УДК 004.94

К. В. Кумунжиев, В. Е. Черненко, А. А. Малыханов

АЛГОРИТМ ПОСТРОЕНИЯ ТРАЕКТОРИИ ДВИЖЕНИЯ АГЕНТА В МОДЕЛИ ТРАНСПОРТНОЙ СИСТЕМЫ

Ульяновский государственный университет
kvk@kvk.ulsu.ru, ChernenkoVE@gmail.com, malyhanov@yandex.ru

Предложен алгоритм построения пути следования агента в пространстве с набором препятствий. Результат работы алгоритма – траектория, отстоящая от любого из препятствий не менее, чем на заданное значение.

Ключевые слова: агентное моделирование, построение траектории движения, объезд препятствий, нахождение пути в двухмерном пространстве.

K. V. Kumunjiev, V. E. Chernenko, A. A. Malykhanov

ALGORITHM OF BUILDING AGENT'S TRAJECTORY IN TRAFFIC MICROSIMULATION MODEL

Ulyanovsk State University

An algorithm of finding way in two-dimensional space with polygonal obstacles is proposed. The algorithm builds a trajectory with distance not less than given value to any obstacle. Proposed algorithm can be used in agent-based models with special relations between agents and environment.

Key words: agent-based simulation, trajectory building, way finding, barriers in 2D space.

Потребность в создании имитационных моделей транспортных систем возрастает с увеличением транспортных потоков. В основе большинства таких моделей лежат алгоритмы, учитывающие пространственные положения агентов и их передвижение, при этом адекватность модели во многом определяется алгоритмом поведения агентов. Более того, целесообразным является перенесение алгоритмической сложности модели именно на логику функционирования агента. В последнее время модели со сложной логикой поведения агента выделяются в особый класс агентных моделей – модели с «умными» агентами. Использование таких моделей оправдано, в частности, при моделировании систем движения автомобилей на мелком и среднем уровне. Важнейшей частью алгоритма функционирования агента в таких моделях является построение траектории движения агента.

Пусть имеется двухмерное непрерывное пространство, а агент представляет собой прямоугольник заданных размеров, находящийся в исходной точке движения S. В пространстве определены конечная точка движения агента D и препятствия, представляющие собой замкнутые многоугольники произвольной формы без самопересечений. Конечное (N штук) множество препятствий обозначим $P = \{ p_i \mid i = ...N \}$. Точки S и D находятся вне любого из препятствий p_i.

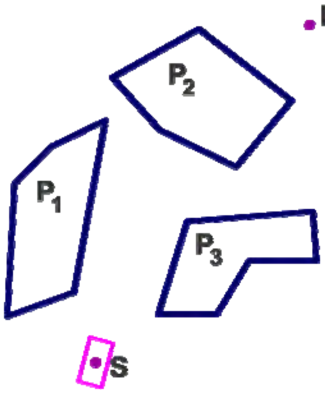

Рис. 1. Иллюстрация к формальной постановке задачи

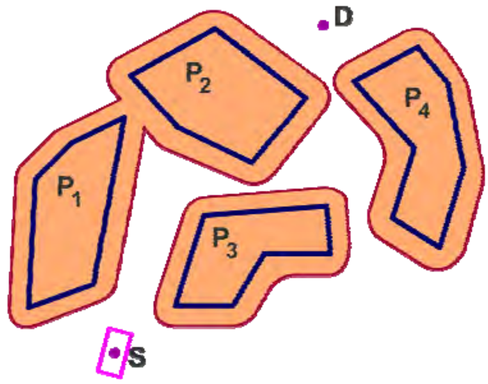

Рис. 2. Схематичное изображение множества F

Необходимо построить ломаную W, начальной точкой которой является точка S, конечной – точка D, удовлетворяющую условию: все точки W должны отстоять от любой из точек препятствий не менее, чем на заданное расстояние d. Это условие объясняется спецификой применения алгоритма – моделирование траектории движения некоторого объекта, при котором необходимо выдерживать безопасную дистанцию до препятствий (например, объезд автомобилем препятствий на дороге). Ломаную W назовем искомой траекторией движения агента.

Существует несколько алгоритмов нахождения пути между двумя точками в двумерном пространстве с барьерами [2, 3]. Некоторые из них [2] основаны на отыскании кривых второго и высших порядков, что является неприемлемым при реализации в AnyLogic, так как агенты в AnyLogic могут перемещаться только по прямолинейным траекториям. Другие алгоритмы находят путь в виде ломанной, однако, она может касаться препятствий, что противоречит поставленной задаче. В данной статье предлагается алгоритм построения траектории движения агента в двухмерном пространстве с учетом текущего положения агента, целевой точки его движения и имеющихся на пути препятствий. Алгоритм позволяет найти траекторию движения агента, отстоящую на заданном расстоянии от любого из препятствий.

Среди всех точек пространства можно выделить множество точек F, с которыми не может пересекаться искомая траектория W. Построим множество F, основываясь на следующих соображениях:

любая внутренняя точка любого препятствия p_i принадлежит F;

любая точка, отстоящая от любого препятствия p_i менее, чем на d, принадлежит F.

Схематично множество F показано на рис. 2.

Множество F имеет сложную структуру: его границы состоят не только из прямых линий, но и дуг окружностей, описанных вокруг точек изгиба ломаных. Учитывать такие формы границ множества F сложно, так как их математическое описание будет содержать уравнения второго порядка, и операции с такими уравнениям внесут в алгоритм дополнительную сложность, что приведет к излишним вычислениям при построении пути объезда препятствий. Следовательно, для исключения из алгоритма нахождения траектории W операций с кривыми второго порядка необходимо произвести аппроксимацию криволинейных участков ломаными. Рис. 3 иллюстрирует необходимые построения для некоторого препятствия p_i.

Возьмем d^*, большее чем d, построим внешнюю границу l_i множества точек, отстоящих от барьера p_i менее, чем на d^*. Рассмотрим кривую l_i вблизи некоторой точки p_i^j изгиба ломаной p_i.

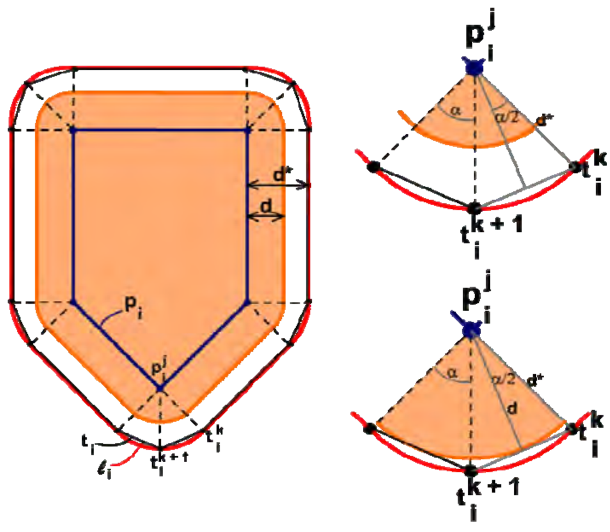

Рис. 3. Дополнительные построения

Для того, чтобы избавиться от дуги окружности, можно приблизить ее некоторой ломаной, причем все ее точки должны находиться на расстоянии не менее чем d от точки p_i^j. Построение ломаной будем производить следующим образом: в качестве первой точки ломаной берется точка пересечения кривой l_i и луча, проведенного из p_i^j перпендикулярно одному из отрезков препятствия p_i; все следующие точки ломаной лежат на пересечении окружности с лучами, исходящими из p_i^j под углами, кратными некоторому значению α. Таким образом, необходимо выбрать d^* и α так, чтобы:

d^* не намного превышало d (чтобы не допустить неточности при построении искомой траектории);

α не было бы слишком малым (во избежание возникновения большого количества точек изгиба ломаной).

Положим $d^* = \beta d$, тогда $\beta = 1 \Big/ \cos\frac{\alpha}{2}$. Сначала выберем угол α, а затем определим соответствующее ему значение β. Как отмечалось, угол α не должен быть слишком малым, однако при больших значениях α увеличивается и β, что приводит к нежелательному увеличению d^*. В результате экспериментов выявлено, что приемлемый результат достигается при значении $\alpha = \frac{\pi}{3}$, тогда $\beta = \frac{2}{\sqrt{3}} \approx 1{,}155$ и $d^* = \frac{2d}{\sqrt{3}}$.

Определив α и d^*, можно построить ломаную, окружающую препятствие p_i и не приближающуюся к p_i ближе, чем на d. Построенную ломаную t_i, состоящую из точек t_i^k, назовем огибающей для препятствия p_i.

Точки ломаной-препятствия p_i обозначим $\left\{p_i^1,...,p_i^{m_i}\right\}$. Аналогично точки огибающей t_i обозначим $\left\{t_i^1,...,t_i^{k_i}\right\}$. Тогда каждой точке t_i^k будет соответствовать некоторая точка p_i^j.

Сформируем множество C из отрезков вида $[p_i^j, t_i^k]$, причем отрезок добавляем в множество, если расстояние от точки t_i^k до любого из препятствий не меньше d^*. Сформируем множество V из конечных точек t_i^k отрезков множества C. Добавим в множество V точки S и D. Сформируем множество отрезков E следующим образом: для всех пар точек $v_i, v_j \in V$ отрезок $[v_i, v_j]$ включаем в E, если он не пересекается ни с одним из отрезков ломаных-препятствий p_i и не пересекается ни с одним из отрезков множества C.

Представим множества V и E в виде графа $G = \left\langle V^*, E^* \right\rangle$, где каждой точке $v \in V$ соответствует вершина графа $v^* \in V^*$, а каждому отрезку $e \in E$ – ребро графа $e^* \in E^*$. Построенный граф изображен на рис. 4.

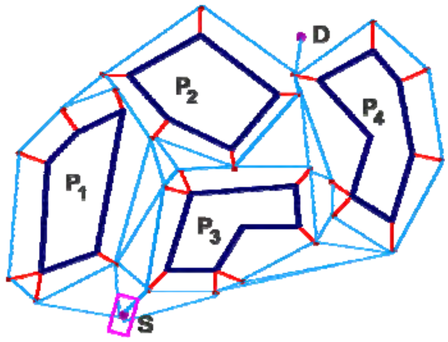

Рис. 4. Граф поиска пути

Причем, $(V \cup E) \cap F = \varnothing$ то есть все точки, соответствующие ребрам и вершинам графа не принадлежат F. Значит, ни одна из точек траектории не приближается к препятствиям ближе, чем на d.

Теперь задача нахождения пути W состоит в поиске кратчайшего пути на графе G, из вершины, соответствующей точке S, в вершину, соответствующую точке D. Найдем такой путь, применив алгоритм Дейкстры [1]. Результатом работы этого алгоритма может стать последовательность $(S^*, v_{q_1}^*, ..., v_{q_k}^*, D^*)$, либо сообщение о том, что путь не найден. В случае положительного результата работы алгоритма путь W строится как объединение отрезков, соответствующих ребрам, связывающим последовательно вершины $(S^*, v_{q_1}^*, ..., v_{q_k}^*, D^*)$.

Предложенный в статье алгоритм положен в основу логики поведения агента в моделях транспортных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Dijkstra E. W.* A note on two problems in connection with graphs, Numerical Mathematics, Vol. 1, Pp. 269–271, Oct 1959.

2. *Kapoor S., Maheshwari S.N.* Efficient algorithms for Euclidean shortest path and visibility problems with polygonal obstacles, Proceedings of 4th Annual ACM Symposium on Computational Geometry, 1988, Pp. 172–182.

3. *Klippel1, K.-F. Richter, S. Hansen.* Wayfinding Choreme Maps, Visual Information and Information Systems. 8th International Conference, VISUAL 2005, Springer, Berlin.