

# Pruebas de Generación de Emboscadas Utilizando A\*mbush

K. Fernandes<sup>1,2</sup> y C. Chang<sup>1</sup>

<sup>1</sup>Departamento de Computación y Tecnología de la Información, Universidad Simón Bolívar, Venezuela

<sup>2</sup>INESC TEC, Porto, Portugal

---

## RESUMEN

*La sección de RESUMEN debe tener el resumen del trabajo en idioma español e inglés. El resumen en español debe preceder al resumen en inglés y debe llevar por título la palabra RESUMEN. Ambas versiones del resumen deben tener alineación justificada, en formato de una columna, y deben estar ubicadas debajo de la afiliación del autor. El título del resumen debe usar una fuente Times tamaño 9, debe estar en negritas, debe estar alienado a la izquierda y debe estar escrito en mayúsculas. La sección de resumen completa, incluyendo ambas versiones y los keywords, no debe exceder las 3.5 pulgadas (8.89 cm) de largo.*

## ABSTRACT

*The ABSTRACT is to be in fully-justified italicized text, between two horizontal lines, in one-column format, below the author and affiliation information. Use the word "Abstract" as the title, in 9-point Times, boldface type, left aligned to the text, fully capitalized. The abstract is to be in 9-point, single-spaced type.*

**Keywords:** A\*, estrategias grupales, búsqueda de caminos, grafos.

---

## 1. Introducción

La generación de conductas inteligentes ha sido un reto constante en el área de Inteligencia Artificial para Videojuegos [MF09], frecuentemente derivado en el desarrollo de acciones preestablecidas que el usuario puede fácilmente identificar después de varias ejecuciones del juego. Esta característica se evidencia aún más común cuando se trata de la generación de movimientos tácticos y estratégicos grupales, los cuales suelen ser sumamente complejos de implementar.

Un problema muy tratado en la literatura es la búsqueda de caminos a un punto común, por parte de grupos de agentes dentro de un juego [MF09]. Este punto suele venir dado por un lugar en el mapa de juego, potencialmente la posición del oponente. El esquema regularmente utilizado es generar caminos de costo mínimo [HNR72] [RN93] hacia este punto, sobre el grafo inducido por el mapa del juego. Es muy probable, que estos caminos confluyan, evitando la diversidad de rutas y exploración del mapa.

Al efectuar una persecución al oponente, la utilización de caminos óptimos como estrategia deja muchos espacios de escapatoria libres, por lo que es de especial interés generar mecanismos de diversificación de rutas que generen situaciones de emboscada.

Anteriormente, fue propuesta una solución para este problema utilizando variantes del algoritmo de A\* [FGC12a] [FGC12b]. Estas variantes muestran ser efectivas en el cálculo de rutas de emboscada. El presente trabajo pretende demostrar que dicho enfoque se adapta bien ante diversos tipos

de grafos de distinto tamaño. Para realizar esta evaluación, se propone una nueva métrica para validar la calidad de una emboscada, que incorpora más información que la métrica planteada originalmente. Esta nueva métrica incluye información de alcanzabilidad por parte de los agentes que pretenden realizar la emboscada a las salidas dejadas como escapatoria. Esto con el fin de no penalizar casos en los cuales el nivel de emboscada alcanzado es el máximo posible dadas las condiciones iniciales.

El resto de este artículo está organizado de la siguiente manera. En la sección 2 se presentan distintas soluciones de la literatura para este problema, en la sección 3 se muestra la definición formal del problema, en conjunto con la métrica anterior y la planteada en el presente trabajo para medir la calidad de una estrategia de emboscada. A continuación, en la sección 4 son mostrados los algoritmos de A\*mbushy sus variantes, anteriormente presentados con mayor detalle por Fernández y colaboradores [FGC12a] [FGC12b]. Los experimentos realizados se incluyen en la sección 5 y, finalmente, las conclusiones y trabajo futuro son presentados en la sección 6.

## 2. Trabajos Relacionados

Este trabajo se centra en el estudio del método expuesto previamente por Fernández y colaboradores [FGC12a] [FGC12b]. Enfoques alternativos a éste fueron propuestos por Silver [Sil06] y por van Toll y colaboradores [vTCG12].

En el trabajo propuesto por Silver [Sil06], denominado

*Space-Time A\**, se plantea una variante de  $A^*$  donde se tiene un grafo no dirigido en forma de cuadrícula, con función de costos común para todos los agentes y constante entre cada par de nodos.

El objetivo principal de dicho trabajo es evitar el paso de dos agentes por un mismo nodo en un instante de tiempo dado. La variación planteada, incluye una extensión en el número de dimensiones de  $A^*$ ; se considera además de la posición de los agentes, el tiempo transcurrido. Dicha variación tiene un costo añadido en tiempo y memoria considerable debido a que se incrementa el tamaño del espacio de búsqueda. Aunque el objetivo final de las soluciones es diferente, la estudiada en el presente trabajo presenta una clara ventaja sobre *Space-Time A\** en varios aspectos. En primer lugar, presenta resuelve con un algoritmo de menor complejidad de cómputo problemas más generales, debido a que permite la utilización de grafos dirigidos con costos de arcos variables. Asimismo se puede trabajar con agentes que tengan grafos distintos entre sí. Por otro lado, en el caso de tener agentes con distintas velocidades, *Space-Time A\** genera un espacio de búsqueda en extremo denso, cuando la velocidad relativa entre los elementos del juego es alta. En cambio,  $A^*mbush$  no se ve afectado por la diferencia de velocidades entre los agentes.

Asignando costos homogéneos para cada arco e incremento infinito sobre nodos o arcos ocupados, el problema que resuelve  $A^*mbush$  se reduce al problema planteado para *Space-Time A\**.

El segundo trabajo mencionado, desarrollado por van Toll y otros [vTCG12], muestra un enfoque basado en densidades para generar diversidad de caminos. Sin embargo, la diversificación de caminos es dada sólo alcanzar un gran número de agentes atravesando una misma área, por lo que se torna infactible en los casos estudiados en este trabajo, donde se pretende alcanzar emboscadas utilizando el mínimo número de agentes posible.

### 3. Definición Formal del Problema

Sea  $G = (V, E)$  un grafo (dirigido o no), donde  $V$  es el conjunto de nodos y  $E$  el conjunto de arcos. Sea  $A$  un conjunto de agentes interesados en alcanzar un punto común  $t \in V$ . Se tiene que cada agente  $a \in A$ , se encuentra en algún nodo del grafo denotado  $pos(a)$ . Además, se define para  $a$  la función de costos de su desplazamiento por el grafo  $\lambda_a : E \rightarrow \mathbb{R}^{\geq 0}$ . Se considera, sin pérdida de generalidad, que  $V$  y  $E$  son comunes para todos los agentes, con el fin de no generar inconsistencias en la información compartida entre estos. Sin embargo, cada agente puede tener función de costos distintas, adaptadas a sus restricciones. Sea  $path(a)$  con  $a \in A$ , el camino que está siguiendo el agente  $a$  hasta el nodo  $t$ .

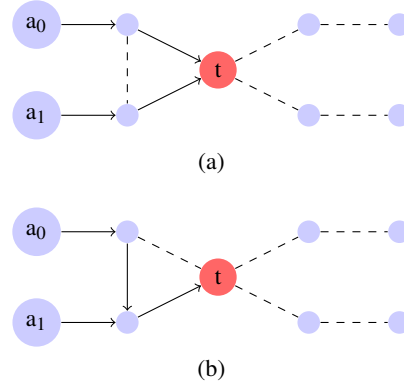
Previamente, el grado de emboscada (1) efectuado por el conjunto de agentes  $A$  al nodo  $t$  se definió como [FGC12a] [FGC12b]:

$$\Phi(t) = \frac{|\{i : path(i) \leq pos(j), \dots, i, t, j \in A\}|}{\min(|\{< i, t > : < i, t > \in E\}|, |A|)} \quad (1)$$

Esta métrica define la emboscada como la proporción de nodos adyacentes al nodo objetivo desde los cuales algún

agente está alcanzándolo. Normalizando este valor con respecto al máximo número de nodos adyacentes a  $t$  mediante los cuales, con el número de agentes disponibles, podría éste ser alcanzado. Por lo tanto,  $0 \leq \Phi(t) \leq 1$ . Bajo estas condiciones, el objetivo de un algoritmo de generación de emboscadas es maximizar el valor de  $\Phi(t)$ , dando prioridad a la diversidad de los caminos considerados por el conjunto de agentes sobre su optimalidad.

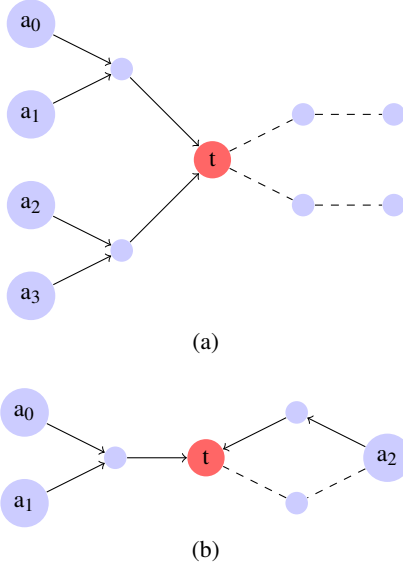
En la figura 1, se pueden observar en el grafo superior dos agentes situados en los nodos izquierdos. Los caminos seleccionados por los agentes hasta el destino  $t$  se presenta con una línea continua, mientras que los arcos no seleccionados se presentan con una línea segmentada. El número de nodos adyacentes al destino es de cuatro. De estos cuatro nodos, sólo dos están siendo considerados en los caminos de los agentes. Sin embargo, es el máximo grado de emboscada alcanzable con el número de agentes implicados. En el escenario inferior se están utilizando dos agentes para cubrir una única escapatoria, por lo que la emboscada no es óptima.



**Figure 1:** Ejemplos de emboscadas con dos agentes con valores de  $\Phi$  de 1.0 para (a) dado que cubre dos caminos y 0.5 para (b) dado que cubre un camino.

Sin embargo, esta métrica penaliza casos donde no es posible alcanzar una mejor emboscada, aún cuando se cuenta con suficientes agentes, debido a la configuración del grafo y de las posiciones iniciales de estos. Ejemplos de estos casos pueden ser observados en la figura 2. En la situación superior, se observa que a pesar de contar con cuatro agentes, las dos salidas restantes no son alcanzables por estos sin pasar antes por el nodo objetivo. Por otra parte, en la situación inferior, a pesar de que todas las salidas son alcanzables por los agentes, no es posible alcanzar una configuración de caminos capaz de cubrir todas las salidas. Para ambos casos, a pesar de que la asignación de caminos es la mejor dadas las condiciones iniciales, la métrica originalmente planteada reporta resultados subóptimos.

Por lo tanto, es necesario utilizar una nueva medida de emboscada para poder discriminar correctamente entre las escapatorias que podrían ser cubiertas por algún agente y aquellas que no. Esta nueva medida, denotada por  $\Phi^*$  (2) busca normalizar a los agentes utilizando el tamaño de la máxima asociación de agentes a los nodos predecesores del objetivo que éstos son capaces de alcanzar. Se define  $\Phi^*$  como:



**Figure 2:** Casos en los cuales la métrica original penaliza emboscadas óptimas. (a) Caso de no alcanzabilidad. (b) Caso de no existencia de mejor asignación posible.

$$\Phi^*(t) = \frac{|\{i : \text{path}(j) \leq \text{pos}(j), \dots, i, t >, j \in A\}|}{|\{y : (x, y) \in \text{MaxMatching}(\text{BipG}(G, A, t))\}|} \quad (2)$$

donde  $\text{BipG}(G, A, t)$  define el grafo bipartito que asocia a agentes con los nodos alcanzables por cada uno que sean predecesores del nodo objetivo (3). Este grafo se construye considerando como nodos el conjunto de los agentes y el conjunto de los nodos predecesores alcanzables por algún agente (4). Por otra parte, los arcos provienen de la matriz de alcanzabilidad entre agentes y predecesores (5).

$$\text{BipG}(G, A, t) = (A \cup \text{Dst}(G, A, t), \text{Edges}(G, A, t)) \quad (3)$$

$$\text{Dst}(G, A, t) = \text{Reachable}(\text{Reduced}(G, t), A) \quad (4)$$

$$\text{Edges}(G, A, t) = \{(x, y) : x \in A \wedge y \in \text{Dst}(G, A, t) \wedge \text{ScopeXY}(G, A, t, x, y)\} \quad (5)$$

$$\text{Reduced}((V, E), t) = (V - \{t\}, E - \{< x, y > : x = t \vee y = t\})$$

$$\text{Reachable}(G, A) = \bigcup_{a \in A} \text{Scope}(G, \text{pos}(a))$$

$$\text{Scope}(\{\{ \}, E\}, v) = \{ \}$$

$$\text{Scope}(G, v) = \{v\} \cup \bigcup_{w \in \text{suc}(G, v)} \text{Scope}(\text{Reduced}(G, v), w)$$

$$\text{ScopeXY}(G, A, t, x, y) = x \in A \wedge y \in \text{Scope}(\text{Reduced}(G, t), \text{pos}(x))$$

y donde, asumiendo que el operador  $\text{argmax}$  retorna el conjunto de mayor tamaño que evalúe como cierta la condición expuesta, se define la máxima asociación de un grafo como (6)

$$\text{MaxMatching}((V, E)) = \text{argmax}_{m \subseteq E} (\text{AtMostOnceSrc}(m) \wedge \text{AtMostOnceDst}(m)) \quad (6)$$

Una implementación real de esta métrica debe usar un algoritmo de cálculo de apareamiento en grafos bipartitos en tiempo polinomial [W\*01]. La definición acá planteada en términos de generación de subconjuntos es utilizada únicamente por motivos de definición formal. En cualquier caso, el costo asociado a calcular la calidad de la emboscada no se ve reflejado en el cálculo de las rutas para los agentes, dado que esto es un procedimiento de validación experimental y no forma parte del algoritmo de A\*mbushni de sus variantes.

#### 4. A\*, Ambush y sus Variantes

En esta sección se muestra el conjunto de adaptaciones realizadas al algoritmo de A\*, presentadas anteriormente en el artículo [FGC12a], para el cálculo de emboscadas. Esta sección se presenta por fin de completitud del trabajo planteado para un mayor entendimiento de los experimentos realizados.

##### 4.1. A\*

El algoritmo de A\* [HNR72] [RN93] [MF09] es una variación del algoritmo de Dijkstra [CLRS09] para cómputos de caminos de costo mínimo. Consiste en un algoritmo de búsqueda informada [RN93], basado en los siguientes elementos:

- $g$ : Es el costo acumulado desde el nodo inicial a un nodo actual  $v$ .
- $\hat{h}$ : Estimado del costo desde el nodo actual  $v$  a la meta.
- $\hat{f} = g + \hat{h}$ : Estimado del costo desde el nodo inicial a la meta, pasando por  $v$ .

Para garantizar optimalidad, la heurística  $\hat{h}$  debe ser admisible [HNR72], es decir, no debe estimar costos mayores al óptimo del grafo. El algoritmo actúa de forma voraz, expandiendo el siguiente nodo no explorado con menor costo estimado  $\hat{f}$ . Este proceso continúa hasta llegar a la meta.

En este caso, la complejidad asintótica en tiempo del algoritmo de A\* es de  $\mathcal{O}(|V| \log(|V|) + |V| * h + |E|)$ , con  $h$  el costo de cómputo de la función  $\hat{h}$ , suponiendo que se cuenta con una implementación eficiente de cola de prioridades tal como un Fibonacci heap [CLRS09] (estructura de montículo que permite acceder al mínimo elemento del montículo y agregar un elemento en tiempo constante; además de eliminar uno en tiempo logarítmico amortizado) y que estamos computando el costo heurístico de cada nodo una sola vez.

##### 4.2. A\*mbush

A\*mbush, presentado como una variación de A\* para el cálculo de emboscadas [FGC12a], consiste en una modificación de la función  $g$ , que favorezca la diversidad de caminos, denominada  $g'$ . Esta modificación busca penalizar aquellos nodos/arcs a través de los cuales pasen más agentes, para

esto, se asume que los agentes pueden establecer comunicación total entre ellos para obtener información compartida del cálculo de los caminos.

Sea  $\Psi(v, i) = 1 + (\#j : j \in A \wedge v \in \text{path}(j))$ , el número de agentes distintos al agente  $i$ , que contienen al nodo  $v$  en sus caminos hasta  $t$ . Se considera como nulo los caminos aún no computados de los otros agentes. Para el nodo inicial, se considera  $g'(\text{pos}(i), i) = 0$ . Sea  $\langle v, w \rangle$  el siguiente arco a considerar en la expansión del nodo  $v$  en una iteración cualquiera del algoritmo, se considera  $g'(w, i) = g'(v, i) + \lambda_i(\langle v, w \rangle) \cdot \Psi(w, i)^2$ .

Dado que  $\Psi(v, i) \geq 1$ , el camino obtenido por A\*mbush es óptimo sobre la nueva definición de  $g'$ , por lo que las propiedades de A\* se mantienen [HNR72]. Sin embargo, sobre la función original de costos, el camino obtenido no es necesariamente óptimo. En términos de eficiencia del algoritmo, si se precomputa la función de incremento de costos  $\Psi$ , el costo de computar  $g'$  es asintóticamente igual al de  $g$ , por lo tanto, la única variación en el costo del algoritmo con respecto a A\*, viene dada por el cómputo inicial de la función  $\Psi$ , dando como complejidad en tiempo  $\mathcal{O}(|V|\log(|V|) + |V| \cdot h + |E| + |A| \cdot |V|)$ .

En el campo específico de los videojuegos, los grafos de interés, vienen dados por la división en polígonos del mapa [MF09] [CS11] (regularmente con pocos lados), según las regiones transitables de éste o cuadrículas y sus adyacencias [MF09] [CS11]. Dado que los polígonos tienen un número reducido de lados, estos grafos, suelen ser poco densos; es decir,  $|E| \in \mathcal{O}(|V|)$ , por lo que el tiempo de ejecución de este método, sobre los grafos de interés en el área de videojuegos, viene dado por  $\mathcal{O}(|V|(\log(|V|) + h + |A|))$ .

#### 4.3. P-A\*mbush

Los caminos calculados por A\*mbush pueden verse más o menos naturales dependiendo del orden en el cual los agentes computan las rutas. Por ejemplo, si un agente cercano al oponente escoge su camino de último, este procederá a recorrer una ruta larga, dando prioridad a la diversificación, mientras que agentes alejados, por los cuales el jugador no suele tener un foco de atención alto, toman rutas óptimas hacia el objetivo. P-A\*mbush se trata de una variación de A\*mbush que considera el orden del cómputo de los caminos de acuerdo a un método de asignación de prioridades P. Este método puede considerar no sólo la distancia sino también otros factores dependientes de la aplicación particular. En este trabajo se considera la prioridad de acuerdo a la distancia al destino.

Otra ventaja, al considerar este método de prioridades, es que P-A\*mbush alcanza siempre la meta en el menor tiempo posible por al menos uno de los agentes. A pesar de que el costo de calcular P-A\*mbush con distancia real es mayor que el costo de A\*mbush para un único agente, el costo amortizado del cálculo de todos los agentes es el mismo que el de A\*mbush. Se sugiere utilizar otro tipo de distancias como la euclidiana en caso de ser necesario disminuir los tiempos de respuesta.

#### 4.4. R-A\*mbush

Buscando minimizar el incremento en el costo de los caminos se presentó R-A\*mbush. El cual plantea ejecutar

A\*mbush una vez alcanzado una cierta distancia mínima al objetivo ( $R$ ) y, antes de alcanzar este punto, utilizar A\*. De esta forma, se evita que los agentes escojan rutas innecesariamente largas desde un inicio.

Sin embargo, esta solución es menos general que la original, dado que requiere la incorporación del radio  $R$ , el cual es variable para cada posible mapa de juego y disposición de los agentes. Es por esto que se plantea la variante SAR-A\*mbush, la cual incorpora la selección del mejor radio  $R$  a partir del cual los agentes comienzan a ejecutar A\*mbush.

#### 4.5. SAR-A\*mbush

SAR-A\*mbush es la última variación estudiada, la cual resuelve el problema de selección del radio  $R$  para el algoritmo de R-A\*mbush. Su nombre proviene de Self Adaptive R-A\*mbush. Este método hace calcular inicialmente a cada agente su camino hasta el destino utilizando A\*. Luego, un conjunto de puntos es escogido de entre los nodos recorridos en el camino mínimo como posibles candidatos para establecer el valor de  $R$ . El algoritmo escoge como mejor  $R$  aquel con mínimo radio que incrementa el grado de emboscada. En caso de empates, el algoritmo considera el balance de los agentes (7) a través de los predecesores del nodo objetivo, con el fin de lograr una distribución más equilibrada de éstos. En la fórmula presentada (7),  $n$  representa el número de agentes que ya han calculado el camino y  $\text{num}(i)$  es el número de agentes que alcanzan a  $t$  a través de  $i$ . La escogencia del menor radio se centra en empezar el camino subóptimo tan tarde como sea posible. Es importante notar que R-A\*mbush se comporta exactamente igual A\* cuando  $R = 0$  y como A\*mbush cuando  $R$  es mayor o igual que la distancia a la meta del agente.

$$1 - \frac{\left( \sum_{i \in \text{pred}(t) \wedge \text{num}(i) > \left\lceil \frac{n}{|\text{pred}(t)|} \right\rceil} : \text{num}(i) - \left\lceil \frac{n}{|\text{pred}(t)|} \right\rceil \right)}{n - \left\lceil \frac{n}{|\text{pred}(t)|} \right\rceil} \quad (7)$$

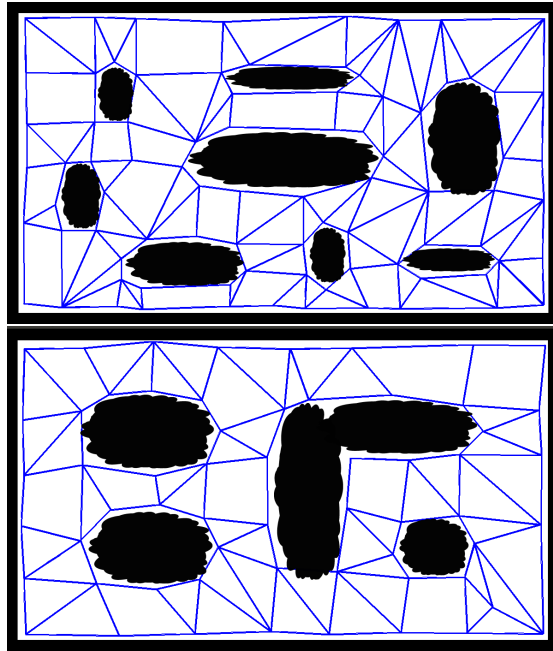
La eficiencia de esta variante esta condicionada al número de nodos considerados como candidatos a radio. Con el fin de demostrar el potencial máximo del algoritmo, se considera cada posible nodo como candidato, sin embargo, para implementación con casos reales se aconseja utilizar un número fijo constante o utilizar algún criterio como selección de los nodos que se encuentran en las posiciones que son potencia de dos, para así minimizar el impacto de recomputar el camino para cada uno.

### 5. Experimentos

Para cada experimento se estudian cinco algoritmos: una implementación base de A\* que sirve de referencia, una implementación de caminos simples aleatorios mediante búsqueda en profundidad, A\*mbush, P-A\*mbush con mecanismo de prioridad determinado por la distancia real de los agentes a la meta y finalmente, SAR-A\*mbush.

presentan resultados del incremento medio porcentual de los caminos. Para cada topología de grafo, se generan aleatoriamente 100 grafos, sobre los cuales se ejecutan experimentos con 100 disposiciones distintas de agentes y del nodo objetivo. Para cada uno de estos algoritmos se muestra el valor

de emboscada utilizando la métrica originalmente propuesta y utilizando la métrica propuesta en el presente trabajo. El número de agentes es variado con el fin de mostrar su impacto en el grado de emboscada y en el incremento derivado de la escogencia de caminos subóptimos. Además se muestran resultados con las dos instancias de mapas (g1 y g2) presentadas en el trabajo previo [FGC12b] con 60 y 85 nodos respectivamente. Estos mapas provienen de la poligonalización del mapa de juego en polígonos. Estas restricciones son ampliamente utilizadas en juegos [MF09]. Estos grafos se pueden visualizar en la figura 3.



**Figure 3:** *Arriba:* Mapa 1 poligonalizado (60 polígonos). *Abajo:* Mapa 2 poligonalizado (85 polígonos).

## 6. Conclusiones y Trabajo Futuro

### References

- [CLRS09] CORMEN T., LEISERSON C., RIVEST R., STEIN C.: *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009. [3](#)
- [CS11] CUI X., SHI H.: Direction oriented pathfinding in video games. *International Journal of Artificial Intelligence & Applications* 2 (2011). [4](#)
- [FGC12a] FERNÁNDEZ K., GONZÁLEZ G., CHANG C.: A\* mbush family: A\* variations for ambush behavior and path diversity generation. In *Motion in Games*. Springer, 2012, pp. 314–325. [1](#), [2](#), [3](#)
- [FGC12b] FERNÁNDEZ K., GONZÁLEZ G., CHANG C.: A mbush: una variación de a para generar emboscadas y diversidad de caminos. *Memorias del II Simposio Científico y Tecnológico en Ciencias de la Computación (SCTC 2012)* (2012). [1](#), [2](#), [5](#)
- [HNR72] HART P., NILSSON N., RAPHAEL B.: Correction to “a formal basis for the heuristic determination of minimum cost paths”. *SIGART Newsletter* 37 (1972), 28–29. [1](#), [3](#), [4](#)
- [MF09] MILLINGTON I., FUNGE J.: *Artificial Intelligence for Games*, 2nd ed. Morgan Kaufmann Publishers, 2009. [1](#), [3](#), [4](#), [5](#)
- [RN93] RUSSELL S., NORVIG P.: *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice-Hall, Englewood Cliffs, NJ, 1993. [1](#), [3](#)

- [Sil06] SILVER D.: Cooperative pathfinding. *AI Game Programming Wisdom 3* (2006), 99–111. [1](#)
- [vTCG12] VAN TOLL W. G., COOK A. F., GERAERTS R.: Real-time density-based crowd simulation. *Computer Animation and Virtual Worlds* 23, 1 (2012), 59–69. [1](#), [2](#)
- [W\*01] WEST D. B., ET AL.: *Introduction to graph theory*, vol. 2. Prentice hall Upper Saddle River, 2001. [3](#)