

УДК 004.942

МЕТОДЫ И АЛГОРИТМЫ ПОИСКА ОПТИМАЛЬНОГО ПУТИ ПЕРЕМЕЩЕНИЯ АГЕНТОВ ПО ТЕРРИТОРИАЛЬНО-РАСПРЕДЕЛЕННЫМ ТОРГОВЫМ ТОЧКАМ

Поправко О.О., Телятников А.О.

*Донецкий национальный технический университет г. Донецк
кафедра автоматизированных систем управления
e-mail: popravkodon@gmail.com*

Аннотация

Поправко О.О., Телятников А.О. Методы и алгоритмы поиска оптимального пути перемещения агентов по территориально-распределенным торговым точкам. В статье рассматривается вопрос нахождения оптимального пути следования торговых агентов для приема заказов на продукцию в магазинах, супермаркетах и других торговых точках согласно определенного расписания, где указано, как часто следует посещать данную торговую точку.

Общая постановка задачи. На текущий момент существует много систем, решающих сходные задачи, в основном они ориентированы на сектор логистики перевозок. Однако большая часть взаимодействует с локальными базами карт местности, что затрудняет масштабирование и регулярное обновление данных карт, что в условиях высокой динамичности процесса наносит ущерб удобству и поддержке данных систем. Таким образом, становится очевидной необходимость в дальнейших разработках и исследовании в данной области научной деятельности. В данной статье представлен материал для получения представления о состоянии и тенденциях развития разработок, выявленных проблемах и недостатках функционирования существующих систем. Также будет обоснован выбор направления исследований этой области и разработана архитектура системы.

Постановка задач исследования. Можно выделить два вида систем оптимизации движения – настольные и ориентированные на доступ через глобальную сеть.

Основным и определяющим различием в данных типах систем является источник информации для составления маршрута. Настольные системы берут информацию из баз данных, расположенных на собственных серверах, или, получая данные по локальной сети. Из преимуществ стоит отметить контроль над доступом к информации, кроме того скорость работы приложений, построенных по настольной клиент – серверной архитектуре может во много раз превышать скорость работы web – ориентированной подсистемы. Недостатком данных систем является необходимость постоянного обновления данных, и их дополнения новыми, что делает такие системы зависимыми от частоты обновления. Это отражается на актуальности сведений, необходимых для выполнения конечной задачи – нахождения оптимального маршрута следования.

Веб – базированные системы лишены данного недостатка, что обосновывает ориентирование разрабатываемой подсистемы на использование поставщиков данных в Интернет. Использование этих данных не требует оплаты, что, несомненно, является также преимуществом, по сравнению с настольными системами. Далее детально рассмотрим критерии при перемещении агентов по территории. В качестве исходных данных принимается расписание посещения агентами торговых точек – маршрутный лист, адреса торговых точек, в некоторых случаях их точные географические координаты (широта, долгота). В результате необходимо получить маршрут, построенный согласно различным критериям:

- расстояние между торговыми точками;
- время, затраченное на поездку;

- расход топлива;
- тип местности.

Работа системы подразумевает наличие поставщика географической информации, такого как Google Maps, который обеспечит систему информацией о дорогах, перекрестках, альтернативных вариантах подъезда к месту назначения. Система, на основании этой информации, сопоставляя ее с информацией о расписании посещения торговых точек, создаст маршрут посещения торговых точек, вычислит показатели, характеризующие выбор маршрута следования. Доступ к аналитической информации будет предоставлен аналитикам, просмотр маршрутов на карте с точным описанием маршрута проезда - агентам.

Схема функционирования системы (иначе говоря, архитектура) представлена на рисунке 1.

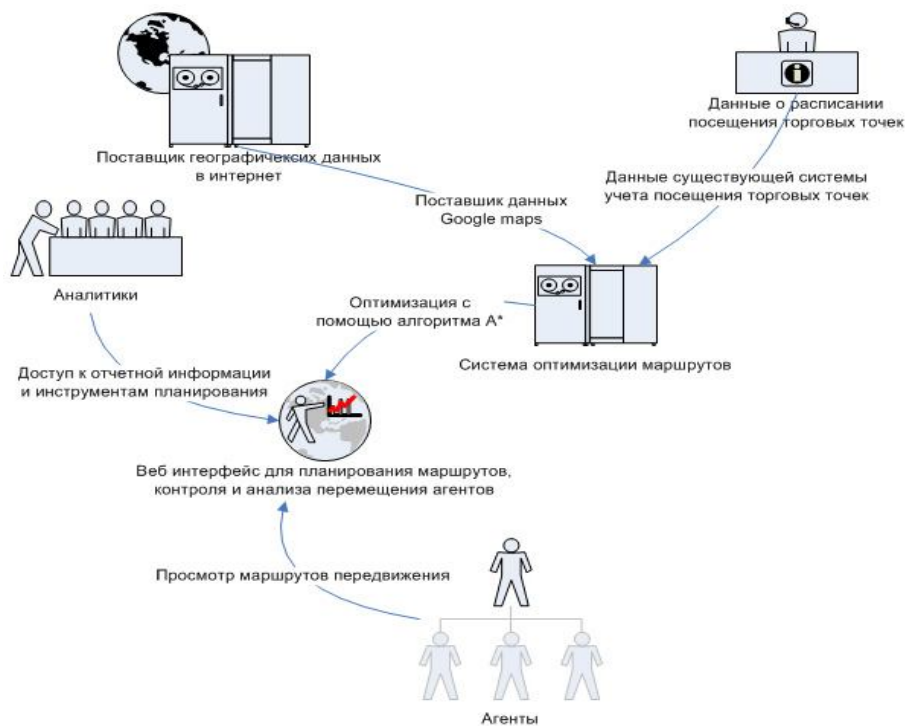


Рисунок 1. Архитектура системы.

После получения маршрута необходимо нанести его на карту местности, с указанием посещаемых торговых точек как пунктов остановки агента.

Для решения поставленной задачи необходимо получить географические координаты путей между торговыми точками, то есть координаты магистралей, улиц, домов, которые агент обязан посетить. После получения таких данных можно подключать алгоритмы поиска оптимальных маршрутов по имеющимся координатам.

Математически, город можно представить как граф, то есть упорядоченная пара $G: = (V, E)$, где:

- V это множество вершин или узлов,
- E это множество рёбер графа.

Вершины V представляют собой перекрестки, пересечения дорог, ребра E - объединяющие их участки дорог. Эта схема применима не только в масштабе города, но и по сути сети дорог страны.

Таким образом, для решения задачи следует использовать алгоритмы на графах, позволяющие найти оптимальный путь через набор вершин.

Решения и результаты исследований. Для решения поставленной задачи существует ряд алгоритмов:

- алгоритм поиска A*;
- алгоритм Дейкстры;
- поиска кратчайших путей алгоритм Флойда;
- генетический алгоритм;

Сравним алгоритмы по критерию скорости выполнения (таблица 1).

Таблица 1. Сравнение алгоритмов по скорости выполнения.

Алгоритм	Баз ГА	С ГА
Алгоритм A*	102	89
Алгоритм Дейкстры	98	87
Алгоритм Флойда	85	76

На основании современных исследований [4], можно сделать вывод о целесообразности использования алгоритма A*, как самого эффективного.

Алгоритм поиска A* - алгоритм поиска по первому наилучшему совпадению на графе, который находит маршрут с наименьшей стоимостью от одной вершины (начальной) к другой (целевой, конечной).

Порядок обхода вершин определяется эвристической функцией «расстояние + стоимость» (обычно обозначаемой как $f(x)$). Эта функция — сумма двух других: функции стоимости достижения рассматриваемой вершины (x) из начальной (обычно обозначается как $g(x)$ и может быть как эвристической, так и нет) и эвристической оценкой расстояния от рассматриваемой вершины к конечной (обозначается как $h(x)$).

Функция $h(x)$ должна быть допустимой эвристической оценкой, то есть не должна переоценивать расстояния к целевой вершине. Например, для задачи маршрутизации $h(x)$ может представлять собой расстояние до цели по прямой линии, так как это физически наименьшее возможное расстояние между двумя точками.

A* пошагово просматривает все пути, ведущие от начальной вершины в конечную, пока не найдёт минимальный. Как и все информированные алгоритмы поиска, он просматривает сначала те маршруты, которые «кажутся» ведущими к цели. От жадного алгоритма (который тоже является алгоритмом поиска по первому лучшему совпадению) его отличает то, что при выборе вершины он учитывает, помимо прочего, весь пройденный до неё путь (составляющая $g(x)$ — это стоимость пути от начальной вершины, а не от предыдущей, как в жадном алгоритме). В начале работы просматриваются узлы, смежные с начальной; выбирается тот из них, который имеет минимальное значение $f(x)$, после чего этот узел раскрывается. На каждом этапе алгоритм оперирует с множеством путей из начальной точки до всех ещё не раскрытых (листовых) вершин графа («множеством частных решений»), которое размещается в очереди с приоритетом. Приоритет пути определяется по значению $f(x) = g(x) + h(x)$. Алгоритм продолжает свою работу до тех пор, пока значение $f(x)$ целевой вершины не окажется меньшим, чем любое значение в очереди (либо пока всё дерево не будет просмотрено). Из множественных решений выбирается решение с наименьшей стоимостью.

Выводы Рассмотрев возможные алгоритмы разрешения задачи, сравнив эффективность, результативность и временные затраты, необходимые для получения результата, оптимальным применимо к задаче является алгоритм A*.

Список литературы

1. Лорьер Ж.-Л. Системы искусственного интеллекта / Пер. с фр. и ред. В. Л. Стефанюка. — М.: Мир, 1991. — С. 238—244. — 20 000 экз. экз. — ISBN 5-03-001408-X
2. Седжвик Р. Фундаментальные алгоритмы 2003. Москва. 2003. — 1100с.
3. Касьянов В.Н., Евстигнеев В.А., Графы в программировании 2003.
4. Tarik Attar. An investigation on routing algorithms for web