

Metodología para Búsqueda y Evasión de Conflictos en Desplazamientos Múltiples sobre Mallado Triangular

J. L. Rico^b, H. J. Puga^a, V. M. Zamudio^a, J. A. Gordillo^b, A. Espinal^a

^aInstituto Tecnológico de León, ^bUniversidad Tecnológica del Suroeste de Guanajuato.
luis7321@ieee.org, pugahector@yahoo.com, vic.zamudio@ieee.org

Resumen

Una solución alternativa orientada a problemas de movilidad de entes artificiales ya sea físicos ó lógicos, se presenta en este trabajo. Ésta es una metodología para la coordinación de desplazamientos simultáneos sobre rutas descritas en un mismo entorno, evitando colisiones entre los móviles. El entorno está modelado a través de un espacio discreto en forma de malla triangular no estructurada, compuesta de un conjunto de vértices y aristas. Los móviles se modelan a través de agentes software, administrados por el ambiente de desarrollo de agentes Java, denominado JADE. Las rutas, descritas como sucesiones de vértices sobre el mallado, en un trabajo previo [1] quedan limitadas a aquellas sin segmentos o aristas comunes entre sí. La actual metodología que aquí se expone está libre de tal restricción, brindando mayor flexibilidad para la implementación en diversas situaciones de control de tráfico múltiple.

1. Introducción

En un ambiente cada vez más dependiente de la tecnología informática y sus áreas afines tales como la robótica y los sistemas inteligentes, el desplazamiento de entidades artificiales en entornos variados es un campo de investigación con múltiples aplicaciones y áreas de oportunidad. Entre ellas se pueden mencionar el control en líneas de producción [2], la exploración de zonas desconocidas o de difícil acceso para el ser humano[3], la gestión inteligente de tráfico vial, aéreo, marítimo o de datos [4][5] y el control de cadenas de suministro a través de agentes móviles[6].

Por otra parte, la necesidad de atender las demandas de una sociedad en continuo cambio, ha motivado a los investigadores a desarrollar nuevas técnicas computacionales que permitan afrontar nuevos retos. En este éste sentido ha surgido la tecnología de agentes [7] como una opción que permite atacar cierto

tipo de problemas de manera más eficiente y con menores costos para el desarrollador [8] [9].

En este artículo se presenta, bajo el paradigma de agentes, una propuesta de solución al problema de navegación de móviles sobre distintas rutas fijas con segmentos comunes, y que pueden generar conflictos de colisión durante el desplazamiento. El problema abordado es parte de la evolución natural de trabajos previamente desarrollados; donde se ha trabajado en la definición de contornos para los mallados [10], la generación de rutas óptimas [11], la evasión de obstáculos externos [12] y la evasión de colisiones entre los mismos agentes cuando sus rutas presentan cruces [1]. En el presente trabajo las diferentes rutas de un caso de prueba carecen de la restricción de contener segmentos comunes entre ellas, como se establecía en [1], esto permite que diferentes caminos contengan una o varias secciones comunes, inclusive de diferentes longitudes; y en cuanto a la orientación del desplazamiento, ésta puede ser en un mismo sentido o en contrasentido. La metodología que se expone ofrece una mayor adaptabilidad para la implementación en diversas situaciones de control de tráfico múltiple.

2. Marco teórico

Para el proceso de búsqueda y evasión de colisiones en desplazamientos múltiples sobre mallado, se requiere de los siguientes elementos:

1. Un mallado. Grafo ponderado no dirigido $M = (V, A)$, donde V es un conjunto de vértices y A es el conjunto de aristas.
2. n agentes a_i $1 \leq i \leq n$ quienes tienen a cargo el recorrido de n rutas sobre el mallado, y un agente adicional a_{n+1} llamado *Agente Central* que es quien coordina los itinerarios de los demás.
3. n rutas fijas de navegación R_i $1 \leq i \leq n$, definidas como una sucesión de vértices a través de la cuales se desplazan los agentes a_i .

Cada ruta R_i se implementa como una matriz r_i de dimensiones $l \times 2$, donde la primera columna contiene las l etiquetas de los vértices $v_{ij} \in M$, $1 \leq i \leq n, 0 \leq j \leq l-1$ que sigue la ruta, y la segunda columna guarda los respectivos valores de tiempo t_{ij} , que son los segundos que el i -ésimo agente permanece en un j -ésimo vértice al recorrer la i -ésima ruta. Los tiempos se definen inicialmente por el usuario y son modificados por el agente central en tiempo de ejecución como parte de la estrategia para evasión de colisiones.

La arquitectura del Sistema Multi-agente está sujeta a las siguientes restricciones, que garantizan el funcionamiento de la estrategia:

- El entorno de navegación es modelado a través de un mallado triangular, que es un grafo ponderado no dirigido.
- Se requiere igual número de rutas y agentes para su recorrido.
- Las rutas son fijas y definidas a priori para cada caso de prueba.
- El tiempo que un agente permanece ocupando un vértice es $t \in N \cup \{0\}$ y puede ajustarse en incrementos fijos $\Delta t \in N$.
- Cada vértice sólo puede alojar a un agente a la vez; excepto en los casos de vértice de inicio o fin de sus respectivas rutas. Esta restricción se ha considerado así porque con frecuencia los puntos de inicio o fin en un trayecto se dan en condiciones más controladas, similar a lo que ocurre en una terminal de autobuses, una estación ferroviaria o en un servidor de datos en una red informática. Sin embargo, dadas las condiciones particulares del problema al que se quiera aplicar la estrategia, esta restricción podría ajustarse.
- La estrategia considera que todos los recorridos inician a un mismo tiempo. Esta condición puede ser alterada en tiempo de diseño por parte del usuario o en tiempo de ejecución por parte del agente administrador como respuesta a la estrategia de evasión de conflicto.
- Los agentes encargados de recorrer las rutas, cuando se desplazan lo hacen siempre a velocidad constante.

3. Metodología

La búsqueda y evasión de conflictos se realiza a partir de cuatro elementos auxiliares que permiten llevar un registro detallado de las tareas realizadas en la ejecución de la metodología, éstos son:

I. Vector de combinaciones (V_{cmb}). La inspección de potenciales puntos de conflicto se realiza por pares de rutas. Así, dado un conjunto de n rutas sobre el mallado, el vector V_{cmb} contiene las h posibles combinaciones de dos rutas:

$$V_{cmb} = [C_k^{(S,T)}]; \quad 1 \leq k \leq h \quad (1)$$

Donde S y T representa los índices de la primera y segunda ruta de cada combinación, $1 \leq S, T \leq n$, $S \neq T$.

II. Matriz de vértices comunes ($M_{vComunes}$). Por cada combinación $C_k^{(S,T)}$, se genera la matriz $M_{vComunes}$ de vértices v_c , $1 \leq c \leq m$ que son comunes a R_S y a R_T . Esta matriz es de tamaño $m \times 4$, donde:

- La primera columna contiene las m etiquetas de cada vértice v_c , que es común R_S y a R_T .
- La segunda columna contiene la posición j que ocupa v_c en la sucesión R_S , y se representa como $j(v_c, R_S)$.
- La tercera columna contiene la posición j que ocupa v_c en la sucesión R_T , y se representa como $j(v_c, R_T)$.
- La cuarta columna se anexa como una adecuación a lo presentado en [1], y contiene un valor entero positivo d que identifica el segmento al cual pertenece el vértice, o bien contiene un cero, si el vértice no pertenece a ningún segmento o trayecto común.

$$M_{vComunes} = \begin{bmatrix} v_1 & j(v_1, R_S) & j(v_1, R_T) & d \\ \vdots & \vdots & \vdots & \vdots \\ v_m & j(v_m, R_S) & j(v_m, R_T) & d \end{bmatrix} \quad (2)$$

III. Vector de t raslape ($V_{traslape}$): Una colisión se presenta cuando dos agentes coinciden algún instante de tiempo sobre un mismo vértice o sobre un segmento común, este tiempo lo denominamos traslape. Para cada v_c en la matriz de vértices comunes, se obtiene un vector llamado $V_{traslape}$ que es de longitud dos, donde

- El primer elemento es la magnitud p que representa el tiempo de traslape, calculado a partir de la suma de los tiempos de recorrido en las aristas de la ruta (pesos de la arista) y los tiempos que cada agente permanece en un vértice (t_{ij}).
- El segundo elemento es el índice i correspondiente a la ruta r_i donde se aplicará el ajuste requerido a fin de evitar la colisión. Este segundo elemento puede ser cero si no hay traslape.

$$V_{\text{traslape}} = [p, i] \quad (3)$$

IV. Matriz de segmentos comunes (M_{sc}). Este elemento no se presenta en [1] y se incluye en este trabajo, para evitar el conflicto de navegación cuando dos o más rutas contienen una o varias secciones comunes.

Para cada combinación $C_k^{(T,S)}$ que contiene los w segmentos g_d , $1 \leq d \leq w$ que son comunes a R_S y a R_T , la matriz de segmentos comunes tiene las siguientes características:

- La primera columna contiene un consecutivo para los w segmentos comunes g_d .
- La segunda columna contiene la etiqueta del vértice donde inicia el segmento común correspondiente.
- La tercera columna contiene la etiqueta del vértice donde finaliza el segmento común correspondiente.
- La cuarta columna contiene un marcador para identificar el sentido de una ruta con respecto a otra en un segmento dado: siendo 1 para rutas en un mismo sentido y -1 para rutas en contrasentido.

$$M_{sc} = \begin{bmatrix} 1 & v_{c\text{inicial}} & v_{c\text{final}} & 1/-1 \\ \vdots & \vdots & \vdots & \vdots \\ w & v_{c\text{inicial}} & v_{c\text{final}} & 1/-1 \end{bmatrix} \quad (4)$$

Con los elementos antes descritos, si ocurre un conflicto entre rutas y las restricciones apropiadas se cumplen, se aplica la estrategia básica para evasión de colisiones, la cual consiste en un ajuste en el tiempo que el agente permanece en el vértice inmediato anterior al choque. Dicho ajuste se verá reflejado como un incremento del t_{ij} correspondiente, en la segunda columna de las matrices r_i .

Con la intención de mostrar un mapa completo del proceso que sigue la estrategia, ésta se presenta a manera de diagrama de flujo en la Figura 1, en la que se muestran los pasos más relevantes de dicho proceso.

El diagrama siguiente sin los bloques punteados, muestra la metodología básica sin la restricción: “Cada arista del mallado puede pertenecer como máximo a una ruta en particular”; que como ya se ha mencionado, limita en cierto modo la disposición de los caminos sobre la malla.

Los bloques marcados con línea punteada y la secuencia de bloques mostrada en la Figura 2, se anexan como parte de la nueva estrategia para extender el dominio de las rutas que acepta el sistema. Esta nueva propuesta permite la navegación de agentes sobre múltiples rutas, las cuales pueden contener

aristas comunes entre sí, como las que se muestran en el caso de prueba de la Figura 3.

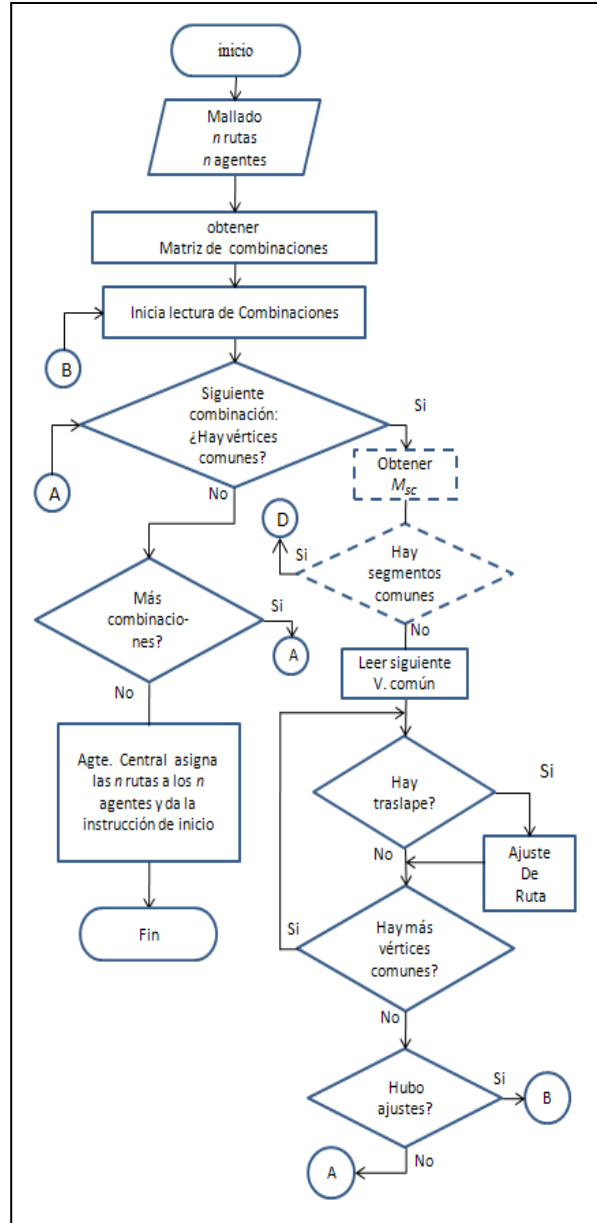


Figura 1. Estrategia básica para búsqueda y resolución de conflictos, con bloques adicionales para ajustarlo a la metodología actual (marcados con línea punteada).

Este caso de prueba, el cual contiene segmentos comunes entre rutas, se analizará como ejemplo para detallar de mejor manera la estrategia. Las rutas R_1 y R_2 están dadas por las sucesiones de vértices:

$$R_1 = \{b, c, d, e, f, h, j, i, k, l, n\}$$

$$R_2 = \{a, c, d, e, f, g, i, j, l, m\}$$

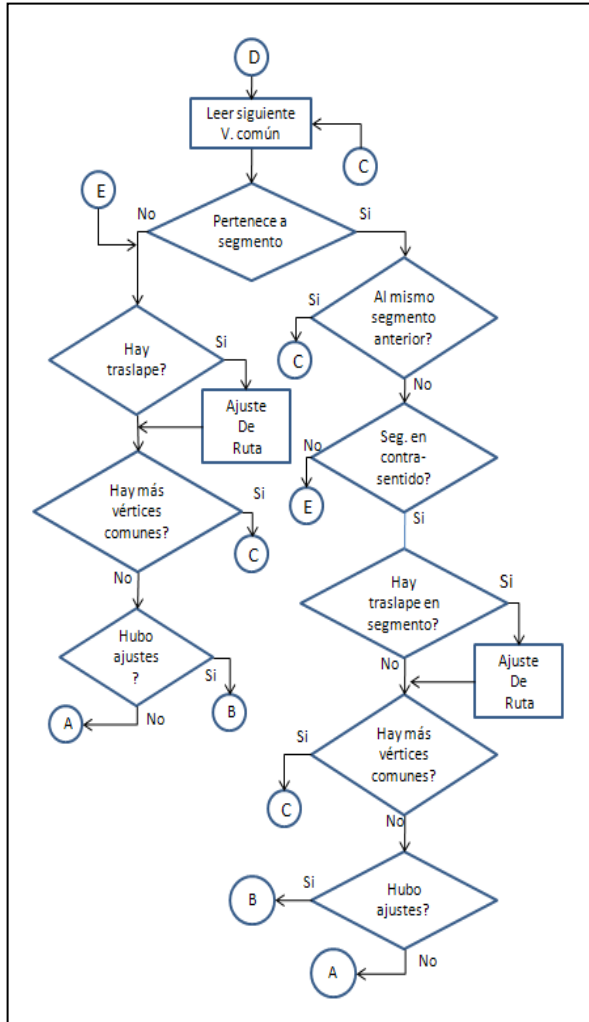


Figura 2. Estrategia para la búsqueda y resolución de colisiones entre rutas con segmentos comunes.

En la Figura 3, los segmentos comunes son dos, el primero es la sucesión $\{c, d, e, f\}$ en el cual ambas rutas se dirigen en el mismo sentido, y el segundo es la sucesión $\{i, j\}$ con dirección en contrasentido; finalmente, se observa el vértice l que es común a ambas rutas y que no pertenece a ningún segmento. Esta figura muestra las diferentes posibilidades que pueden presentarse en un caso de prueba cualquiera, pudiendo dar lugar a otras más complejas, a través de una combinación de estas configuraciones básicas.

Las dos rutas se ingresan en el sistema a través de sus correspondientes matrices r_1 y r_2 , en las que puede observarse para este ejemplo, un $t_{ij} = 1$ en todos los

vértices, excepto para los extremos de las rutas, donde éste tiene un valor de cero. Las rutas descritas se muestran a continuación:

$$r_1 = \begin{bmatrix} b & 0 \\ c & 1 \\ d & 1 \\ e & 1 \\ f & 1 \\ h & 1 \\ j & 1 \\ i & 1 \\ k & 1 \\ l & 1 \\ n & 0 \end{bmatrix} \quad r_2 = \begin{bmatrix} a & 0 \\ c & 1 \\ d & 1 \\ e & 1 \\ f & 1 \\ g & 1 \\ i & 1 \\ j & 1 \\ l & 1 \\ m & 0 \end{bmatrix}$$

De manera resumida se muestran los pasos a seguir para la búsqueda y resolución de colisiones:

PASO 1. Una vez que se han instanciado en JADE el mallado, las rutas y los agentes requeridos, se obtiene el vector de combinaciones, que en este caso debido a que sólo hay dos rutas, sólo contiene un elemento:

$$V_{cmb} = [C_1^{(1,2)}] \quad (5)$$

PASO 2. Se inicia la búsqueda de vértices comunes para la única combinación en (5), lo cual genera:

$$M_{vComunes} = \begin{bmatrix} c & 1 & 1 & 1 \\ d & 2 & 2 & 1 \\ e & 3 & 3 & 1 \\ f & 4 & 4 & 1 \\ i & 6 & 6 & 2 \\ j & 7 & 7 & 2 \\ l & 9 & 8 & 0 \end{bmatrix} \quad (6)$$

Donde la cuarta columna se llena con los resultados del paso 3

PASO 3. Se procede a la búsqueda de segmentos comunes, con lo cual se genera la siguiente matriz:

$$M_{sc} = \begin{bmatrix} 1 & c & f & 1 \\ 2 & i & j & -1 \end{bmatrix} \quad (7)$$

En ésta se muestran los dos segmentos mencionados anteriormente, sus vértices de inicio y fin, y el indicador de orientación. Debido a que la matriz no está vacía, el segundo bloque punteado en la Figura 1 dirige la estrategia hacia la secuencia de la Figura 2.

PASO 4. En esta fase se evalúa cada vértice común en (6), para verificar si pertenece o no a algún segmento común. De no ser así, la estrategia evalúa el traslape, y lo corrige si es necesario, incrementando el t_{ij} en el vértice inmediato anterior al vértice del conflicto. El criterio bajo el cual se decide la ruta que será modificada, es aleatorio cuando ambos agentes llegan al vértice al mismo tiempo, de otro modo, el agente que llega después es quien debe esperar. Debido a que el vértice c pertenece al segmento común 1 y que es el primero ubicado en este segmento, se procede al siguiente paso.

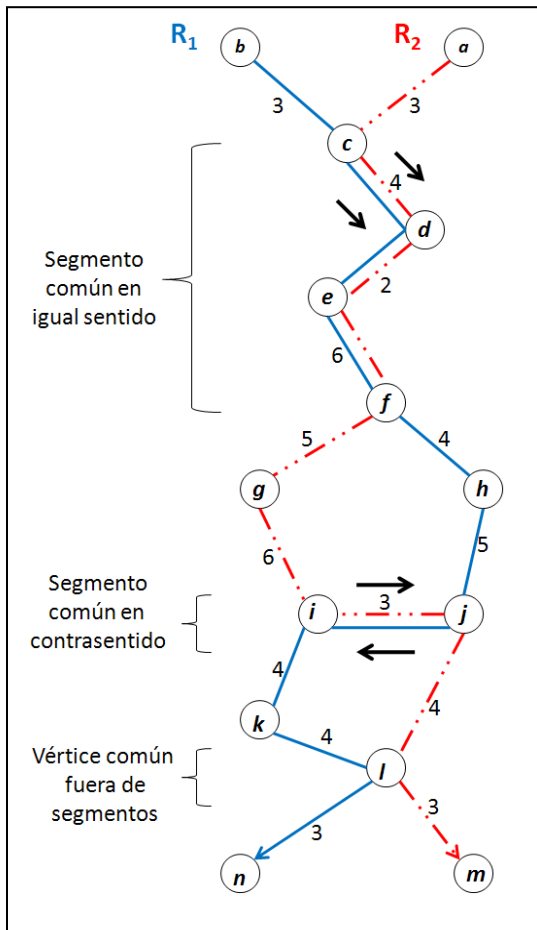


Figura 3. Caso de prueba con segmentos comunes de diferente longitud y sentido.

PASO 5. Se verifica el sentido de R_1 y R_2 en el segmento 1, que para este caso será en igual sentido.

Si una colisión ocurre en segmentos cuyas rutas tienen igual sentido, bastará con corregir el traslape en el primer vértice del segmento, y éste quedará completamente libre de colisiones debido a que los agentes se desplazan todos a velocidad constante. Por lo tanto al hacer el ajuste para evadir el conflicto en el

vértice c , la ruta 2 es aleatoriamente seleccionada para ser modificada, quedando como se muestra a continuación:

$$r_1 = \begin{bmatrix} b & 1 \\ c & 1 \\ d & 1 \\ e & 1 \\ f & 1 \\ h & 1 \\ j & 1 \\ i & 1 \\ k & 1 \\ l & 1 \\ n & 0 \end{bmatrix}$$

Lo que implica un retraso adicional de un segundo en el vértice b .

PASO 6. La estrategia identifica los vértices d , e y f como parte del mismo segmento ya corregido; por otra parte la situación cambia cuando evalúa el vértice i , identificado como parte de un segundo segmento, que a diferencia del primero, la dirección de las rutas es en contrasentido. Esto dirige al algoritmo a otra sección.

PASO 7. Si las rutas están en contrasentido, habrá que calcular el traslape de tiempo p en el segmento completo. El cálculo de p mostrado en la ecuación (3), se describe en la sección 3,

Agente 1. Ingreso al segmento: segundo 30

Salida del segmento: segundo 35

Agente 2

Ingreso al segmento: segundo 31

Salida del segmento: segundo 36

Entonces el vector traslape del segundo segmento común es:

$$V_{\text{traslape}_{g_2}} = [4,2]$$

PASO 8. Se procede al ajuste en la ruta dos incrementando 4 segundos al tiempo de espera en el vértice g , quedando la ruta como se muestra continuación:

$$r_2 = \begin{bmatrix} a & 0 \\ c & 1 \\ d & 1 \\ e & 1 \\ f & 1 \\ g & 5 \\ i & 1 \\ j & 1 \\ l & 1 \\ m & 0 \end{bmatrix}$$

PASO 9. La estrategia continúa verificando si hay más vértices comunes al final del segundo segmento común, esto nos lleva finalmente a la evaluación del vértice l , el cual no pertenece a ningún segmento.

PASO 10. Se identifica una última colisión en el vértice l , el cual se corrige haciendo un ajuste en la ruta l nuevamente. Los itinerarios finales quedan así:

$$r_1 = \begin{bmatrix} b & 1 \\ c & 1 \\ d & 1 \\ e & 1 \\ f & 1 \\ h & 1 \\ j & 2 \\ i & 1 \\ k & 1 \\ l & 1 \\ n & 0 \end{bmatrix} \quad r_2 = \begin{bmatrix} a & 0 \\ c & 1 \\ d & 1 \\ e & 1 \\ f & 1 \\ g & 5 \\ i & 1 \\ j & 1 \\ l & 1 \\ m & 0 \end{bmatrix}$$

4. Resultados.

Una solución satisfactoria es aquella que contiene un conjunto de rutas cuyos tiempos descritos en la segunda columna (t_{ij}), permitan el recorrido completo de cada ruta sin problemas de colisión entre agentes.

Los resultados obtenidos para el caso de prueba de ejemplo, muestran este proceso de ajuste en los tiempos t_{ij} , a partir de los cuatro elementos descritos en la Sección 3 y la metodología planteada en las Figuras 1 y 2. Otros casos de prueba con diferentes condiciones se han evaluado, obteniendo resultados satisfactorios en todos ellos. Sin embargo queda aún trabajo por hacer para garantizar la robustez de tal metodología.

5. Conclusiones.

Con este trabajo se muestra la viabilidad de un Sistema Multi-agente para la coordinación de múltiples desplazamientos en un mismo entorno. La

evasión de colisiones se ha resuelto de manera satisfactoria, aplicando la actual metodología en la resolución de diferentes casos prototipo como el que aquí se expone de ejemplo. Aún cuando los resultados de la estrategia han sido favorables hasta el momento, actualmente se está trabajando en pruebas que validen la existencia de una solución para cualquier caso de prueba que se apegue a las restricciones requeridas.

Por otra parte, la solución que se obtiene para un problema en particular no siempre es única; esto se debe a que el Sistema Multi-agente, designa de manera aleatoria el agente que ha de ceder el paso cuando dos de ellos llegan al mismo tiempo sobre un vértice. Así pues, un trabajo futuro será la búsqueda de la mejor solución; es decir, aquella que optimice en tiempo los itinerarios finales.

5.1 Agradecimientos. Los autores agradecen a PROMEP y DGEST el apoyo proporcionado para realizar esta investigación (proyecto DGEST No. 3528.10-P)

6. Referencias.

- [1] J. L. Rico, J. H. Puga, V. M. Zamudio, M. Ornelas, J. M. Carpio, J. A. Gordillo, (2011); "Propuesta basada en Sistemas Multi-agente para desplazamientos múltiples sobre mallado triangular como mapa de ruta." VIII Encuentro Participación de la Mujer en la Ciencia. Marzo 2011. ISBN 978-607-95228-2-7
- [2] Bussmann s., Schild k., (2000); "Self-organising manufacturing control: An industrial application of agent technology", Proc,4th Int. Conf. Multi-Agent Systems, Boston, MA 87-94.
- [3] Rekleitis Ioannis; Bedwani Jean-Luc; Dupuis Erick (2007); "Over-the-Horizon, Autonomous Navigation for Planetary Exploration", IEEE/RSJ International Conference on Intelligent Robots and Systems San Diego, CA, USA, Oct 29 - Nov 2.
- [4] Adel Sadek; Nagi Basha (2006): "Self-Learning Intelligent Agents For Dynamic Traffic Routing On Transportation Networks" Dept. of Civil and Environmental Engineering and Dept of Computer Science University of Vermont
- [5] Lucas A (1997): "Decision Support systems for arrivals flow Management" International technical conference, Prague Czech Republic.
- [6] Brugali D., Menga G., Galarraga S., (1998) "Inter-Company Supply Chains Integration via Movable Agents". Proceedings of PROLAMAT'98, Italia.
- [7] Mas Ana (2005): "Agentes Software y Sistemas Multi-Agente: Conceptos, Arquitecturas y Aplicaciones", editorial PEARSON Prentice Hall, ISBN:84-205-4367-5
- [8] Julián V.; Botti V. (2000): "Agentes Inteligentes: el siguiente paso en la inteligencia Artificial" NOVATICA, Especial 25 Aniversario, Edición digital ATI 2000.

- [9] Wooldridge Michael; Jennings Nicholas R. (1995): "Intelligent agents: theory and practices", The Knowledge Engineering; Review, Vol. 10:2, 1995, 115-152.
- [10] López Cruces J. M.; Puga Soberanes H.J.; Carpio Valadez J.M. (2008): "Definición de contornos a través de curvas paramétricas utilizando técnicas Heurísticas" Tesis de Maestría, Instituto Tecnológico de León. León Guanajuato, México.
- [11] Ávila C. Victor Germán; Puga Hector J. (2008): "Generación de caminos en mapas 2D utilizando técnicas sobre mallado automático no estructurado" Tesis de Maestría, Instituto Tecnológico de León.
- [12] Ruiz R.; Puga S. (2011): "Heurísticas para evasión de obstáculos en caminos sobre mallado no estructurado" Tesis de Maestría, Instituto Tecnológico de León. León Guanajuato, México.