otra. [6]
- En robótica móvil han sido aplicados considerando, usualmente, el espacio correspondiente a \mathcal{R}^2 , la distancia Euclideana entre puntos y el principio del vecino más próximo.

Diagramas de Voronoi I

- Una de las principales construcciones dentro de la Geometría Computacional y una de las técnicas más usuales en la construcción de mapas de trayectorias.
- Sea S un conjunto de entidades geométricas en \mathcal{R}^d y una distancia métrica $\|\cdot\|$, el diagrama de Voronoi correspondiente a S es la partición de \mathcal{R}^d en la máxima cantidad de regiones, tales que los puntos pertenecientes a cada región de Voronoi cumplen la condición de estar más cerca de una sola entidad de S que de ninguna otra. [6]
- En robótica móvil han sido aplicados considerando, usualmente, el espacio correspondiente a \mathcal{R}^2 , la distancia Euclídeana entre puntos y el principio del vecino más próximo.

Diagramas de Voronoi II

- Sea $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de puntos no alineados en el plano y $d(p_i, p_j)$ la distancia Euclideana entre los puntos p_i y p_j , la *región de Voronoi* $R(p_i)$ generada por el punto p_i queda definida por:

$$R(p_i) = \{p \in \mathcal{R}^2; d(p, p_i) \leq d(p, p_j), \forall j \neq i\} \quad (1)$$

- A los puntos p_i se les denomina *generadores de Voronoi*.
- Al conjunto de todas las regiones de Voronoi $R(p_1), R(p_2), \dots, R(p_n)$, *diagrama de Voronoi* de P .
- A una frontera común a dos regiones se le denomina *arista de Voronoi*.
- A un punto de corte entre tres o más aristas, *vértice de Voronoi*.

Diagramas de Voronoi II

- Sea $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de puntos no alineados en el plano y $d(p_i, p_j)$ la distancia Euclideana entre los puntos p_i y p_j , la *región de Voronoi* $R(p_i)$ generada por el punto p_i queda definida por:

$$R(p_i) = \{p \in \mathcal{R}^2; d(p, p_i) \leq d(p, p_j), \forall j \neq i\} \quad (1)$$

- A los puntos p_i se les denomina *generadores de Voronoi*.
- Al conjunto de todas las regiones de Voronoi $R(p_1), R(p_2), \dots, R(p_n)$, *diagrama de Voronoi* de P .
- A una frontera común a dos regiones se le denomina *arista de Voronoi*.
- A un punto de corte entre tres o más aristas, *vértice de Voronoi*.

Diagramas de Voronoi II

- Sea $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de puntos no alineados en el plano y $d(p_i, p_j)$ la distancia Euclideana entre los puntos p_i y p_j , la *región de Voronoi* $R(p_i)$ generada por el punto p_i queda definida por:

$$R(p_i) = \{p \in \mathcal{R}^2; d(p, p_i) \leq d(p, p_j), \forall j \neq i\} \quad (1)$$

- A los puntos p_i se les denomina *generadores de Voronoi*.
- Al conjunto de todas las regiones de Voronoi $R(p_1), R(p_2), \dots, R(p_n)$, *diagrama de Voronoi* de P .
- A una frontera común a dos regiones se le denomina *arista de Voronoi*.
- A un punto de corte entre tres o más aristas, *vértice de Voronoi*.

Diagramas de Voronoi II

- Sea $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de puntos no alineados en el plano y $d(p_i, p_j)$ la distancia Euclideana entre los puntos p_i y p_j , la *región de Voronoi* $R(p_i)$ generada por el punto p_i queda definida por:

$$R(p_i) = \{p \in \mathcal{R}^2; d(p, p_i) \leq d(p, p_j), \forall j \neq i\} \quad (1)$$

- A los puntos p_i se les denomina *generadores de Voronoi*.
- Al conjunto de todas las regiones de Voronoi $R(p_1), R(p_2), \dots, R(p_n)$, *diagrama de Voronoi* de P .
- A una frontera común a dos regiones se le denomina *arista de Voronoi*.
- A un punto de corte entre tres o más aristas, *vértice de Voronoi*.

Diagramas de Voronoi II

- Sea $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de puntos no alineados en el plano y $d(p_i, p_j)$ la distancia Euclideana entre los puntos p_i y p_j , la *región de Voronoi* $R(p_i)$ generada por el punto p_i queda definida por:

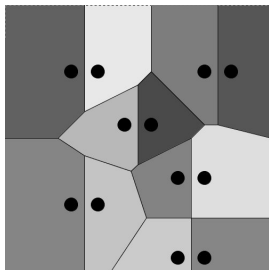
$$R(p_i) = \{p \in \mathcal{R}^2; d(p, p_i) \leq d(p, p_j), \forall j \neq i\} \quad (1)$$

- A los puntos p_i se les denomina *generadores de Voronoi*.
- Al conjunto de todas las regiones de Voronoi $R(p_1), R(p_2), \dots, R(p_n)$, *diagrama de Voronoi* de P .
- A una frontera común a dos regiones se le denomina *arista de Voronoi*.
- A un punto de corte entre tres o más aristas, *vértice de Voronoi*.

Diagramas de Voronoi III

Dos propiedades de interés en el contexto de la planificación de caminos para robots móviles:

- 1 Toda arista de Voronoi pertenece a la mediatriz del segmento de recta formado por los dos puntos generadores de las regiones que determinan la arista.
- 2 Todo vértice de Voronoi está ubicado exactamente en el centro del círculo en cuyo perímetro se encuentran los puntos generadores de las regiones que determinan el vértice.



Algoritmos Genéticos I

- 1950: Considerar que la evolución natural podría utilizarse como una herramienta efectiva para resolver problemas de ingeniería.
- 1960: *John Holland*, concibió los AG como técnicas probabilísticas de búsqueda que imitaran la evolución natural de las especies basándose en el principio de supervivencia de los individuos más aptos. [7, 8]
- En la actualidad, los conceptos centrales, entre las muy diversas implementaciones de la idea original de *Holland*, son: cromosoma, población, función de *fitness* y operadores genéticos.

Algoritmos Genéticos I

- 1950: Considerar que la evolución natural podría utilizarse como una herramienta efectiva para resolver problemas de ingeniería.
- 1960: *John Holland*, concibió los AG como técnicas probabilísticas de búsqueda que imitaran la evolución natural de las especies basándose en el principio de supervivencia de los individuos más aptos. [7, 8]
- En la actualidad, los conceptos centrales, entre las muy diversas implementaciones de la idea original de *Holland*, son: cromosoma, población, función de *fitness* y operadores genéticos.

Algoritmos Genéticos I

- 1950: Considerar que la evolución natural podría utilizarse como una herramienta efectiva para resolver problemas de ingeniería.
- 1960: *John Holland*, concibió los AG como técnicas probabilísticas de búsqueda que imitaran la evolución natural de las especies basándose en el principio de supervivencia de los individuos más aptos. [7, 8]
- En la actualidad, los conceptos centrales, entre las muy diversas implementaciones de la idea original de *Holland*, son: cromosoma, población, función de *fitness* y operadores genéticos.

Algoritmos Genéticos II

- Operadores genéticos:

- selección
- mutación
- cruzamiento.

- Algoritmo:

- 1 Población inicial
- 2 Mientras !parar hacer
 - 1 mutación(p_m) o cruzamiento(p_c)
 - 2 selección

Algoritmos Genéticos II

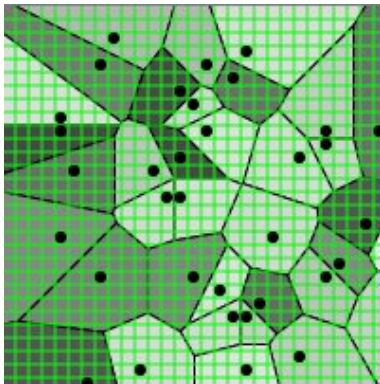
- Operadores genéticos:
 - selección
 - mutación
 - cruzamiento.
- Algoritmo:
 - 1 Población inicial
 - 2 Mientras !parar hacer
 - 1 mutación(p_m) o cruzamiento(p_c)
 - 2 selección

Agenda

- 1 Motivación
 - Antecedentes
 - Introducción
 - El problema de planificar un camino
 - Trabajos previos
- 2 Nuestra propuesta
 - Implementación
 - Resultados

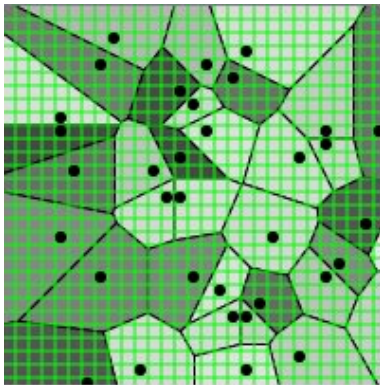
Modelado del entorno

- Totalmente observable, estático, determinístico, secuencial, discreto y mono-agente.
- Grilla regular, celdas libres u obstáculos.
- Diagrama de Voronoi generado por el conjunto de obstáculos.
- Robot como partícula. [6]



Modelado del entorno

- Totalmente observable, estático, determinístico, secuencial, discreto y mono-agente.
- Grilla regular, celdas libres u obstáculos.
- Diagrama de Voronoi generado por el conjunto de obstáculos.
- Robot como partícula. [6]



Aplicación del AG I

- Cromosomas: cada camino se codifica como una secuencia de celdas libres de largo variable.
- Población inicial:
 - generar el DV a partir de los obstáculos presentes en el entorno.
 - identificar las regiones que contienen la posición inicial y final.
 - Dijkstra para todas las combinaciones de puntos de partida y llegada.
 - Sorteo de caminos y celdas.
- Función de *fitness*:
 - factibles: largo, seguridad y suavidad.
 - infactibles: largo, tasa de infactibilidad y grado de infactibilidad.

Aplicación del AG I

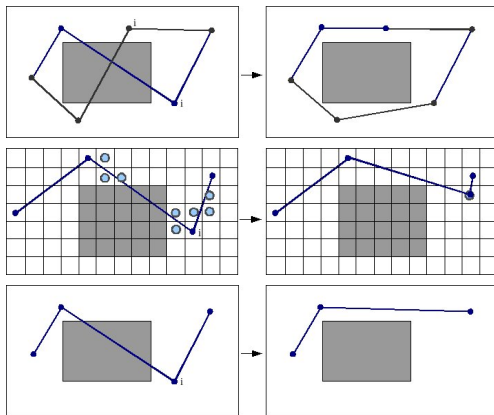
- Cromosomas: cada camino se codifica como una secuencia de celdas libres de largo variable.
- Población inicial:
 - generar el DV a partir de los obstáculos presentes en el entorno.
 - identificar las regiones que contienen la posición inicial y final.
 - Dijkstra para todas las combinaciones de puntos de partida y llegada.
 - Sorteo de caminos y celdas.
- Función de *fitness*:
 - factibles: largo, seguridad y suavidad.
 - infactibles: largo, tasa de infactibilidad y grado de infactibilidad.

Aplicación del AG I

- Cromosomas: cada camino se codifica como una secuencia de celdas libres de largo variable.
- Población inicial:
 - generar el DV a partir de los obstáculos presentes en el entorno.
 - identificar las regiones que contienen la posición inicial y final.
 - Dijkstra para todas las combinaciones de puntos de partida y llegada.
 - Sorteo de caminos y celdas.
- Función de *fitness*:
 - factibles: largo, seguridad y suavidad.
 - infactibles: largo, tasa de infactibilidad y grado de infactibilidad.

Aplicación del AG II

- Operadores genéticos:



Plataforma utilizada I

CPU	Intel Pentium Dual Core E2140 - 1.60GHz
RAM	2GB DDR
Swap	3GB
Disk	160 GB

Cuadro: Plataforma de hardware.

Plataforma utilizada II

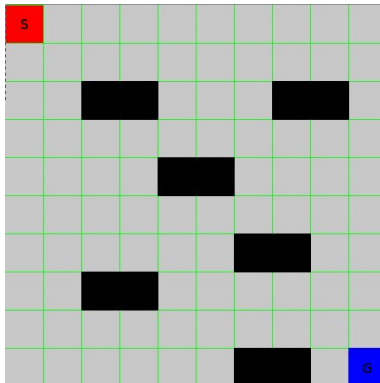
OS	GNU/Linux - Xubuntu 2.6.31-22
IDE	Eclipse Helios 3.6
Lenguaje	Java 1.6
AG Library	JGAP 3.3.3
Graph Library	JGrapht 0.8.1
Voronoi Library	Quickhull3d 1.4
Visor	Processing 1.0.7

Cuadro: Plataforma de software.

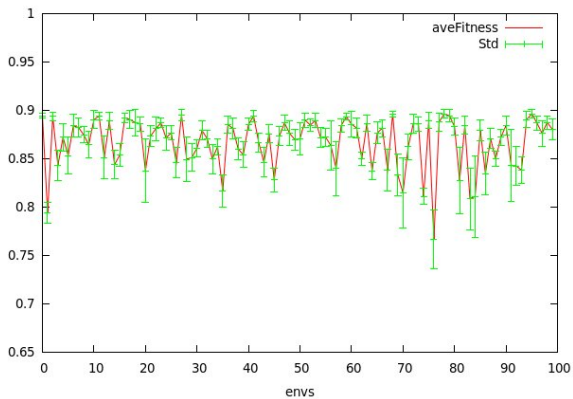
Agenda

- 1 Motivación
 - Antecedentes
 - Introducción
 - El problema de planificar un camino
 - Trabajos previos
- 2 Nuestra propuesta
 - Implementación
 - Resultados

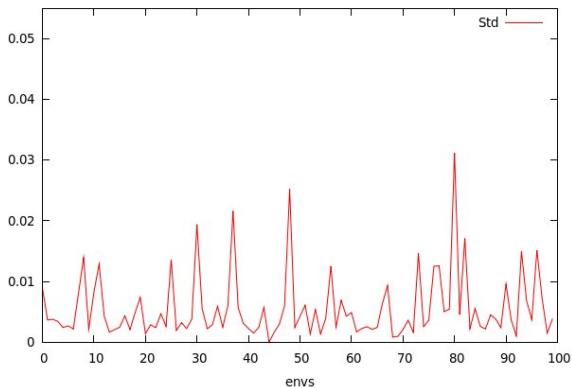
Entorno de prueba



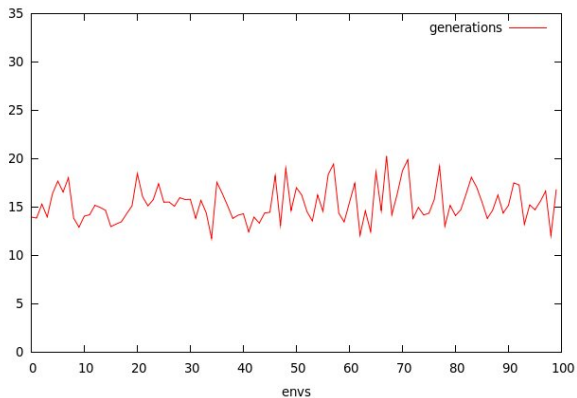
Fitness



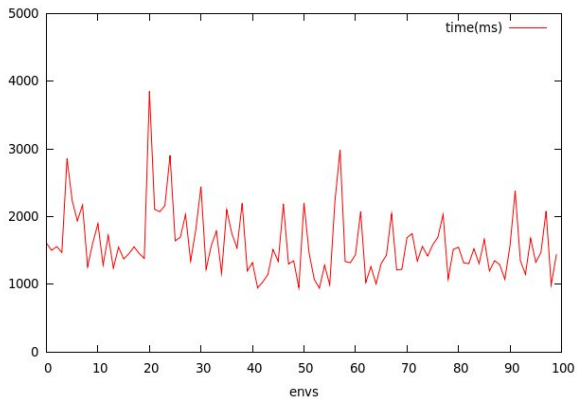
Desviación estándar



Generaciones



Tiempo



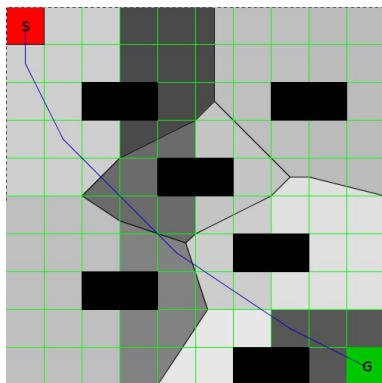
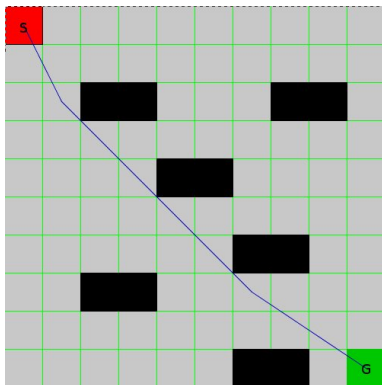
Comparativo numérico I

	AG sobre DPF	AG sobre DV
Generación del entorno (ms)	1632	1217
Evolución del AG (ms)	275.79	1299
Generaciones promedio	12.51	17.09
<i>Fitness</i> promedio	0.82990	0.84669
Desviación estándar	0.01067	0.00567
Mejor <i>Fitness</i>	0.84583	0.85459
Desviación al mejor	0.01267	0.00790
Largo promedio	656.27	657.63
Seguridad promedio	253.58	280.46
Suavidad promedio	0.29111	0.21318

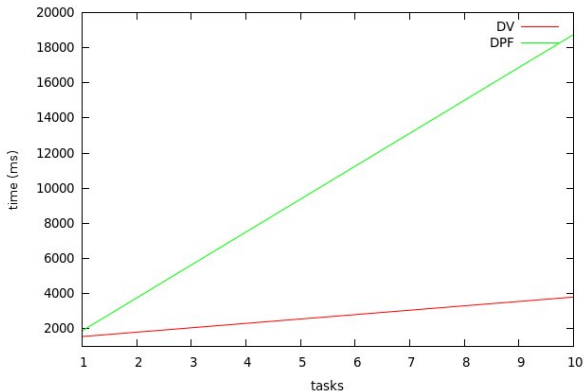
Comparativo numérico II

AG sobre DV	Parámetros de DPF	Parámetros ad-hoc
Generación del entorno (ms)	1305	1136
Evolución del AG (ms)	223	265
Generaciones promedio	15	15.45
<i>Fitness</i> promedio	0.83437	0.83720
Desviación estándar	0.01376	0.01169
Mejor <i>Fitness</i>	0.85433	0.85433
Desviación al mejor	0.01996	0.01714
Largo promedio	662.44	661.07
Seguridad promedio	291.19	287.55
Suavidad promedio	0.29058	0.27583

Comparativo gráfico



Comparativo multi-tarea



Conclusiones

- Factibilidad desde el punto de vista metodológico y en aplicabilidad práctica con robots reales.
- Mejora significativa en el desempeño manteniendo la calidad de las soluciones alcanzadas.

Conclusiones

- Factibilidad desde el punto de vista metodológico y en aplicabilidad práctica con robots reales.
- Mejora significativa en el desempeño manteniendo la calidad de las soluciones alcanzadas.

Trabajo futuro

- Desarrollar pruebas en entornos reales.
- Generalizar la librería de DV para soportar el modelado de cuerpos bidimensionales.
- Probar otras alternativas con AG.
- Considerar el aspecto dinámico.
- Portar el algoritmo a una plataforma robótica.

Trabajo futuro

- Desarrollar pruebas en entornos reales.
- Generalizar la librería de DV para soportar el modelado de cuerpos bidimensionales.
- Probar otras alternativas con AG.
- Considerar el aspecto dinámico.
- Portar el algoritmo a una plataforma robótica.

Trabajo futuro

- Desarrollar pruebas en entornos reales.
- Generalizar la librería de DV para soportar el modelado de cuerpos bidimensionales.
- Probar otras alternativas con AG.
- Considerar el aspecto dinámico.
- Portar el algoritmo a una plataforma robótica.

Trabajo futuro

- Desarrollar pruebas en entornos reales.
- Generalizar la librería de DV para soportar el modelado de cuerpos bidimensionales.
- Probar otras alternativas con AG.
- Considerar el aspecto dinámico.
- Portar el algoritmo a una plataforma robótica.





Trabajo futuro

- Desarrollar pruebas en entornos reales.
- Generalizar la librería de DV para soportar el modelado de cuerpos bidimensionales.
- Probar otras alternativas con AG.
- Considerar el aspecto dinámico.
- Portar el algoritmo a una plataforma robótica.



Agradecimientos

- Co-Autores
 - Gonzalo Tejera
 - Martín Pedemonte
 - Serrana Casella
- Colegas
 - Eduardo Grampín
 - Federico Andrade



Referencias I

-  D. Kumar Pratihari and L. C. Jain, *Intelligent Autonomous Systems*, vol. 275.
Springer, 2010.
-  S. Russell and P. Norvig, *Artificial Intelligence: A modern approach*.
Pearson, 2010.
-  J.-C. Latombe, *Robot Motion Planning*.
Kluwer Academic Publishers, 1991.
-  W. L. Roque and D. Doering, “Trajectory planning for lab robots based on global vision and voronoi roadmaps,” *Robotica*, vol. 23, no. 4, pp. 467–477, 2005.

Referencias II

-  Y. Zhang, L. Zhang, and X. Zhang, “Mobile robot path planning base on the hybrid genetic algorithm in unknown environment,” in *ISDA '08: Proceedings of the 2008 Eighth International Conference on Intelligent Systems Design and Applications*, vol. 2, (Washington, DC, USA), pp. 661–665, IEEE Computer Society, Nov. 2008.
-  R. Wein, J. P. van den Berg, and D. Halperin, “The visibility–voronoi complex and its applications,” *Computational Geometry: Theory and Applications*, vol. 36, no. 1, pp. 66–87, 2007.

Referencias III

-  D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*.
Upper Saddle River, NJ, USA: Addison-Wesley Professional, 1989.
-  M. Mitchell, *An Introduction to Genetic Algorithms*.
Cambridge, MA, USA: MIT Press, 1998.